

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
UNIVERSITY OF ADRAR
FACULTY OF MATTER SCIENCES, MATHEMATICS AND COMPUTER SCIENCE
DEPARTMENT OF COMPUTER SCIENCE



DOCTORAL THESIS

Submitted in partial fulfillment of the requirements for the degree of Doctor

In Computer Science

Specialization: Artificial Intelligence

Presented by:

TABBI Amina

THESIS TITLE

=====
**Contribution of a Bio-inspired Meta-heuristic for Optimization of
Localization in Wireless Sensor Networks**
=====

Publicly defended on May 13, 2026, before the examination committee:

Name	Academic Rank	Institution	Role
Mr. KADDI Mohammed	Professor	University of Adrar, Algeria	Chair
Mr. RABHI Seddik	Associate Professor	University of Adrar, Algeria	Supervisor
Mr. MAMOUNI El Mamoun	Associate Professor	University of Adrar, Algeria	Examiner
Mr. OUAHAB Abdelwhab	Associate Professor	University of Adrar, Algeria	Examiner
Mr. KABOU Salheddine	Associate Professor	University of Bechar, Algeria	Examiner
Mr. SABA Djamel	Research Director	URERMS – Adrar , Algeria	Examiner

Academic Year: 2025 – 2026

I would like to dedicate this thesis to my beloved mother.

To my beloved brothers Khalil, Haider, and Mokhtar.

To my beloved sister Soumia.

To my beloved nieces Rinad and Maria.

To my beloved nephew Mohammed Ikbel.

And to my beloved friends.

Acknowledgments

Alhamdulillah, all praise is due to Allah for His countless blessings and for granting me the strength and perseverance to complete this thesis.

I would first like to express my deepest gratitude to my beloved mother for her endless support, encouragement, and prayers throughout my academic journey.

I would also like to sincerely thank my supervisor, Mr. Seddik Rabhi, for his trust, continuous guidance, valuable advice, and significant contributions to this research.

I extend my special thanks to Professor Dwi Joko Suroso for warmly welcoming me to his university, for his guidance and constant support during my short research stay, and for his remarkable kindness and humanity, which I will always deeply appreciate.

I would also like to express my special thanks to my dear friend Samira for her encouragement and motivation that inspired me to take the PhD examination. Her continuous support and belief in me have been truly valuable throughout this journey.

Finally, I would like to thank everyone who supported me directly or indirectly during the completion of this work.

ملخص

يُعدّ تحديد المواقع في شبكات المستشعرات اللاسلكية (WSNs) مهمةً أساسيةً تؤثر بشكل مباشر في دقة الشبكة وأدائها. ورغم تعدد تقنيات التحديد المطروحة في الأدبيات، لا تزال قيود عديدة قائمة، في حين أثبتت خوارزميات التحسين المستوحاة من الطبيعة فعاليتها الكبيرة في هذا المجال. في هذا العمل، يقترح خوارزميتان لتحسين تحديد مواقع العقد غير المعروفة. الخوارزمية الأولى، SLnO-CLA، تدمج خوارزمية مركز الثقل SLnO: Sea Lion مع خوارزمية تحسين أسد البحر CLA: Centroid Localization Algorithm Optimization. وتُستخدم SLnO لتحسين تقدير الموضع الأولي الذي توفره CLA وضبطه بدقة، بهدف توجيه عملية البحث وتحسين دقة التحديد وتسريع التقارب. أما الخوارزمية الثانية، IMP-SLnO (Improved Sea Lion Optimization)، فهي نسخة محسنة من SLnO تهدف إلى تعزيز الاستقرار والأداء العام من خلال آليات تطوير متعددة تدعم التنوع وتقوي الاستكشاف وتسرع التقارب. قُيِّم أداء SLnO-CLA ومقارنته بخوارزمية CLA الكلاسيكية عبر ٣٠ تشغيلاً مستقلاً باستخدام نشرات شبكة متطابقة. وتُظهر النتائج أن SLnO-CLA تُقلِّل متوسط خطأ التحديد بنسبة 93.7% مقارنةً بـ CLA المنفردة. كما قُيِّم أداء IMP-SLnO ومقارنته بخوارزمية SLnO الأصلية وعدد من الخوارزميات المعروفة، منها تحسين الذئب الرمادي GWO: Grey Wolf Optimization، وتحسين سرب الجسيمات PSO: Particle Swarm Optimization، وخوارزمية اليراعة (FA: Firefly Algorithm)، و SLnO-CLA. وتحقق IMP-SLnO متوسط خطأ تحديد قدره 0.2459 متر، بتحسين نسبته 82.1% عن SLnO الأصلية، متفوقةً على جميع الخوارزميات المقارنة في جميع المقاييس المُبلَّغ عنها.

الكلمات المفتاحية: تحديد المواقع، شبكات المستشعرات اللاسلكية، خوارزميات الإلهام الحيوي، خوارزميات التحسين، تحسين أسد البحر، تحديد مواقع العقد

Résumé

La localisation dans les réseaux de capteurs sans fil (Wireless Sensor Networks, WSNs) constitue une tâche fondamentale qui influence directement la précision et les performances du réseau. Malgré les nombreuses techniques de localisation proposées dans la littérature, plusieurs limitations subsistent, tandis que les algorithmes d'optimisation inspirés de la nature ont démontré une grande efficacité dans ce domaine. Dans ce travail, deux méthodes de localisation sont proposées afin d'améliorer l'estimation de la position des nœuds inconnus. La première méthode, appelée SLnO-CLA, combine l'algorithme de localisation par centroïde (Centroid Localization Algorithm, CLA) avec l'algorithme d'optimisation Sea Lion Optimization (SLnO). L'algorithme SLnO est utilisé pour affiner et optimiser l'estimation initiale de position fournie par CLA, afin d'améliorer la précision de localisation et d'accélérer la convergence. La seconde méthode, nommée IMP-SLnO (Improved Sea Lion Optimization), représente une version améliorée de SLnO visant à renforcer la stabilité et les performances globales grâce à plusieurs mécanismes d'amélioration favorisant la diversité de la population, l'exploration de l'espace de recherche et la rapidité de convergence. Les performances de SLnO-CLA sont évaluées et comparées à celles de l'algorithme classique CLA à travers 30 exécutions indépendantes réalisées sur des déploiements de réseau identiques. Les résultats montrent que SLnO-CLA réduit l'Erreur Moyenne de Localisation (Mean Localization Error, MLE) de 93,7 % par rapport à CLA utilisé seul. Les performances d'IMP-SLnO sont également évaluées et comparées à celles de SLnO ainsi qu'à plusieurs algorithmes bio-inspirés bien connus, notamment Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO), Firefly Algorithm (FA) et SLnO-CLA. IMP-SLnO obtient une erreur moyenne de localisation de 0,2459 m, représentant une amélioration de 82,1 % par rapport à SLnO, et surpasse l'ensemble des algorithmes comparés selon tous les indicateurs de performance rapportés.

Mots-clés : Localisation, Réseaux de capteurs sans fil (WSNs), Algorithmes bio-inspirés, Algorithmes d'optimisation, Sea Lion Optimization (SLnO), Localisation des nœuds.

Abstract

Localization in Wireless Sensor Networks (WSNs) is a fundamental task that directly influences network accuracy and performance. Although many localization techniques have been proposed, several limitations remain, while bio-inspired optimization algorithms have shown strong effectiveness in this domain. In this work, two localization algorithms are proposed to improve the positioning of unknown nodes. The first, SLnO-CLA, combines the Centroid Localization Algorithm (CLA) with the Sea Lion Optimization (SLnO) algorithm. SLnO is used to optimize and refine the initial position estimate provided by CLA, improving localization accuracy, and accelerating the convergence. The second algorithm, referred to as IMP-SLnO, is an enhanced version of SLnO that focuses on improving stability and overall performance through multiple enhancement mechanisms that increase diversity, strengthen exploration, and speed up convergence. The performance of SLnO-CLA is evaluated and compared with the classical CLA algorithm across 30 independent runs using identical network deployments. Results show that SLnO-CLA reduces the Mean Localization Error (MLE) by 93.7% compared to plain CLA. The performance of IMP-SLnO is further evaluated against baseline SLnO and several well-known bio-inspired algorithms including Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO), Firefly Algorithm (FA), and SLnO-CLA. IMP-SLnO achieves an MLE of 0.2459 m, representing an 82.1% improvement over baseline SLnO, and outperforms all comparison algorithms across all reported metrics.

Keywords: Localization, Wireless Sensor Networks (WSNs), Bio-inspired Algorithms, Optimization Algorithms, Sea Lion Optimization (SLnO), Node Localization.

Contents

List of Figures	iv
List of Tables	vi
List of Abbreviations	viii
General Introduction	9
1 Wireless Sensor Networks: An Overview	11
1.1 Introduction	11
1.2 Sensor Node	11
1.3 Wireless Sensor Networks	12
1.3.1 Definition	12
1.3.2 Node Deployment in WSN	14
1.3.3 Types of WSNs	14
1.3.4 Topologies in WSNs	16
1.3.5 Applications in Wireless Sensor Networks	17
1.3.6 WSNs challenges	20
1.4 Conclusion	21
2 Localization Techniques in WSN	23
2.1 Introduction	23
2.2 Localization Fundamentals	23
2.2.1 Positioning of Nodes	23
2.2.2 GPS in WSN localization	24
2.2.3 System Model (Anchor Target Nodes)	24
2.3 Classification of WSN Localization Algorithms	25
2.3.1 According to the Calculation Location	25
2.3.2 Mobility-Based Localization Algorithms	26
2.3.3 According to Utilization of Extra Hardware	26

2.3.4	According to the Connectivity	28
2.3.5	According to the Granularity of Position Estimation	29
2.3.6	Anchor-Based vs. Anchor-Free Localization	29
2.4	Localization Process: Main Phases and Methodologies in WSNs	30
2.4.1	Distance Estimation Phase	30
2.4.2	Position Estimation	34
2.5	Performance Metrics of Localization Algorithms	45
2.6	Challenges in Wireless Sensor Network Node Localization	47
2.7	Conclusion	48
3	Localization Optimization Using Bio-Inspired Techniques	49
3.1	Introduction	49
3.2	General Optimization Formulation	49
3.3	Bio-Inspired Meta-heuristic Techniques (BIAs)	51
3.3.1	General Metalogy of BIAs	51
3.4	A review of BIA based Localization Optimization	52
3.4.1	Particle Swarm Optimization	52
3.4.2	Grey Wolf Optimizer (GWO)	58
3.4.3	Salp Swarm Optimization (SSA)	63
3.4.4	FireFly Algorithm (FA)	69
3.4.5	Artificial Bee Colony (ABC) Algorithm	73
3.4.6	Discussion	78
3.5	Conclusion	85
4	The Proposed Node Localization Methods	86
4.1	Introduction	86
4.2	Sea Lion Bio-inspired Optimization Technique	86
4.2.1	Detecting and Tracking Phase	87
4.2.2	Vocalization Phase	88
4.3	The proposed SLnO-CLA node localization	89
4.4	Localization Model	90
4.4.1	System Architecture	90
4.4.2	Distance Calculation Model	90
4.4.3	Localization Optimization Formulation	91
4.4.4	The detailed SLnO-CLA node localization	91
4.5	Experimental Results for SLnO-CLA	92
4.5.1	Evaluation Metrics	92

4.5.2	Simulation Parameters	94
4.5.3	Experimental Analysis	95
4.6	The proposed Improved SLnO (IMP-SLnO)	100
4.6.1	Population Initialization with the half OBL Strategy	100
4.6.2	Adaptive Control Parameter C'	101
4.6.3	Enhanced Exploration with Lévy Flight	102
4.6.4	Probabilistic strategy-selection mechanism	104
4.7	IMP-SLnO node localization	106
4.8	Experimental Results for IMP-SLnO	106
4.8.1	Comparison of Initial Population Diversity Between SLnO and IMP-SLnO	108
4.8.2	Experimental Analysis IMP-SLnO vs. SLnO	110
4.8.3	Experimental Analysis of IMP-SLnO vs. Multiple Algorithms	114
4.9	Conclusion	118
	Conclusion Générale	120
	Contributions	121
	Bibliography	122

List of Figures

1.1	Essential and Optional Components of a Sensor Node	13
1.2	The architecture of WSNs	14
1.3	The Five Types of Wireless Sensor Networks	15
1.4	The three fundamental WSN topologies	17
1.5	Different Wireless Sensor Networks Applications	18
1.6	Environmental applications in WSNs	19
2.1	Overview of classification metrics for WSN localization algorithms	27
2.2	(a) one-way ToA ranging technique, (b) two-way ToA ranging technique. .	31
2.3	The Principle of TDoA ranging technique.	33
2.4	Trilateration example showing intersection of three distance circles	36
2.5	Multilateration	36
2.6	Triangulation	37
2.7	The concept of the bounding box method	38
2.8	Example of the Centroid Localization Algorithm (CLA)	39
2.9	Concept of the APIT localization method.	40
2.10	The concept of the DV-Hop method	41
3.1	Classification of BIAs	51
3.2	The BIAs general process	53
4.1	Flow chart of SLnO-CLA for node localization process.	93
4.2	Comparison of localization error for each target node for CLA and SLnO-CLA.	95
4.3	Position estimated by CLA algorithm	96
4.4	Position estimated by SLnO-CLA.	97
4.5	MLE for both CLA and SLnO-CLA	98
4.6	LSR for both CLA and SLnO-CLA algorithms	98
4.7	The MLE vs number of anchors	99

4.8	The MLE vs transsmission range	100
4.9	Lévy flights tracks.	103
4.10	The IMP-SLnO algorithm's Flowchart	107
4.11	Node Localization General Process in WSNs	108
4.12	Initial population distribution comparison across four runs	109
4.13	Example of node deployment and localization results.	111
4.14	Comparison of localization error for each target node for IMP-SLnO and SLnO	112
4.15	The Mean Localization Error vs. Number of Anchors	112
4.16	The Mean Localization Error vs. different Transmission Range	113
4.17	MLE comparison across 30 runs	116
4.18	Min-Max Error Range Visualization	116
4.19	The LSR for all algorithms	117
4.20	The computation time average (CT(s)) across 30 runs	117

List of Tables

1.1	Summary of Node Components: Primary Functions and Key Elements . .	13
2.1	Comparison Between Centralized and Distributed Localization Methods . .	26
2.2	Comparison Between Range-Based and Range-Free Localization Methods .	28
2.3	Range-Based Localization Techniques	35
3.1	A Comparative Synthesis of Bio-Inspired Localization Algorithms	79
4.1	Simulation Parameters	95
4.2	Performance comparison between CLA and SLnO-CLA algorithms	97
4.3	The MLE for SLnO-CLA and CLA under different numbers of anchors . .	98
4.4	The MLE for SLnO-CLA and CLA under different Transssmission Range .	99
4.5	Localization Errors (LE) Comparison of SLnO and IMP-SLnO	110
4.6	The MLE for SLnO and IMP-SLnO under different number of anchors . .	113
4.7	The MLE for SLnO and IMP-SLnO under different Transssmission Range .	114
4.8	Algorithm-Specific Heuristic Parameters	115
4.9	Comprehensive Statistical Comparison Over 30 Independent Runs	115

List of Algorithms

1	The pseudocode of PSO	55
2	The pseudocode of (GWO)	60
3	The pseudocode of SSA	65
4	The pseudocode of FA	70
5	Artificial Bee Colony (ABC) Algorithm	75
6	Sea Lion Optimization Algorithm (SLnO)	87
7	IMP-SLnO Algorithm	105

List of Abbreviations

Acronyms / Abbreviations

ADC	Analog-to-Digital Converter
AoA	Angle of Arrival
APIT	Approximate Point-In-Triangulation
AUV	Autonomous Underwater Vehicle
CLA	Centroid Localization Algorithm
DV-Hop	Distance Vector Hop
FA	Firefly Algorithm
GPS	Global Positioning System
GWO	Grey Wolf Optimizer
IMP-SL _{NO}	Improved Sea Lion Optimization
IoT	Internet of Things
LOS	Line of Sight
LSR	Localization Success Rate
MDS	Multidimensional Scaling
MLE	Mean Localization Error
MWSNs	Multimedia Wireless Sensor Networks
NLOS	Non-Line of Sight
PSO	Particle Swarm Optimization
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
SL _{NO}	Sea Lion Optimization
SSA	Salp Swarm Algorithm
TD _{OA}	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
WSN	Wireless Sensor Network

General Introduction

Wireless Sensor Networks (WSNs) have become an essential technology in many fields, including environmental monitoring, healthcare, military operations, and disaster management. In most of these applications, knowing the exact location of sensor nodes is crucial because the value of collected data depends on knowing where it comes from. However, WSNs often contain large numbers of sensor nodes that are randomly deployed in vast or hard-to-reach areas, making it difficult to determine their exact positions.

Although GPS provides a straightforward solution for localization, equipping every sensor node with GPS is not practical. GPS modules consume too much energy, increase costs significantly, and work poorly indoors or in obstructed environments. To overcome these limitations, a common strategy is to equip only a small subset of nodes—called anchor nodes—with GPS or known positions, while the remaining nodes estimate their positions based on information received from these anchors through localization algorithms.

In recent years, researchers have increasingly treated node localization as an optimization problem. Bio-inspired algorithms, which imitate natural behaviors and work with populations of candidate solutions, have shown strong potential for solving this problem. These algorithms offer effective search capabilities, reasonable computational speed, and improved accuracy, making them well-suited for handling the challenges of WSN localization.

The main objective of this thesis is to develop effective bio-inspired localization methods that can accurately and reliably determine sensor node positions in WSNs. To achieve this objective, two node localization methods based on the SLnO algorithm are proposed for accurate and stable range-based localization in WSNs. SLnO was selected for three main reasons. First, it is a relatively recent metaheuristic optimizer characterized by a clear biological inspiration and a simple mathematical structure, making it suitable for targeted modifications and hybridization strategies. Second, unlike well-established algorithms such as GWO and PSO, which have been extensively investigated in WSN localization, SLnO has received comparatively limited attention in this domain, representing

a promising research opportunity. Third, although SLnO has demonstrated competitive performance on several benchmark optimization problems, preliminary experiments conducted in this work indicated that its localization accuracy in WSN environments could still be improved, particularly when compared with strong optimizers such as GWO.

The first proposed method, SLnO-CLA, is a hybrid localization approach that combines the Centroid Localization Algorithm (CLA) with the SLnO algorithm to improve localization accuracy. The second proposed method, IMP-SLnO, is an enhanced version of SLnO that incorporates several improvement strategies to increase population diversity, strengthen exploration capability, and accelerate convergence. These enhancements make the proposed algorithm more effective and stable for estimating node positions under various network conditions.

To evaluate our proposed methods, we conducted comprehensive comparisons with existing algorithms, including the classical CLA and several well-known bio-inspired optimization techniques. Performance was assessed using standard metrics such as average localization error, success rate, and computation time, with all algorithms tested under identical network conditions to ensure fair comparison.

This thesis is organized into four chapters. Chapter 1 provides an overview of Wireless Sensor Networks, including their architecture, applications, and main challenges. Chapter 2 presents the localization problem in WSNs and reviews existing localization techniques. Chapter 3 discusses optimization approaches, with particular focus on bio-inspired algorithms and their application to localization problems. Chapter 4 describes our proposed methods, presents the experimental results, and provides detailed performance comparisons. Finally, we conclude with a summary of our contributions and suggestions for future research directions.

Chapter 1

Wireless Sensor Networks: An Overview

1.1 Introduction

Wireless sensor networks (WSNs) have emerged as an essential technology for surveillance and data collection in diverse, complex, and dynamic environments. These networks are made up of specially dispensed sensor nodes that collaboratively sense, process, and transmit data to a central unit or base station. This chapter aims to present an introduction to sensor nodes and provide a comprehensive overview of WSNs, covering their types, topologies, applications, and the challenges they encounter.

1.2 Sensor Node

A sensor node is a compact, autonomous device designed to monitor various environmental or physical conditions, such as temperature, pressure, and motion. It collects these data and processes them locally before wirelessly transmitting them within the network [1–4]. Generally, a sensor node consists of a sensing unit, a processing unit, a communication module (transceiver), and a power supply, representing the essential components of its architecture [5, 6].

1. **Sensing unit:** This unit is responsible for capturing physical or environmental phenomena. It comprises a sensor to detect these phenomena and an analog-to-digital converter (ADC) that transforms the collected analog signals into digital data [2].
2. **Processing unit:** The processing unit functions as the central component of the

sensor node, responsible for data processing and enabling cooperation among sensor nodes in the network to accomplish designated tasks. It often includes a micro-controller and a memory unit for local data storage, which helps reduce the size of transmitted messages and conserve energy.

3. **Communication unit (transceiver):** This unit is responsible for handling the transmission and reception of data between nodes within the network. It generally comprises a radio transceiver that enables wireless communication, allowing nodes to interact efficiently within the network.
4. **Power supply:** It provides energy to all components of the sensor node, typically through a non-rechargeable or a rechargeable battery. Efficient energy management is essential for extending the node's operational lifespan, particularly in environments where battery replacement is impractical.

In addition to these essential components, optional components such as localization systems, mobility mechanisms, and energy harvesting devices can be integrated into sensor nodes to further improve their functionality, introduce supplementary features, and optimize their performance according to the application's specific requirements. Localization systems provide positional data, mobility mechanisms offer node movement for necessary positioning, and energy harvesters convert environmental energy into usable electric power.

These components collaborate to enable nodes to perform monitoring, processing, and communication functions effectively within the limitations of restricted battery life, memory capacity, and computation resources. Figure 1.1 illustrates the architecture of a sensor node, showcasing both essential and optional components, whereas Table 1.1 provides a structured summary of each component, including the primary functions and key components.

1.3 Wireless Sensor Networks

1.3.1 Definition

A WSN is a system of wirelessly connected sensor nodes that cooperate to monitor and relay data on environmental and physical parameters [8, 9]. The architecture of a WSN is depicted in figure 1.2, wherein data gathered by sensor nodes is transmitted either directly to the base station (sink node) or through intermediary nodes, ensuring efficient coverage even in extensive areas [10]. The base station serves as the central hub, consolidating and

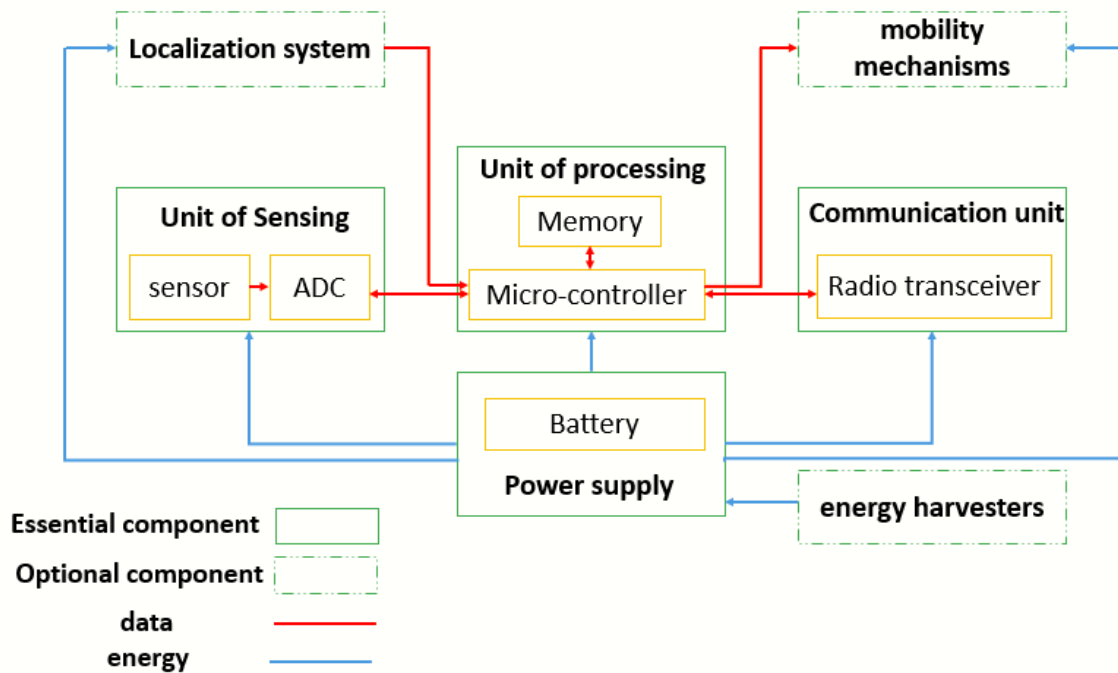


Figure 1.1: Essential and Optional Components of a Sensor Node [7]

Table 1.1: Summary of Node Components: Primary Functions and Key Elements [1–6]

Nodes components	Primary function	Key Components
Sensing unit	Collects data from the environments	Sensor, ADC
Processing unit	Data computation, algorithms execution, decision making	Microprocessor, Memory unit (RAM, ROM)
Communication unit	Transmission and reception of data	Radio transceiver
Power supply	Managing energy use	Batteries, power managements circuit
Optional components	Enhance functionality, adds extra features	GPS positioning, energy harvesters, dynamic movement for improved coverage

processing the incoming data. It also functions as the communication interface between the sensor network and the user, utilizing satellite or internet connectivity. This structure allows WSN to operate autonomously, requiring less human involvement [11].

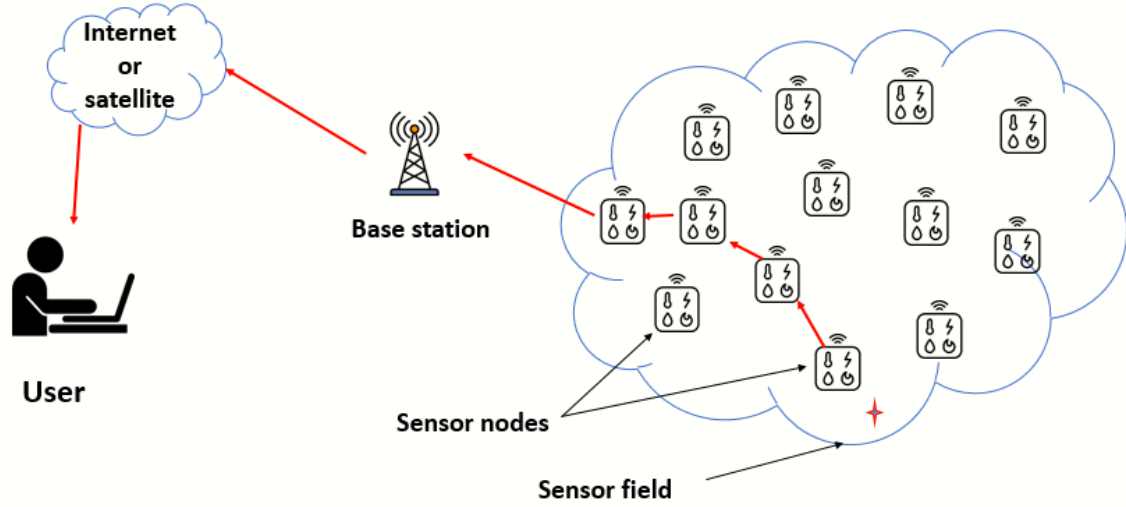


Figure 1.2: The architecture of WSNs [12]

1.3.2 Node Deployment in WSN

Deployment in WSN refers to the placement of sensor nodes within a designated area to enable efficient monitoring and data acquisition [13]. The deployment strategy is primarily contingent upon the application's requirements and the prevailing environmental conditions [14]. Two primary deployment strategies exist: deterministic and random [15]

- **Deterministic deployment.** In this strategy, sensor nodes are installed at fixed and predefined locations. This approach simplifies localization and helps achieve good area coverage. It is particularly suitable for scenarios where manual placement is possible and precise coverage is critical.
- **Random deployment.** In contrast, random deployment involves distributing sensor nodes without a fixed pattern, often using techniques such as aerial drops. This strategy is cost-effective and well-suited for inaccessible or hard-to-reach areas. However, achieving complete coverage often requires deploying a higher density of nodes to compensate for uneven distribution [16].

1.3.3 Types of WSNs

WSNs are categorized based on their deployment environments, which influence their design and operational challenges [7, 17–20]. Figure 1.3 presents the five types of WSNs.

Terrestrial WSNs Terrestrial WSNs consist of sensor nodes deployed on land, either systematically or randomly. Nodes are powered by limited batteries, often supplemented

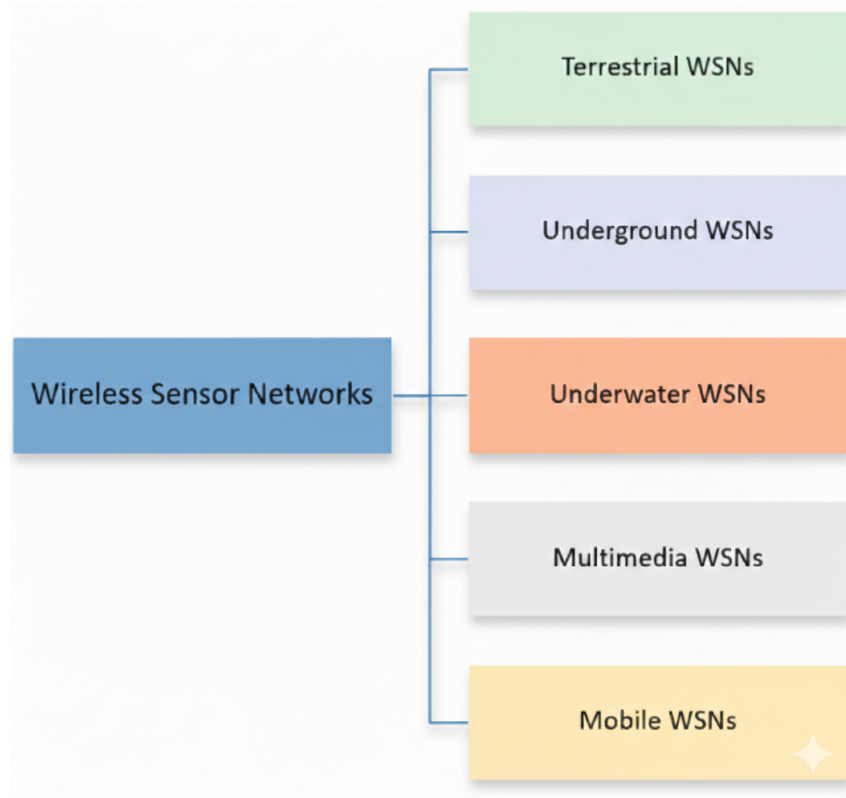


Figure 1.3: the Five Types of Wireless Sensor Networks [17]

with renewable energy sources such as solar panels to overcome the constraints of non-rechargeable batteries.

Underground WSNs In underground WSNs, sensors are placed beneath the surface to monitor soil, minerals, or underground activities. These networks require above-ground sinks to facilitate data transmission. Underground WSNs face significant challenges in wireless communication due to the high signal attenuation and loss caused by geological materials such as soil and rocks. The inaccessibility of buried nodes makes battery recharging nearly impossible, making energy-efficient communication protocols essential for maintaining network longevity and reliable data transmission.

Underwater WSNs Underwater WSNs involve deploying sensor nodes beneath water surfaces, often in deep ocean environments. These networks face unique challenges due to the complexities of underwater operations. The high cost of underwater devices limits node deployment, and data collection often relies on autonomous underwater vehicles (AUVs). Communication is conducted using acoustic waves, which suffer from limited bandwidth, long propagation delays, high latency, and signal degradation. Additionally, underwater sensors typically have limited, non-rechargeable power sources, highlighting

the need for energy-efficient communication techniques.

Multimedia WSNs Multimedia WSNs (MWSNs) are composed of cost-effective sensor nodes equipped with audio, video, and imaging capabilities to monitor and record multimedia data. These nodes are often deployed in a structured manner to ensure optimal coverage and data acquisition. Despite their advantages, MWSNs encounter challenges such as high energy consumption, data compression requirements, and quality of service (QoS) considerations for reliable multimedia delivery.

Mobile WSNs Mobile WSNs feature sensor nodes capable of movement, allowing them to navigate and interact dynamically with their surroundings. These movable nodes can autonomously adjust their positions and reconfigure the network topology while performing sensing, processing, and communication tasks. However, mobile WSNs face challenges including effective deployment, mobility control, accurate localization of mobile nodes, movement regulation, ensuring adequate sensing coverage, minimizing energy usage during movement, maintaining network connectivity, and efficient data distribution.

1.3.4 Topologies in WSNs

WSNs are commonly structured using three key topologies [21–24]: centralized, distributed, and decentralized. Figure 1.4 illustrates the structural design of these topologies.

Centralized topology in WSNs In a centralized topology, all nodes communicate directly with a single central unit, commonly known as the base station or fusion center, which processes and aggregates data received from all nodes. This topology simplifies the network by centralizing control and data management and facilitates the identification of data errors. However, all nodes must remain within the communication range of the base station, which limits scalability. Additionally, continuous data transmission to the central unit may cause excessive energy consumption, especially when much of the transmitted data is redundant [26].

Distributed topology in WSNs In a distributed topology, all nodes operate equally, performing computations and sharing data with neighboring sensors within their communication range. By distributing information processing across the network rather than relying on a single unit, the system achieves greater resilience to failures.

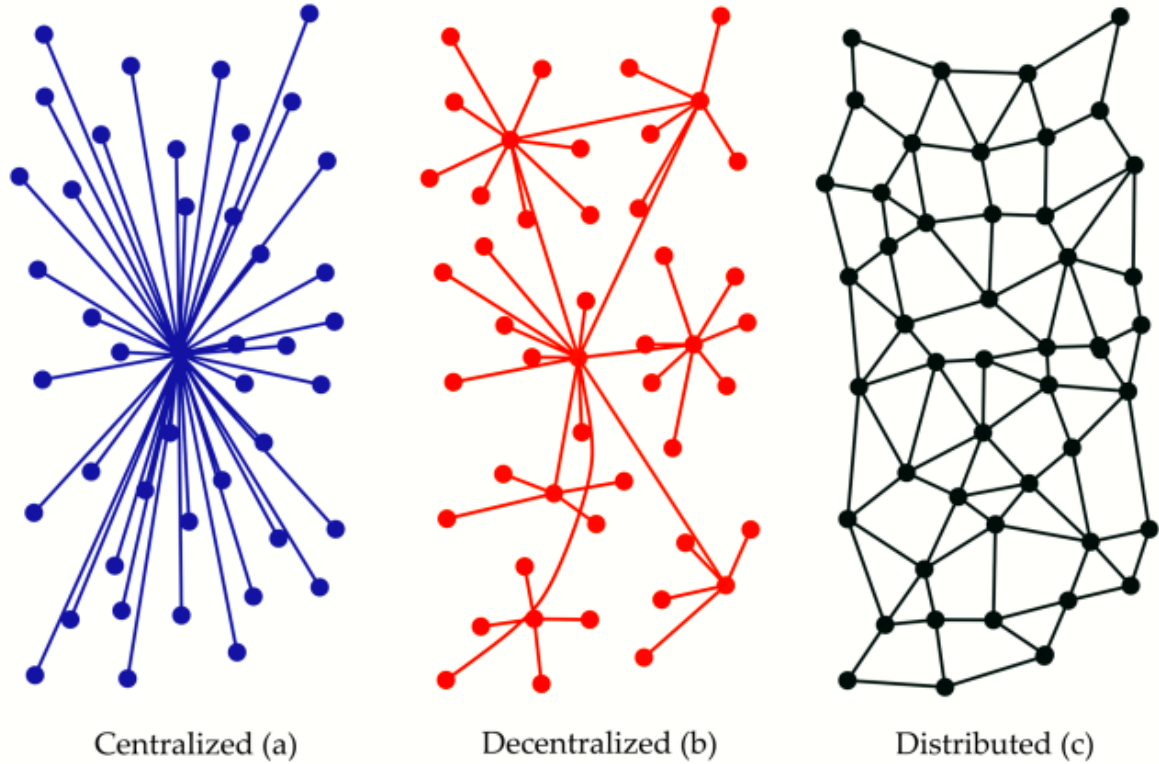


Figure 1.4: The three fundamental WSN topologies [25]

Decentralized topology in WSNs The decentralized, or cluster-based topology, combines features of both centralized and distributed structures. The network is divided into clusters, with each cluster managed by its own fusion center. Sensors within a cluster share data locally, which is processed by the cluster’s central unit. The outcomes from all central units are then consolidated to form the final decision [23].

1.3.5 Applications in Wireless Sensor Networks

WSNs are increasingly utilized in various fields due to their ability to collect and transmit real-time data. The flexibility of WSNs allows them to tackle field-specific challenges, harnessing their unique features to enhance processes and outcomes. Below, we highlight some of the main applications that demonstrate the practical uses of WSNs, while Figure 1.5 represents these different applications.

- **Environmental applications**

WSNs play a vital role in environmental applications by monitoring various conditions such as air quality, soil moisture, water pH, and greenhouse gases. These networks are also instrumental in tracking animal movements, detecting natural disasters like floods, fires, and earthquakes, and supporting agricultural practices through the observation of plant

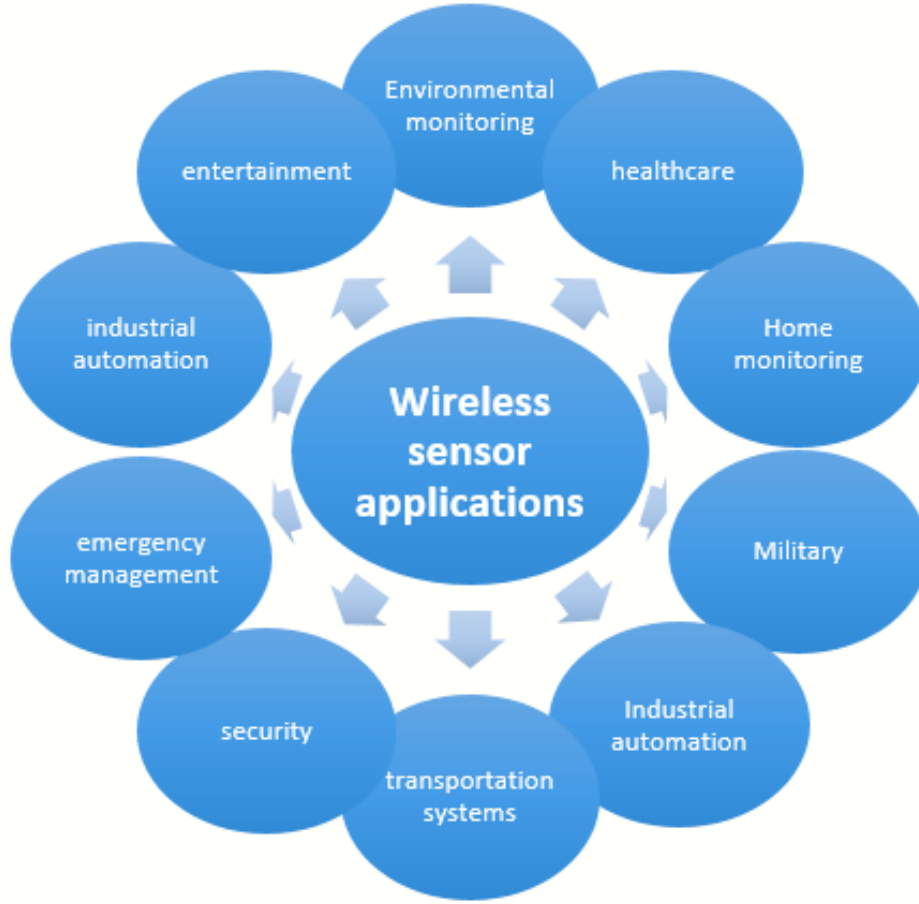


Figure 1.5: Different Wireless Sensor Networks Applications

growth, irrigation, and nutrient levels [20]. Furthermore, WSNs are utilized in traffic management, weather forecasting, and pollution monitoring, providing real-time data that enhance decision-making and environmental management. By enabling efficient, remote data collection, WSNs contribute significantly to the understanding of ecological processes and the development of effective hazard response systems. Figure 1.6 depicts several environmental applications of these networks.

- **Health applications**

WSNs have transformed healthcare by enabling advanced patient monitoring and diagnostics. These networks are used in wearable devices to track body movements, physiological data, and overall health, as well as in implanted sensors for gathering critical health information. They enable real-time telemonitoring of vital signs, facilitate precise drug administration, and provide diagnostic support. WSNs are particularly effective in monitoring patients with chronic conditions, disabilities, or diseases like Alzheimer's. By seamlessly transmitting data to healthcare providers, WSNs enhance care quality, enable timely interventions, and support better health management across hospitals, homes, and



Figure 1.6: Some of environmental applications of wireless sensor networks

emergency scenarios.

- **Military applications**

WSNs have become indispensable in military operations, providing real-time capabilities for battlefield surveillance, intrusion detection, and monitoring critical parameters [27]. These networks enable the detection of enemy movements, assessment of battlefield damage, and tracking of friendly forces [3, 28], enhancing situational awareness and operational efficiency. Additionally, WSNs play a crucial role in identifying and responding to threats such as Nuclear, Biological, and Chemical (NBC) attacks, as well as monitoring sensitive areas, including borders and landmine fields [29, 30]. Their deployment facilitates robust intelligence gathering and supports strategic decision-making in complex and dynamic scenarios.

- **Home applications**

WSNs serve a pivotal role in creating intelligent and interconnected home environments. By embedding sensors into household devices such as refrigerators, air conditioners, and lighting systems, WSNs enable seamless communication and control, allowing both local and remote management by homeowners. These networks support functionalities such as automated temperature regulation, motion-activated lighting, and advanced security monitoring. Additionally, WSNs monitor device performance, detect anomalies, and provide timely alerts for potential hazards [3, 31, 32].

- **Other applications**

In addition to the fields mentioned above, these networks are also utilized in numerous other areas, including transportation systems, security, emergency management, indus-

trial automation, and entertainment such as theater production [33, 34].

1.3.6 WSNs challenges

The advancement of WSNs is hindered by various challenges [7, 12, 14, 18, 28, 34–36], including the following:

Energy constraint WSNs operate under strict energy limitations, as sensor nodes are typically equipped with finite power sources, primarily batteries, which are often irreplaceable or non-rechargeable in remote locations such as forests, deserts, and military zones. The operational lifespan of a sensor node is directly dependent on its battery life, making efficient power management a critical concern.

Quality of service (QoS) requirements QoS in WSNs refers to the network’s capability to deliver accurate and timely information from the sensor node to the base station. The QoS requirements vary across applications, encompassing factors such as data accuracy, transmission delay, and reliability. However, achieving QoS in WSNs is challenging due to the inherent constraints of sensors, including limited energy, processing, and communication capabilities.

Scalability Scalability is a crucial challenge in WSN design, as these networks can consist of thousands of sensor nodes deployed across various environments. As the network grows in size, the performance should scale proportionally without significant degradation. This requires the development of efficient algorithms and protocols that can handle increasing communication costs and resource constraints while ensuring reliable and effective operation.

Dynamical topologies and challenging environmental conditions WSNs are prone to dynamic topology challenges due to sensor node failure, which may arise from hardware malfunctions, energy depletion, or physical damage. These networks are often deployed in challenging and remote environments such as volcanic regions, underwater locations, battlefields, and areas with hazardous biological or chemical exposure. Harsh environmental factors, including extreme temperatures, high humidity, vibrations, and dust, can further impact node performance and network stability. To ensure reliable operation, sensor nodes must be designed to endure such demanding conditions and adapt effectively to changes in network topology.

Security WSNs are susceptible to a variety of security threats, which can broadly be classified into passive and active attacks. Passive attacks involve eavesdropping on communication, such as traffic analysis or content interception. Active attacks, on the other hand, include actions such as data modification, fabrication, or disruption, including node capture or routing attacks. Given that many WSNs are employed in sensitive applications like military or surveillance systems, ensuring secure communication is critical to protect against unauthorized access and denial-of-service attacks. The inherent limitations of WSNs, including low energy and computational resources, further complicate the implementation of robust security measures. Therefore, designing effective security protocols to ensure the confidentiality, integrity, and availability of data is essential for the reliable operation of these networks.

Localization In many WSN applications, the value of collected data depends heavily on knowing the precise location where the event or physical quantity was detected. Without location information, the sensed data lacks context and relevance. Localization is therefore a critical component in WSNs, as it ensures that the data can be accurately interpreted and applied in the decision-making process. This will be explored in greater detail in Chapter 2, where we delve into various methods and techniques for accurate positioning in these networks.

Cost production In WSNs, a large number of nodes are typically deployed, making cost a critical factor in their implementation. If the production cost per node is too high, the network may become financially impractical compared to traditional sensing solutions, making cost management a key consideration in WSN design.

Data compression Reducing the size of data before transmission can significantly decrease communication time and conserve network resources. Incorporating data compression or filtering techniques at the node level optimizes performance by lowering the volume and frequency of transmitted data.

1.4 Conclusion

This chapter provided an overview of WSNs, emphasizing their significance in diverse applications necessitating monitoring and data acquisition in complex environments. We discussed the key aspects of WSNs, including the deployment of sensor nodes, which are often placed randomly in many applications. Given the random deployment, the need for an effective localization system becomes clear to ensure accurate data interpretation

and optimal network operation. This review chapter sets the stage for the next chapter, which delves into the concept of localization in WSNs, its importance, and the techniques employed to achieve precise positioning of sensor nodes.

Chapter 2

Localization Techniques in WSN

2.1 Introduction

In many WSN applications, it is not enough to simply collect data; the values of that data are often closely linked to the location where it was gathered. Whether it's monitoring environmental changes, tracking movements, or detecting critical events, knowing where something happens is as important as what is happening. For instance, in fire detection scenarios, when a sensor detects a fire and sends a message to the base station, that message must include the location of the detecting node. Without knowing where the event occurred, the information becomes practically useless. Consequently, localization is crucial for the efficient operation of WSNs and the fulfillment of their intended function.

In this chapter, the localization challenge in WSNs is described, along with the key features and limitations. Several methods used to estimate node position are summarized, followed by the factors used to evaluate their performance.

2.2 Localization Fundamentals

2.2.1 Positioning of Nodes

The localization, or positioning, problem refers to the task of estimating the locations of nodes that have been randomly deployed, without any prior knowledge of their positions, in a target area. At first glance, this might seem like a simple task, especially in small, controlled environments. However, real-world deployments often occur over vast or difficult terrains, where nodes are dropped from the air over forests, mountains, or disaster zones. In such cases, determining where each node ends up is very difficult, and it becomes a serious challenge in WSN.

2.2.2 GPS in WSN localization

A straightforward solution to the node localization challenge in WSN is to equip each sensor with a GPS module [37]. While GPS has long been used as a standard technique for position estimation in various wireless systems [38], its application in WSNs is often impractical due to several limitations. First, GPS units are relatively expensive, and the incorporation of such a module into every node would make the overall deployment significantly costly, which runs counter to the low-cost nature of WSNs [39, 40]. Second, GPS devices consume substantial power, which can drastically shorten the operational lifetime of battery-powered nodes. Third, the performance of GPS decreases in situations with weak satellite signals or a complete lack of them, e.g., indoors, in dense forests, or underground [41, 42]. Such limitations demonstrate the need for more scalable, cost-effective, and energy-efficient alternative localization techniques.

2.2.3 System Model (Anchor Target Nodes)

In WSNs, nodes are generally categorized according to their knowledge of location information:

Anchor Nodes (A)

Also known as reference, beacon, or landmark nodes, these nodes have prior knowledge of their exact coordinates. This information is typically obtained through manual placement or by equipping the nodes with GPS devices.

Target Nodes (T)

Occasionally referred to as unknown, dumb, or free nodes, these lack any prior knowledge of their locations, which must be estimated through a localization process.

Consider a WSN deployed in a two-dimensional plane. The network is represented by a graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of all sensor nodes, and E is the set of edges between nodes that can directly communicate with each other.

An edge $e = (v_i, v_j) \in E$ exists if and only if the Euclidean distance between v_i and v_j is less than or equal to a predefined communication range r . This condition indicates that the nodes are neighbors:

$$(v_i, v_j) \in E \iff \text{distance}(v_i, v_j) \leq r$$

If the distance between two nodes exceeds r , no direct connection is established between them, and they are not considered neighbors.

Each node v_i has coordinates $(x_i, y_i) \in \mathbb{R}^2$. Let $A \subset V$ denote the set of anchor nodes, with known positions (x_a, y_a) , and let $T = V \setminus A$ denote the set of target (unknown) nodes, whose positions (x_t, y_t) are to be estimated.

The aim of the localization task is to accurately estimate the coordinates (x_t, y_t) for as many target nodes $t \in T$ as possible.

2.3 Classification of WSN Localization Algorithms

Localization algorithms in WSNs are categorized based on several key aspects. These include the presence or absence of anchor nodes, the location where the position calculations are performed, the structure and mobility of the network, and the hardware requirements [37, 39, 41, 43–46]. Figure 2.1 provides an overview of the key metrics commonly used to classify and distinguish localization algorithms in WSNs.

2.3.1 According to the Calculation Location

There exist two main categories based on where the position estimation is carried out: centralized and distributed methods.

Centralized Localization

In centralized approaches, sensor nodes report their measurements, such as distances to nearby nodes, to a base station. This central unit collects the data from all nodes and calculates their positions across the entire system before sending the estimated positions back to the nodes[47].

Distributed Localization

In distributed approaches, each node estimates its own position independently. This method relies solely on local information obtained from nearby anchor nodes or neighboring nodes, without involving a central authority.

Table 2.1 summarizes the main differences between centralized and distributed localization methods. Centralized algorithms often achieve higher accuracy in small networks due to their ability to perform comprehensive data analysis and global optimization. However, they suffer from limited scalability and high energy costs. In large networks, data

loss caused by multi-hop communication to the sink node can further degrade localization accuracy. On the other hand, distributed algorithms have higher scalability and energy efficiency but tend to have lower accuracy and require more complex coordination mechanisms [48, 49].

Table 2.1: Comparison Between Centralized and Distributed Localization Methods [47–49]

Feature	Centralized Localization	Distributed Localization
Energy and Complexity	Higher	Lower
Scalability	Limited, not suitable for large networks	Highly scalable, suitable for large networks
Accuracy	High, due to global data availability	Lower
Reliability	Prone to central point failures or disruptions in multi-hop data flow	Resilient to individual node failures and packet loss
Design Complexity	Simpler algorithm design	More complex
Suitability	Small to medium networks with sufficient resources	Large-scale, resource-constrained WSNs

2.3.2 Mobility-Based Localization Algorithms

The majority of existing localization approaches are designed for static WSNs, where both target and anchor nodes remain in fixed positions. However, the rise of applications such as tracking moving objects or monitoring dynamic environments necessitates new types of localization methods that can handle mobility.

Mobility introduces additional challenges in position estimation due to the constant changes in node placement. At the same time, it offers advantages by enabling multiple measurements during movement, thereby enhancing localization accuracy. Localization methods that address mobility in WSNs are generally classified into three categories:

- Mobile target nodes with fixed anchor nodes
- Fixed target nodes with mobile anchor nodes
- Both mobile target and anchor nodes

2.3.3 According to Utilization of Extra Hardware

Considering the hardware requirements and the techniques applied to estimate measurements between nodes, localization algorithms are commonly classified into **range-based**

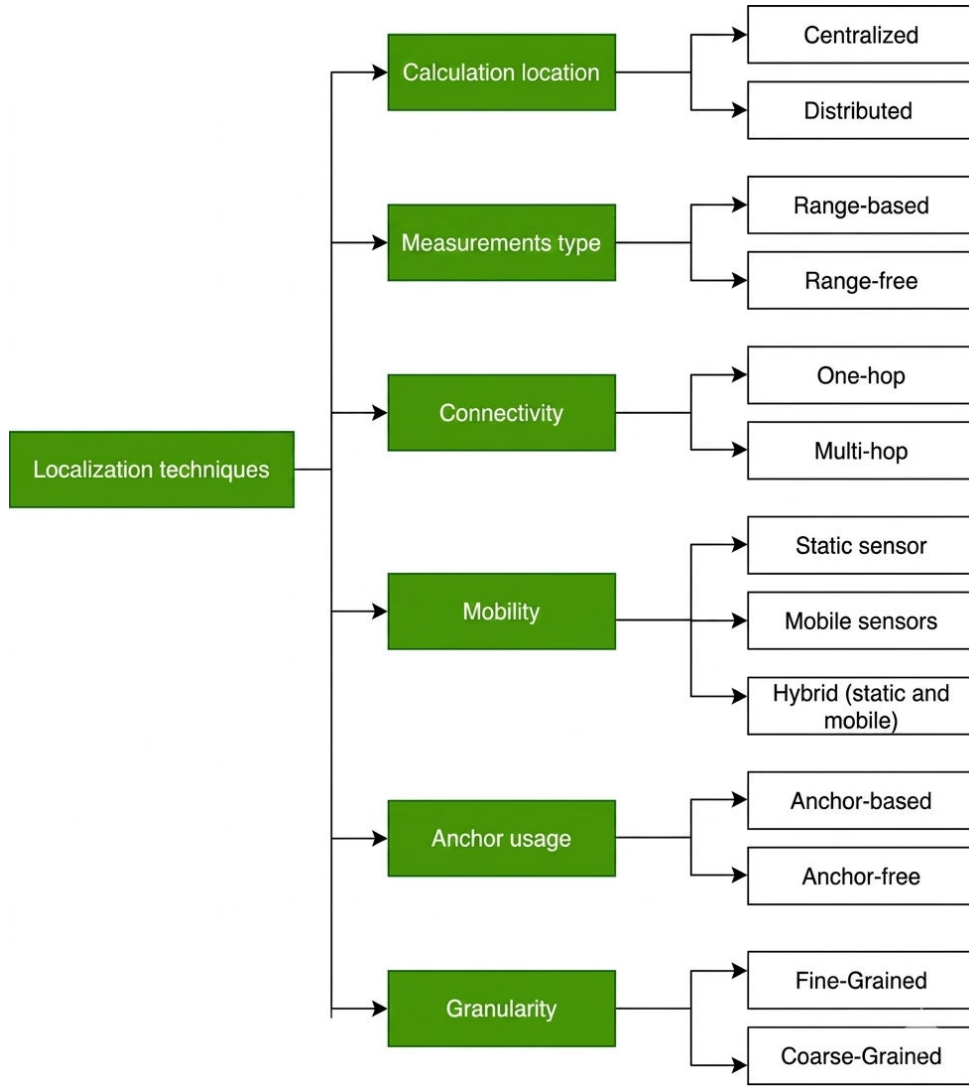


Figure 2.1: Overview of classification metrics for WSN localization algorithms [43]

and **range-free** methods. Table 2.2 summarizes the key differences between these two classes.

Range-Based Methods

Range-based localization methods estimate node positions by measuring either the distance or the angle between sensor nodes. These measurements are typically obtained using techniques such as Time of Arrival (TOA) [50], Time Difference of Arrival (TDOA) [51], and Received Signal Strength (RSS) [52]. While these methods offer high accuracy, they require additional hardware, which leads to increased computational and energy costs.

Range-Free Methods

Range-free localization methods determine the positions of target nodes using connectivity information and hop counts rather than relying on precise distance or angle measurements. These techniques are easier to implement, do not require extra hardware, and impose lower energy and computational demands. However, they typically offer lower localization accuracy compared to range-based methods. Common examples of range-free techniques include the Centroid [53] and DV-Hop [54] algorithms.

Table 2.2 summarizes the key differences between range-based and range-free localization methods across several important aspects. The choice between these two approaches depends on the application requirements and the available resources of the sensor nodes.

Table 2.2: Comparison Between Range-Based and Range-Free Localization Methods[50–54]

Aspect	Range-Based	Range-Free
Measurement	Physical distance/angle (ToA, AoA, RSS)	Connectivity and hop counts
Hardware	Additional ranging hardware needed	Minimal
Accuracy	High (fine-grained)	Low to medium (coarse-grained)
Energy and Complexity	Higher	Lower
Scalability	Limited by hardware	More scalable for large networks
Robustness	Sensitive to environmental factors	More robust

2.3.4 According to the Connectivity

Based on how unknown nodes connect with anchors, localization algorithms can also be divided into one-hop and multi-hop methods.

One-Hop Methods

In one-hop localization methods, each unknown node must have direct connectivity with multiple anchor nodes. It then estimates its position using the known coordinates of these anchors, as in the Centroid method.

Multi-Hop Methods

In contrast, multi-hop methods do not require unknown nodes to be in direct contact with anchors. Instead, they receive information indirectly through intermediate nodes and use this data to estimate their positions, as in the DV-Hop algorithm. Although multi-hop methods are more scalable and require fewer anchors, they often suffer from reduced accuracy due to the accumulation of errors across multiple hops.

2.3.5 According to the Granularity of Position Estimation

Localization techniques are commonly classified into coarse-grained or fine-grained approaches.

Coarse-Grained Localization

Coarse-grained localization methods aim to provide approximate position estimates with limited accuracy, often sufficient for applications where exact coordinates are not essential.

Fine-Grained Localization

Conversely, fine-grained localization techniques aim to achieve high positional precision by computing exact coordinates. This degree of precision typically requires increased communication overhead, computational complexity, and reliance on detailed measurements. Fine-grained methods often employ sophisticated algorithms to convert such measurements into accurate position coordinates.

2.3.6 Anchor-Based vs. Anchor-Free Localization

Localization techniques can also be broadly classified according to the availability of anchor nodes, resulting in anchor-based and anchor-free approaches.

Anchor-Based Localization

Anchor-based schemes utilize anchor nodes as spatial references to aid in localizing unknown nodes within the network. The precision of such methods is significantly influenced by both the number of deployed anchors and their spatial distribution.

Anchor-Free Localization

In contrast, anchor-free schemes function without relying on any anchor nodes. These methods help reduce deployment costs and are particularly advantageous in environments

where GPS signals are inaccessible or unreliable.

2.4 Localization Process: Main Phases and Methodologies in WSNs

The process of determining the accurate geographical locations of sensors in WSN is generally divided into sequential phases, each with its own specific goals and techniques. In this section, we focus on the two primary phases: Distance Estimation and Position Estimation, and discuss the methods commonly used in each.

2.4.1 Distance Estimation Phase

The first step in many localization algorithms, especially range-based ones, is to estimate the distance between unknown and anchor nodes. This is often the most important step. The accuracy of the final position coordinates relies significantly on how well the distances between the nodes are estimated. Several methods can be applied to compute such distances, and the most commonly used techniques include RSSI, ToA, TDoA, and AoA [55].

2.4.1.1 Time of Arrival

Time of Arrival (ToA), or Time of Flight (ToF), is a commonly investigated approach to estimating the distance between two wireless sensor devices based on precise signal timing. The basic rule behind ToA is straightforward: if the exact time at which a signal is sent is known, and the exact arrival time at the receiver is also known, then the signal's propagation time can be calculated [56]. Since the propagation speed is known—for example, a radio frequency signal travels at approximately 3×10^8 meters per second, while an acoustic signal propagates at about 343 meters per second—the travel time can then be directly converted into a distance estimate [44].

In real-life situations, the ToA measurements can be performed in two ways [42] as illustrated in figure 2.2 :

- **One-way ToA** calculates the difference between the transmission time of a signal and its arrival time at the receiver. This method requires the clocks of both the transmitter and the receiver to be highly synchronized, normally to within microseconds, to achieve a good distance reading.

- **Two-way ToA**, also known as round-trip time determination, relaxes the strict synchronization requirement by having the signal travel from the transmitter to the receiver and then back again. By measuring the total round-trip time and accounting for processing delays, the distance can be calculated.

Mathematically, the distance between two nodes, i and j , for one-way ToA is given by:

$$d_{ij} = \tau \cdot v \quad (2.1)$$

Where $\tau = t_2 - t_1$, with t_1 and t_2 being the transmission and reception times, respectively, and v is the signal propagation speed.

For two-way ToA, the distance is determined by:

$$d_{ij} = \frac{\tau_1 - \tau_2}{2} \cdot v \quad (2.2)$$

Where $\tau_1 = t_4 - t_1$ and $\tau_2 = t_3 - t_2$.

While ToA can be highly accurate under ideal conditions, its demand for precise synchronization and specialized hardware can make it less suitable for large-scale, low-cost deployments or for networks with mobile nodes.

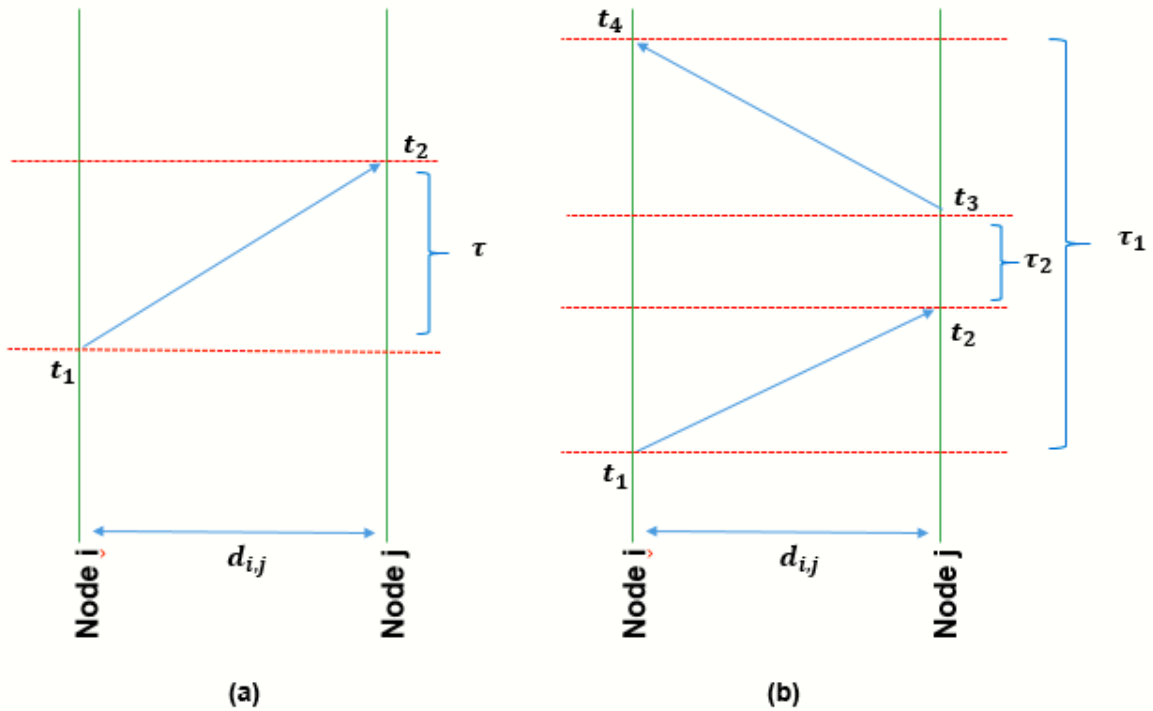


Figure 2.2: (a) one-way ToA ranging technique, (b) two-way ToA ranging technique.

2.4.1.2 Time Difference of Arrival

Time Difference of Arrival (TDoA) ranging method estimates distance between nodes by comparing two different signals' time of arrivals that propagate at different speeds, usually a radio wave and an acoustic (ultrasonic) wave. Unlike ToA, it does not require tightly synchronized clocks between the transmitter and the receiver, because the same receiver's local clock measures the arrival times of both signals. Typically, an anchor node transmits the radio and acoustic signals simultaneously or nearly simultaneously (see figure 2.3). The receiving node then measures the difference in arrival times of these two signals [57] and estimates distance using this time difference and the fixed delay [44] caused by hardware or processing, as follows:

$$d_{ij} = (T_{\text{acoustic}} - T_{\text{radio}} - t_{\text{delay}}) \cdot v \quad (2.3)$$

Or more generally:

$$d_{ij} = \tau \cdot v \quad \text{where } \tau = T_{\text{acoustic}} - T_{\text{radio}} - t_{\text{delay}} \quad (2.4)$$

Where,

- T_{acoustic} is the time when the acoustic signal arrives,
- T_{radio} is the time when the radio signal arrives,
- t_{delay} is the known fixed delay between the transmission of the two signals due to hardware or system processing (often zero or negligible if signals are sent simultaneously),
- v is the speed of sound in the medium.

TDoA positioning can provide better ranging accuracy for shorter distances but needs extra hardware such as speakers and microphones. It is also sensitive to environmental conditions, such as temperature and humidity, which affect the speed of sound [58].

2.4.1.3 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) is a cost-effective [59] and commonly utilized technique for calculating the distance between sensors in WSNs. It measures the power level of the received radio signal, which decreases as the distance between the transmitter and the receiver increases due to signal attenuation. In other words, when a sensor node transmits a signal, the strength of that signal decreases as it travels through space because

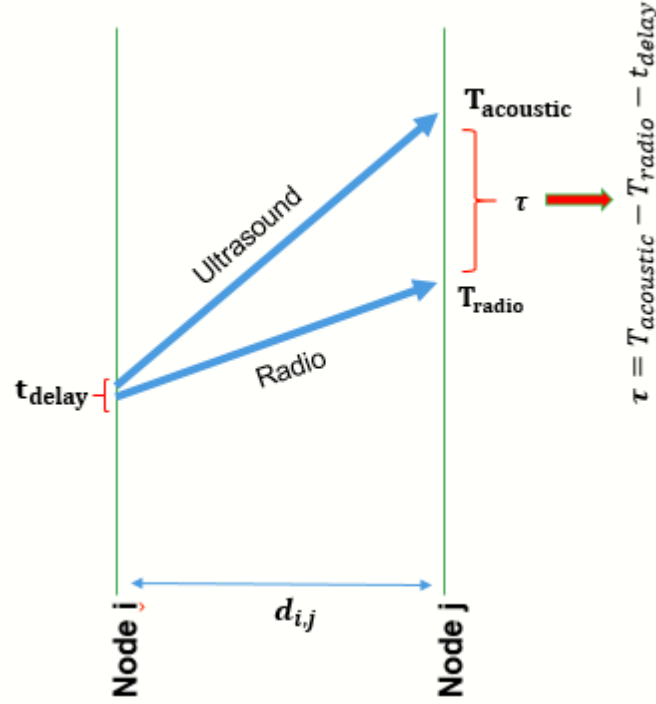


Figure 2.3: The Principle of TDoA ranging technique.

of path loss. Therefore, it's possible to estimate how far two nodes are from each other by measuring how weak a signal is when it reaches the receiver node. This distance is typically modeled using a path loss model [60, 61]. One widely used model is the log-distance path loss model, which can be expressed as:

$$PL(d) = PL(d_0) + 10n \log \left(\frac{d}{d_0} \right) + X_\sigma \quad (2.5)$$

Where,

- $PL(d)$ is the path loss at distance d (in dB),
- $PL(d_0)$ is the path loss at a known reference distance d_0 ,
- n is the path loss exponent that depends on the environment,
- X_σ is a random variable representing shadowing effects,
- d is the distance between the transmitter and the receiver.

Since RSSI is the received power level, it is directly related to this path loss as:

$$RSSI(d) = P_t - PL(d) \quad (2.6)$$

Where P_t is the known transmit power.

By rearranging this model and ignoring shadowing for simplicity, the distance d between two nodes can be estimated as:

$$d = d_0 \cdot 10^{\left(\frac{P_t - \text{RSSI}(d) - PL(d_0)}{10n}\right)} \quad (2.7)$$

However, RSSI is very sensitive to environmental factors such as obstacles and interference, which can cause large variations in measurements for the same distance. Another challenge is that the assumption of uniform signal power decay often does not hold true in real-world scenarios. Consequently, accurately estimating the path loss parameters, such as the path loss exponent, becomes difficult in environments where signal propagation is non-uniform.

2.4.1.4 Angle of Arrival

Instead of measuring distance like previous approaches, Angle of Arrival (AoA), which is a direction-based method, works by detecting the angle from which a signal reaches a sensor with a known position. This technique requires anchors to have antenna arrays [62] that are capable of detecting the direction from which the signal comes relative to a known reference, like true north or a local axis [63]. By combining angle measurements from at least two or three anchors, the unknown node's position can be computed using geometric principles, typically through triangulation.

Though the results of the AoA-based method can be extremely precise [64], the method depends on specialized hardware like antenna arrays, which increases cost and design complexity. It is also subject to reflections, signal scattering, or non-line-of-sight conditions, which can distort the true arrival angle.

A comparative summary of these ranging techniques, including their principles, advantages, disadvantages, accuracy levels, and typical use cases, is provided in Table 2.3, adapted from [55, 58, 62, 65–69]

2.4.2 Position Estimation

After obtaining the approximate distance from the previous phase, the position estimation phase uses these estimates along with the known locations of anchors to calculate the precise location of nodes with unknown positions.

Table 2.3: Range-Based Localization Techniques [55, 58, 62, 65–69]

Localization Technique	Principle	Advantages	Disadvantages	Accuracy	Typical Use Cases
RSSI	Distance estimation based on signal power attenuation	Low cost, low energy, no extra hardware	Environmentally sensitive, low accuracy, needs calibration	Low to medium	Large-scale, low-cost WSNs
ToA	Measures signal travel time from sender to receiver	High accuracy in LOS	Requires precise synchronization, complex hardware	High	LOS environments, industrial settings
TDoA	Measures time difference of arrival of two signals (radio/acoustic)	No transmitter clock sync needed	Needs additional hardware (speakers/mics)	High	Indoor, NLOS environments
AoA	Measures signal arrival angle	Directional info, fewer anchors	More expensive and complex due to antenna arrays	High (sub-meter w/UWB)	Robotics, drones, directional apps

2.4.2.1 Trilateration

Trilateration is an often-utilized method for locating an unknown node's position in the network. It utilizes distance readings from three anchors whose locations are known and are not in a direct line. Each of the unknown nodes falls somewhere on the perimeter of a circle whose radius equals the measured distance from the anchor to the unknown node.

By drawing three such circles—one for each anchor—the estimated location of the unknown node is where all three circles intersect, as shown in Figure 2.4. This forms a set of three equations that can be solved to find the unknown node's coordinates:

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = d_1^2 \\ (x - x_2)^2 + (y - y_2)^2 = d_2^2 \\ (x - x_3)^2 + (y - y_3)^2 = d_3^2 \end{cases}$$

Where (x_i, y_i) represents the coordinates of the three anchors ($i = 1, 2, 3$), and d_1, d_2, d_3 are the distances from the unknown node (x, y) to each anchor.

2.4.2.2 Multilateration

Multilateration extends the concept of trilateration by utilizing more than three anchor nodes to find the location of an unknown sensor. Rather than depending on just three points, the distance measurements from several anchors are combined to enhance the accuracy of the location estimate. This technique is particularly popular for stationary node networks due to its simplicity of implementation and relative cost-effectiveness compared

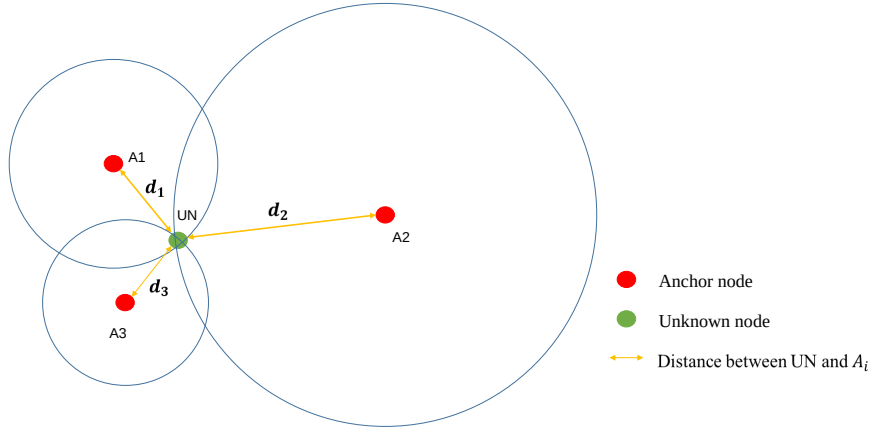


Figure 2.4: Trilateration example showing intersection of three distance circles [5]

to techniques such as triangulation, as depicted in Figure 2.5.

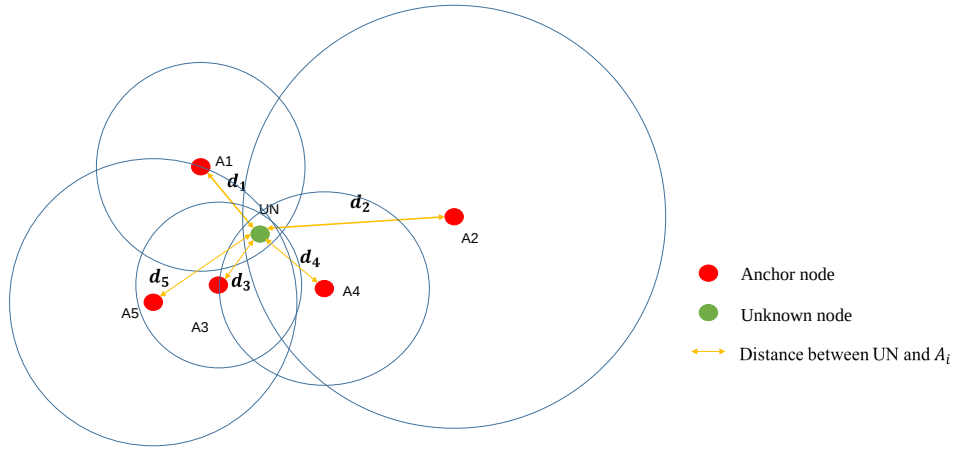


Figure 2.5: Multilateration [5]

2.4.2.3 Triangulation

Triangulation is a positioning technique employed to determine the location of a sensor node using angle measurements rather than distances. Angle information from at least three reference anchor nodes is often required to execute this technique. Using each anchor's Angle of Arrival (AoA) information, the node's location is calculated by applying basic trigonometric principles, specifically the laws of sines and cosines [70, 71]. The anchors and the unknown node form a triangle, as depicted in Figure 2.6, and the geometric

relationships within this triangle allow the position of the unknown node to be found. These relationships are mathematically expressed as:

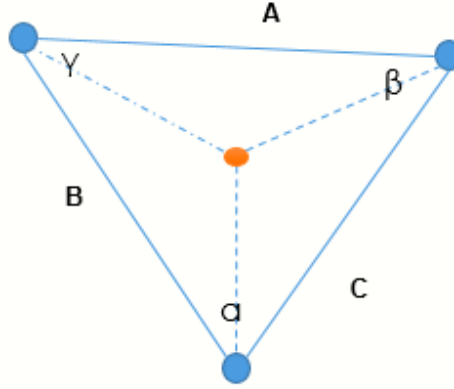


Figure 2.6: Triangulation[5]

Law of sines:

$$\frac{A}{\sin \alpha} = \frac{B}{\sin \beta} = \frac{C}{\sin \gamma} \quad (2.8)$$

Laws of cosines:

$$\begin{cases} A^2 = B^2 + C^2 - 2BC \cos \alpha \\ B^2 = A^2 + C^2 - 2AC \cos \beta \\ C^2 = A^2 + B^2 - 2AB \cos \gamma \end{cases} \quad (2.9)$$

Although triangulation offers exceptional location accuracy in optimal conditions, the method is susceptible to environmental factors, such as signal reflections and obstructions, which can skew angle measurements.

2.4.2.4 Bounding Box

Bounding box is a simple, range-based position estimation method [72]. Instead of calculating one exact point right away, it narrows down where the unknown node could be by overlapping possible areas [42]. For each anchor, an approximate distance to the unknown node is estimated using a ranging technique such as RSSI or ToA. Around each anchor, a square (bounding box) is drawn with the anchor (x_i, y_i) at its center and side lengths equal to twice the estimated distance d_i . This means the unknown node (x_{UN}, y_{UN}) is assumed to lie somewhere inside this square, as shown in Figure 2.7. Mathematically, each bounding box is defined as:

$$x_{UN} \in [x_i - d_i, x_i + d_i], \quad y_{UN} \in [y_i - d_i, y_i + d_i]$$

By overlapping the bounding boxes from all anchors, the intersection area defines the zone where the unknown node (x_{UN}, y_{UN}) is most likely located. This intersection region is computed by finding:

$$\begin{cases} x_{\min} = \max(x_i - d_i) \\ x_{\max} = \min(x_i + d_i) \\ y_{\min} = \max(y_i - d_i) \\ y_{\max} = \min(y_i + d_i) \end{cases}$$

If $x_{\min} > x_{\max}$ or $y_{\min} > y_{\max}$, then the bounding boxes do not overlap, indicating inconsistent distance estimates. Otherwise, the center of gravity (centroid) of the intersection rectangle is taken as the estimated position of the unknown node (x_{UN}, y_{UN}) , and it is given by:

$$x_{UN} = \frac{x_{\min} + x_{\max}}{2}, \quad y_{UN} = \frac{y_{\min} + y_{\max}}{2}$$

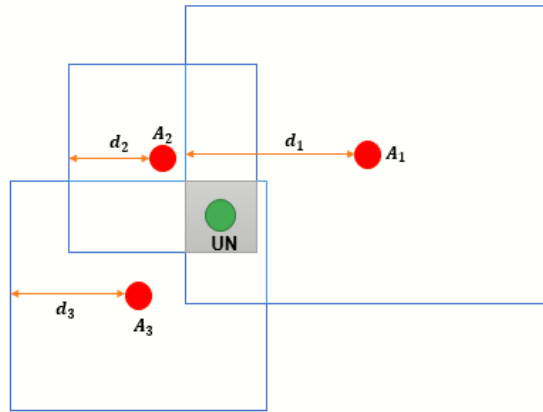


Figure 2.7: The concept of the bounding box method

This method does not require any complex calculation, but its accuracy depends on the precision of the initial phase of distance estimates.

2.4.2.5 GPS-less / Centroid Localization Algorithm (CLA)

This method [53] is considered one of the simplest range-free position estimation techniques in WSNs. It does not require any specific measurements of distance or angles

between nodes. Instead, it estimates the position of an unknown node (x_{UN}, y_{UN}) using only the known coordinates of surrounding anchor nodes.

The basic idea is that an unknown node estimates its location as the geometric centroid (average) of the coordinates of all anchors within its communication range. This means, if an unknown node can receive signals from m anchors at coordinates (x_i, y_i) for $i = 1, 2, \dots, m$, then the estimated position (x_{UN}, y_{UN}) is calculated as:

$$x_{UN} = \frac{1}{m} \sum_{i=1}^m x_i, \quad y_{UN} = \frac{1}{m} \sum_{i=1}^m y_i$$

Figure 2.8 illustrates the concept of the centroid (CLA). Particular attention is given to the region marked by the red triangle, where the magenta star indicates the estimated position of the unknown node calculated using CLA of unknown sensor node (depicted as blue circles) number “3”, which has three anchor nodes (red squares) numbered “8”, “10”, “3” within its transmission range.

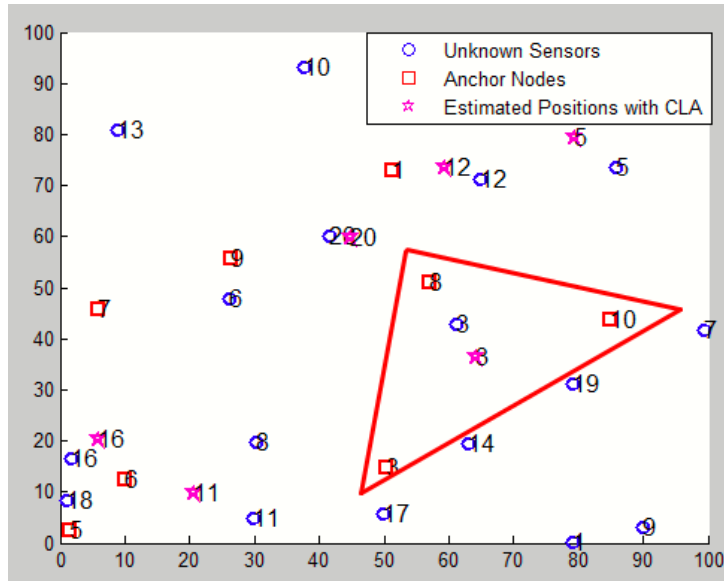


Figure 2.8: Example of the Centroid Localization Algorithm (CLA)

2.4.2.6 The APIT

The APIT (Approximate Point-In-Triangulation) [45] is a range-free position estimation method that estimates the position of an unknown node without requiring any specific measurements. Instead, it relies on connectivity information and the relative positions of anchors.

The algorithm starts by selecting all possible triangles formed by any three anchor nodes within the communication range of the unknown node. If there are N anchors, the

number of possible triangles is given by $\binom{N}{3}$.

For each triangle, the node uses an approximate point-in-triangulation test (PIT) to decide whether it is inside or outside the triangle. This is done by observing how the received signal strength changes as the node moves slightly. If moving in any direction causes all distances from the node to the triangle's corners to increase or decrease together, the node is considered outside the triangle. Conversely, if moving in any direction causes the node to get closer to some anchors but farther from others, the node is considered inside.

After testing all triangles, the algorithm keeps only those where the node is estimated to be inside. The intersection (overlapping area) of these triangles defines the possible region where the node may be located [73]. Finally, the node's estimated position is computed as the center of gravity (centroid) of the intersection area.

The main concept of the APIT method is presented in Figure 2.9.

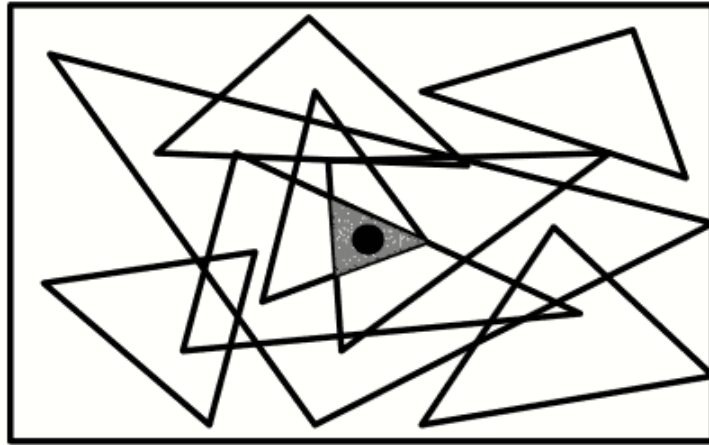


Figure 2.9: Concept of the APIT localization method.

2.4.2.7 DV-HOP

The DV-Hop algorithm, first introduced by Niculescu and Nath in 2003 [54], remains the most used range-free position estimation method for localizing nodes in WSNs. It operates in three main phases:

Phase 1: Determining Minimum Hop Counts

Initially, the anchor nodes begin by broadcasting packets containing their IDs, coordinates, and a hop count initialized to zero. As these packets spread through the network, each node records the coordinates and hop count from the first packets it gets from each anchor. However, if a node receives multiple packets from the same anchor through different paths, it applies the following rules to guarantee that the shortest path information is preserved:

- If the hop count in the received packet is less than the currently stored hop count, the node updates its record, increments the hop count by one, and rebroadcasts the packet to its neighbors.
- Otherwise, the packet is discarded.

This flooding process ensures that every node learns the minimum number of hops required to reach each anchor.

For instance, as shown in Figure 2.10, consider anchor nodes AN1, AN2, AN3, and an unknown node UN. The true distance between AN1 and AN2, AN2 and AN3, and AN1 and AN3 are 30 m, 65 m, and 90 m, respectively. The connections between nodes are depicted using red dashed lines. After the first phase, the unknown node determines its minimum hop counts to AN1, AN2, and AN3 as 3, 2, and 3, respectively.

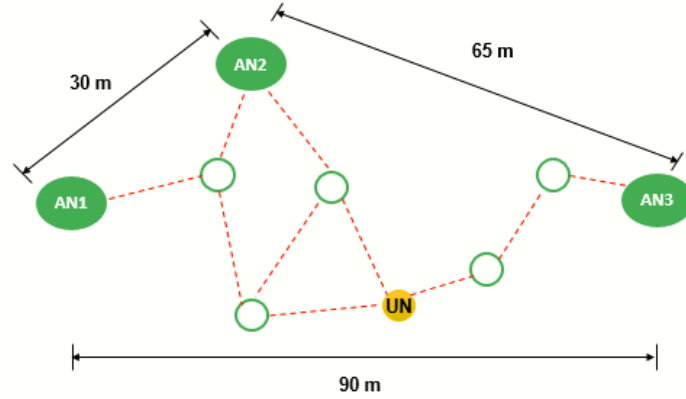


Figure 2.10: The concept of the DV-Hop method

Phase 2: Average Hop Size Calculation and Distance Estimation

This phase involves two main tasks: calculating the average hop distance and using it to estimate the physical distance between nodes.

First, each anchor AN_i computes its mean hop distance, denoted by $AHopSize_i$, by dividing the sum of actual distances to other anchors by the total hop counts to those anchors. Once calculated, this value is broadcast throughout the network. The formula used to calculate $AHopSize_i$ for anchor AN_i is:

$$AHopSize_i = \frac{\sum_{i \neq j}^M \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{i \neq j}^M H_{i,j}}$$

Here, M is the total number of anchors, (x_i, y_i) and (x_j, y_j) are the coordinates of anchors i and j , respectively, and $H_{i,j}$ is the hop count between them.

For instance, referring to the example in Figure 2.10, the AHopSize_i for anchors AN1, AN2, and AN3 are calculated as:

$$\text{HopSize}_{\text{AN1}} = \frac{30 + 90}{2 + 6} = 15 \text{ m}$$

$$\text{HopSize}_{\text{AN2}} = \frac{30 + 65}{2 + 5} = 13.5 \text{ m}$$

$$\text{HopSize}_{\text{AN3}} = \frac{90 + 65}{5 + 6} = 14 \text{ m}$$

Next, each unknown node uses the AHopSize from the first anchor it receives, which is typically the closest anchor. This value is then multiplied by the hop count between the unknown node U and each anchor AN_i to estimate the physical distances:

$$d_{Ui} = \text{AHopSize}_U \times H_{U,i}$$

Where AHopSize_U is the average hop size assigned to the unknown node U , and $H_{U,i}$ is the minimum hop count from the node U to anchor i .

Applying this to the example in Figure 2.10, the minimum hop counts from the node UN to anchor AN1 is 3, to AN2 is 2, and to AN3 is 3. Since AN2 offers the shortest hop path, its hop size (13.5 m) is used. The estimated distances are therefore:

$$d_{\text{UN,AN1}} = 13.5 \times 3 = 40.5 \text{ m}$$

$$d_{\text{UN,AN2}} = 13.5 \times 2 = 27 \text{ m}$$

$$d_{\text{UN,AN3}} = 13.5 \times 3 = 40.5 \text{ m}$$

Phase 3: Estimating the Positions of Unknown Nodes The goal of this phase is to calculate the coordinates of each unknown node by applying the multilateration technique. Each unknown node determines its location by using the estimated distances to several anchors, obtained from the previous phase.

Let (x_U, y_U) be the coordinates of the unknown node U , while (x_i, y_i) denote the coordinates of the i^{th} anchor, where $i = 1, 2, 3, \dots, M$. The distance from node U to each anchor can be expressed by the following set of equations:

$$(x_1 - x_U)^2 + (y_1 - y_U)^2 = d_{U,1}^2$$

$$(x_2 - x_U)^2 + (y_2 - y_U)^2 = d_{U,2}^2$$

$$\vdots$$

$$(x_M - x_U)^2 + (y_M - y_U)^2 = d_{U,M}^2$$

To convert this nonlinear system into a linear form suitable for computation, the final equation is subtracted from each of the preceding equations. This yields a linear system of equations:

$$\begin{aligned} x_1^2 - x_M^2 - 2(x_1 - x_M)x_U + y_1^2 - y_M^2 - 2(y_1 - y_M)y_U &= d_{U,1}^2 - d_{U,M}^2 \\ x_2^2 - x_M^2 - 2(x_2 - x_M)x_U + y_2^2 - y_M^2 - 2(y_2 - y_M)y_U &= d_{U,2}^2 - d_{U,M}^2 \\ &\vdots \\ x_{M-1}^2 - x_M^2 - 2(x_{M-1} - x_M)x_U + y_{M-1}^2 - y_M^2 - 2(y_{M-1} - y_M)y_U &= d_{U,M-1}^2 - d_{U,M}^2 \end{aligned}$$

This system can be compactly represented in matrix form as:

$$AX = B$$

Where:

$$A = \begin{bmatrix} 2(x_1 - x_M) & 2(y_1 - y_M) \\ 2(x_2 - x_M) & 2(y_2 - y_M) \\ \vdots & \vdots \\ 2(x_{M-1} - x_M) & 2(y_{M-1} - y_M) \end{bmatrix} \quad (2.10)$$

$$X = \begin{bmatrix} x_U \\ y_U \end{bmatrix} \quad (2.11)$$

$$B = \begin{bmatrix} x_1^2 - x_M^2 + y_1^2 - y_M^2 + d_{U,M}^2 - d_{U,1}^2 \\ x_2^2 - x_M^2 + y_2^2 - y_M^2 + d_{U,M}^2 - d_{U,2}^2 \\ \vdots \\ x_{M-1}^2 - x_M^2 + y_{M-1}^2 - y_M^2 + d_{U,M}^2 - d_{U,M-1}^2 \end{bmatrix} \quad (2.12)$$

Finally, the coordinates (x_U, y_U) are then found by solving this system using the least squares method:

$$X = (A^T A)^{-1} A^T B$$

Where the superscript T denotes the matrix transpose.

Figure 8 depicts the overall DV-Hop algorithm workflow, illustrating its three core phases.

Although its simplicity, low computational demand, and lack of additional ranging hardware make DV-Hop cost-effective and suitable for large-scale WSNs—where energy

efficiency and deployment cost are crucial—its accuracy can suffer, especially in small or irregularly deployed networks. For instance, two nodes might have identical hop counts to all anchors, resulting in the same estimated position, even though they may actually be far apart. Consequently, much research since DV-Hop was proposed has focused on enhancing DV-Hop to achieve higher accuracy [74].

2.4.2.8 MDS-MAP

MDS-MAP is an extension of classical multidimensional scaling (MDS), a data analysis technique that originated in the fields of psychology and psychometrics, where it was initially developed to visualize the similarity or dissimilarity between objects. In the context of WSN localization, MDS-MAP adapts MDS to estimate the position of sensors based on pairwise distance information [75].

The process begins by assembling a distance matrix that contains estimated distances between every pair of nodes. Since it is rarely feasible to obtain direct distance measurements for all pairs, MDS-MAP estimates those distances using techniques such as RSSI or indirectly through shortest path algorithms like Dijkstra's. Classical MDS is then applied to this matrix to generate a relative map of the network, maintaining the overall topology but without the absolute position.

To convert this relative configuration into absolute coordinates, a sufficient number of anchors is required. This alignment is achieved by applying linear transformations, including translation, rotation, and scaling. In a two-dimensional scenario, at least three anchors are required, while four or more are needed for a three-dimensional scenario. MDS-MAP is robust to incomplete or noisy distance data and naturally supports three-dimensional deployments. However, it may become computationally intensive for large networks, and its localization accuracy depends strongly on the quality and precision of the estimated distances.

2.4.2.9 Sum Dist Min Max

In the SumDistMinMax method, the localization process unfolds in two main phases: Sum-Dist and Min-Max. First, each anchor broadcasts a message containing its unique identifier, known coordinates, and a path length that has been set to zero.

When a sensor node, let's call it X , receives this broadcast, it measures the range to the sender using, for example, RSSI or another ranging method. It then adds this measured range to the path length carried in the message, updates the total path length, and forwards the message to its neighbors only if the new path length is shorter than previously recorded ones. This process allows each node to gather an estimated shortest

distance to each anchor. Among multiple path estimates, only the minimum sum of range (SumDist) for each anchor is retained.

Next, each node applies the Min-Max method to approximate its own coordinates. The Min-Max method works by defining a box or interval within which the node must lie, based on the estimated distances to multiple anchors. For each anchor, the node calculates a possible range that accounts for measurement uncertainties. The overlapping regions from all anchors form a confined area that is likely to contain the node's true position. The estimated location is then determined by calculating the centroid of this intersection region.

2.5 Performance Metrics of Localization Algorithms

Evaluating the performance of a localization technique requires more than just examining its accuracy. Since sensor nodes face several limitations—such as hardware costs, limited energy supply, and restricted capabilities—other performance metrics like complexity, convergence rate, coverage, scalability, and robustness should also be taken into account [70, 71, 76–81].

Localization Accuracy

In the design of localization algorithms, achieving high positional accuracy is a primary objective. This performance is typically evaluated based on the average error between the estimated coordinates of target nodes and their real ones. To enhance accuracy, algorithms must minimize this error. However, striving for higher accuracy may impact other system parameters, making it necessary to carefully balance accuracy with other operational requirements.

Complexity

The complexity metric reflects the algorithm's requirements in terms of processing time and memory usage. A good localization algorithm should be quick to execute, easy to implement, and should not rely on heavy computations or large memory. Reducing complexity helps extend the lifetime of battery-powered nodes, which in turn enhances the overall network performance. Algorithms like centroid and trilateration address this complexity challenge, albeit at the cost of increased localization errors.

Computation Time

Computation time refers to the total amount of time sensor nodes spend performing the calculations needed to estimate the positions of unknown nodes. These steps include calculating distances between nodes, solving equations to compute positions, performing mathematical operations such as matrix decomposition (used in methods like MDS), and updating estimations in iterative algorithms. It only accounts for the time used in calculations and does not include communication delays. Since sensor nodes typically have limited battery and processing power, minimizing computation time is crucial to saving energy and making the localization system faster and more practical for real-time use.

Convergence Rate

This metric measures how quickly a positioning method reaches a stable and accurate solution—that is, how fast the estimated positions converge to the true node locations. It is typically quantified by the number of iterations required to meet a defined accuracy threshold. In simple terms, it reflects the algorithm’s efficiency in reaching a solution, rather than its actual runtime.

Latency

Latency refers to the total time from the start of the localization process until the target nodes determine their final estimated positions. This includes time spent on computation, communication delays, and synchronization overhead needed for coordination among nodes.

Localization Success Rate

LSR in WSN localization refers to the percentage of sensor nodes in the network whose positions are successfully estimated. For a network of 100 nodes, 100% success means all nodes are localized, while 70% indicates 70 nodes have known positions.

Scalability

Scalability describes a localization scheme’s ability to maintain its performance as the network size increases. It refers to how well the algorithm handles a larger number of nodes or a larger deployment area without a significant rise in localization error, computational cost, or communication overhead. A scalable algorithm can localize large-scale deployments efficiently and reliably.

Robustness

Robustness refers to the ability of a localization algorithm to accurately estimate node positions even under difficult conditions. In WSNs, signals may be noisy or obstructed (e.g., indoors), and some nodes may become inactive due to battery depletion. Under such circumstances, a robust algorithm can still estimate node positions using limited or uncertain data [82].

2.6 Challenges in Wireless Sensor Network Node Localization

Despite the significant progress in WSN localization techniques and performance evaluation methods, achieving reliable and accurate node positioning remains challenging due to the inherent limitations of WSNs. These limitations arise from hardware constraints, environmental factors, and network characteristics, which directly affect localization performance and system efficiency. The principal challenges are outlined below [83].

1. **Resource Constraints:** Sensor nodes operate with minimal processing power, limited memory, restricted communication capabilities, and finite battery energy. These limitations require localization methods to be computationally efficient and energy-aware so that accurate positioning can be achieved without exhausting node resources .
2. **Signal Behavior in Real Environments:** Wireless signals are heavily influenced by environmental conditions such as path loss, multipath propagation, interference, and physical obstacles. These factors cause variations in measurements—such as received signal strength, travel time, or angle estimation—which reduces localization accuracy. Addressing these signal-related uncertainties is essential for reliable positioning.
3. **Non-Line-of-Sight (NLOS) Conditions:** In many deployments, nodes may not have a direct line-of-sight, and signals often reach receivers through reflections. NLOS effects distort measurements and can introduce substantial localization errors. Developing techniques capable of recognizing and mitigating NLOS conditions remains a major challenge.
4. **Large-Scale Deployment:** WSNs often contain a high number of nodes distributed over large areas. As network size grows, localization algorithms must scale

effectively while keeping computation and communication overhead manageable. Ensuring timely and efficient localization across large deployments is therefore critical.

5. **Mobility and Dynamic Topology:** In some applications, nodes may move or the network structure may change over time. Frequent updates to node positions and connectivity complicate the localization process. Algorithms must adapt quickly to maintain accurate estimates in dynamic environments.

2.7 Conclusion

This chapter elucidates the issue of node localization in WSNs and delineates the methods employed by these networks to determine the positions of unknown nodes. First, we presented the general system model used in localization tasks, highlighting the role of anchor nodes with known positions. We also discussed why relying on GPS is often impractical in WSNs due to high energy consumption, signal unavailability in certain environments, and additional hardware costs. Various classification approaches were discussed based on the location of computation, the use of additional hardware, network connectivity, localization granularity, and dependence on reference nodes. Performance metrics were introduced to evaluate localization techniques. Metrics such as accuracy, complexity, computation time, and scalability help determine the strengths and limitations of each approach. Subsequently, the localization process was divided into two main phases: distance estimation and position estimation, highlighting common methods such as ToA, RSSI, trilateration, DV-Hop, and others. However, after reviewing these techniques, it is clear that no single method performs optimally across all performance metrics. For example, range-based techniques offer high precision but often require additional hardware, higher energy consumption, and increased computational complexity. Consequently, researchers have focused on improving these algorithms to enhance localization accuracy while maintaining acceptable computation time and system complexity. One promising research direction is the use of bio-inspired optimization algorithms, which are inspired by natural collective behaviors observed in biological systems. In the next chapter, we will explore these bio-inspired methods and examine how they can improve localization performance.

Chapter 3

Localization Optimization Using Bio-Inspired Techniques

3.1 Introduction

A continuation of the theoretical background on traditional localization schemes in WSNs presented in the previous chapter, and recognizing their inherent limitations, researchers have turned to alternative strategies, namely optimization algorithms, and in particular bio-inspired techniques, which constitute the main focus of this chapter. These algorithms have proven as a promising category of optimization strategies to solve the localization problem in WSN, due to their ability to search effectively through complex solution spaces and offer near-optimal position estimates of unknown nodes at a lower computation cost. Their flexibility and efficiency have led to an increasing research focused on investigating and expanding their potential in this field.

The purpose of this chapter is to introduce the general notion of optimization, provide an overview of Bio-Inspired Algorithms (BIAs), highlight their categories and general working mechanisms, and explain how these approaches have been applied by researchers to address or enhance the localization problem in WSNs, as well as to handle its associated challenges.

3.2 General Optimization Formulation

Optimization refers to the process of finding the best possible values for a set of decision variables so that a given objective function is minimized or maximized. Consider a function

$$f : \mathbb{R}^D \rightarrow \mathbb{R},$$

which is to be minimized, where D represents the number of variables in the problem. The goal is to identify a vector $\mathbf{x}_0 \in \mathbb{R}^D$ such that

$$f(\mathbf{x}_0) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathbb{R}^D,$$

while also satisfying the problem's constraints.

These constraints may include:

Inequality constraints:

$$g_p(\mathbf{x}_0) \leq 0, \quad p = 1, 2, \dots, P,$$

Equality constraints:

$$h_q(\mathbf{x}_0) = 0, \quad q = 1, 2, \dots, Q,$$

where P and Q are the numbers of inequality and equality constraints, respectively. Additionally, each variable is restricted to lie within its allowable bounds:

$$x_i^l \leq x_i \leq x_i^u, \quad i = 1, 2, \dots, D,$$

where $[x_i^l, x_i^u]$ defines the permissible interval for the i -th variable.

Many real-world problems, such as timetabling, path planning, packing, the traveling salesman, trajectory optimization, and engineering design, can be formulated as optimization problems. The complexity of these problems depends on factors like the number of variables, the size of the solution space, and the specific constraints. Solving high-complexity problems often requires more time and computational resources. Methods for solving optimization problems are generally classified as deterministic or stochastic. Metaheuristics are a group of stochastic optimization methods designed to tackle complex and nonlinear problems by searching for high-quality solutions without guaranteeing the global optimum [84]. They are widely used because they are flexible, easy to implement, and can quickly produce good results across a broad range of applications.

These methods are organised into several families based on their underlying principles or search mechanisms. Among these categories, bio-inspired algorithms (BIAs) form a major group, drawing inspiration from natural processes and biological behaviours. This class of metaheuristics is the focus of the current study.

3.3 Bio-Inspired Meta-heuristic Techniques (BIAs)

BIAs have gained prominence as effective tools for solving complex optimization problems. They are a subclass of metaheuristics that emulate strategies observed in nature, such as evolutionary adaptation, swarm behavior, foraging strategies, and immune system responses. BIAs are inherently stochastic, population-based, and adaptive, allowing them to explore large and complex search spaces efficiently.

BIAs are commonly grouped into several major families. One widely referenced classification, presented by Somvanshi et al. [85], organises these techniques into eight overarching categories. These include evolutionary-based algorithms, swarm intelligence-based algorithms, plant or ecosystem-inspired algorithms, predator-prey and foraging models, neural-inspired methods, physics-based approaches, human-inspired techniques, and hybrid models, as illustrated in Figure 3.1.

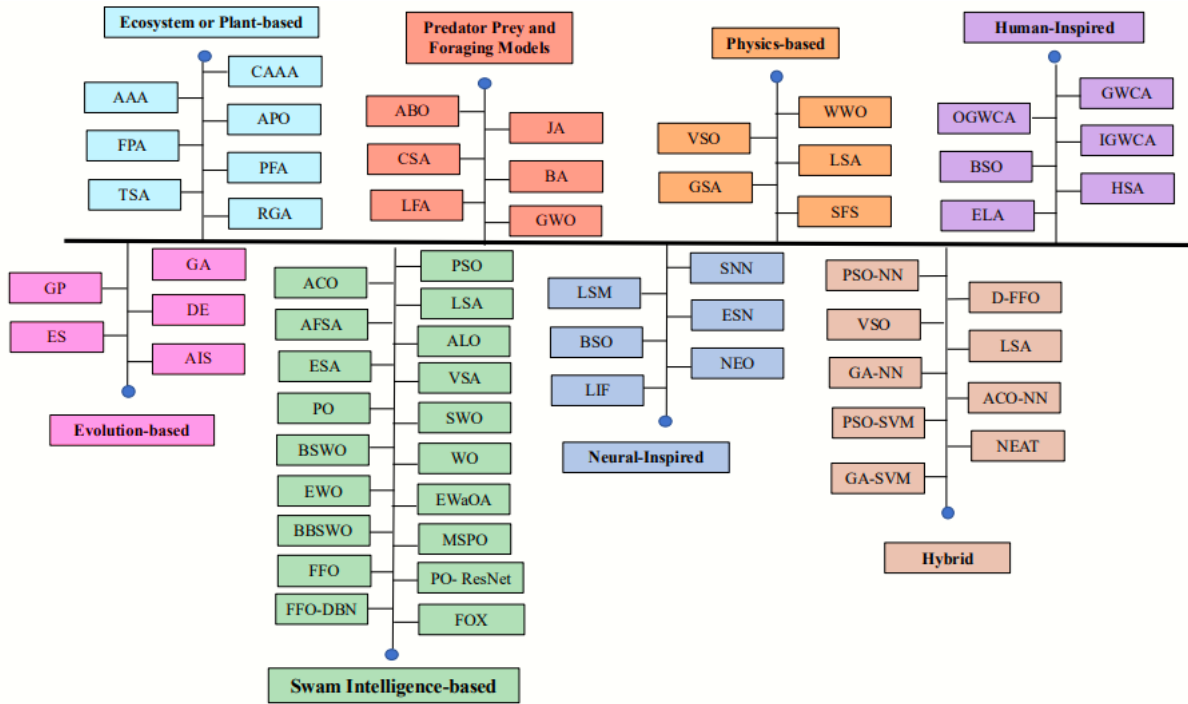


Figure 3.1: Classification of BIAs (adapted from Somvanshi et al. [85])

3.3.1 General Metalogy of BIAs

Although different BIAs may model nature's behaviors differently, they usually follow a similar step-by-step process [86]. Randomness plays a key role in these algorithms, especially when starting the search with an initial group of candidate [87] solutions. the general BIAs's process is shown in Figure 3.2

- **Phase 1: Initialization of Candidate Solutions**

At the very beginning, the algorithm creates a group (called a population) of possible solutions. These solutions might be generated by randomly selecting values within set boundaries or using specific mathematical rules to make sure the solutions are valid. This initial group helps the algorithm explore many different parts of the problem's solution space, making it less likely to miss good solutions. Once the population is ready, each solution is evaluated based on a fitness or objective function that measures how good it is. The solutions are then ranked from best to worst. The algorithm also sets criteria for when to stop searching, such as reaching a maximum number of iterations or achieving a desired level of accuracy.

- **Phase 2: Solution Generation and Improvement**

Next, the algorithm produces new candidate solutions based on the existing ones. This is done by combining information from previous solutions, adding random variations, and applying algorithm-specific rules and parameters. This phase focuses more on exploitation, which means improving solutions in promising regions while still allowing some exploration with randomness. After generating new solutions, the algorithm evaluates and ranks them again to see if it has found better candidates.

- **Phase 3: Best Solution Selection and Stopping Criteria**

Finally, the algorithm compares the current best solutions with those newly generated and selects the top-performing ones to form the population for the next round. Then, it checks if the stopping criteria, like maximum iterations or sufficient solution quality, are met. If not, the process repeats by going back to generate new candidates. If yes, the algorithm ends, returning the best solution it found.

3.4 A review of BIA based Localization Optimization

In this part, we review five widely used BIAs, highlighting recent relevant studies and examining how researchers have applied these techniques—either in their original form or through enhanced variants—to address the sensor node localization problem.

3.4.1 Particle Swarm Optimization

The algorithm was proposed by Kennedy and Elbehart [88]. This algorithm is derived from the collective behavior exhibited by birds when they flock together in nature. Since its introduction, PSO has been improved and adapted for various environments and applied

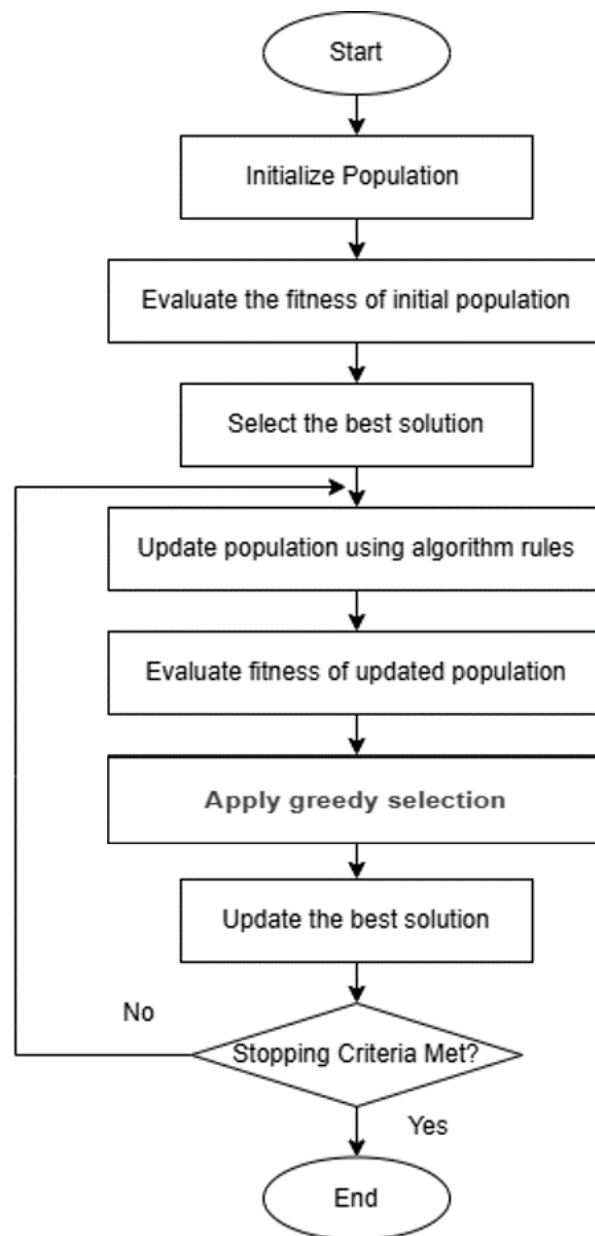


Figure 3.2: The BIA's general process

to solve optimization problems in many fields. To identify the best solution to a problem, in the PSO algorithm, a group of particles travels through a search space. With a unique position and velocity, every particle represents a possible solution to the problem. A particle's velocity describes how it moves around the search space, whereas its position relates to a possible solution. Then Particles update their velocities and positions iteratively, influenced by their previous velocities, personal best positions (the stored best position they have personally found), and global best positions (the best position found by the entire swarm). This iterative process allowing particles to explore the search space and

move toward the best solutions (positions). The position and velocity of each particle are updating using:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (\text{pbest}_i - x_i^t) + c_2 \cdot r_2 \cdot (\text{gbest} - x_i^t) \quad (3.1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3.2)$$

where:

- v_i^t and x_i^t represent the velocity and position of the i -th particle at iteration t .
- pbest_i is the local best position found by the i -th particle.
- gbest is the global best position found by the entire swarm.
- w , c_1 , and c_2 denote the inertia weight, cognitive coefficient, and social coefficient, respectively.
- r_1 and r_2 are random values in the range $[0, 1]$ that introduce stochastic behavior to diversify the search process.

This algorithm is simple to implement, computationally efficient, and capable of producing high-quality solutions with relatively fast convergence [89], making it very suited for optimizing node locations in WSNs.

The main steps of this algorithm are summarized in **Algorithm 1**, which presents its pseudocode for clarity and better understanding.

Algorithm 1 The pseudocode of PSO

- 1: Initialize particles with random positions and velocities
- 2: Evaluate fitness of each particle
- 3: Initialize personal best ($pbest$) and global best ($gbest$)
- 4: **while** (termination condition not met) **do**
- 5: **for** each particle i **do**
- 6: Update velocity:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest - x_i^t)$$

- 7: Update position:

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

- 8: Update personal best ($pbest_i$) if current fitness is better
 - 9: **end for**
 - 10: Update global best ($gbest$) from all particles
 - 11: **end while**
 - 12: **return** Global best position found
-

3.4.1.1 Bio-Inspired Node Localization in Wireless Sensor Networks

In their study [90], the authors presented a decentralized iterative localization algorithm by utilizing bioinspired algorithms to addresses the challenge of determining the positions of randomly deployed sensor nodes in a WSN. Specifically, they investigate PSO and bacterial foraging algorithms (BFA) to address the localization issue in WSNs. Their study includes the examination of two cases. In the first one, every localizable node utilizes PSO to determine its own localization. In the second, every localizable node uses BFA to determine its own localization. Both PSO and BFA aim to identify the optimal values for the coordinates x and y that minimize the objective function. Moreover, in each iteration, the settled nodes act as references and send their location coordinates, similar to how beacons (anchors) do. The finding indicates that the suggested algorithms have a tradeoff problem. While the PSO method achieves approximately 3-4x faster determination of node coordinates, the BFA approach accomplishes this task with greater accuracy with fewer non-localized nodes and lower errors.

3.4.1.2 Particle Swarm Based Node Localization in Wireless Sensor Networks

Authors in [91] suggest PSO-IDV, an enhanced DV-Hop technique that utilizes PSO, to reduce localization error and improve precision in locating unknown nodes in a WSN. This is achieved by modifying the average hop size of the anchor node in the PSO-IDV algorithm by appending a correction factor to the previous hop size value of a specific

anchor node. The PSO is then used to further improve accuracy. The correction factor is determined as the difference between the estimated distance calculated by the original DV-Hop algorithm and the actual distance between two anchor nodes. Based on simulation results, the PSO-IDV algorithm outperformed DV-Hop and DV-Hop based on Genetic Algorithm (GADV-Hop) methods as well as DV-Hop utilizing PSO algorithms in terms of convergence rates and accuracy in locating unknown nodes in WSNs.

3.4.1.3 A swarm intelligence based distributed localization technique for wireless sensor network

This work [9] by Ramesh et al. introduces a distributed method for localization in WSNs that emphasizes energy efficiency. The first thing that the author did was a practical study of the energy requirements for computation and transmission carried out in a typical sensor node, where they found out that transmission is significantly more power-intensive than computation. This finding motivated the authors to propose an energy-efficient localization scheme that reduces communications overhead and makes more use of local computation. The localization task was regarded as optimization problem and solved with two swarm intelligence methods: PSO and Comprehensive Learning PSO. Findings from simulations reveal that although PSO reaches a solution more quickly, CLPSO delivers greater positional accuracy since precision is more critical than speed in one-time localization processes, the study identifies CLPSO as the more suitable algorithm, particularly in large-scale or challenging environments.

3.4.1.4 Distributed Localization for Wireless Sensor Networks Using Binary Particle Swarm Optimization (BPSO)

Here [92], Zain et al. Present a distributed localization method for WSNs using BPSO and its modified version. The paper thoroughly compare PSO algorithm with the proposed ones algorithms through simulations conducted in Matlab, evaluating key metrics such as localization errors and computation time. The results demonstrate that BPSO reduces computation time by nearly 59% compared to PSO with only an 11% increase in error, while the modified BPSO achieves a further 39% reduction in time over basic BPSO while maintaining equivalent accuracy.

3.4.1.5 A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks

This study [93] focuses on solving the profound issue of proper localization of sensor nodes (SNs) in WSNs placed in obstructed environments, which causes significant deformation of signal reading as well as the inaccuracy of localization. To overcome these difficulties, the authors propose an innovative hybrid approach named Node Segmentation with Improved PSO, called NS-IPSO. The main idea of this approach is to segment the deployment area into smaller regions based on how often they appear in the shortest path between anchor nodes. This segmentation enables nodes within the same group to communicate more directly, thereby reducing detours caused by obstacles and improving the distance calculation accuracy. Then, the authors limit the localization search area using the bounding box method. They also design a new fitness function that incorporates the inverse of hop counts between anchor and unknown nodes. This weighting ensures that anchors closer in terms of hops have a stronger influence on localization, which means it yields more accurate distance estimates. Furthermore, PSO is improved by monitoring particles whose positions remain unchanged over iterations and randomly repositioning them within the defined search space (bounding box). This strategy reduces the likelihood of particles being trapped in local optima, thus accelerating convergence and improving solution quality. Numerical results across different sensor deployments (C-, H-, and S-shaped node distributions) and varying anchor and unknown node counts, consistently demonstrate that NS-IPSO outperforms existing algorithms, including HDPSO, Hybrid PSO, ADNL, WSLA, and min-max PSO, in terms of mean location error, with improvement reaching 91.12%.

3.4.1.6 DV-Hop localization methods for displaced sensor nodes in wireless sensor network using PSO

The paper [94] proposes a new model for localizing the displaced nodes in WSNs based on improved DV-Hop mechanism coupled with PSO. In their model, the authors suppose a random of anchor and unknown nodes on a network of 100 x 100 m², with a dislocated subarea of 50 x 50 m². The nodes, anchor or unknown, within this subarea need to be re-localized when events cause their original locations to be change. To overcome it, the framework uses selected anchors, which are nodes posed not change and credibly act as a reference point after the displacement. The proposed model is organized into several methods. The first restricts localization to displaced unknown nodes by using only anchor nodes inside the displaced region. The second expands this by using all anchors in the

entire network for displaced node localization. In later methods, the unknown nodes that are successfully localized in the first round are promoted to act as reference points for subsequent localization. To estimate the positions, DV-HOP first computes distances using the corrected hop sizes, which are calculated by incorporating the error between estimated and true anchor distances. PSO then is applied to find and optimize node positions. After simulations, the results show that the promotion-based strategy records nearly 81% less elapsed time and 67% lower energy consumption compared with the traditional approaches, while keeping the localization error at a comparable level.

3.4.1.7 A sensor deployment approach for target coverage problem in wireless sensor networks

The paper [95] addresses the sensor deployment problem in WSNs, which is a critical pre-deployment challenge closely related to but distinct from the post-deployment localization problem. While localization seeks to discover unknown node positions after random deployment, the deployment problem aims to determine optimal sensor node locations beforehand to ensure target coverage while maximizing network lifetime. The authors focus on solving this problem under three coverage constraints: simple (each target covered by at least one sensor), K-coverage (each target covered by at least K sensors), and Q-coverage (each target T_j covered by at least q_j sensors) by using two improved versions of PSO: cooperative PSO (CPSO) and cooperative PSO enhanced with fuzzy logic (CPSO_fuzzy). CPSO decomposes the n -dimensional optimization problem into n one-dimensional problems, each optimized by a separate swarm, improving scalability and solution quality. CPSO_fuzzy further refines CPSO by dynamically adjusting acceleration coefficients in the velocity update equation using a fuzzy logic controller. This dynamic adaptation balances global and local search capabilities during iterations, thus avoiding premature convergence and leading to better quality solutions. The simulations, performed on an area of $500 \times 500 \text{ m}^2$ with 25 targets and a varying number of sensor nodes (100–250), showed superiority of CPSO and CPSO_fuzzy over the existing methods in terms of network lifetime across all versions of the coverage problem, with CPSO_fuzzy consistently achieving the longest network lifetime.

3.4.2 Grey Wolf Optimizer (GWO)

The Grey Wolf Optimizer (GWO), introduced by Mirjalili and colleagues in 2014 [96], takes inspiration from the social structure and hunting techniques of grey wolves (*Canis lupus*). This algorithm, influenced by nature, replicates the leadership structure of wolf packs (alpha, beta, and delta wolves) and their cooperative hunting tactics, including

tracking, encircling, and attacking prey. By leveraging these principles, the GWO algorithm can navigate candidate solutions in both the exploration and exploitation phases of the search space.

The algorithm starts by generating a population of N grey wolves, where each wolf corresponds to a candidate vector $\vec{X}_i$ in the n -dimensional search domain. According to the fitness function, the best solution is identified as the alpha ($\vec{X}_\alpha$), the next as the beta ($\vec{X}_\beta$), and the third as the delta ($\vec{X}_\delta$). These three dominant wolves guide the rest of the pack (omega) toward better positions. The distance between an individual and a leader wolf is expressed as:

$$\vec{D}_j = \left| \vec{C}_j \cdot \vec{X}_j(t) - \vec{X}_i \right| \quad (3.3)$$

where $\vec{X}_j(t)$ is the position of one of the j -th leading wolves (α, β, δ), $\vec{X}_i$ is the current wolf's position, and $\vec{C}_j = 2 \cdot \vec{r}_2$ is a coefficient vector generated from a random vector $\vec{r}_2 \in [0, 1]$.

Each wolf then adjusts its position based on the guidance of the three leaders in the following way:

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \quad (3.4)$$

$$\vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \quad (3.5)$$

$$\vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \quad (3.6)$$

$$\vec{X}_i(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (3.7)$$

where $\vec{A}_j = 2 \cdot \alpha \cdot \vec{r}_1 - \alpha$, and α decreases linearly from 2 to 0 over the iterations. This parameter controls the coefficient $\vec{A}_j$, which affects how wolves move toward or away from the prey (candidate solution). Meanwhile, $\vec{r}_2 \in [0, 1]$ is a randomly generated vector.

This updating mechanism encourages the wolves to gather around the most promising candidate solutions while still maintaining diversity in their search through random coefficients. The key steps of this algorithm are outlined in Algorithm 2, which provides its pseudocode for clarity.

Algorithm 2 The pseudocode of (GWO)

```
1: Initialize the grey wolf population  $X_i$  randomly
2: Initialize  $a$ ,  $A$  and  $C$ 
3: Determine the fitness of each wolf  $X_i$ 
4:  $X_\alpha$  = the best solution
5:  $X_\beta$  = the second best solution
6:  $X_\delta$  = the third best solution
7: while  $i \leq \text{MaxIter}$  do
8:   for each wolf  $X_i$  do
9:     Update the position of the current wolf using equation 3.7
10:  end for
11:  Update  $a$ ,  $A$  and  $C$ 
12:  Determine the fitness of each wolf  $X_i$ 
13:  Update  $X_\alpha$ ,  $X_\beta$  and  $X_\delta$ 
14:   $i = i + 1$ 
15: end while
16: return  $X_\alpha$ 
```

3.4.2.1 A Novel Localization Algorithm Based on Grey Wolf Optimization for WSNs

GWO-LA is the algorithm proposed in [97] that applies the bio-inspired GWO technique directly to the problem of localization in WSNs. The authors detailed the mechanism of GWO and explained how it is adapted for node localization. Simulation experiments were used to evaluate and compare the suggested algorithms with the classical PSO-LA in simulated network environments with varying parameters, such as ranging errors, anchor node ratios, and communication radii. The results indicate that GWO-LA outperformed PSO-LA in all performance metrics. In fact, the proposed GWO-LA improved the average localization error by about 11% across different error scenarios. Plus, it needed fewer anchor nodes and smaller communication radii to match the accuracy of PSO-LA. Overall, this study highlights how effective and practical the GWO approach is for localization in WSNs, presenting a promising alternative to conventional intelligent computing methods.

3.4.2.2 Localization for Wireless Sensor Networks Assisted by Two Mobile Anchors with Improved Grey Wolf Optimizer

Cui et al. [98] introduced a clever localization algorithm for WSNs that effectively balances accuracy with energy consumption by utilizing two mobile anchors that navigate along a specially designed path known as Double Anchor SCAN (DASCAN) and an improved GWO. The DASCAN splits the region-of-interest (ROI) into grids, allowing the two mobile anchors to move simultaneously in a zigzag pattern along adjacent rows, creating

virtual anchors arranged in equilateral triangles, which significantly improve distance estimation. They further proposed the Multiple-Disturbance Strategy GWO (MSD-GWO), which enhances the classical GWO through nonlinear weight adjustments, random perturbations, and mirror wolves to effectively steer the optimization away from local optima. The energy consumption and localization rate of DASCAN were measured against other well-known path panning algorithms (GTURNS, GSCAN, PP-MMAN, H-CURVES, M-CURVES, SCAN) through extensive MATLAB simulations. For localization errors, MDS-GWO was compared with PSO, trilateration, Whale Optimization Algorithms (WOA), and the standard GWO. The results show that DASCAN consumes less energy (approximately 30.1%) than similar multi-anchor paths (GSCAN, GTURN, and PP-MMA), but with about same accuracy. Despite the slightly higher energy consumption than that of single-anchor paths (M-CURVES, H-CURVES, SCAN), DASCAN still leads to a 32.3% reduction in localization error. Meanwhile, MDS-GWO is the top performer among compared localization algorithms, with an average reduction of 25.5% in localization error. In addition, the proposed method is robust to changes in radio irregularity, radio range, and fading effects.

3.4.2.3 Repulsion-based grey wolf optimizer with improved exploration and exploitation capabilities to localize sensor nodes in 3D wireless sensor network

The article [99] proposes a practical and novel method for addressing the difficult task of defining sensor locations in 3-dimensional spaces. The main contribution of the authors is the creation of the Repulsion-based GWO (R-GWO), an enhanced GWO that significantly outperforms the standard algorithm by incorporating the worst-performing solution (omega wolf). This addition creates repulsion forces that guide the search away from non-promising areas, allowing global or near-global solutions to be found more effectively. The system uses just one physical anchor node equipped with GPS and then created six virtual anchors that are projected in different orientations around it in an umbrella-shaped projection (also referred to as a parasol projection). This setup is intended to alleviate line-of-sight (LoS) issues and to satisfy the minimum anchor requirement for 3D localization. These anchors then transmit signals to the target nodes, which are randomly deployed within a three-layer cubic architecture, in order to estimate distances base on received signal strength indicator (RSSI) values, while also considering signal irregularities and fading. The results clearly show that R-GWO does better than several well-known meta-heuristics, such as PSO, BBO, FA, FPA, adaptive FPA, and standard GWO, since it reaches the lower average localization error and converges more quickly.

3.4.2.4 Improved Grey Wolf Optimization Based Node Localization Approach in Underwater Wireless Sensor Networks

This paper [100] presents the Improved Grey Wolf Optimization-Based Node Localization (IGWONL-UWSN) technique, which is an innovative hybrid metaheuristic aimed at solving the challenges of node localization in Underwater WSNs. The authors have made a significant contribution by combining the Grey Wolf Optimizer with a Dimension Learning-based Hunting (DLH) search strategy, which is specifically designed to improve convergence and reduce localization errors. The simulation results highlight its effectiveness, showing notable enhancements in both the number of localized nodes and accuracy compared to existing algorithms. While the study presents some interesting findings, it does come with some important limitations. The main "improvement" to the GWO algorithm is somewhat of a mystery, as the way the DLH strategy is integrated isn't clearly explained or detailed. This raises questions about how reproducible the work is and its overall scientific credibility. Additionally, the evaluation uses a basic static 2D model and depends on RSSI-based ranging, which misses out on crucial real-world factors like 3D dynamics, node movement, and the often unreliable nature of acoustic signal strength in underwater environments. So, even though the results look promising, the practical significance and validity of the proposed method are limited by these major oversights and simplifications.

3.4.2.5 Wireless Sensor Network Localization Incorporating Gray Wolf Optimization and DV-Hop Algorithm

In this paper [101], the authors present a high-precision hybrid localization model for WSNs, combining an improved version of the DV-Hop algorithm with an enhanced Grey Wolf Optimization (GWO) method. The two main innovations introduced by the authors systematically address the primary shortcomings of the traditional DV-Hop approach. First, they optimize the distance estimation process by adopting hopping distance error correction with the help of RSSI, applying dynamic communication radius adaptation for fine-grained hop counting, and using centroid weighting calculations. The second enhancement focuses on the coordinate calculation phase. Instead of sticking with the traditional least-squares method, which can lead to errors in large-scale networks, the authors opt for an improved version of the Grey Wolf Optimizer (GWO). To minimize the chances of premature convergence, they incorporate elements from PSO, such as inertia weights and a personal-best (pBest) memory component. Additionally, chaotic mapping is used during the initialization phase to expand the search space and improve solution diversity. According to the findings, the proposed model achieves an average localization

error of less than 0.18 meters while maintaining reasonable computational costs.

3.4.2.6 GWO Model for Optimal Localization of IoT-Enabled Sensor Nodes in Smart Parking Systems

The work [102] introduces an innovative multi-objective optimization model that employs Grey Wolf Optimization (GWO) to determine the optimal placement of IoT-enabled wireless sensor nodes in smart parking systems. A key factor in these systems is the accurate positioning of sensor nodes, particularly anchor nodes, as this directly influences vehicle detection and ensures efficient communication within the network. Conventional single-objective localization methods primarily focus on distance constraints while often neglecting geometric topology constraints, which can lead to reduced localization accuracy. The proposed approach addresses the node localization problem as a multi-objective optimization task by simultaneously considering both distance and geometric topology constraints to minimize localization error. The Multi-Objective Grey Wolf Optimization Localization Algorithm (MOGWOLA) integrates the hierarchical hunting strategies of grey wolves with Pareto optimality principles to effectively reduce localization error and minimize the number of anchor nodes required, thus lowering deployment costs. Extensive simulations demonstrate that MOGWOLA outperforms existing techniques—such as Generalized Regression Neural Network (GRNN), Particle Swarm Optimization-based Localization (PSO-L), and Decision Theory-Based Localization Algorithm (DTBLA)—by achieving up to a 26% reduction in localization error while exhibiting faster convergence rates. Furthermore, the algorithm can localize a higher number of sensor nodes and adapts effectively to varying transmission ranges and node densities, making it well-suited for large-scale smart parking deployments.

3.4.3 Salp Swarm Optimization (SSA)

Salp Swarm Algorithm is a bio-inspired method that was inspired by the salps' movements and feeding behavior, which are transparent marine animals belonging to the Salpidae family. These animals look like jellyfish in both shape and locomotion, and navigate through the water by contracting their cylindrical bodies to push water behind them. Even though there is little scientific knowledge on salps, largely due to their deep-sea habitats that make them hard to study in labs, their amazing collective behavior prompted the development of SSA.

Salps establish themselves in their natural habitat in organized groups called salp chains which help them to move more easily and make better use of their energy in foraging. The SSA captures this mechanism by categorizing individuals into types: Leader

(head) and followers. The leader, positioned at the front of the chain, directs the motion toward the food source, while the remaining salps follow sequentially. In the optimization framework, each salp's location is expressed in an n -dimensional search space according to the problem variables. The leader then adjusts its coordinates relative to the food source using the following expression:

$$x_j^1 = \begin{cases} F_j + c_1 \cdot [(ub_j - lb_j) \cdot c_2 + lb_j], & \text{if } c_3 \geq 0 \\ F_j - c_1 \cdot [(ub_j - lb_j) \cdot c_2 + lb_j], & \text{if } c_3 < 0 \end{cases} \quad (3.8)$$

where x_j^1 represents the leader's coordinate in the j -th dimension, and F_j denotes the position of the food source in the same dimension. The symbols ub_j and lb_j correspond to the upper and lower bounds of the search space, respectively. The terms c_1 , c_2 , and c_3 are random control parameters that determine the step size and movement direction. Among these, c_1 is particularly crucial because it balances exploration and exploitation. Its value is adjusted dynamically throughout the iterations using:

$$c_1 = 2 \cdot e^{-\left(\frac{4l}{L}\right)^2} \quad (3.9)$$

where l is the current iteration index and L is the maximum number of iterations.

For the followers, the position is updated based on a simplified version of Newton's law of motion. Assuming the initial velocity is zero and the time between iterations is one unit, the update equation is:

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}), \quad i \geq 2 \quad (3.10)$$

Algorithm 3 provides the pseudocode representation of SSA.

Algorithm 3 The pseudocode of SSA

```
1: Initialize the salp population randomly
2: Evaluate the fitness of each salp
3: while (termination condition not met) do
4:   Identify the leader (best solution so far)
5:   Update leader  $F$  (best position so far)
6:   Update  $c_1$ 
7:   for each salp  $i$  do
8:     if  $i = 1$  then
9:       Update leader using equation 3.8
10:    else
11:      Update follower  $i$  using equation 3.10
12:    end if
13:  end for
14:  Evaluate and update the food source (best position found so far)
15:  Update convergence curve
16: end while
17: return Best position found
```

3.4.3.1 SALP Swarm algorithm for node localization in wireless sensor networks

In [8], SSA was applied to solve the problem of precise node localization in WSNs. The authors compared the SSA-based approach with some of the most popular swarm intelligence methods, namely PSO, FA, Butterfly Optimization Algorithm (BOA), and GWO. The findings showed that SSA significantly boosted the accuracy of node positioning, reduced the computation time, and increased the number of nodes that were successfully localized compared to other algorithms. Moreover, the research emphasized how crucial precise localization is for improving the overall efficiency of WSNs, positioning SSA as a more effective method for achieving this goal.

3.4.3.2 Localization by Salp Swarm Optimization with Doppler Effect in Wireless Sensor Networks

In this paper [103], the authors presents a novel node localization approach for WSNs (WSNs) by integrating the Salp Swarm Optimization Algorithm with the Doppler Effect. This method, called LOSSADE (Localization by Salp Swarm Optimization Algorithm with Doppler Effect), takes advantage of the natural swarming behavior of salps along with the principles of the Doppler effect to continuously improve the accuracy of localization. In their hybrid method, they imagine the lead salp and the target location (the prey) are like a moving observer and a sound source. As the lead salp moves, the "frequency"

of the signals between it and the prey changes. This change in frequency gives the algorithm intelligent information about the salp's speed and direction relative to the target. They use this Doppler frequency to create a new, smarter formula for updating the salp's position. Instead of just moving randomly, the salp's movement is now scaled by this frequency, which helps guide it more directly. The localization model starts with computing unknown node positions via trilateration, then enhanced it by the optimization process that minimizes localization error. The simulation results indicate that LOSSADE outperforms other algorithms like SSA, GWO, BSA, EHO, and DV-Hop by achieving lower localization errors, quicker convergence, and a higher localization rate.

3.4.3.3 A novel approach of localization with single mobile anchor using quantum-based Salp swarm algorithm in wireless sensor networks

This study [104] introduces a novel range-based localization strategy for WSNs in smart city mobility scenarios. The core of the proposed method leverages a single mobile anchor node that traverses the network area following a deterministic Hilbert space-filling curve. This specific trajectory is chosen for its properties of providing comprehensive coverage without path repetition, thereby mitigating beacon signal collisions and reducing the impact of Line-of-Sight propagation issues. The localization process is initiated when a target node enters the communication radius of the mobile anchor. The distance between them is first estimated using the Received Signal Strength indicator. To enhance localization accuracy and overcome the limitation of having only a single physical anchor, the method projects six virtual anchor nodes in a ring formation around the anchor node, each separated by a 60-degree angle from the anchor. This configuration allows for the formation of an equilateral triangle between the mobile anchor and a pair of virtual anchors, enabling the application of the trilateration method to compute an initial centroid estimate of the target's position. Subsequently, the estimated position is refined using a metaheuristic optimization algorithm to minimize localization error. The proposed method employs the Quantum-based Salp Swarm Algorithm. QBSSA is utilized to optimize the estimated coordinates by stochastically dispersing a population of salp particles around the calculated centroid and iteratively converging towards the most probable location of the target node. The performance of this integrated system was evaluated through simulation and benchmarked against established techniques like PSO and H-best Particle Swarm Optimization. The results demonstrate that the proposed QBSSA-based outperformed both PSO and HPSO by achieving higher accuracy, faster computation, and better localization coverage. It achieved a localization error of 29%, significantly outperforming PSO (45%) and HPSO (38%)., localized about twice as many nodes, and lowered computation time

to 0.0294 seconds, proving it to be the most efficient and reliable approach among the compared algorithms.

3.4.3.4 Large scale WSNs node localization based on salp swarm algorithm using density peak clustering strategy

The article [105] titled “Large-scale WSNs Node Localization Based on Salp Swarm Algorithm Using Density Peak Clustering Strategy” presents a comprehensive framework that tackles one of the central issues in large-scale WSNs (WSNs): achieving accurate node localization while maintaining low energy consumption and communication overhead. The study integrates two complementary mechanisms—an improved Density Peak Clustering (IDPC) approach and an enhanced Salp Swarm Algorithm (ISSA)—to improve scalability, convergence, and localization precision. The authors propose a block-based non-ranging localization model that adaptively partitions the network into subregions of equal radius. This adaptive division minimizes total communication hops by balancing intra-region and inter-region data exchange. Through a detailed mathematical formulation, the paper derives an optimization function that determines the ideal number of subregions and the maximum communication hops per region, ensuring efficient energy distribution and reduced network congestion. Within each subregion, localization is modeled as an optimization problem, where the weighted distance error between location-aware and location-unaware nodes serves as the objective function. The improved salp swarm algorithm is then applied to solve this objective. However, the authors recognize the weaknesses of standard SSA, notably its tendency toward premature convergence when facing complex, multi-modal problems. To address this, the ISSA integrates clustering behavior inspired by density peak analysis. The IDPC component adaptively determines hyperparameters through an anomaly-based cutoff distance index and a two-stage approximation process, which enhances clustering stability and accelerates computation. This mechanism dynamically identifies leader and follower agents within the salp population, allowing the algorithm to better balance exploration and exploitation during the search process. The proposed ISSA-based localization method showed clear improvements over existing algorithms. It achieved faster convergence and significantly lower localization errors—reducing them by about 65.83% and 23.93% compared to benchmark methods—while maintaining stable accuracy even as network size, noise, and communication parameters varied. Though slightly more computationally demanding, it delivered consistently higher precision across all large-scale WSN scenarios.

3.4.3.5 DV-Hop Algorithm Based on Multi-Objective Salp Swarm Algorithm Optimization

Over the past few years, localization has remained one of the most persistent challenges in WSNs. This paper [106] tackles that issue by presenting an improved approach known as the Improved Multi-Objective Salp Swarm Algorithm DV-Hop (IMSSA-DV-Hop). In essence, the work builds upon the well-known DV-Hop technique — a range-free method appreciated for its simplicity — but one that still struggles to deliver high positioning accuracy. The authors attempt to close this gap by embedding a multi-objective optimization framework into the DV-Hop process, supported by an enhanced version of the Salp Swarm Algorithm (SSA). The IMSSA-DV-Hop algorithm features a subdivided hop count that better reflects the distances between nodes. It also uses a weighted average hop distance, adjusted by a minimum mean-square error criterion, and shifts the traditional single-objective DV-Hop model into a multi-objective optimization challenge. The multi-objective salp swarm algorithm (MSSA) is refined with a solid point-set initialization strategy, fuzzy selection methods in the solution repository, Levy flight for updating leader positions, and adaptive follower update strategies inspired by mayfly mating behavior. In order to evaluate its performance, the authors model two network topologies, including a square random and C-shaped random deployment. In both, IMSSA-DV-Hop has lower localization errors than classical DV-Hop and its variants which are better, including SSA-DV-Hop, GWO-DV-Hop, and ISSA-DV-Hop. The error reduction depending on the network conditions such as transmission range, node density and number of beacon nodes is in the range of 19 to 56 percent. In general, the results suggest that the IMSSA-DV-Hop algorithm does not only improve accuracy but has a good stability in different deployment conditions, which is why it is a viable improvement to offer to the current WSN application.

3.4.3.6 Application on Target Localization Based on Salp Swarm Algorithm

This study [107] introduces a new approach to target localization for Time-Difference-of-Arrival (TDOA) passive positioning problems, where the Salp Swarm Algorithm (SSA) is employed as the main optimization tool. The central goal is to enhance both the accuracy and the convergence reliability of source localization, addressing the well-known shortcomings of traditional and intelligent optimization methods such as PSO and its improved variant (IPSO). To achieve this, the paper first develops a clear mathematical representation of the TDOA-based localization problem. It models the measurements under Gaussian noise and formulates the estimation task within a maximum likelihood framework. The SSA is then adapted to this nonlinear optimization environment by

generating an initial population within defined search limits, allowing each salp to iteratively adjust its position, and progressively selecting the most promising candidates—or “food sources”—according to their fitness values. A series of simulation experiments are conducted to evaluate performance. Findings indicate that although PSO and IPSO converge faster during the initial stages, they tend to converge early and slow down during the latter stages. SSA on the other hand has a slower initial convergence but has a more constant and predictable improvement over the course of the process, and reaches higher fitness values. This indicates the successful global search and local refinement of SSA. In terms of localization accuracy, SSA is always better in comparison to the rest of the algorithms, particularly when the noise of measurements rises.

3.4.4 FireFly Algorithm (FA)

The algorithm (FA) was originally proposed by Xin-She Yang [11]. The algorithm takes inspiration from the mesmerizing flashing patterns and behaviors displayed by fireflies during the night. The FA is basically based on three main principles: First, all fireflies in the algorithm are considered unisex, meaning that any firefly can be attracted to any other firefly that seems brighter, irrespective of their gender. Secondly, the brightness of a firefly is assessed by evaluating an objective function, which acts as a measure to quantify its suitability or efficiency in solving the given problem. Lastly, the attractiveness between fireflies is directly correlated with their brightness, and both quantities steadily decrease as the distance between fireflies grows. The FA utilizes the principles to guide the search for an optimal solution to an optimization problem. The pseudocode of this technique is presented in **Algorithm 4** below. The position update of a firefly i attracted to a brighter firefly j is given by:

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \epsilon_i \quad (3.11)$$

where:

- β_0 is the attractiveness at distance $r = 0$,
- γ is the light absorption coefficient,
- r_{ij} is the distance between firefly i and j ,
- α is a random parameter,
- ϵ_i is a vector of random numbers drawn from a uniform distribution.

Algorithm 4 The pseudocode of FA

```
1: Initialize a population of fireflies with random positions
2: Evaluate the brightness (objective value) of each firefly
3: while (termination condition not met) do
4:   for each firefly  $i$  do
5:     for each firefly  $j$  do
6:       if firefly  $j$  is brighter than firefly  $i$  then
7:         Move firefly  $i$  towards  $j$  using equation 3.11
8:       end if
9:     end for
10:  end for
11:  Update brightness values of all fireflies
12: end while
13: return Best solution found
```

3.4.4.1 A K-means based firefly algorithm for localization in sensor networks

Liu and Chen proposed a K-means-based Firefly Algorithm (KFA) [108] aimed at enhancing localization accuracy in sensor networks. This research expands upon the multilateration algorithm (MA), characterized by its simplicity and hardware efficiency, yet it is limited by poor convergence and accuracy. The authors proposed a weighted K-means method to enhance the fitness function of the MA and mitigate the impact of significant distance errors among anchor nodes. Additionally, an inertia weight factor was introduced to modulate the attractiveness of fireflies and improve search speed, while a novel update strategy was formulated to enhance global exploration and mitigate premature convergence. Simulation experiments were performed to assess the KFA in comparison to MA, improved PSO (IPSO), improved GA (IGA), and standard FA. The results indicate that the proposed algorithm exhibits superior convergence, enhanced stability, and reduced localization error relative to alternative algorithms. The KFA demonstrated consistent performance across different configurations of anchor nodes, communication ranges, and measurement errors, indicating robustness in both sparse and dense network environments. The study demonstrates that the KFA significantly enhances localization precision and computational efficiency in sensor network applications.

3.4.4.2 Node Localization in Wireless Sensor Network using Firefly Algorithm

The article [109] examines the utilization of the Firefly Algorithm (FA) for node localization in WSN. Precise localization of nodes is crucial for the performance of WSN, and the FA is introduced as a means to enhance both the accuracy and computing time of

node localization. In this algorithm, the localization error is used as a fitness function, aiming to minimize the computed distance between anchor nodes and nearby unknown nodes, as well as their true distance. The algorithm starts by initializing a population of fireflies, where each firefly represents a potential solution or location for a sensor node. Fireflies move towards brighter fireflies, which represent better solutions. The brightness is determined by the fitness function, with a lower localization error indicating brighter fireflies. Through optimizing the positioning of sensor nodes using the FA, the proposed approach endeavors to decrease the localization error and elevate the precision of node localization in WSNs.

3.4.4.3 Optimized Range-Free 3D Node Localization in Wireless Sensor Networks using Firefly Algorithm

Kumar and Kaur have introduced a range-free 3D node localization technique for WSNs using the FA combined with a Fuzzy Logic System (FLS) [110]. The study focuses on overcoming the nonlinearity between received signal strength (RSS) and distance—a key issue in localization accuracy—while keeping the hardware cost low. In their approach, nodes are randomly deployed across three layers: the top layer contains anchor nodes with known coordinates, while the middle and bottom layers contain target nodes. Edge weights between anchors and targets are calculated and optimized using FLS to model RSS-based uncertainty, then refined by the FA to minimize localization errors. Simulations were conducted in a 101010 unit 3D environment, using 40 target nodes and 20 anchors. The proposed RFFA + Fuzzy method was benchmarked against conventional localization algorithms such as Centroid, Weighted Centroid, and a Hybrid Particle Swarm Optimization (HPSO) combined with fuzzy logic. Results showed a significant reduction in average localization error (0.0283), outperforming all compared approaches.

3.4.4.4 Parallel Firefly Algorithm for Localization Algorithm in Wireless Sensor Network

Here, in [111], the authors suggest a method for localizing nodes in a WSN using measurements of received signal strength (RSS) and the Parallel Firefly Algorithm (PFA). The parallelized FA method is a revised iteration of the basic FA optimization algorithm. In the parallelized version, the fireflies are partitioned into subgroups, and each subgroup undergoes independent evolution using the FA optimization process. Every subgroup possesses its own fireflies and the solution that is closest to being the greatest. Once a predetermined number of iterations has been completed, communication is triggered between subgroups. Better fireflies from one subgroup are exchanged with poorer

fireflies from neighboring subgroups to improve the overall quality of the solution. The experimental results of the proposed algorithm show that the proposed method with a communication strategy increases accuracy in comparison with the original FA, PSO, and GA methods for finding the best solution.

3.4.4.5 RSS-Based Indoor Positioning Utilizing FireFly Algorithm in Wireless Sensor Networks

Amer in [59] has introduced a novel indoor positioning system that leverages the Received Signal Strength (RSS) measurements in ZigBee WSNs. The proposed system addresses the critical challenge of dynamic path loss exponent (PLE) in complex indoor environments by employing the Firefly Algorithm (FFA) for adaptive parameter estimation. Moreover, the system integrates Kalman Filtering to perform noise reduction on both the RSS data and barometric pressure sensor measurements, thereby enhancing the robustness of the positioning solution. A significant advantage of the proposed approach is that it operates effectively without the need for manual pre-data collection or exhaustive radio surveys, which are typically required by fingerprinting methods. The author has utilized the FFA to periodically estimate the PLE for each communication channel, which allows for a more accurate construction of the distance matrix between mobile and anchor nodes. Additionally, the barometer sensor found in commercial smartphones is employed to estimate the altitude of the mobile node, providing crucial vertical positioning information by calculating height based on atmospheric pressure changes. The experimental evaluation has demonstrated that the proposed algorithm can determine the position with an error of less than two and a half meters. The obtained results have verified that the hybrid FFA/KF method provides superior performance and achieves a lower root mean square error compared to traditional methods that rely on a constant PLE value.

3.4.4.6 Two Stage Wireless Sensor Node Localization Using Firefly Algorithm

The authors describe an approach based on the FA to improve accuracy in WSNs [112]. The proposed method operates in two main stages. In the first stage, sensor nodes are localized using four anchors placed at the corners of the 3D area. The FA is then used to minimize the error between the calculated and measured distances obtained from RSSI. In the second stage, these anchors are moved to new optimized positions to enhance geometry—ensuring wider angles between anchors and sensors and shorter distances to reduce RSSI error. A final localization is carried out using both the original and new anchor positions, creating a virtual network of 8 anchors. This combination not only refines position

estimates but also resolves the flip ambiguity problem, where a node's estimated position can be mirrored over a line of anchors, producing a false but mathematically plausible location. Simulations show that the FA achieves higher localization accuracy than methods like PSO and the Bat Algorithm, even when RSSI measurements are affected by Gaussian noise. By dynamically repositioning anchors, the algorithm delivers more precise results than setups that rely only on static anchors—even those using a larger number of them.

3.4.5 Artificial Bee Colony (ABC) Algorithm

The ABC algorithm is a bio-inspired optimization method inspired by the foraging behavior of honey bees. In this model, food sources represent possible solutions, and their nectar amount indicates the fitness or quality of those solutions. The algorithm involves three types of bees: employed, onlooker, and scout. Consider a space with D dimensions, where SN denotes the total number of food sources. The position of the i -th food source can be described as a vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, with $i \leq SN$.

The search begins with the scout bee phase, where initial solutions are generated randomly within the given limits as:

$$x_{ij} = x_{ij}^{\min} + rand(0, 1) \times (x_{ij}^{\max} - x_{ij}^{\min}) \quad (3.12)$$

where $x_{ij}^{\min}$ and $x_{ij}^{\max}$ are the lower and upper bounds, respectively.

Next, the employed bee phase explores neighboring solutions by adjusting parameters of the current solution according to:

$$v_{ij} = x_{ij} + \phi \times (x_{ij} - x_{kj}) \quad (3.13)$$

where x_i and x_k are two different food sources, and $\phi \in [-1, 1]$ is a random control coefficient. The new candidate solution v_i is compared to the original x_i using a greedy selection rule, which mean only the better (fitter) one is retained for the next cycle.

In the onlooker bee phase, bees select food sources based on their fitness probability using the roulette wheel method defined as:

$$p_i = \frac{fit(x_i)}{\sum_{j=1}^{SN} fit(x_j)} \quad (3.14)$$

where the fitness value of the i -th food source, denoted as $fit(x_i)$, is computed based on the objective function value F_i corresponding to the i -th food source. It is defined as:

$$fit(x_i) = \begin{cases} \frac{1}{1 + F_i}, & F_i \geq 0, \\ 1 + |F_i|, & F_i < 0, \end{cases} \quad (3.15)$$

They then apply the same update rule described in Equation (3.13) to improve these selected solutions. Solutions that fail to improve after a certain number of trials (called the limit) are abandoned and replaced by new random ones through the Equation (3.12).

3.4.5.1 Centroid Based Localization Utilizing Artificial Bee Colony Algorithm

The centroid localization technique (CLA) is a widely used technique for the localization of sensors in wireless sensor networks. It is an inexpensive and easy-to-implement technique that estimates the position of unknown sensor nodes based on the coordinates of reference nodes. However, this technique suffers from a significant localization errors. To address this issue, the ABC algorithm was utilized in this study [113] to evaluate the CLA in terms of localization error. The ABC algorithm is used to optimize the CLA's localization error. Simulation results reveal that ABC and DE-based CLA have a lower localization error than basic and PSO-based methods. In contrast to basic CLA, ABC-based CLA requires seven times more computation time.

3.4.5.2 A Connectivity Weighting DV-Hop Localization Algorithm Using Modified Artificial Bee Colony Optimization

In [114], the authors developed an enhanced DV-Hop algorithm for localizing unknown nodes (UNs) in WSNs, called the connectivity weighting DV-Hop, which is further improved by a modified ABC optimization. Traditional DV-Hop suffers from localization errors mainly due to inaccurate AHD (average hop distance) estimation and distance errors in multilateration. Therefore, the authors proposed three main improvements to address these issues. First, they calculate the AHD of anchor nodes using the minimum mean squared distance error instead of standard unbiased estimators, which gives more accurate distance estimates. Next, for unknown nodes, they did not just take the AHD from the nearest anchor as before. They developed a better method that uses information from several nearby anchors. Each anchor's importance (or weight) depends on how close it is and how well it is connected to other nodes. Closer and more connected anchors have a greater effect, giving a more balanced and realistic distance estimate. Finally, they modified the ABC algorithm for optimizing the multilateral localization. The standard ABC algorithm generates new solutions around the current food source and a randomly

Algorithm 5 Artificial Bee Colony (ABC) Algorithm

Require: Number of food sources SN , Limit for abandonment, $MaxIter$

```
1: Initialize the population of food sources  $x_i$  randomly using Eq. (3.12)
2: Evaluate the fitness  $fit(x_i)$ 
3: Initialize  $trial_i = 0$  for each food source
4:  $iter = 0$ 
5: while  $iter < MaxIter$  do
6:   Employed Bee Phase:
7:   for each employed bee do
8:     Generate a new candidate  $v_i$  using Eq. (3.13)
9:     Evaluate the fitness  $fit(v_i)$ 
10:    if  $fit(v_i) > fit(x_i)$  then
11:      Replace  $x_i$  with  $v_i$ 
12:       $trial_i = 0$ 
13:    else
14:       $trial_i = trial_i + 1$ 
15:    end if
16:  end for
17:  Onlooker Bee Phase:
18:  Calculate probability  $p_i$  for each food source using Eq. (3.14)
19:  for each Onlooker Bee do
20:    if  $rand(0,1) < p_i$  then
21:      Generate a new candidate  $v_i$  using Eq. (3.13)
22:      Evaluate the fitness  $fit(v_i)$ 
23:      if  $fit(v_i) > fit(x_i)$  then
24:        Replace  $x_i$  with  $v_i$ 
25:         $trial_i = 0$ 
26:      else
27:         $trial_i = trial_i + 1$ 
28:      end if
29:    end if
30:  end for
31:  Scout Bee Phase:
32:  for each food source  $i = 1$  to  $SN$  do
33:    if  $trial_i > Limit$  then
34:      Replace  $x_i$  with a new random solution using Eq. (3.12)
35:      Evaluate fitness  $fit(x_i)$ 
36:       $trial_i = 0$ 
37:    end if
38:  end for
39:  Memorize the best solution achieved so far ( $x_{best}$ )
40:   $iter = iter + 1$ 
41: end while
42: return  $x_{best}$ 
```

selected neighbor, which can limit the search space and lead to premature convergence. The authors changed the search equation so that it now uses a random individual from the entire population as the search center rather than a local neighbor. This modification substantially expands the search space and enhances global exploration capability. The authors tested their improved DV-Hop method through simulations and found that it performs better than other DV-Hop versions and optimization methods such as least squares (LS), iterative nonlinear least squares (INLS), and standard ABC, providing more accurate and reliable localization results.

3.4.5.3 Localization of Sensor Nodes in 3D Wireless Sensor Networks with a Single Anchor by an Improved Adaptive Artificial Bee Colony (iaABC) Algorithm

Locating sensor nodes precisely without utilizing numerous expensive physical GPS-equipped anchors is a major problem in three-dimensional WSNs, which this study [115] aims to solve. With just one anchor, the authors provide innovative solution that can determine the locations of one hundred target nodes. They achieve this by employing an "umbrella projection" technique, where the single anchor creates a ring of six virtual anchors around it. The core bio-inspired method employed is the iaABC algorithm. This algorithm is an enhanced version of the classic ABC, which mimics the foraging behavior of bees. The key weakness of the standard ABC is that its search direction is guided by a random parameter, which can be inefficient. The authors first made the algorithm "adaptive," dynamically adjusting its parameters to better balance the search between exploring new areas and refining good ones. Their major innovation, the "improved" iaABC, replaces the random parameter with an "intelligent coefficient." This smart parameter is calculated by a fuzzy PID controller that guides the search direction towards better solutions. Experimental results on the CEC 2022 benchmark suite demonstrate that iaABC surpasses state-of-the-art algorithms such as Co-PPSO, SSA, SCA, AFPA, and other ABC variants in convergence and robustness. When applied to 3D WSN localization with one fixed anchor and six virtual anchors, iaABC accurately located all 100 moving target nodes, achieving the lowest average positioning error among its competitors. Although slightly slower than the basic ABC, its substantial improvement in accuracy makes it a strong choice for scenarios where precision is more critical than speed.

3.4.5.4 Implementation of an Efficient Artificial Bee Colony Algorithm for Node Localization in Unmanned Aerial Vehicle Assisted Wireless Sensor Networks

This research [116] tackles the challenge of pinpointing the location of sensor nodes using a drone, formally known as an Unmanned Aerial Vehicle (UAV), to overcome challenges such as high deployment costs and low accuracy in GPS-denied or obstructed environments. The core innovation lies in a two-stage optimization strategy that significantly improves accuracy. The key insight is that the drone's flying height is not just an operational detail but a critical parameter that systematically influences all distance measurements. If the drone flies too low, signals can be obstructed; if it flies too high, small measurement errors are greatly magnified. Therefore, the process begins by optimizing this flying height before any attempt is made to locate the unknown sensors. In the first stage, the system is calibrated using a single reference node placed at a known location in the center of the area. The drone flies along its path, and the algorithm adjusts its hypothetical flying height in a computer model. It seeks the one optimal height where the distance estimates calculated from the drone's signals most closely match the actual, known distances to the reference point. This process identifies the perfect flying altitude that minimizes systemic measurement error. Once this ideal height is determined and set for the drone's flight path, the second stage begins. Now, the algorithm focuses exclusively on finding the unknown X and Y coordinates of the target sensor nodes. Using the pre-optimized height, the system calculates much more accurate horizontal distances from the drone to each unknown node. The proposed approach leverages metaheuristic optimization techniques, particularly the ABC algorithm, to minimize localization errors effectively. Comparative simulation results reveal that ABC consistently achieves lower Average Localization Error (ALE) and Root Mean Square Error (RMSE) than DEA and classical multilateration.

3.4.5.5 Artificial Bee Colony Algorithm Based Multi-Objective Node Placement for Wireless Sensor Network

R.K. Jena in [117] introduces a multi-objective optimization framework for efficient sensor node placement using the Artificial Bee Colony (ABCMO) algorithm. Recognizing that node placement directly influences coverage, connectivity, and energy consumption, the author formulates a model that simultaneously optimizes these conflicting objectives. The proposed method considers three sensor modes (X, Y, and Z), each with varying sensing and energy parameters, to ensure balanced monitoring of a two-dimensional field. By integrating multiple performance metrics—field coverage, energy consumption, and clustering efficiency—into a single fitness function, the ABCMO algorithm identifies Pareto-optimal

solutions that best satisfy diverse network requirements. An external archive is employed to store non-dominated solutions, maintaining diversity through a comprehensive learning strategy and crowding distance control. Simulation results on a 1010 grid of 100 nodes reveal that ABCMO achieves faster convergence and superior performance compared to PSO, ACO, and TPSMA. Overall, the study demonstrates that the ABCMO approach effectively enhances WSN deployment by maximizing coverage and minimizing energy use through an intelligent, adaptive optimization process.

3.4.6 Discussion

To summarize the research papers reviewed in this chapter, Table 3.1 compiles the essential information for each bio-inspired localization approach. The entries are organized by algorithm, outlining the core innovation introduced in each study, the deployment scenario considered (i.e., whether localization was carried out in a 2D or 3D environment and under static or mobile node conditions), the evaluation metrics employed (see abbreviations below), and the primary performance claim, which captures the main result demonstrating the method's effectiveness—such as improved accuracy, faster convergence, reduced computation time, or higher energy efficiency.

The evaluation metrics used across the surveyed works are abbreviated as follows: ALE (Average Localization Error), APT (Average Processing Time), CR (Convergence Rate), CT (Computation Time), EC (Energy Consumption), ET (Elapsed Time), LE (Localization Error), LR (Localization Rate), LSR (Localization Success Ratio), MLE (Mean Localization Error), NL (Number of Localized Nodes), and RMSE (Root Mean Square Error).

Table 3.1: A Comparative Synthesis of Bio-Inspired Localization Algorithms

PARTICLE SWARM OPTIMIZATION (PSO) & VARIANTS				
Algorithm & Citation	Core Innovation / Hybridization	Scenario (2D/3D, Mobile/Static)	Evaluation Metrics	Primary Performance Claim
Kulkarni et al. (2009) (PSO vs. BFA) [90]	Uses an iterative, distributed approach where settled nodes become new references	Static, 2D	NL, LE, CT	PSO is 3–4faster; BFA is more accurate.
Sharma et al. (2017) (PSO-IDV) [91]	DV-Hop enhanced with PSO and a hop-size correction factor.	Static, 2D	MLE, LE, CR	60-70% reduction in error compared to basic DV-Hop.
(Zain et al., 2014) BPSO/Modified BPSO [92]	Operates in a binary search space for efficiency.	Static, 2D	LE, CT	59% less computation time, 11% higher error.
(Sharma et al., 2017) CLPSO [9]	Employs Comprehensive Learning PSO (CLPSO) to avoid premature convergence.	Static, 2D	LR, NL, CT	Higher localization accuracy than standard PSO, but at the cost of significantly longer computational time.
Phoemphon et al. (2020) (NS-IPSO) [93]	Hybrid: Node Segmentation + Improved PSO	Static, 2D with obstacles	MLE, robustness	Highest accuracy, especially in obstructed environments (C, H, S-shapes). Achieves 59-91% lower error than other algorithms.
Kanwar et al. (2020) PSO for Displaced Nodes [94]	smart Re-localization work: Localizes only displaced nodes and uses promoted anchors to reduce problem complexity.	Static (that get displaced), 2D	ET, EC	81% less time and 67% less energy than traditional PSO DV-Hop.

Table 3.1 (Continued)

GREY WOLF OPTIMIZER (GWO) & VARIANTS				
Algorithm & Citation	Core Innovation / Hybridization	Scenario (2D/3D, Mobile/Static)	Evaluation Metrics	Primary Performance Claim
Zhang ET AL., (2020) GWO-LA [97]	Direct application of Grey Wolf Optimizer (GWO)	Static, 2D		11% lower average error than PSO.
Cui et al. (2022) (DASCAN & MDS-GWO) [98]	a two-mobile-anchor (DASCAN path) with an improved GWO (MDS-GWO) using multiple disturbance strategies.	mobile anchors, 2D	LE, EC, LR	30% less energy than 3-anchor paths and 25% higher accuracy than PSO/GWO.
Abuaddouet al., (RGWO)[99]	virtual anchor projection + incorporates Repulsion from worst solution	Mobile (single anchor), 3D	LE, CS	Lower error & faster convergence in 3D.
Jeyaseela et al. (2024) IGWONL-UWSN[100]	Improved Grey Wolf Optimizer (IGWO) with Dimension Learning-based Hunting (DLH)	Static, 3D	LN, LE	Outperforms existing methods
Xu (2024) IGWOA [101]	Hybrid: Improved DV-Hop + GWO enhanced with PSO concepts.	Static, 2D with obstacles	ALE, PA, RT, RunT	Outperforms traditional and other improved DV-Hop with ALE: of 0.20 m
Ghorpade et al.(2020) MOGWOLA [102]	Multi-Objective Grey Wolf Optimization (MOGWO) using Pareto optimal front to simultaneously minimize distance and geometric topology constraints.	2D, Smart Parking, Static	RMSE, LN, CT	Error Reduction: 26%, 20%, and 17% lower than GRNN, PSO, and DTBLA, respectively, Localized more nodes, and. Faster convergence

Table 3.1 (Continued)

SALP SWARM ALGORITHM (SSA) & VARIANTS				
Algorithm & Citation	Core Innovation / Hybridization	Scenario (2D/3D, Mobile/Static)	Evaluation Metrics	Primary Performance Claim
Kanoosh et al.(2019) SSA [8]	Direct application of SSA	Static, 2D	MLE, CT, NL	Higher accuracy & more localized nodes vs. PSO, FA, GWO.
Kathiroli et al. (2021) LOSSADE [103]	Hybridized SSA with Doppler Effect for smarter position updates.	Static, 2D	NL, LR CR, LE	Outperforms other metaheuristics (SSA, GWO, BSA, EHO) and DV-Hop
Rani et al. (2023) QB-SSA [104]	Quantum-based Salp Swarm Algorithm+ SINGLE ANCHOR+ VIRTUAL ANCHORS	Mobile anchor, 2D	NL, LE, ALE, CT	Mobile Efficiency: Outperforms PSO and HPSO in accuracy and speed.
Liu et al. (2025) ISSA-IDPC [105]	Hybrid: Improved SSA + Density Peak Clustering.	Static, 2D (large scale)	LE, ME, RMSE, CT	Scalability: 65% error reduction; stable in large networks.
Liu et al. (2023) IMSSA-DV-Hop [106]	Hybrid: Improved Multi-Objective Salp Swarm Algorithm (IMSSA) hybridized with DV-Hop	Static, 2D (square/Chaped)	ALE, PA, RT, RunT	Outperforms traditional and other improved DV-Hop with ALE: of 0.20 m

Table 3.1 (Continued)

FIREFLY ALGORITHM (FA) & VARIANTS				
Algorithm & Citation	Core Innovation / Hybridization	Scenario (2D/3D, Mobile/Static)	Evaluation Metrics	Primary Performance Claim
Liu et al., (2018) (KFA) [108]	Hybrid: K-means clustering + FA with inertia weight.	Static, 2D	LE (%), CS, Stb(std)	Higher localization accuracy, faster convergence, better stability than IPSO, IGA, FA
Kaur et al. (2018) FA [109]	Direct application of FA	Static, 2D	LE, CT, NL	Not compared against other algorithms.
Kumar et al., (2015) FLSFA [110]	Uses FLS to model edge weights from RSS and FA to optimize them	3D, Static (Anisotropic Environment)	LE, ALE	Achieves the lowest location error among the compared range-free methods.
Sai et al., (2015) pFA [111]	Subpopulations with communication	2D, Static	LE, NL	Outperforms GA, PSO, and FA in accuracy.
Amer et al. (2016) FFA + Kalman Filter [59]	FFA is used to adaptively estimate the PLE, while KF is used for sensor fusion and position prediction.	Indoor Mobile (Smart-phone)	3D, RMSE, LE, CDF of error	Position error < 2.5 meters, outperformed fixed PLE method
Tuba et al., (2018) Two-Stage FA [112]	A two-stage localization strategy using semi-mobile anchor nodes. The FA is applied to both node localization and anchor position optimization	3D, Static (Semi-Mobile Anchors)	ALE, Stb (Std), Min/Max Error	Position error < 2.5 meters, outperformed fixed PLE method

Table 3.1 (Continued)

Artificial Bee Colony (ABC) & VARIANTS				
Algorithm & Citation	Core Innovation / Hybridization	Scenario (2D/3D, Mobile/Static)	Evaluation Metrics	Primary Performance Claim
Gupta et al., (2019) ABC-based CLA [113]	Hybrid: CLA with ABC optimization to minimize localization error.	Static, 2D	LE CT	Reduces avg. localization error by 95% vs. basic CLA;
Chen et al. (2019) RABC [114]	hybrid of the Modified (DV-Hop + ABC (RABC) with random-center search to avoid prematurity).	Static, 2D	ALE, CT, CDF	Outperforms LS, INLS, and basic ABC
Ekmekci et al. (2025) iaABC [115]	Umbrella Projection Method + Improved Adaptive ABC with intelligent coefficient (ic) via fuzzy-PID	3D, Mobile	LE, CEC 2022 BFP	successfully locates 100 mobile nodes with a single anchor; better accuracy than ABC/aABC but slower.
jena (2014) ABCMO [117]	Multi-objective ABC with Pareto optimization, external archive, and comprehensive learning strategy	2D, Static	FC, EC, SPC, OCE, SOE	faster convergence; better energy efficiency and coverage; optimal node mode selection

From Table 3.1, it is evident that a predominant focus of bio-inspired localization research has been on static, two-dimensional (2D) deployments. Algorithms such as PSO [90]–[94], GWO [97], Salp Swarm Algorithm (SSA) [8], [106], [104], [103], and ABC [114], and others were initially tested in 2D environments.

While directly applying these algorithms often produced promising results [8],[91], [95], and [109], the most notable improvements arise from hybrid approaches that combine algorithmic strengths or integrate AI techniques to overcome challenges such as local optima, high-dimensional search spaces, and environmental obstacles.

Across the reviewed studies, enhanced bio-inspired algorithms consistently achieve lower localization errors than both their basic counterparts and traditional techniques such as trilateration, least-squares estimation, and standard DV-Hop. Some studies combine multiple bio-inspired algorithms to leverage their respective advantages. For example, the Improved Grey Wolf Optimizer with PSO concepts (IGWOA) [101] incorporates inertia weight and personal best into GWO to enhance convergence and avoid local optima. Other studies integrate AI or machine-learning components, such as LOSSADE [103], which employs the Doppler Effect to guide salp movements more intelligently.

Despite the significant advancements achieved through bio-inspired and hybrid localization approaches, several methodological limitations can be observed across the reviewed studies.

First, many works evaluate algorithm performance using a single execution or a very limited number of runs. Since metaheuristic algorithms are inherently stochastic, their results may vary across independent executions due to random initialization and probabilistic search mechanisms. Consequently, performance assessments based on only a few trials may not accurately reflect algorithm stability, convergence consistency, or overall reliability.

Second, a noticeable variation exists in the deployment configurations adopted in different studies. In many cases, anchor nodes and unknown nodes are randomly positioned for each algorithm independently, leading to non-identical spatial distributions across experiments. Because localization accuracy is strongly influenced by network geometry, anchor placement, and node distribution, results obtained under different deployment realizations cannot be directly compared. Variations in spatial configuration may significantly affect connectivity structure, distance relationships, and error propagation behavior, potentially introducing evaluation bias.

These observations indicate that, although bio-inspired methods demonstrate promising localization capabilities, a rigorous evaluation framework based on controlled deployment scenarios and statistically reliable performance analysis remains necessary.

3.5 Conclusion

To conclude this chapter, we presented the basic idea of optimization and gave an overview of the main bio-inspired algorithms used for localization in WSNs. From the studies reviewed, it is clear that these algorithms are widely used because they can explore the search space efficiently and provide good estimates of unknown node positions. Their flexibility and ability to deal with complex problems explain why they have become an important research direction in WSN localization. This chapter also showed how different researchers have applied these algorithms, either in their original form or through improved versions, to solve or enhance the localization process. The examples discussed here help to give a complete picture of how bio-inspired methods are being used in this field. However, the stochastic nature of metaheuristic algorithms and the influence of deployment configurations highlight the importance of using multiple independent runs and consistent experimental settings to obtain more reliable and comparable results.

Based on this background, the next chapter, we will introduce the two improved localization methods proposed in this thesis, both based on the Sea Lion Optimization (SLnO) algorithm. The proposed methods are evaluated using a unified deployment configuration and multiple independent experimental runs to ensure fair and consistent comparison with existing techniques. This allows a clearer demonstration of the effectiveness of the proposed approaches in WSNs localization.

Chapter 4

The Proposed Node Localization Methods

4.1 Introduction

Over the previous chapters, we discussed the fundamental concepts of WSNs and the importance of accurate localization. In Chapter 3, we reviewed different bio-inspired optimization algorithms and how they have been applied to solve the localization problem, while also identifying some limitations in existing studies.

To address these challenges, in this chapter we propose two improved localization methods based on the Sea Lion Optimization (SLnO) algorithm. The first method, SLnO-CLA, combines the Centroid Localization Algorithm with SLnO to help guide the search process and improve localization accuracy. The second method, IMP-SLnO, is an improved version of the standard SLnO algorithm, designed to further enhance performance and stability. In the following sections, we present the details of the proposed methods and evaluate their performance through simulations under the same experimental conditions and multiple independent runs.

4.2 Sea Lion Bio-inspired Optimization Technique

The Sea Lion Optimization Algorithm (SLnOA) is a bio-inspired metaheuristic optimization algorithm designed to emulate the distinctive hunting behaviors of sea lions. It was initially introduced by Raja Masadeh and associates in 2019 [118]. It's important to highlight that SLnOA is not the same as the Lion Optimization Algorithm (LOA), which was developed earlier in 2016 [119]. In this section, we'll break down the core mechanisms of SLnOA and share its pseudo-code in Algorithm 6.

Algorithm 6 Sea Lion Optimization Algorithm (SLnO)

```

Initialize Sea Lions Population  $X_i$  ( $i = 1, 2, 3, \dots, n$ )
Calculate the fitness of each search agent and select the best agent
 $X_{best} = \text{best search agent}$ 
while Current iteration < Max iterations do
    Update parameter  $C$  (decreases linearly from 2 to 0)
    Calculate Leader Sound using Eqs. ((4.3)-(4.5))
    for each search agent do
        if Leader Sound < 0.25 then
            if  $|C| < 1$  then
                Update the position of the current search agent by Eq. ((4.2))
            else
                Select a random search agent ( $X_{rand}$ )
                Update the position of the current search agent by Eq. ((4.8))
            end if
        else
            Update the position of the current search agent by Eq. ((4.6))
        end if
        Check if any search agent goes beyond the search space
        Calculate the fitness of each search agent
        Replace the old solution if the new solution has better fitness.
    end for
    Update  $X_{best}$  with the best current solution.
    Current iteration = Current iteration + 1
end while
return  $X_{best}$ 

```

4.2.1 Detecting and Tracking Phase

Sea lions can sense where a prey is and then signal the rest of their group to help with the chase. The one that spots the prey first naturally takes the lead. In SLnOA, this idea is modeled by treating the best solution found at the current iteration as the “prey,” meaning it represents the most promising point in the search space. Once this best search agent is identified, the remaining agents adjust their positions relative to it. This process is described using the following equations:

$$\vec{\text{Dist}} = \left| 2\vec{B} \cdot \vec{P}(t) - \vec{SL}(t) \right| \quad (4.1)$$

$$\vec{SL}(t+1) = \vec{P}(t) - \vec{\text{Dist}} \cdot \vec{C} \quad (4.2)$$

where t and $t+1$ refer to the current and next iterations. The vector $\vec{C}$ decreases linearly from 2 to 0 throughout the algorithm, and $\vec{B}$ is a random vector within $[0, 1]$.

Here, $\vec{P}(t)$ is the position of the best solution so far, $\vec{SL}(t)$ is the position of a sea lion agent, and $\vec{Dist}$ represents the distance between them. Using Eq. ((4.2)), each agent moves closer to the best candidate, allowing the population to search around the most promising area.

4.2.2 Vocalization Phase

As noted earlier, the sea lion that first detects the prey becomes the leader of the hunt. To gather the rest of the group, it produces vocal signals while moving toward the target. SLnOA represents this behavior mathematically using Eqs. ((4.3)), ((4.4)), and ((4.5)):

$$\vec{SP}_{\text{leader}} = \left| \frac{\vec{V}_1(1 + \vec{V}_2)}{\vec{V}_2} \right| \quad (4.3)$$

$$\vec{V}_1 = \sin \theta \quad (4.4)$$

$$\vec{V}_2 = \sin \phi \quad (4.5)$$

In this model, $\vec{SP}_{\text{leader}}$ represents the leader's sound speed. The term $\vec{V}_1$ corresponds to the sound generated underwater, while $\vec{V}_2$ refers to sound produced in air. The value $\sin \phi$ represents the portion of the sound reflected in the air, and $\sin \theta$ describes the part refracted in the water. The angles ϕ and θ denote the reflection and refraction angles.

Attacking Phase (Exploitation Phase)

To model how sea lions attack their prey, SLnOA considers two main behaviors:

a. Dwindling encircling technique: This behavior is controlled by the parameter $\mathbf{C}$ in Eq. ((4.2)). As the iterations progress, $\mathbf{C}$ decreases linearly from 2 down to 0. When this value becomes smaller, the leader moves closer to the prey, trying to surround it. Because of this, a sea lion (search agent) can update its position anywhere along the line between its current location and the position of the best agent.

b. Circle-based position updating: In nature, sea lions often attack schools of fish by circling around them and striking from the edges. SLnOA captures this action using Eq. ((4.6)). Here, $|\vec{P}(t) - \vec{SL}(t)|$ is the distance between the agent and the best solution (the prey), and m is a random number in the range $[-1, 1]$:

$$\vec{SL}(t+1) = |\vec{P}(t) - \vec{SL}(t)| \cdot \cos(2\pi m) + \vec{P}(t). \quad (4.6)$$

Searching for Prey (Exploration Phase)

In contrast to the attacking phase, where sea lions adjust their positions based on the best agent, during exploration each agent moves according to a randomly chosen sea lion. This encourages the group to look for new prey, or in optimization terms, to explore different regions of the search space. When the parameter $\mathbf{C}$ is larger than 1, the algorithm performs a wide-ranging search to help find the global best solution. The process is described by the following equations:

$$\text{Dist} = \left| 2\vec{B} \cdot \vec{SL}_{\text{rand}} - \vec{SL}(t) \right|, \quad (4.7)$$

$$\vec{SL}(t+1) = \vec{SL}_{\text{rand}} - \text{Dist} \cdot \vec{C} \quad (4.8)$$

where $\vec{SL}_{\text{rand}}$ is a randomly selected sea lion from the current population, $\mathbf{SL}(t)$ is the current position of the agent, and Dist measures the distance between them.

4.3 The proposed SLnO-CLA node localization

To achieve accurate node position estimation in WSNs, this thesis proposes SLnO-CLA, a hybrid localization approach that combines the computational simplicity of the Centroid Localization Algorithm (CLA) with the optimization capability of the Sea Lion Optimization (SLnO) algorithm.

In the proposed SLnO-CLA framework, the conventional CLA (described in detail in Chapter 2, Section 4) is first applied to obtain an initial estimate of the target node's coordinates based on the positions of neighboring anchor nodes. This initial estimate provides a coarse but computationally efficient approximation of the node location.

The SLnO algorithm is then employed to refine and optimize this initial estimate by minimizing a localization objective function based on distance measurements between the target node and its neighboring anchors. In this refinement phase, the CLA-based position is used as the center of the SLnO search space, around which the population of sea lion agents is initialized and iteratively updated to obtain a more accurate location estimate.

The mathematical formulation of the localization problem, including the system architecture, distance calculation model, optimization objective, and the detailed step-by-step SLnO-CLA localization procedure, is presented in the following section.

4.4 Localization Model

4.4.1 System Architecture

Our study focuses on a monitoring region that is square, with dimensions $X \times Y$. A total $N = M + T$ sensor nodes are scattered randomly across this region. Among them, M nodes know their exact locations and are referred to as anchor nodes. The positions of the remaining T nodes are unknown and are estimated through the localization process.

Every sensor node is assumed to have the same transmission range, denoted by R , which means it can send and receive data to any node within a distance of R units. The M anchors periodically broadcast messages containing both their positions and unique node IDs. This allows nearby unknown nodes to identify which anchors they can receive signals from. Each anchor node is assigned its ID at the time of deployment. When an unknown node receives these broadcast messages, it measures its distance from each anchor.

4.4.2 Distance Calculation Model

Each unknown node estimates its distance to the anchor nodes within its communication range. Following a common assumption in many WSN studies, these distances are estimated using an RSSI-like model. Instead of real hardware measurements, the true distances are computed and Gaussian noise is added to reflect the uncertainty normally found in RSS-based ranging. This approach allows realistic measurement errors to be simulated in a simple and controlled way.

For an unknown node positioned at (x, y) and an anchor node i located at (x_i, y_i) , the actual distance is computed using the Euclidean formula:

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}. \quad (4.9)$$

The corresponding measured distance $\hat{d}_i$ is simulated by adding zero-mean Gaussian noise:

$$\hat{d}_i = d_i + n_i, \quad (4.10)$$

where n_i is a random variable drawn from a normal distribution $N(0, \sigma_i^2)$. The noise variance σ_i^2 is chosen to be proportional to the true distance, reflecting the practical observation that measurement uncertainty tends to increase as nodes are farther apart:

$$\sigma_i = \gamma \cdot d_i, \quad (4.11)$$

with γ representing the noise factor. This approach ensures that the distance estimates realistically reflect measurement uncertainties, providing the optimization algorithm with input data similar to what would be obtained in an actual deployment.

4.4.3 Localization Optimization Formulation

The localization problem is formulated as an optimization task in which the objective is to minimise the discrepancy between the measured distances and the distances computed from candidate coordinates. Let (x_i, y_i) represent the candidate location of the unknown node, and let (x_j, y_j) denote the coordinates of the j -th anchor within its communication range. The objective function is defined as the mean squared error over all available anchors:

$$\text{Obj}(x_i, y_i) = \frac{1}{M} \sum_{j=1}^M \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \hat{d}_{ij} \right)^2. \quad (4.12)$$

Only cases with $M \geq 3$ are considered, as an unknown node requires at least three anchors within its transmission range to be localizable. The coordinate pair (x_i, y_i) that yields the smallest value of this objective function is taken as the estimated position of the unknown node.

4.4.4 The detailed SLnO-CLA node localization

The proposed SLnO-CLA localization framework operates according to the following sequence of steps and the overall flowchart is illustrated in Figure 4.1.

Step 1: Network Deployment N target nodes with unknown positions and M anchor nodes with known coordinates are randomly deployed over the monitoring region. All nodes are assumed to have an identical communication range R .

Step 2: Distance Measurement For each target node, distance measurements to neighboring anchor nodes are obtained according to the distance calculation model described in Section 4.4.2. These measurements incorporate Gaussian noise to emulate RSSI-based ranging uncertainty.

Step 3: Localizability Check A target node is considered localizable if it has at least three anchor nodes within its communication range.

Step 4: Initial Estimation (CLA) For each localizable node, the Centroid Localization Algorithm (CLA) is applied to compute an initial position estimate based on the coordinates of the neighboring anchors.

- Step 5: Population Initialization** The initial population of sea lion agents is generated around the CLA-based estimate, which serves as the center of the search region.
- Step 6: Optimization (SLnO)** The Sea Lion Optimization (SLnO) algorithm refines the initial estimate by minimizing the localization objective function defined in Eq. (4.12), which quantifies the mismatch between measured and estimated distances to the anchors.
- Step 7: Best Solution Selection** After reaching the maximum number of iterations, the best-performing sea lion agent is selected as the estimated position of the target node.
- Step 8: Reference Set Update** Once a target node is successfully localized, its estimated position is added to the reference set and can be used as an additional anchor in subsequent localization iterations.
- Step 9: Iterative Localization** Steps (3) through (8) are repeated until all target nodes are localized or no further nodes satisfy the localization condition.
- Step 10: Performance Evaluation** The localization process is evaluated using the Mean Localization Error (MLE) and the Localization Success Ratio (LSR), which are presented and discussed in the experimental results section.

4.5 Experimental Results for SLnO-CLA

In this section, the performance of the proposed SLnO-CLA localization algorithm is evaluated and compared with the classical Centroid Localization Algorithm (CLA). The main objective of this comparison is to demonstrate the benefit of integrating the Sea Lion Optimization (SLnO) algorithm with CLA in order to improve localization accuracy and reliability under identical network conditions.

In all simulations, the deployment of anchor and target nodes is identical for the two algorithms to ensure a fair comparison. Therefore, any observed differences in performance can be attributed solely to the localization algorithms themselves.

4.5.1 Evaluation Metrics

The following performance metrics are used to evaluate and compare the proposed localization algorithms.

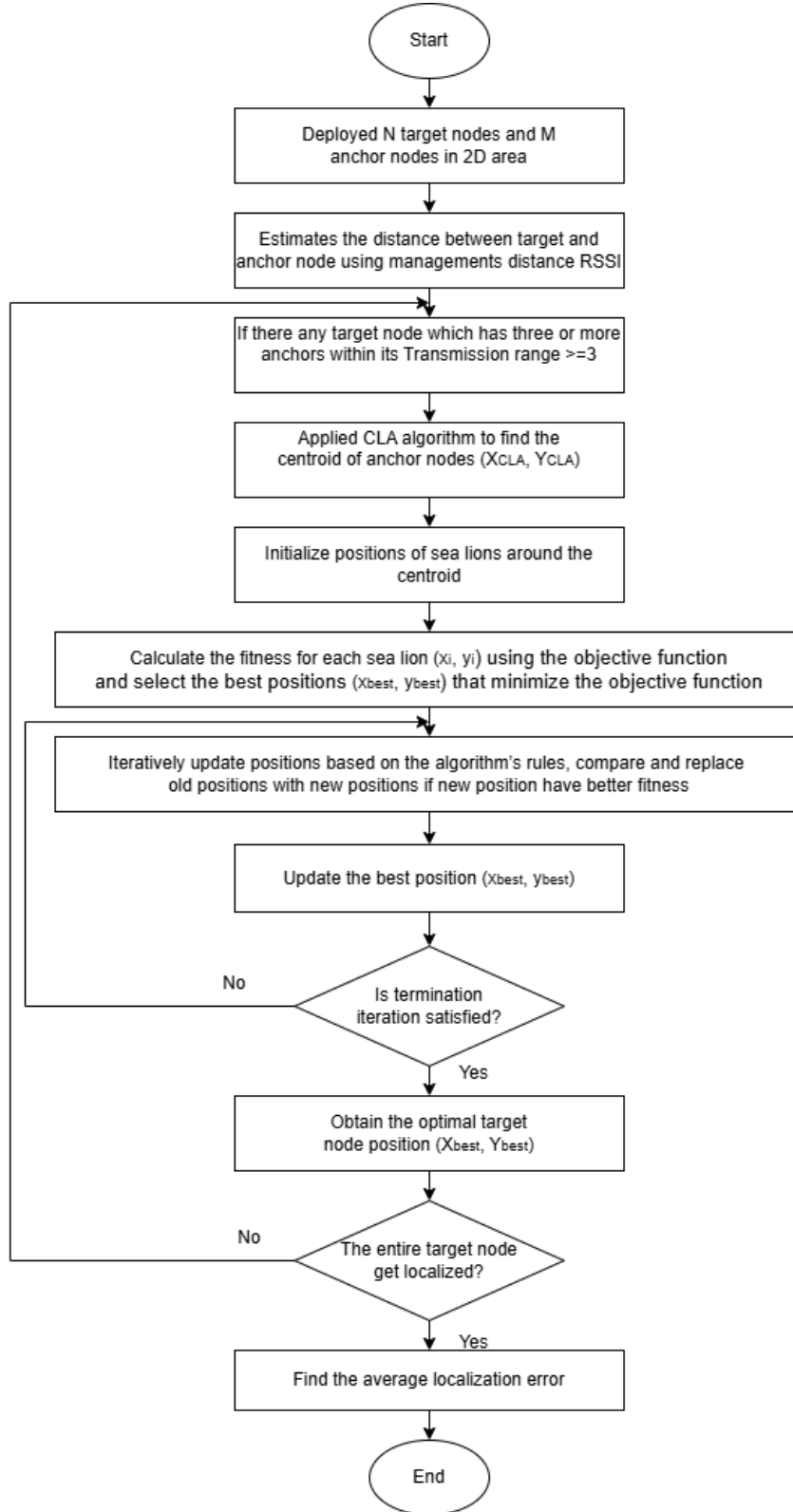


Figure 4.1: Flow chart of SLnO-CLA for node localization process.

4.5.1.1 Localization Error (LE)

Localization Error (LE) measures the accuracy of the estimated position of each target node. It is defined as the Euclidean distance between the true position and the estimated position of a target node, as given in Equation (4.13):

$$LE_i = \sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2} \quad (4.13)$$

where (x_i, y_i) represents the true coordinates of the i -th target node, and (X_i, Y_i) denotes its estimated position.

4.5.1.2 Mean Localisation Error (MLE)

MLE reflects the overall localization accuracy of the algorithm. It is computed as the average of the localization errors over all successfully localized nodes, as expressed in Equation (4.14):

$$MLE = \frac{1}{N_L} \sum_{i=1}^{N_L} \sqrt{(X_i - x_i)^2 + (Y_i - y_i)^2} \quad (4.14)$$

where N_L is the number of localized nodes, (x_i, y_i) denotes the true position, and (X_i, Y_i) represents the estimated position of each target node.

4.5.1.3 Localisation Success Ratio (LSR)

LSR indicates the percentage of target nodes that are successfully localized and is calculated using Equation (4.15):

$$LSR = \frac{\text{Number of target nodes localized}}{\text{Total number of target nodes}} \times 100\% \quad (4.15)$$

A higher LSR value indicates better localization capability and reliability.

4.5.2 Simulation Parameters

The proposed localization approach is implemented in MATLAB R2013b on a computer equipped with an Intel Core i5 processor, 8 GB RAM, and running Windows 10. The simulation parameters used in all experiments are summarized in Table 4.1.

Table 4.1: Simulation Parameters

Parameter	Value
Simulation area	$100 \times 100 \text{ m}^2$
Transmission range (R)	25 m
Anchor nodes	25
Target nodes	100
noise	2%
population size	30
Maximum iterations	50

4.5.3 Experimental Analysis

To analyze localization accuracy, a two-dimensional WSN consisting of 100 target nodes and 25 anchor nodes is considered. All nodes are randomly deployed within a $100 \times 100 \text{ m}^2$ area, the transmission range of each node is fixed at 25 m, and a 2% measurement noise level is considered. The population size along with the maximum iteration are 30 and 50, respectively.

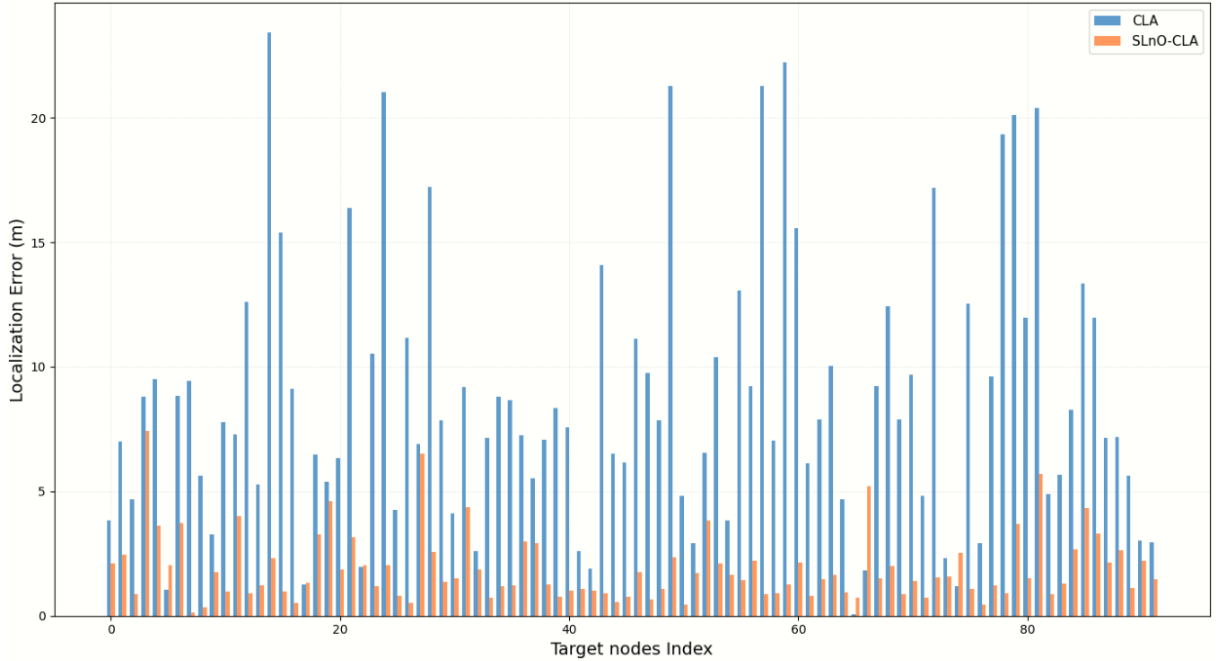
**Figure 4.2:** Comparison of localization error for each target node for CLA and SLnO-CLA.

Figure 4.2 presents the localization error of each target node obtained using CLA and SLnO-CLA. It can be seen that CLA produces large localization errors for many target nodes. In several cases, the error is greater than 10 m, and for some nodes it exceeds 20 m. High error values are observed at target nodes 16, 26, 55, 64, 67, 87, and 89, where

the localization error reaches approximately 23.44 m, 21.03 m, 21.28 m, 21.26 m, 22.21 m, 19.34 m, and 20.39 m, respectively.

On the other hand, SLnO-CLA shows much better accuracy for almost all target nodes. The localization error of SLnO-CLA remains below 7.5 m for all nodes, and most errors lie in the range of 0.5–3 m. For the nodes where CLA shows the largest errors (nodes 16, 26, 55, 64, 67, 87, and 89), SLnO-CLA reduces the error to about 2.32 m, 2.03 m, 2.35 m, 0.89 m, 1.28 m, 3.70 m, and 5.71 m, respectively.

The visualization of the real positions of target and anchor nodes, as well as the estimated positions using CLA and SLnO-CLA algorithms, is shown in Figures 4.3 and 4.4, respectively. The connecting purple lines are used to highlight the significant error between the real positions and the estimated ones.

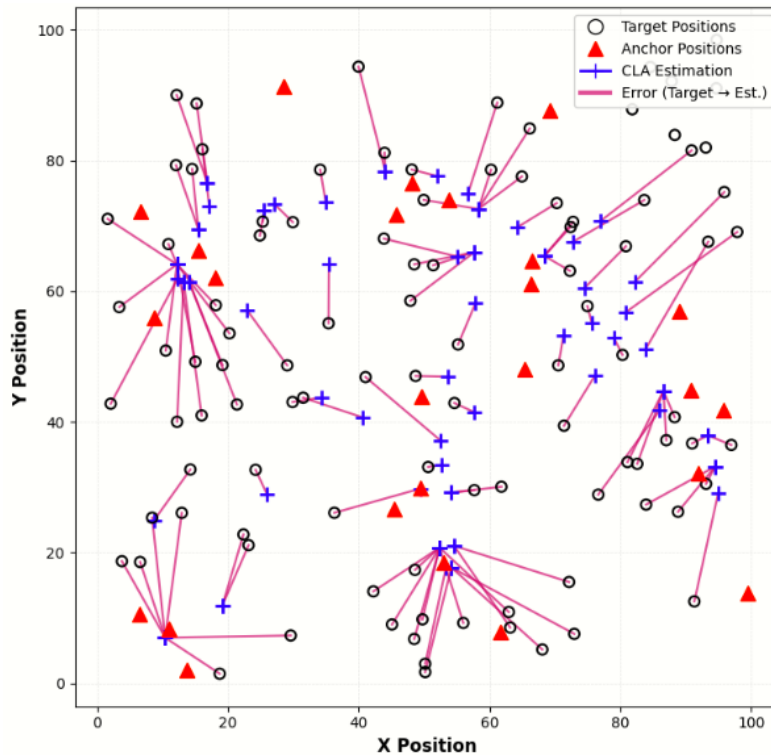


Figure 4.3: Position estimated by CLA algorithm

To provide a quantitative performance comparison between the classical CLA and the proposed SLnO-CLA, a comprehensive statistical analysis across 30 runs is presented in Table 4.2. This table reports the Mean Localization Error (MLE) and Localization Success Ratio (LSR) for both algorithms. In addition, the variations of MLE and LSR are illustrated graphically in Figures 4.5 and 4.6 to better visualize the performance differences between the two methods.

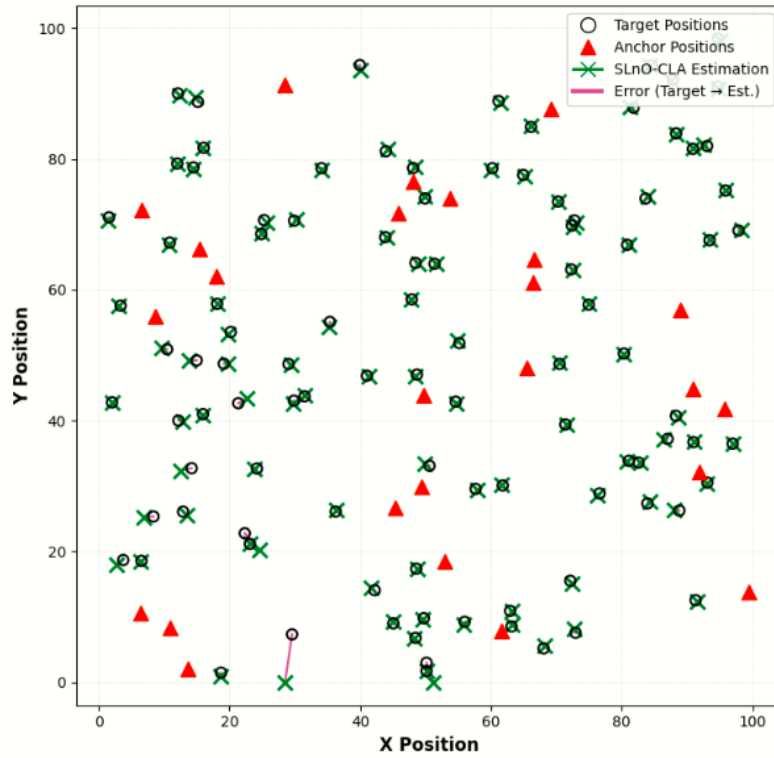


Figure 4.4: Position estimated by SLnO-CLA.

Metric	CLA	SLnO-CLA
Mean Error	10.0256	0.6339
Success Rate	94.7%	99.9%

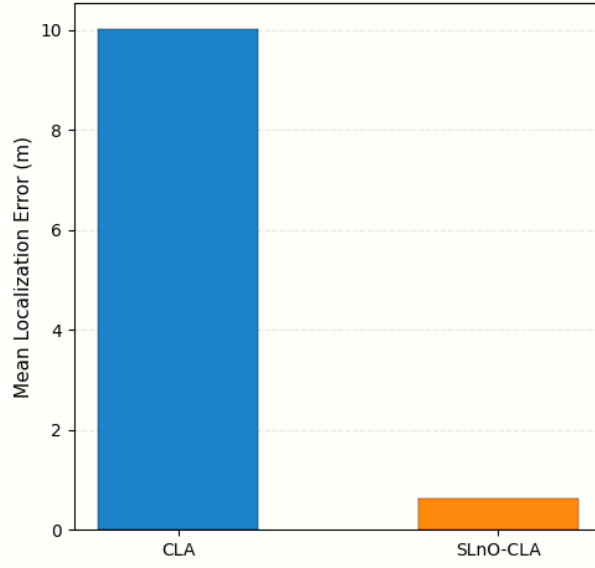
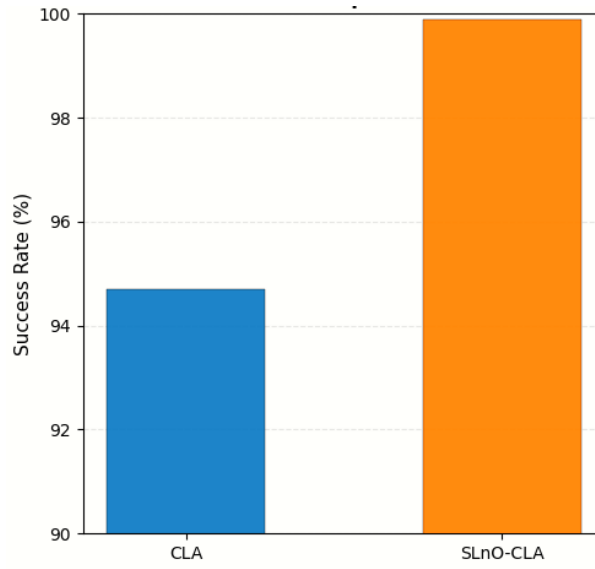
Table 4.2: Performance comparison between CLA and SLnO-CLA algorithms

From Table 4.2 and Figure 4.5, it can be noticed that CLA produces a high MLE of 10.03 m. In contrast, SLnO-CLA achieves a significantly lower MLE of 0.6339 m. This represents a remarkable error reduction of 93.7% compared to CLA.

Regarding the LSR shown in Figure 4.6, both algorithms achieved good results in terms of the percentage of localized nodes. However, SLnO-CLA algorithm attained a higher LSR of 99.9% compared to CLA with LSR of 94.7%.

To study the effect of the number of anchor nodes, we tested different anchor counts using 100 sensor nodes deployed in a 100 m 100 m area with a communication radius of 25 m. The number of anchors starts at 25 and increases by 3 up to 40. The resulting MLE for each anchor setting is shown in Table 4.3 and Figure 4.7.

From Figure 4.7, it can be observed that the MLE decreases as the number of anchor nodes increases. This behavior is expected, since having more anchors increases the probability that each target node is connected to three or more nearby anchors, which

**Figure 4.5:** MLE for both CLA and SLnO-CLA**Figure 4.6:** LSR for both CLA and SLnO-CLA algorithms

	MLE (m)	SLnO-CLA	CLA
A=25		0.6339	10.0256
A=28		0.6175	9.3399
A=31		0.6245	9.8468
A=34		0.6419	9.5019
A=37		0.4577	9.2807
A=40		0.3421	8.7328

Table 4.3: The MLE for SLnO-CLA and CLA under different numbers of anchors

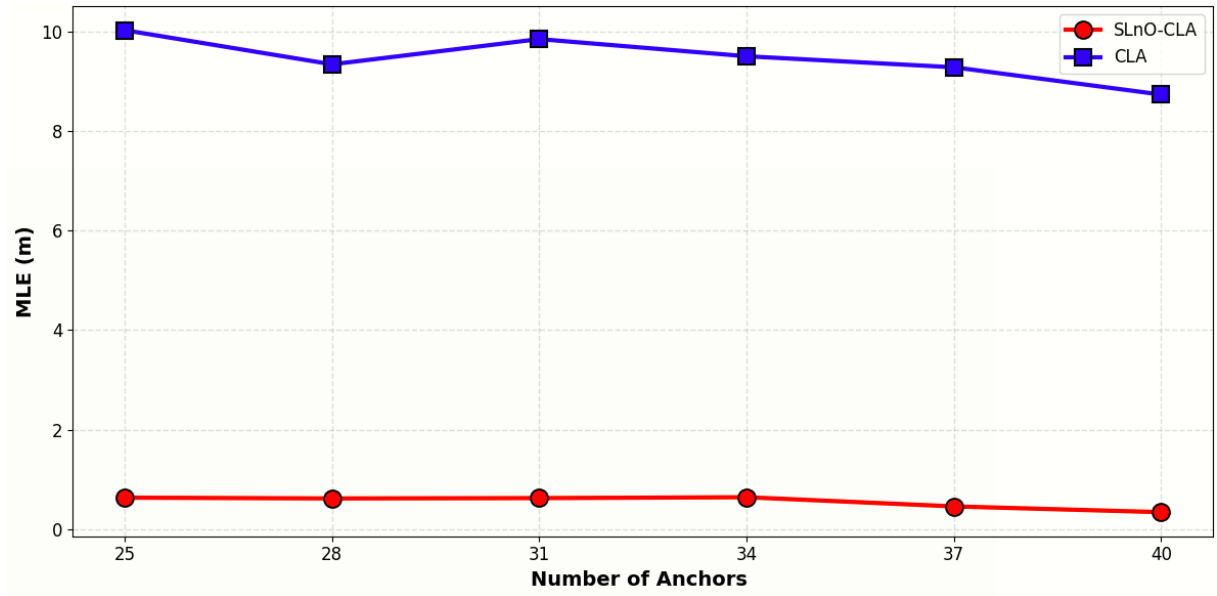


Figure 4.7: The MLE vs number of anchors

improves the accuracy of position estimation. Nevertheless, for all anchor ratios, SLnO-CLA consistently achieves a lower MLE compared to the classical CLA.

Notably, SLnO-CLA with 25 anchors (0.6339 m) achieves better accuracy than CLA with 40 anchors (8.7328 m), demonstrating that algorithmic optimization provides greater benefit than simply increasing hardware resources. This 92.7% improvement in accuracy with 37.5% fewer anchors highlights the cost-effectiveness of the proposed approach.

To assess how transmission range (TR) affects WSN localization, both SLnO-CLA and CLA were evaluated in a simulation using 100 target nodes and 25 anchors. The communication radius R was varied from 20 m to 40 m with increments of 5 m. Table 4.4 and Figure 4.8 summarize the resulting average localization errors (MLE).

From the results shown in Figure 4.8, both algorithms benefit from increased transmission range, as larger coverage radius improves node connectivity. However, SLnO-CLA achieves consistently lower MLE than CLA across all tested transmission ranges.

MLE (m)	SLnO-CLA	CLA
TR=20	0.6767	9.1817
TR=25	0.6339	10.0256
TR=30	0.5021	10.6534
TR=35	0.5289	12.5036
TR=40	0.4724	11.4215

Table 4.4: The MLE for SLnO-CLA and CLA under different Transssmission Range

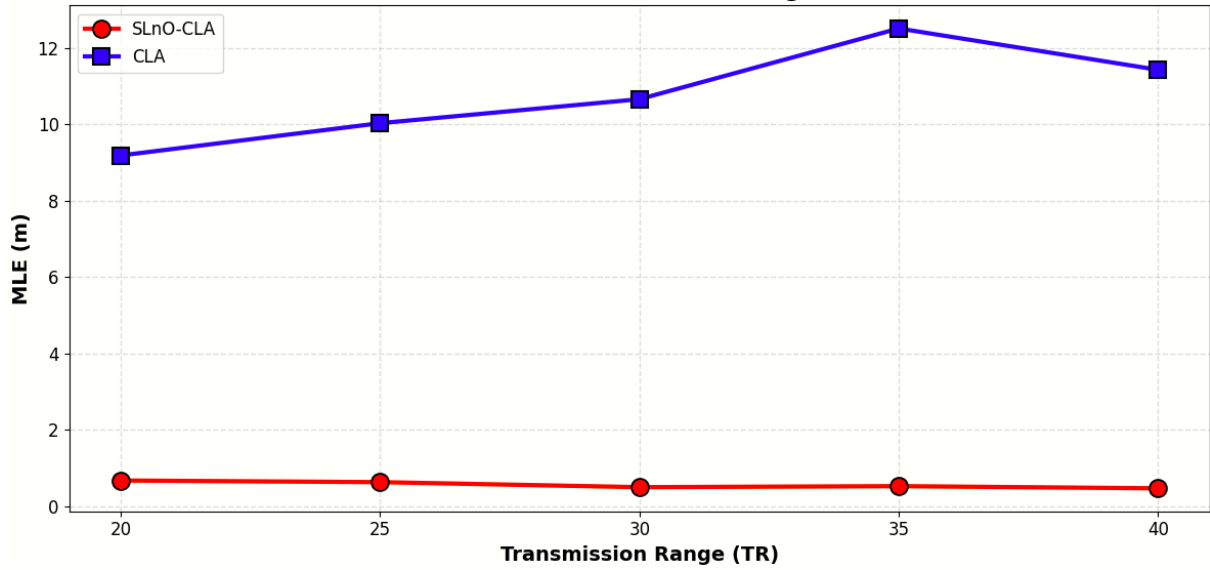


Figure 4.8: The MLE vs transssmission range

4.6 The proposed Improved SLnO (IMP-SLnO)

Although the SLnO algorithm is effective as an optimization technique, it still shares some common limitations like many bio-inspired algorithms, such as the tendency to become trapped in local optima and the need for a large number of iterations to reach stable solutions. To overcome these issues in the context of node localization, we propose an enhanced version called IMP-SLnO. This algorithm introduces four main enhancements. First, an Opposition-Based Learning (OBL) strategy is applied to generate more diverse initial positions, allowing the algorithm to explore the search space more effectively from the beginning of the optimization process. Second, adaptive parameter control is incorporated to dynamically balance exploration and exploitation during the search process. Third, Levy flight-based exploration is used to strengthen the global search capability, helping the algorithm escape local optima and cover the search space more efficiently. Finally, a PSO-based strategy is occasionally used to help the algorithm focus on promising solutions and speed up convergence. The key procedures of the IMP-SLnO are outlined in the pseudocode in Algorithm 7, while its operational flow is illustrated in Fig. 4.10

4.6.1 Population Initialization with the half OBL Strategy

In our proposed method, the Opposition-Based Learning (OBL) strategy is adapted to maintain a diverse population while keeping computational cost low. Unlike the classical OBL introduced by Tizhoosh [120], this adaptation preserves exploration capability

without introducing additional overhead.

Let a candidate solution in an n -dimensional search space be defined as

$$p = (x_1, x_2, \dots, x_n), \quad x_i \in [LB, UB], \quad i = 1, \dots, n. \quad (4.16)$$

The classical OBL defines the opposite point p^{opp} by reflecting each component within its bounds:

$$x_i^{\text{opp}} = LB + UB - x_i. \quad (4.17)$$

Both p and p^{opp} are evaluated using the fitness function $f(\cdot)$, and the one with better fitness is retained for the next generation:

$$p \leftarrow \begin{cases} p^{\text{opp}}, & \text{if } f(p^{\text{opp}}) < f(p), \\ p, & \text{otherwise.} \end{cases}$$

In our adaptation, instead of generating an entire population and evaluating all opposites for selection, the initial population of size N is divided into two equal halves. The first half ($N/2$) is generated randomly. For each randomly generated solution, its opposite is computed and assigned directly to the second half of the population.

Through this construction, the population becomes

$$\{p_1, p_2, \dots, p_{N/2}, p_1^{\text{opp}}, p_2^{\text{opp}}, \dots, p_{N/2}^{\text{opp}}\},$$

ensuring that the search covers both sides of the domain in a well-balanced manner from the very beginning. This approach preserves the fundamental spirit of OBL by promoting strong initial exploration while avoiding the computational burden of evaluating and ranking all candidates at this stage. As a result, the method achieves diversity and broad coverage of the search space at a lower computational cost, making it particularly suitable for large-scale WSN localization scenarios.

4.6.2 Adaptive Control Parameter C

In the original SLnO algorithm, exploration is allowed only when the Sound Pressure (SP) value is below a certain threshold, specifically $SP < 0.25$. This stochastic generation introduces variability into the search behavior, allowing the algorithm to probabilistically enter the exploration phase.

Within this exploration-allowed region, the control parameter C determines whether the algorithm performs exploration or exploitation. In the original SLnO, C decreases

linearly from an initial value of 2 to 0 according to

$$C = 2 - t \left(\frac{2}{\text{max_iterations}} \right), \quad (4.18)$$

where t denotes the current iteration. Exploration occurs when $|C| \geq 1$, while exploitation occurs when $|C| < 1$. Since C decreases linearly, the condition $|C| \geq 1$ holds only during the first half of iterations, limiting the duration of exploration.

To enhance exploration, we introduce a nonlinear, quadratic adaptation of C :

$$C = C_{\text{initial}} - (C_{\text{initial}} - C_{\text{final}}) \left(\frac{t}{\text{max_iterations}} \right)^2. \quad (4.19)$$

Because the squared term $(t/\text{max_iterations})^2$ increases slowly at the beginning, C decreases minimally during early iterations. Solving $C = 1$ for this nonlinear form gives

$$\frac{t}{\text{max_iterations}} = \sqrt{0.5} \approx 0.707, \quad (4.20)$$

indicating that $|C| \geq 1$ is maintained for a longer portion of the exploration-allowed region compared to the linear case. This extension allows the algorithm to perform exploration for a significantly longer period, improving its ability to investigate the search space before committing to local refinement.

The nonlinear adaptation thus produces a more natural transition from exploration to exploitation: the algorithm spends more time performing global search initially and gradually focuses on local refinement as promising areas are identified. This improved balance supports better convergence, reduces stagnation, and enhances localization accuracy in dense and noisy WSN deployment scenarios.

4.6.3 Enhanced Exploration with Lévy Flight

To further improve the global search ability and reduce the risk of premature convergence in our proposed IMP-SLnO algorithm, we incorporate a population-splitting levy mechanism. Lévy flight represents a stochastic movement pattern characterized by frequent short steps and occasional long jumps [121] (see Figure 4.9). This mechanism is inspired by the foraging behavior of predators, which often encounter few and scattered targets and must balance local search with long-distance moves. In optimization, Lévy flights help search agents explore the solution space more efficiently and avoid becoming trapped in local minima, improving both diversification and exploitation [122].

In our approach, and rather of applying Lévy flight to all search agents at every

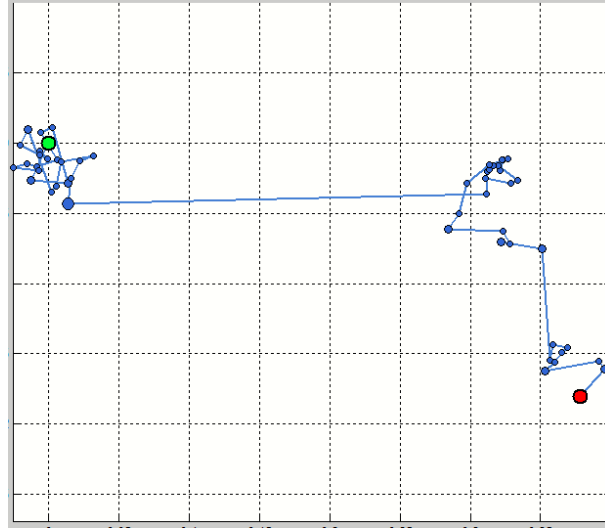


Figure 4.9: Lévy flights tracks.

repetition [123], we divide the population of n sea lions into two equal groups: Group A = $\{1, 2, \dots, \lfloor n/2 \rfloor\}$ and Group B = $\{\lfloor n/2 \rfloor + 1, \dots, n\}$. In each repetition, Group A updates its positions using only the original IMP-SLnO rule, while Group B uses the IMP-SLnO update combined with a Lévy flight step. In the next repetition, the roles are switched, so the agents that used Lévy flight now apply the normal IMP-SLnO rule, and the others use the IMP-SLnO update plus the Lévy step. The mathematical expression of this strategy is given as follows.

For even iterations (t even):

$$X_i^{\text{candidate}} = \begin{cases} \text{IMP_SLnO}(X_i^t, X_{\text{best}}^t) + \alpha \cdot L, & i \in \text{Group A} \\ \text{IMP_SLnO}(X_i^t, X_{\text{best}}^t), & i \in \text{Group B} \end{cases} \quad (4.21)$$

For odd iterations (t odd):

$$X_i^{\text{candidate}} = \begin{cases} \text{IMP_SLnO}(X_i^t, X_{\text{best}}^t), & i \in \text{Group A} \\ \text{IMP_SLnO}(X_i^t, X_{\text{best}}^t) + \alpha \cdot L, & i \in \text{Group B} \end{cases} \quad (4.22)$$

where X_i^t is the position of agent i at iteration t , X_{best}^t is the global best position found so far, $\text{IMP_SLnO}(\cdot)$ denotes the IMP-SLnO update rule, L is the Lévy flight step vector, α is the Lévy scaling factor, and $f(\cdot)$ is the fitness function

This mechanism allows the search to benefit from both exploration and exploitation in each repetition, making it easier for the algorithm to escape local minima while maintaining stable convergence behavior

The Lévy flight step vector L can be generated using Mantegna's algorithm as follows:

$$s = \frac{\mu}{|\nu|^{1/\beta}}, \quad \mu \sim N(0, \sigma_\mu^2), \quad \nu \sim N(0, \sigma_\nu^2), \quad \beta = 1.5, \quad (4.23)$$

where the variables μ and ν are drawn from normal distributions with

$$\sigma_\nu = 1, \quad \sigma_\mu = \left[\frac{\Gamma(1 + \beta) \cdot \sin(\pi\beta/2)}{\Gamma((1 + \beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right]^{1/\beta}. \quad (4.24)$$

This design allows sea lion agents to perform occasional long-distance moves while mostly making small adjustments, enhancing diversification and increasing the probability of finding the global optimum.

4.6.4 Probabilistic strategy-selection mechanism

The final improvement in the proposed IMP-SLnO is the integration of a PSO-based refinement step. PSO is known for its fast convergence because each particle updates its position based on its personal best and the global best positions. To take advantage of this behavior without reducing the exploratory strength of SLnO, a probabilistic strategy-selection mechanism is used. In each iteration, for each agent, the SLnO update rule (incorporating OBL, adaptive C, and Lévy flight) is applied with a probability of 0.8, while the PSO velocity update is applied with a probability of 0.2. The PSO update follows Equation (3.2), as presented in the previous chapter. This design keeps IMP-SLnO as the main search engine, preserving wide exploration through Levy flights and natural movement patterns, while PSO is applied only occasionally to improve exploitation. Using PSO too often could lead to early convergence, which is undesirable in WSN localization.

Algorithm 7 IMP-SLnO Algorithm

```

1: Initialization
2: Generate  $P_{rand}$  of size  $N/2$  randomly
3: Compute opposite population  $P_{opp}$  via Eq. (4.17)
4:  $Population \leftarrow [P_{rand}, P_{opp}]$ 
5: Evaluate fitness of all individuals, set  $X_{best}$  as best individual
6: Split population into  $GroupA$  and  $GroupB$  of equal size
7: Initialize  $pbest_i \leftarrow X_i$  for all  $i$ 
8: while  $iter < T_{max}$  do
9:     Compute  $C$  via Eq. (4.19) and  $SP$  via Eq.(4.5)
10:    for each agent  $X_i$  in population do
11:        Draw  $r_1 \sim \mathcal{U}(0, 1)$ 
12:         $X_{temp} \leftarrow X_i$ 
13:        if  $r_1 \leq 0.2$  then
14:            Update  $X_{temp}$  using PSO rule Eq. (3.2)
15:        else
16:            if  $SP < 0.25$  then
17:                if  $|C| < 1$  then
18:                    Update  $X_{temp}$  using Eq. (4.2)
19:                else
20:                    Select  $X_{rand}$  randomly from population
21:                    Update  $X_{temp}$  using Eq. (4.8)
22:                end if
23:            else
24:                Update  $X_{temp}$  using Eq. (4.6)
25:            end if
26:            if ( $iter$  even and  $X_i \in GroupB$ ) or ( $iter$  odd and  $X_i \in GroupA$ ) then
27:                Generate Lévy step  $L$ 
28:                 $X_{temp} \leftarrow X_{temp} + \alpha \cdot L$ 
29:            end if
30:        end if
31:        Enforce boundary constraints on  $X_{temp}$ 
32:        if  $f(X_{temp}) < f(X_i)$  then
33:             $X_i \leftarrow X_{temp}$ 
34:            if  $f(X_i) < f(pbest_i)$  then
35:                 $pbest_i \leftarrow X_i$ 
36:            end if
37:            if  $f(X_i) < f(X_{best})$  then
38:                 $X_{best} \leftarrow X_i$ 
39:            end if
40:        end if
41:    end for
42:     $iter \leftarrow iter + 1$ 
43: end while
44: return  $X_{best}$ 
    
```

Restricting PSO to 20% of agents per iteration provides controlled exploitation without causing stagnation, ensuring that the algorithm maintains a productive balance between exploration and convergence accuracy.

4.7 IMP-SLnO node localization

To gain the main goal of WSN localization, the IMP-SLnO node localization process is carried out in the following steps, which are also summarized in Figure 4.11:

1. N target nodes and M anchor nodes are placed randomly within the network area. As mentioned earlier, all nodes, whether targets or anchors, have the same communication range R .
2. Each target node calculates its distance to each anchor node using Eqs (4.9) and (4.10).
3. When a target node has three or more anchors within its transmission range, this node is called a localisable node.
4. Every localizable node runs the IMP-SLnO algorithm independently to determine its position.
5. The algorithm identifies the best positions for all localisable nodes by minimizing the ranging error, as defined by the objective function in Equation (4.12). Once a node is localized, it functions as an anchor, helping other nodes to localize.
6. Steps 3–5 are repeated until all the target nodes get localized or no more target node can be localized.
7. The localization process is evaluated using the Mean Localization Error (MLE) and the Localization Success Ratio (LSR).

4.8 Experimental Results for IMP-SLnO

To evaluate performance, our analysis begins by comparing the proposed IMP-SLnO algorithm with the baseline SLnO. Afterward, the comparison is extended to other classical optimisation methods to demonstrate the improvements achieved by the enhanced version. In additionally, the same experimental setup described earlier is adopted for IMP-SLnO, where each algorithm is run 30 independent times using identical network deployments.

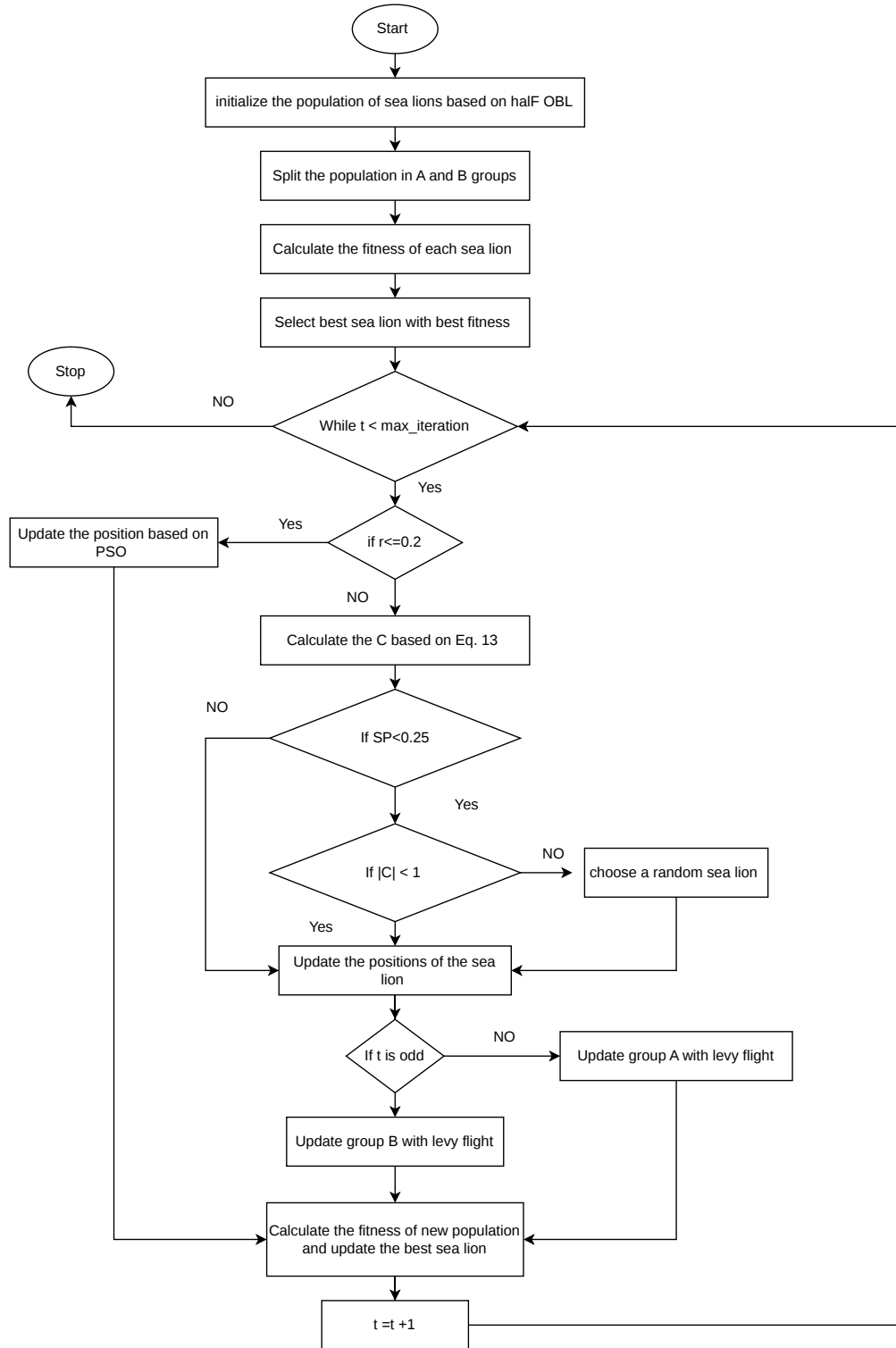


Figure 4.10: The IMP-SLno algorithm's Flowchart

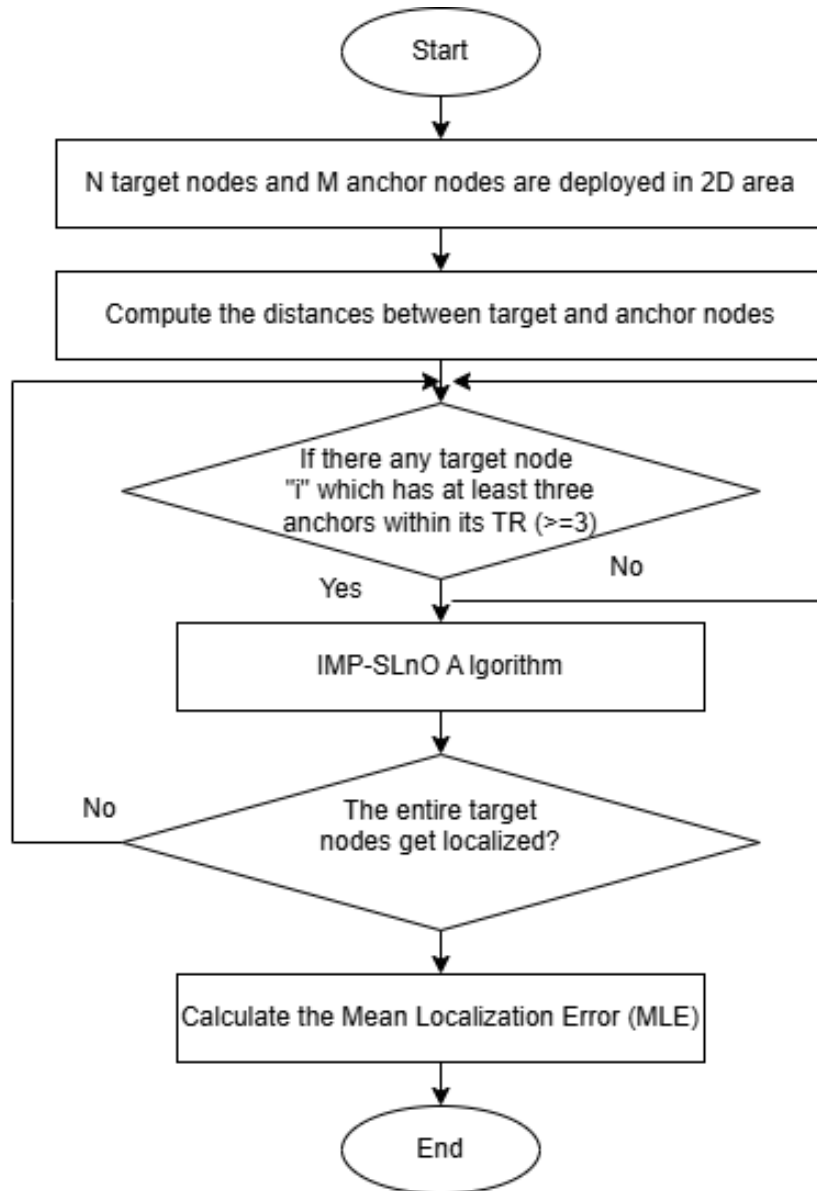


Figure 4.11: Node Localization General Process in WSNs

4.8.1 Comparison of Initial Population Diversity Between SLnO and IMP-SLnO

Before comparing the performance of both algorithms, we first examine the diversity and coverage of their initial populations. Figure 4.12 illustrates the initial population distributions generated across four different runs. The SLnO algorithm (left) begins with random random initialization, whereas IMP-SLnO (right) uses a half-OBL.

Across the runs, the IMP-SLnO initialization produces a noticeably wider and more even spread of individuals. In the SLnO case (left), the randomly generated individuals often cluster together, leaving some regions of the search space, especially the edges, less

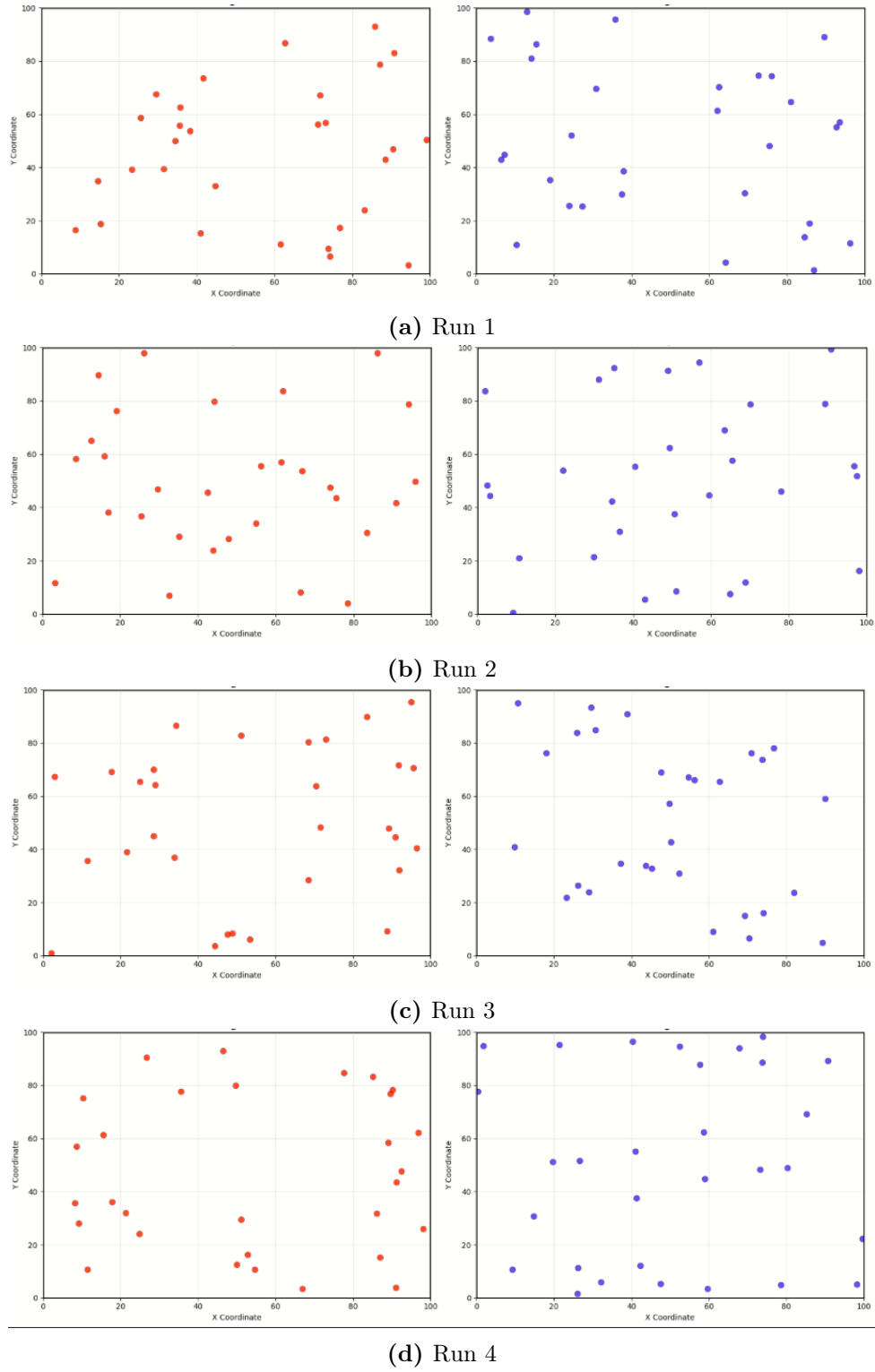


Figure 4.12: Initial population distribution comparison across four runs: SLnO with random initialization (left) versus IMP-SLnO with half opposition-based initialization (right).

covered. In contrast, the half-OBL initialization (right) distributes the population more

uniformly. This happens because each randomly generated position is paired with its opposite position, which helps fill areas that would otherwise remain unexplored. As a result, IMP-SLnO starts the optimization process with a more diverse and better-distributed population. This gives it a stronger initial exploration capability compared to the standard SLnO.

4.8.2 Experimental Analysis IMP-SLnO vs. SLnO

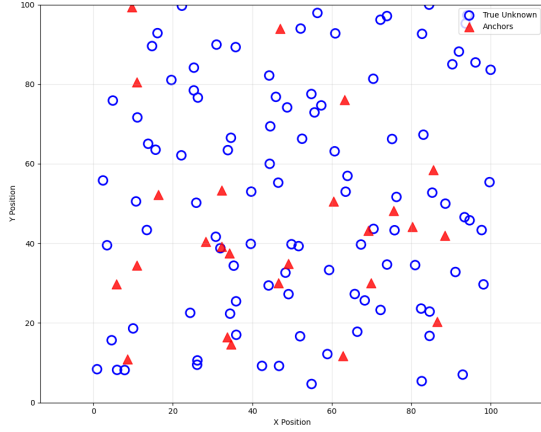
Here, all sensor nodes were randomly placed within a 100×100 area, consisting of 25 anchor nodes and 100 unknown nodes. The transmission range for all nodes was fixed at 25 meters and remained unchanged throughout the experiments. The population size and maximum number of iterations were set to 30 and 50, respectively. The parameters of the deployment setup are summarized in Table 4.1, and the results are presented using both tables and graphical illustrations.

Based on this setup, Figure 4.13 illustrates an example of a randomly generated deployment scenario along with the estimated positions of the unknown nodes. The random deployment of anchor and unknown nodes is shown in Subfigure 4.13a, while Subfigures 4.13b and 4.13c display the positions estimated by IMP-SLnO and basic SLnO, respectively, providing a visual comparison of their localization accuracy, where the connecting lines in red serve to emphasize the significant discrepancy between the real positions and the estimated positions. Meanwhile, the localization error (LE) for each target node is depicted in Figure 4.14, and the detailed numerical values of the LE are comprehensively presented in Table 4.5. To improve visual clarity and allow detailed analysis, any localization errors (LE) shown in Figure 4.14 exceeding 5 meters are capped at this threshold. This ensures that smaller errors remain visible and are not overshadowed by extreme outliers, making it easier to observe variations among most target nodes.

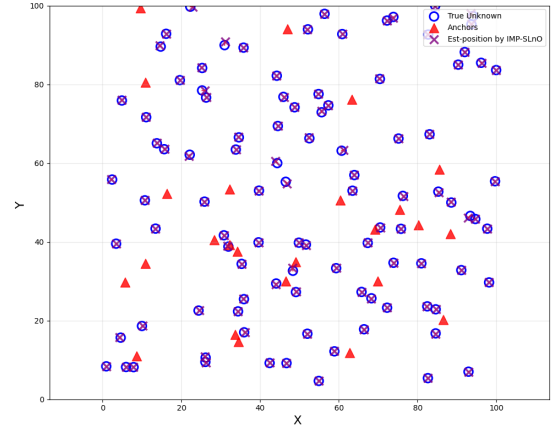
Table 4.5: Localization Errors (LE) Comparison of SLnO and IMP-SLnO

Algorithm	Min LE	Max LE	AVR LE
SLnO	0.0073	16.7627	1.0028
IMP-SLnO	0.0040	4.0831	0.1828

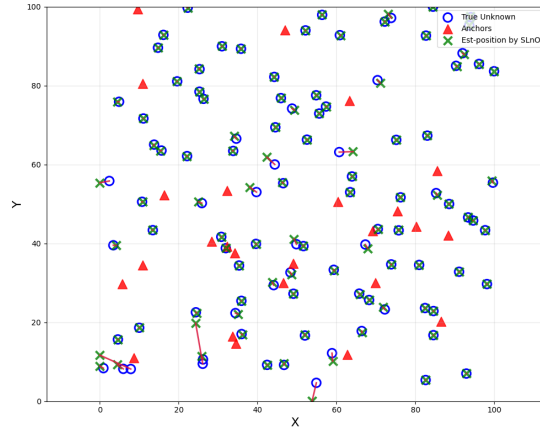
Since a lower localization error (LE) reflects higher accuracy, the figure shows that our IMP-SLnO algorithm achieves smaller errors for most target nodes when compared with the basic SLnO. As we can see, about 25 nodes have localization errors larger than 1m with SLnO, and 3 nodes (nodes: 45, 54, and 66) even exceed 5m. In comparison, with IMP-SLnO, only 1 node has an error above 1m, and none reach 5m. Nevertheless, it is important to note that the localization error of IMP-SLnO is not the smallest for



(a) Random deployment of anchor and unknown nodes.



(b) Estimated positions using IMP-SLnO.



(c) Estimated positions using basic SLnO.

Figure 4.13: Example of node deployment and localization results. Subfigure (a) shows the random placement of anchor and unknown nodes. Subfigures (b) and (c) show the estimated positions obtained using IMP-SLnO and basic SLnO, respectively.

all target nodes. For example, for target nodes 32, 62, and 81, the localization error obtained with SLnO is smaller than that calculated by IMP-SLnO. It can be seen from Table 4.5 that SLnO exhibits large fluctuations in localization errors across nodes (range: 0.0073–16.763 m), while IMP-SLnO keeps the errors much smaller (range: 0.004–4.083 m). On average, IMP-SLnO achieves a localization error of 0.1870 m, representing an 81.8% reduction compared to SLnO's 1.0028 m.

To conduct a thorough evaluation of the proposed algorithm in terms of mean localization error (MLE), experiments were performed by varying both the transmission range and the anchor density. The same experimental settings used for evaluating SLnO-CLA were adopted to ensure a fair comparison. Tables 4.6 and 4.7, together with Figures 4.15 and 4.16, present the results of this comparative analysis.

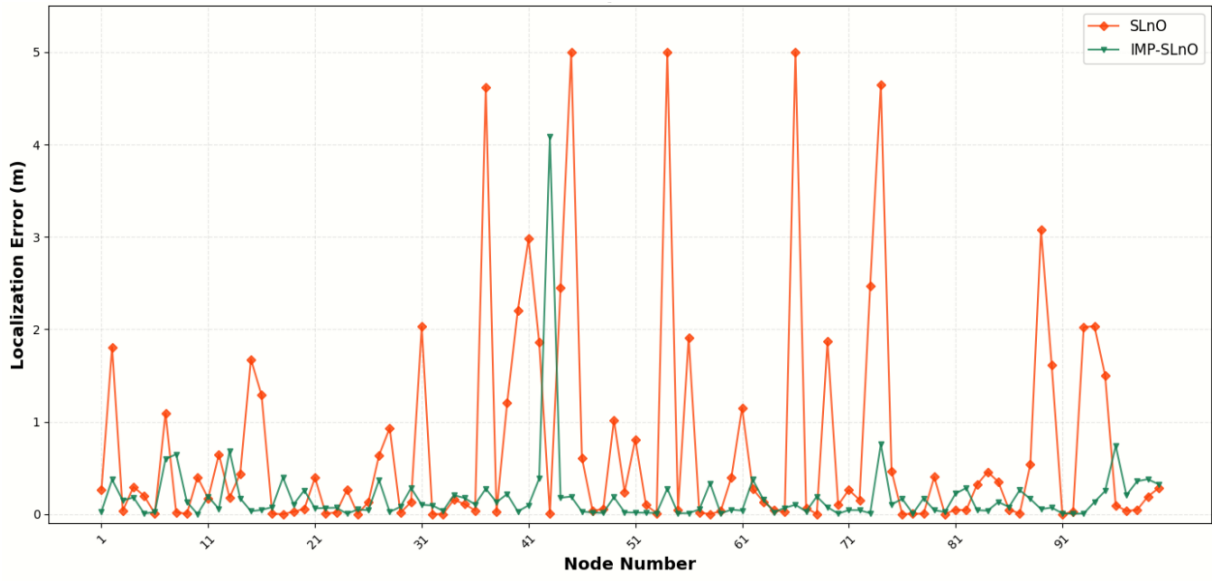


Figure 4.14: Comparison of localization error for each target node for IMP-SLNO and SLNO

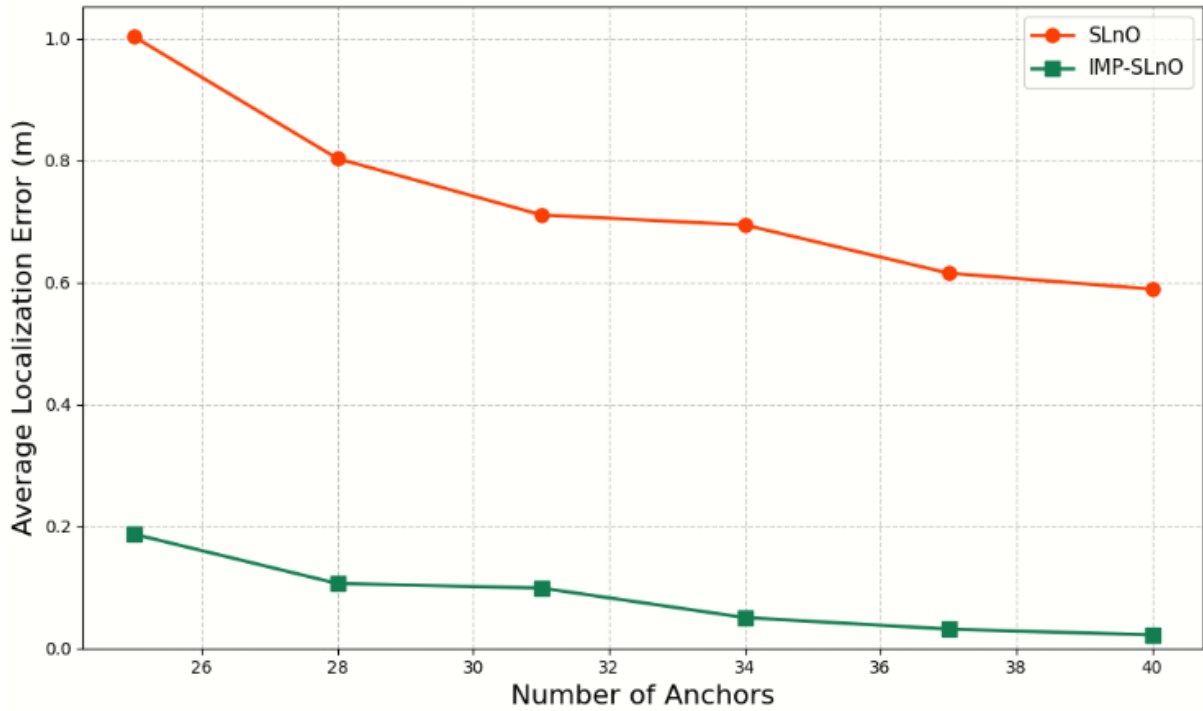


Figure 4.15: The Mean Localization Error vs. Number of Anchors

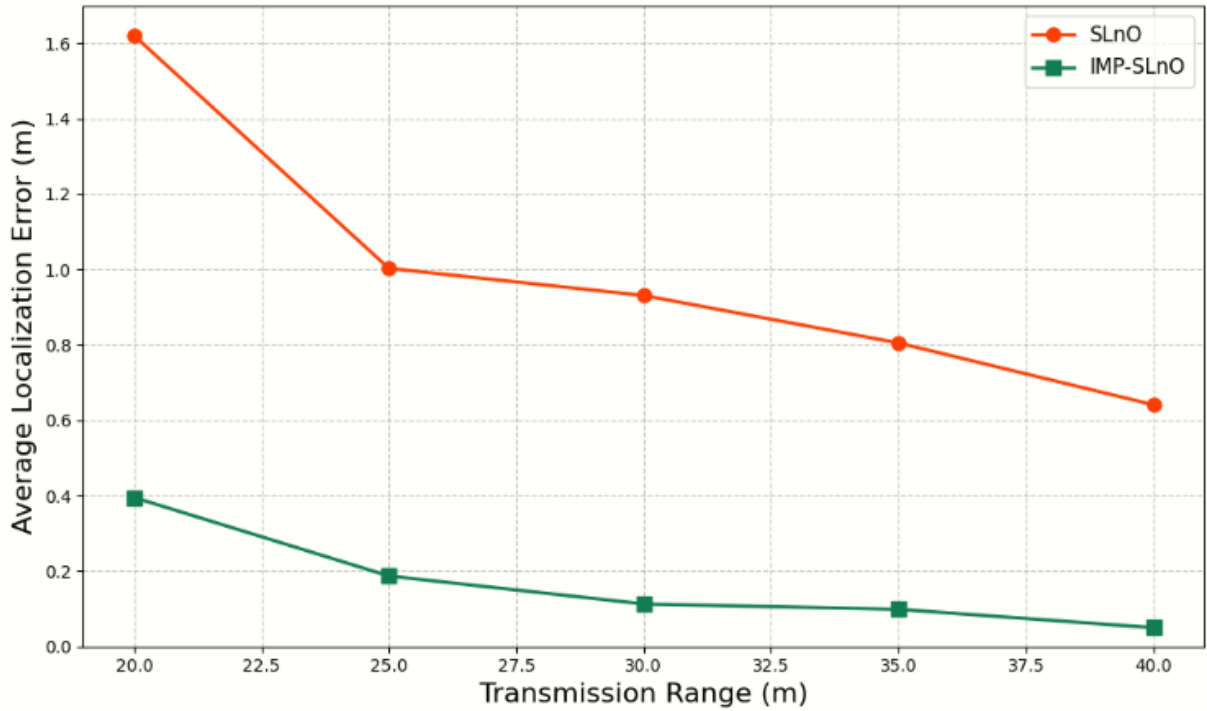
From Figure 4.15 the MLE decreases for both algorithms as the anchor density increases.

A notable observation in Figure 4.15 is that SLNO with 40 anchors (MLE = 0.5888 m) still does not reach the accuracy achieved by IMP-SLNO with only 25 anchors (MLE = 0.1871 m). This indicates that IMP-SLNO can achieve accurate localization even with

Table 4.6: The MLE for SLnO and IMP-SLnO under different number of anchors

MLE (m)	IMP-SLnO	SLnO
A=25	0.1871	1.0029
A=28	0.1061	0.8029
A=31	0.0988	0.7106
A=34	0.0503	0.6940
A=37	0.0314	0.6151
A=40	0.0219	0.5888

fewer anchors, which suggests the potential to reduce the number of required anchor nodes. Furthermore, the IMP-SLnO consistently maintains lower error values across all configurations, reflecting its stable performance.

**Figure 4.16:** The Mean Localization Error vs. different Transmission Range

From Figure 4.16, it is clear that increasing the transmission range improves localization accuracy for both algorithms. At the same time, Table 4.7 shows that IMP-SLnO consistently achieves the lowest errors, with its MLE reduced by more than 75% across all tested ranges compared to SLnO.

Table 4.7: The MLE for SLnO and IMP-SLnO under different Transmission Range

MLE (m)	IMP-SLnO	SLnO
TR=20	0.3953	1.6202
TR=25	0.1871	1.0029
TR=30	0.1129	0.9310
TR=35	0.0987	0.8051
TR=40	0.0499	0.6413

4.8.3 Experimental Analysis of IMP-SLnO vs. Multiple Algorithms

In this part, we further evaluate the performance of the IMP-SLnO algorithm by comparing it with several classical bio-inspired optimisation methods, including FA, PSO, GWO, and our previously proposed SLnO-CLA. The heuristic parameters used for each comparison algorithm are listed in Table 4.8. All algorithms were executed 30 independent times under the same network settings and on the same deployment of target and anchor nodes to ensure a fair comparison. The sensor nodes were randomly placed in a 100×100 area containing 10 anchor nodes and 50 unknown nodes. A fixed transmission range of 25 meters was used for all nodes and a noise measurement of 2%, while each optimisation method operated with a population size of 30 and a maximum of 50 iterations. The statistical summary of all runs is reported in Table 4.9. To provide a clearer interpretation of these quantitative results, several figures are included. Figure 4.17 compares the MLE values across the thirty runs for all algorithms. The range of minimum and maximum errors is illustrated in Figure 4.18, giving a better view of how widely each method can fluctuate in accuracy. The computation time for each algorithm is presented in Figure 4.20, highlighting the differences in speed between the optimisation methods. Finally, Figure 4.19 displays the localisation success rate (LSR) across of the six algorithm, showing how reliably each algorithm manages to localise the unknown nodes.

4.8.3.1 Mean Localization Error and Standard Deviation Analysis

Table 4.9 and Figure 4.17 summarize the average localization performance of all tested algorithms. Among them, IMP-SLnO achieves the lowest mean localization error (MLE) of 0.2459 m, which indicates the highest accuracy. This represents a significant improvement over the other algorithms: 55.9% better than GWO (0.5582 m), 91.1% better than PSO (2.7530 m), 68.3% better than FA (0.7752 m), and 59.2% better than SLnO-CLA (0.6027 m). These results demonstrate that the proposed IMP-SLnO provides substantially more precise position estimates than other bio-inspired methods under identical

Table 4.8: Algorithm-Specific Heuristic Parameters

Algorithm	Parameter	Value
PSO	Inertia weight w	0.7
	Cognitive coefficient c_1	1.4
	Social coefficient c_2	1.4
GWO	Control parameter a	Decreases linearly 2→0
FA	Attractiveness β_{min}	0.2
	Absorption γ	1.0
	Randomness α	0.5
IMP-SLnO	Inertia weight w	0.7
	PSO coefficients c_1, c_2	1.5
	PSO probability	0.2
	$C_{initial}$	2
	C_{final}	0
	β	1.5
	α	$0.1 \times (\text{UB} - \text{LB})$

Table 4.9: Comprehensive Statistical Comparison Over 30 Independent Runs

Algorithm	MLE	Std Error	Min Error	Max Error	SRL (%)	CT
GWO	0.5582	0.5039	0.0237	33.6397	99.4	1.5
PSO	2.7530	1.5591	0.0159	27.8894	66.4	0.47
FA	0.7752	0.5709	0.0133	28.0654	99.6	5.24
SLnO-CLA	0.6027	0.3690	0.0393	7.4129	99.2	1.98
Imp-SLnO	0.2459	0.2025	0.0330	4.9376	99.4	2.77
Centroid	13.0966	0.0000	2.8507	22.0011	54.0	0.01

simulation conditions.

In addition, IMP-SLnO records the smallest standard deviation of 0.2025 m, which reflects very stable performance. This represents a 59.8% improvement in stability compared to GWO (0.5039 m), 87.0% better than PSO (1.5591 m), 64.5% better than FA (0.5709 m), and 45.12% better than SLnO-CLA (0.3690 m). The low standard deviation confirms that IMP-SLnO produces consistent localization results and is significantly less sensitive to randomness in the search process compared to other algorithms.

4.8.3.2 Minimum–Maximum Error Range Analysis

The minimum and maximum localization errors reported in Table 4.9 and Figure 4.18 provide further insight into algorithm stability. Imp-SLnO maintains a low maximum

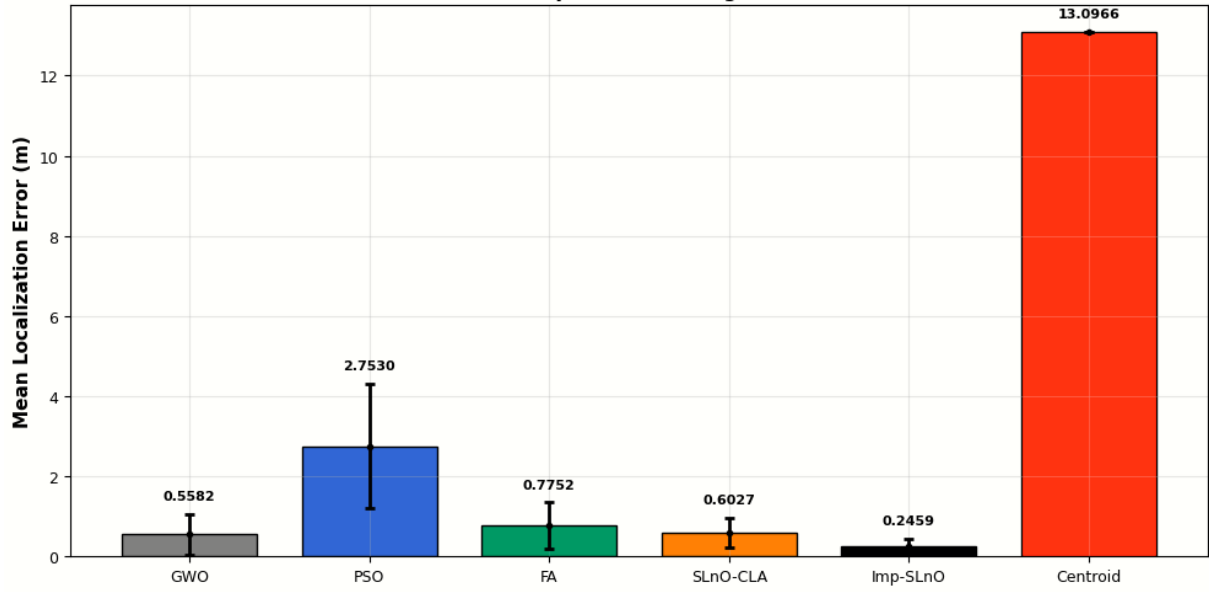


Figure 4.17: MLE comparison across 30 runs

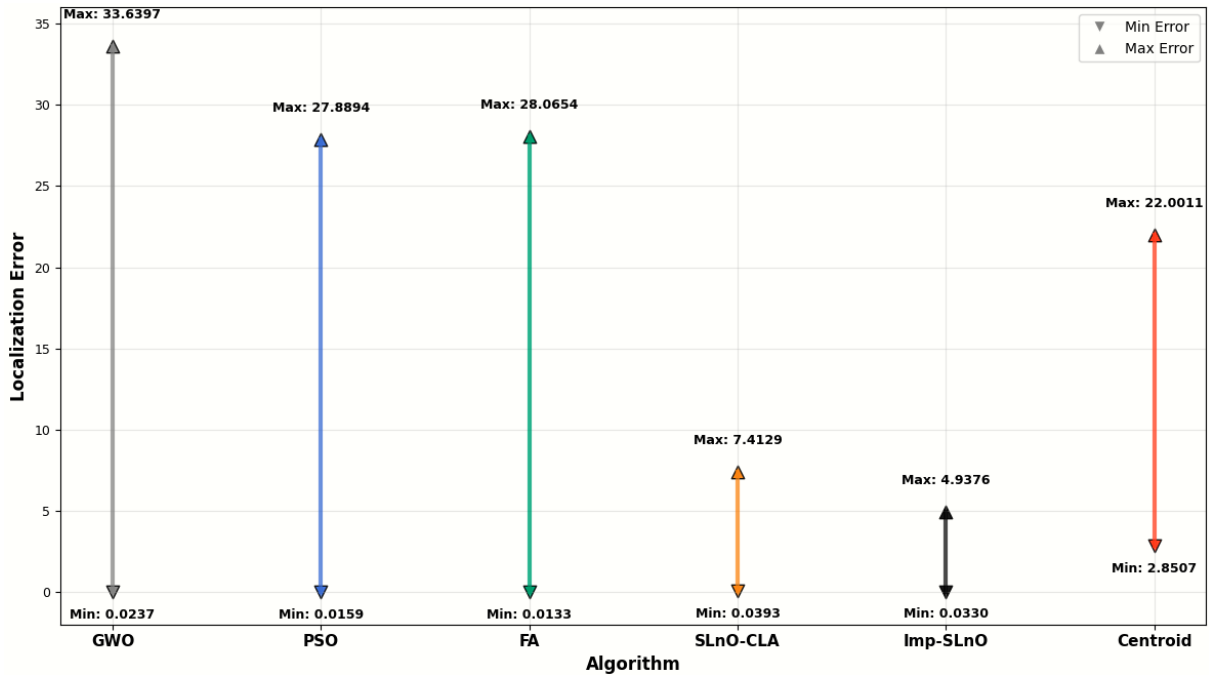


Figure 4.18: Min-Max Error Range Visualization

error of 4.9376, which is much smaller than the maximum errors of GWO (33.6397), PSO (27.8894), and FA (28.0654). This indicates that Imp-SLnO effectively escape the local optima.

SLnO-CLA also shows a clear improvement compared to classical bio-inspired algorithms, with a maximum error of 7.4129. However, Imp-SLnO further reduces this value, demonstrating better control over extreme error cases. Overall, the narrow error range of

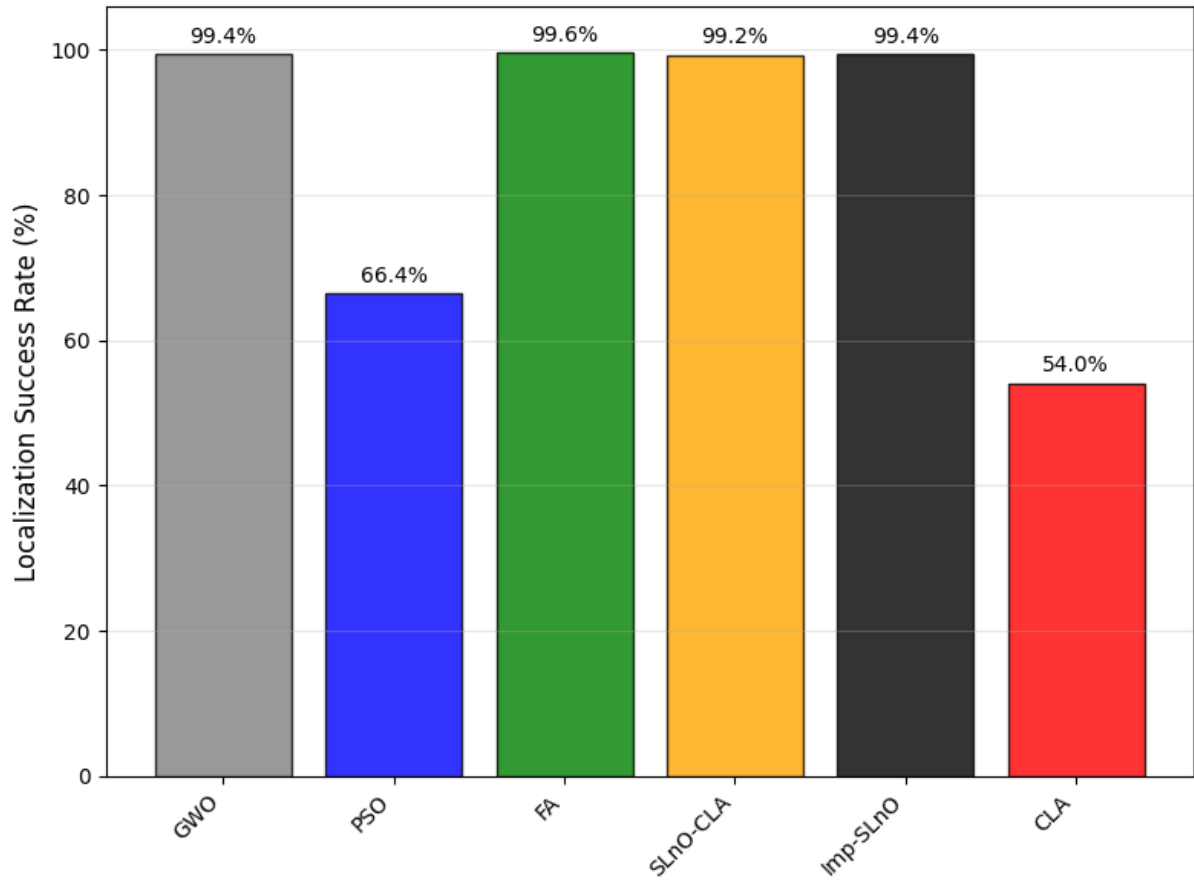


Figure 4.19: The LSR for all algorithms

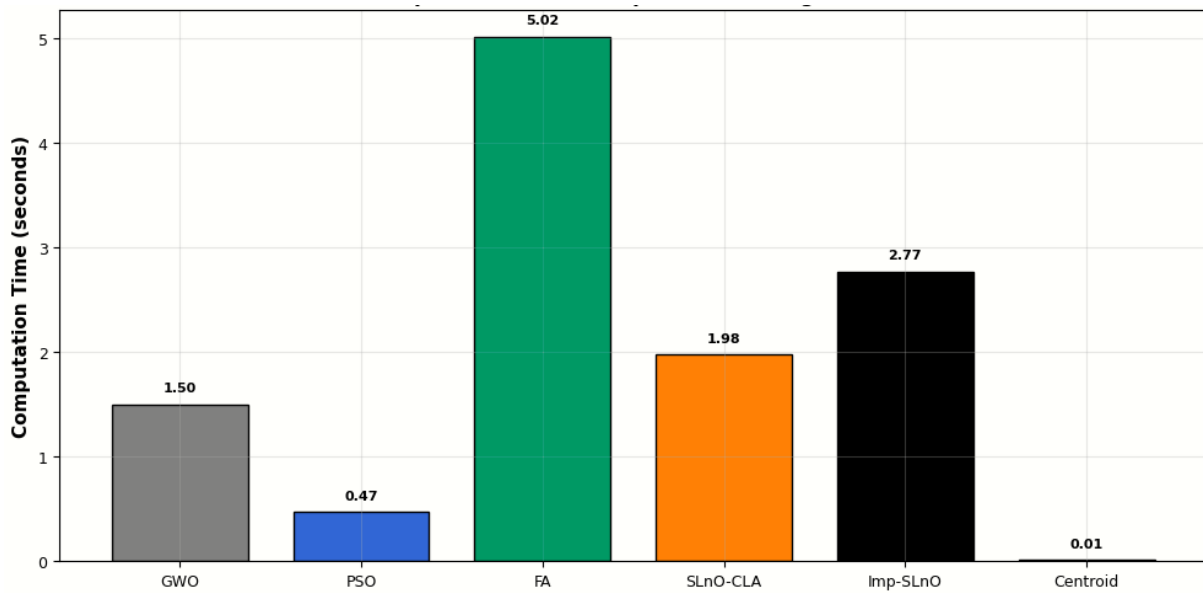


Figure 4.20: The computation time average (CT(s)) across 30 runs

Imp-SL_nO confirms its robustness and stable behavior across all localized nodes.

4.8.3.3 Localisation Success Rate (LSR) Analysis

The localization success rate (LSR) results presented in Table 4.9 and Figure 4.19 show that most bio-inspired algorithms achieve high success rates. FA achieves the highest LSR of 99.6%, followed by GWO and IMP-SL_nO at 99.4%. SL_nO-CLA also performs well with an LSR of 99.2%, while PSO records a lower value of 66.4%. In contrast, the Centroid method achieves only 54.0%.

Overall, IMP-SL_nO combines a high success rate with the lowest localization error and strong stability, making it the most effective method among the compared algorithms.

4.8.3.4 Computation Time Analysis

In terms of computation time, Figure 4.20 shows the computation time of the compared methods. The Centroid method is the fastest (0.01 s), but its localization accuracy is very low (MLE = 13.0966 m) and the success rate is limited (54.0%).

Among the optimization-based methods, PSO is the fastest (0.47 s), followed by GWO (1.5 s). However, this faster execution comes with reduced accuracy, especially for PSO, which shows higher localization error and lower success rate. GWO provides better accuracy than PSO but still does not match the performance of the proposed method.

FA requires the highest computation time (5.24 s), making it the slowest among the compared algorithms, while its accuracy (0.7752 m) remains lower than that of GWO, SL_nO-CLA, and Imp-SL_nO.

SL_nO-CLA requires 1.98 s and achieves reasonable accuracy (0.6027 m) with a high success rate (99.2%). The proposed Imp-SL_nO requires 2.77 s, which is slightly higher than SL_nO-CLA, but it provides significantly better accuracy (0.2459 m) and lower standard deviation (0.2025 m).

Although Imp-SL_nO is slower than simpler algorithms such as PSO and GWO, the additional computation time remains acceptable and is compensated by the clear improvement in localization accuracy and stability.

4.9 Conclusion

In this chapter, two new localization algorithms for WSNs were proposed, namely SL_nO-CLA and IMP-SL_nO. Both approaches are based on the bio-inspired SL_nO algorithm and aim to improve the accuracy of node localization.

The first approach, SLnO-CLA, combines the centroid localization method with SLnO in order to enhance the accuracy of classical centroid-based localization. The second approach, IMP-SLnO, introduces several improvement strategies to strengthen the exploration ability and stability of the basic SLnO algorithm.

Simulation results show that SLnO-CLA provides better localization performance than the classical centroid method, while IMP-SLnO further improves accuracy and stability compared to SLnO-CLA and other existing bio-inspired algorithms. These results confirm the effectiveness of the proposed approaches for localizing unknown nodes in WSNs.

General conclusion

Localization plays a central role in the effectiveness of Wireless Sensor Networks, especially when nodes are randomly deployed and precise position information cannot be obtained using GPS. Traditional localization techniques often suffer from limited accuracy and sensitivity to noise, which motivates the use of optimization-based approaches.

In this thesis, localization in WSNs was addressed using bio-inspired metaheuristic algorithms. Two algorithms were proposed. The first one, SLnO-CLA, combines the simplicity of the Centroid Localization Algorithm with the exploration capability of the Sea Lion Optimization algorithm in order to improve centroid-based localization accuracy. The second contribution, IMP-SLnO, is an enhanced version of the original SLnO, designed to provide better exploration, improved stability, and faster convergence when estimating unknown node positions.

Different simulations were conducted under identical deployment conditions and repeated over multiple independent runs to account for the stochastic nature of metaheuristic algorithms. The results showed that SLnO-CLA significantly improves the accuracy of classical centroid localization, while IMP-SLnO achieves the lowest localization error and the most stable performance among all compared methods. Although some algorithms offer lower computation time, they generally exhibit higher variability, whereas IMP-SLnO provides a better balance between accuracy, stability, and efficiency.

Despite these promising results, this work is limited to two-dimensional static networks. Future work may extend the proposed algorithms to three-dimensional environments, mobile sensor networks, and more realistic energy models.

Overall, this thesis confirms that bio-inspired metaheuristic algorithms are effective tools for localization in Wireless Sensor Networks and that the proposed SLnO-CLA and IMP-SLnO algorithms provide reliable and accurate solutions under various deployment scenarios.

Contributions

This appendix summarizes the scientific contributions produced during the course of this thesis.

Journal Articles

- A. Tabbi and S. Rabhi, “Optimization of Centroid-Based Location Using Sea Lion Optimization Algorithm in Wireless Sensor Networks,” *Automatic Control and Computer Sciences*, vol. 59, pp. 610–622, 2025. doi:10.3103/S0146411625701172.

International Conference Papers

- A. Tabbi and S. Rabhi, “Localization in Wireless Sensor Networks Using Bio-Inspired Techniques: A Survey,” presented at the *15th International Hybrid Conference on Science, Computing, Chemistry & Management (ICSCCM 2024)*.

National Conference Papers

- A. Tabbi and S. Rabhi, “A Review on the Application of Metaheuristic Algorithms in WSN Localization,” presented at the *Second National Conference on Applications of Artificial Intelligence (A2I-25)*.
- A. Tabbi and S. Rabhi, “Grey Wolf Optimizer-Based Accurate Node Localization in WSNs,” presented at the *First National Conference on Artificial Intelligence and Socio-Economic Applications (AISEA '25)*.

Bibliography

- [1] Z. Lalama, “Application des méthaheuristiques pour l’optimisation de la localisation dans les réseaux de capteurs sans fil,” Ph.D. dissertation, Université A. Mira de Béjaïa, Béjaïa, Algeria, December 2022.
- [2] L. Hamami and B. Nassereddine, “Application of wireless sensor networks in the field of irrigation: A review,” *Computers and Electronics in Agriculture*, vol. 179, p. 105782, 11 2020. [Online]. Available: <https://doi.org/10.1016/j.compag.2020.105782>
- [3] M. R. Senouci and A. Mellouk, “Wireless sensor networks,” in *Elsevier eBooks*, 1 2016, pp. 1–19. [Online]. Available: <https://doi.org/10.1016/b978-1-78548-099-7.50001-5>
- [4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1389128601003024>
- [5] S. Rabhi, “Optimisation des algorithmes de localisation dans les réseaux de capteurs sans fil,” Ph.D. dissertation, Université Ferhat Abbas Sétif 1, Sétif, Algeria, 2020.
- [6] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of Things security: A survey,” *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 4 2017. [Online]. Available: <https://doi.org/10.1016/j.jnca.2017.04.002>
- [7] M. K. Singh, S. I. Amin, S. A. Imam, V. K. Sachan, and A. Choudhary, “A Survey of Wireless Sensor Network and its types,” *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 326–330, 10 2018. [Online]. Available: <https://doi.org/10.1109/icacccn.2018.8748710>
- [8] H. M. Kanoosh, E. H. Houssein, and M. M. Selim, “SALP Swarm algorithm for node localization in wireless sensor networks,” *Journal of Computer Networks and Communications*, vol. 2019, pp. 1–12, 2 2019. [Online]. Available: <https://doi.org/10.1155/2019/1028723>

- [9] M. V. Ramesh, P. L. Divya, R. V. Kulkarni, and R. Manoj, "A swarm intelligence based distributed localization technique for wireless sensor network," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, ser. ICACCI '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 367–373. [Online]. Available: <https://doi.org/10.1145/2345396.2345457>
- [10] M. Manickam and S. Selvaraj, "Range-based localisation of a wireless sensor network using Jaya algorithm," *IET Science Measurement Technology*, vol. 13, no. 7, pp. 937–943, 2 2019. [Online]. Available: <https://doi.org/10.1049/iet-smt.2018.5225>
- [11] X.-S. Yang, *Firefly algorithms for multimodal optimization*, 1 2009. [Online]. Available: https://doi.org/10.1007/978-3-642-04944-6_14
- [12] C. Gupta and A. Kumar, "Wireless Sensor Networks: A review," *International Journal of Sensors Wireless Communications and Control*, vol. 3, no. 1, pp. 25–36, 11 2013. [Online]. Available: <https://doi.org/10.2174/22103279112029990001>
- [13] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 5 2007. [Online]. Available: <https://doi.org/10.1016/j.adhoc.2007.05.003>
- [14] L. Pavithra, M. Abdullah, S. P. Prakash, S. Karthick, B. Ragavi, and V. D. Nandhini, "Wireless Sensor Networks: A review on sensor deployment and routing protocols for different application," *IOP Conference Series Materials Science and Engineering*, vol. 1084, no. 1, p. 012052, 3 2021. [Online]. Available: <https://doi.org/10.1088/1757-899x/1084/1/012052>
- [15] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7333–7373, 1 2020. [Online]. Available: <https://doi.org/10.1007/s11227-020-03166-5>
- [16] J. Mao, X. Jiang, and X. Zhang, "Analysis of node deployment in wireless sensor networks in warehouse environment monitoring systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, 12 2019. [Online]. Available: <https://doi.org/10.1186/s13638-019-1615-x>
- [17] M. R. Islam, P. Bharti, and R. Caytiles, "Wireless Sensor Networks: A Descriptive Study on its Types, Topologies and Applications," *Journal of innovative*

- technology convergence.*, vol. 3, no. 2, pp. 53–66, 12 2021. [Online]. Available: <https://doi.org/10.69478/jitc2021v3n2a06>
- [18] C. Mallick and S. Satpathy, “Challenges and design Goals of Wireless Sensor Networks: A state-of-the-art review,” *International Journal of Computer Applications*, vol. 179, no. 28, pp. 42–47, 3 2018. [Online]. Available: <https://doi.org/10.5120/ijca2018916667>
- [19] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, “Wireless sensor networks: a survey on recent developments and potential synergies,” *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, 10 2013. [Online]. Available: <https://doi.org/10.1007/s11227-013-1021-9>
- [20] T. Ojha, S. Misra, and N. S. Raghuwanshi, “Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges,” *Computers and Electronics in Agriculture*, vol. 118, pp. 66–84, 9 2015. [Online]. Available: <https://doi.org/10.1016/j.compag.2015.08.011>
- [21] R. V. Saraswathi, L. P. Sree, and K. Anuradha, “Key management schemes in wireless sensor networks: a survey,” *CiiT International Journal of Wireless Communication*, vol. 8, no. 05, 2016.
- [22] M. Matin and M. Islam, “Overview of wireless sensor network,” in *Wireless Sensor Networks - Technology and Protocols*, M. A. Matin, Ed. London: IntechOpen, 2012, ch. 1. [Online]. Available: <https://doi.org/10.5772/49376>
- [23] D. Alshamaa, “Indoor Localization of Sensors: Application to Dependent Elderly People,” Theses, Université de technologie de Troyes, Nov. 2018. [Online]. Available: <https://hal.science/tel-02568691>
- [24] N. Truong, U. Jayasinghe, T.-W. Um, and G. M. Lee, “A survey on trust computation in the internet of things,” *Journal of the Korean Institute of Communications and Information Sciences (J-KICS)*, vol. 33, pp. 10–27, 2016.
- [25] G. H. Alves, G. C. Guimarães, and F. A. M. Moura, “Battery Storage Systems Control Strategies with Intelligent Algorithms in Microgrids with Dynamic Pricing,” *Energies*, vol. 16, no. 14, p. 5262, 7 2023. [Online]. Available: <https://doi.org/10.3390/en16145262>
- [26] K. Maraiya and K. Kant, “Architectural based data aggregation techniques in wireless sensor network: A comparative study,” *International Journal of Computer Science and Engineering*, vol. 3, no. 3, pp. 1131–1138, 2015.

- [27] S. R. B. Prabhu, M. Pradeep, and E. Gajendran, "Military applications of wireless sensor network system," *A Multidisciplinary Journal of Scientific Research & Education*, vol. 2, no. 12, 2016.
- [28] A. Jangra, Swati, Richa, and P. Jangra, "Wireless sensor network (wsn): Architectural design issues and challenges," *International Journal on Computer Science and Engineering*, vol. 2, 12 2010.
- [29] U. Jain and M. Hussain, "Securing Wireless Sensors in Military Applications through Resilient Authentication Mechanism," *Procedia Computer Science*, vol. 171, pp. 719–728, 1 2020. [Online]. Available: <https://doi.org/10.1016/j.procs.2020.04.078>
- [30] M. P. Đurišić, Z. Tafa, G. Dimić, and V. Milutinović, "A survey of military applications of wireless sensor networks," in *2012 Mediterranean Conference on Embedded Computing (MECO)*, 2012, pp. 196–199.
- [31] D. Galar and U. Kumar, "Sensors and data acquisition," in *Elsevier eBooks*, 1 2017, pp. 1–72. [Online]. Available: <https://doi.org/10.1016/b978-0-12-811153-6.00001-4>
- [32] A. M. Popescu, G. I. Tudorache, B. Peng, and A. H. Kemp, "Surveying position based routing protocols for wireless sensor and ad-hoc networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 4, no. 1, 4 2022. [Online]. Available: <https://doi.org/10.17762/ijcnis.v4i1.85>
- [33] M. Franceschinis, L. Gioanola, M. Messere, R. Tomasi, M. A. Spirito, and P. Civera, "Wireless sensor networks for intelligent transportation systems," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, 2009, pp. 1–5.
- [34] M. S. Manshahia, "Wireless sensor networks: A survey," *International Journal of Scientific and Engineering Research*, vol. 7, no. 4, pp. 710–716, 2016.
- [35] M. S. Obaidat and S. Misra, "Principles of wireless sensor networks." Cambridge University Press, 12 2014. [Online]. Available: <https://doi.org/10.1017/cbo9781139030960>
- [36] S. Naskar, "Wireless sensor networks challenges and solutions," in *Wireless Sensor Networks - Design, Applications and Challenges*, J. Sen, M. Yi, F. Niu, and H. Wu, Eds. London: IntechOpen, 2023, ch. 5. [Online]. Available: <https://doi.org/10.5772/intechopen.109238>

- [37] J. Kuriakose, S. Joshi, and V. I. George, “Localization in Wireless Sensor Networks: A survey,” *arXiv (Cornell University)*, 1 2014. [Online]. Available: <https://arxiv.org/abs/1410.8713>
- [38] V. Sneha and M. Nagarajan, “Localization in Wireless Sensor Networks: A review,” *Cybernetics and Information Technologies*, vol. 20, no. 4, pp. 3–26, 11 2020. [Online]. Available: <https://doi.org/10.2478/cait-2020-0044>
- [39] S. Kagi and B. S. Mathapati, “Localization in wireless sensor networks: A compact review on state-of-the-art models,” in *Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Jan. 2021, pp. 5–12. [Online]. Available: <https://doi.org/10.1109/ICICT50816.2021.9358793>
- [40] P. Agrawal, R. K. Ghosh, and S. K. Das, “Localization of wireless sensor nodes using proximity information,” in *Proceedings of the 6th International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2007, pp. 485–490. [Online]. Available: <https://doi.org/10.1109/ICCCN.2007.4317866>
- [41] M. Farooq-I-Azam and M. N. Ayyaz, “Location and position estimation in wireless sensor networks,” in *CRC Press eBooks*, 12 2012, pp. 179–214. [Online]. Available: <https://doi.org/10.1201/b13092-8>
- [42] N. Zaarour, N. Hakem, and N. Kandil, “Localization context-aware models for wireless sensor network,” in *Emerging Trends in Wireless Sensor Networks*, V. K. Parimala, Ed. London: IntechOpen, 2022, ch. 1. [Online]. Available: <https://doi.org/10.5772/intechopen.103893>
- [43] M. Khelifi, S. Moussaoui, S. Silmi, and I. Benyahia, “Localisation algorithms for wireless sensor networks: a review,” *International Journal of Sensor Networks*, vol. 19, no. 2, p. 114, 1 2015. [Online]. Available: <https://doi.org/10.1504/ijsnnet.2015.071632>
- [44] E. Niewiadomska-Szynkiewicz, “Localization in wireless sensor networks: Classification and evaluation of techniques,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 2, pp. 281–297, 6 2012. [Online]. Available: <https://doi.org/10.2478/v10006-012-0021-x>
- [45] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser.

- MobiCom '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 81–95. [Online]. Available: <https://doi.org/10.1145/938985.938995>
- [46] E. Saad, M. Elhosseini, and A. Y. Haikal, “Recent achievements in sensor localization algorithms,” *Alexandria Engineering Journal*, vol. 57, no. 4, pp. 4219–4228, 11 2018. [Online]. Available: <https://doi.org/10.1016/j.aej.2018.11.008>
- [47] A. Karbasi, “From centralized to distributed sensor localization,” in *Proceedings of the 2010 ACM Workshop on Wireless of the Students, by the Students, for the Students*, ser. S3 '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 5–8. [Online]. Available: <https://doi.org/10.1145/1860039.1860042>
- [48] P. D. Patil and S. S. Raut, “Distributed localization in wireless sensor networks,” *International Journal of Engineering Science Invention (IJESI)*, vol. 2, no. 6, Version-2, pp. 30–36, 2013, accessed: July 22, 2025. [Online]. Available: [https://www.ijesi.org/papers/Vol%202\(6\)/Version-2/Q026201030106.pdf](https://www.ijesi.org/papers/Vol%202(6)/Version-2/Q026201030106.pdf)
- [49] S. P. Singh and S. C. Sharma, “Critical analysis of distributed localization algorithms for wireless sensor networks,” *International Journal of Wireless and Microwave Technologies (IJWMT)*, vol. 6, no. 4, pp. 72–83, 2016, accessed: July 22, 2025. [Online]. Available: <https://www.mecs-press.org/ijwmt/ijwmt-v6-n4/v6n4-7.html>
- [50] S. K. Meghani, M. Asif, and S. Amir, “Localization of wsn node based on time of arrival using ultra wide band spectrum,” in *WAMICON 2012 IEEE Wireless Microwave Technology Conference*, 2012, pp. 1–4.
- [51] N. W. Meng, N. L. Xie, and N. W. Xiao, “Decentralized TDOA sensor pairing in multihop wireless sensor networks,” *IEEE Signal Processing Letters*, vol. 20, no. 2, pp. 181–184, 1 2013. [Online]. Available: <https://doi.org/10.1109/lsp.2013.2237823>
- [52] G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques,” *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 1 2007. [Online]. Available: <https://doi.org/10.1016/j.comnet.2006.11.018>
- [53] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 1 2000. [Online]. Available: <https://doi.org/10.1109/98.878533>
- [54] D. Niculescu and B. Nath, “Dv based positioning in ad hoc networks,” *Telecommun. Syst.*, vol. 22, no. 1–4, p. 267–280, Jan. 2003. [Online]. Available: <https://doi.org/10.1023/A:1023403323460>

- [55] F. Khan and S. K. Nguang, “Distributed localization algorithm for wireless sensor networks using range lookup and subregion stitching,” *IET Wireless Sensor Systems*, vol. 11, no. 5, pp. 179–205, 3 2021. [Online]. Available: <https://doi.org/10.1049/wss2.12020>
- [56] N. Patwari, “Location estimation in sensor networks.” University of Michigan, 2005.
- [57] S. Hayward, K. Van Lopik, C. Hinde, and A. West, “A survey of indoor location technologies, techniques and Applications in industry,” *Internet of Things*, vol. 20, p. 100608, 8 2022. [Online]. Available: <https://doi.org/10.1016/j.iot.2022.100608>
- [58] I. Daramouskas, I. Perikos, M. Paraskevas, V. Lappas, and V. Kapoulas, “Performance analysis for time difference of arrival localization in long-range networks,” *Smart Cities*, vol. 7, no. 5, pp. 2514–2541, 2024. [Online]. Available: <https://www.mdpi.com/2624-6511/7/5/98>
- [59] B. Amer and A. Nouredin, “Rss-based indoor positioning utilizing firefly algorithm in wireless sensor networks,” in *2016 11th International Conference on Computer Engineering Systems (ICCES)*, 2016, pp. 329–333.
- [60] N. Chuku and A. Nasipuri, “RSSI-Based localization schemes for wireless sensor networks using outlier detection,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, p. 10, 1 2021. [Online]. Available: <https://doi.org/10.3390/jsan10010010>
- [61] T. Wu, “Review of RSSI-based positioning algorithm and accuracy,” *Frontiers in Computing and Intelligent Systems*, vol. 2, no. 1, pp. 10–12, 11 2022. [Online]. Available: <https://doi.org/10.54097/fcis.v2i1.2485>
- [62] M. Zaidi, I. Bouazzi, M. Usman, M. Z. M. Shamim, N. Singh, and V. K. Gunjan, “Cooperative Scheme TOA-RSSI and variable anchor positions for sensors localization in 2D environments,” *Complexity*, vol. 2022, no. 1, 1 2022. [Online]. Available: <https://doi.org/10.1155/2022/5069254>
- [63] S. Mohapatra, S. Kar, and S. Behera, “Different approaches of angle of arrival techniques in wireless sensor networks,” *International Journal of Engineering Research*, vol. 2, no. 2, 2013.
- [64] R. Zhang, J. Liu, X. Du, B. Li, and M. Guizani, “AOA-Based Three-Dimensional Multi-Target localization in industrial WSNs for LOS conditions,” *Sensors*, vol. 18, no. 8, p. 2727, 8 2018. [Online]. Available: <https://doi.org/10.3390/s18082727>

- [65] Y. Miao, H. Wu, H. Zhu, and Y. Song, “Localization accuracy of farmland wireless sensor network localization algorithm based on received signal strength indicator,” *Ingénierie des systèmes d’information*, vol. 23, no. 5, pp. 69–80, 10 2018. [Online]. Available: <https://doi.org/10.3166/isi.23.5.69-80>
- [66] M. I. M. Ismail, R. A. Dzyauddin, S. Samsul, N. A. Azmi, Y. Yamada, M. F. M. Yakub, and N. A. B. A. Salleh, “An RSSI-based Wireless Sensor Node Localisation using Trilateration and Multilateration Methods for Outdoor Environment,” *arXiv (Cornell University)*, 1 2019. [Online]. Available: <https://arxiv.org/abs/1912.07801>
- [67] H. K. Maheshwari, I. G. Tudorache, and A. A. Memon, “Simple and effective on-line rssi calibration method in wireless sensor networks (wsns),” *Sindh University Research Journal (Science Series)*, vol. 45, no. 1, pp. 95–100, 2013.
- [68] G. Blumrosen, B. Hod, T. Anker, D. Dolev, and B. Rubinsky, “Enhanced calibration technique for RSSI-based ranging in body area networks,” *Ad Hoc Networks*, vol. 11, no. 1, pp. 555–569, 8 2012. [Online]. Available: <https://doi.org/10.1016/j.adhoc.2012.08.002>
- [69] M. A. Landolsi, H. R. Khan, A. S. Al-Ahmari, and A. H. Muqaibel, “On the performance of wireless sensor network time difference of arrival localization under realistic interference and timing estimation errors,” *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, p. 155014771983253, 2 2019. [Online]. Available: <https://doi.org/10.1177/1550147719832537>
- [70] A. Paul and T. Sato, “Localization in Wireless Sensor Networks: A survey on algorithms, measurement techniques, applications and challenges,” *Journal of Sensor and Actuator Networks*, vol. 6, no. 4, p. 24, 10 2017. [Online]. Available: <https://doi.org/10.3390/jsan6040024>
- [71] N. A. Alrajeh, M. Bashir, and B. Shams, “Localization techniques in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 6, p. 304628, 6 2013. [Online]. Available: <https://doi.org/10.1155/2013/304628>
- [72] F. Mekelleche and H. Haffaf, “Classification and comparison of Range-Based localization techniques in wireless sensor networks,” *Journal of Communications*, 1 2017. [Online]. Available: <https://doi.org/10.12720/jcm.12.4.221-227>
- [73] S. M. Hosseinirad, M. Niazi, J. Pourdeilami, S. K. Basu, and A. A. Pouyan, “On improving apit algorithm for better localization in wsn,” *Journal of AI and Data Mining*, vol. 2, no. 2, pp. 97–104, 2014.

- [74] E. Shakshuki, A. A. Elkhail, I. Nemer, M. Adam, and T. Sheltami, “Comparative study on range free localization algorithms,” *Procedia Computer Science*, vol. 151, pp. 501–510, 1 2019. [Online]. Available: <https://doi.org/10.1016/j.procs.2019.04.068>
- [75] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, “Localization from mere connectivity,” in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, ser. MobiHoc ’03. New York, NY, USA: Association for Computing Machinery, 2003, p. 201–212. [Online]. Available: <https://doi.org/10.1145/778415.778439>
- [76] L. L. De Oliveira, G. H. Eisenkraemer, E. A. Carara, J. B. Martins, and J. Monteiro, “Mobile Localization Techniques for Wireless Sensor Networks: Survey and recommendations,” *ACM Transactions on Sensor Networks*, vol. 19, no. 2, pp. 1–39, 9 2022. [Online]. Available: <https://doi.org/10.1145/3561512>
- [77] B. Joda, K. A. Yahya, and B. A. Stephen, “WIRELESS SENSOR NETWORK LOCALIZATION TECHNIQUES PERFORMANCE,” *Engineering and Technology Journal*, vol. 08, no. 07, 7 2023. [Online]. Available: <https://doi.org/10.47191/etj/v8i7.02>
- [78] B. A. Shahal and M. N. Abdullah, “A review of localization algorithms based on software defined networking approach in wireless sensor network,” *Measurement Sensors*, vol. 27, p. 100772, 5 2023. [Online]. Available: <https://doi.org/10.1016/j.measen.2023.100772>
- [79] E. Niewiadomska-Szynkiewicz, “Localization in wireless sensor networks: Classification and evaluation of techniques,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 2, pp. 281–297, 6 2012. [Online]. Available: <https://doi.org/10.2478/v10006-012-0021-x>
- [80] M. Allen, S. Baydere, E. Gaura, and G. Kucuk, “Evaluation of localization algorithms,” in *IGI Global eBooks*, 1 2009, pp. 348–379. [Online]. Available: <https://doi.org/10.4018/978-1-60566-396-8.ch014>
- [81] N. Sharma and V. Gupta, “Meta-heuristic based optimization of WSNs Localisation Problem- a Survey,” *Procedia Computer Science*, vol. 173, pp. 36–45, 1 2020. [Online]. Available: <https://doi.org/10.1016/j.procs.2020.06.006>

- [82] Q. Xiao, K. Bu, Z. Wang, and B. Xiao, “Robust localization against outliers in wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 9, no. 2, pp. 1–26, 3 2013. [Online]. Available: <https://doi.org/10.1145/2422966.2422981>
- [83] L. A. Caceres Najarro, I. Song, and K. Kim, “Fundamental limitations and state-of-the-art solutions for target node localization in wsns: A review,” *IEEE Sensors Journal*, vol. 22, no. 24, pp. 23 661–23 682, 2022.
- [84] V. Tomar, M. Bansal, and P. Singh, “Metaheuristic algorithms for optimization: A brief review,” *Engineering Proceedings*, vol. 59, no. 1, 2023. [Online]. Available: <https://www.mdpi.com/2673-4591/59/1/238>
- [85] S. Somvanshi, M. M. Islam, S. A. Javed, G. Chhetri, K. S. Islam, T. I. Chowdhury, S. B. B. Pollock, A. K. Dutta, and S. Das, “A comprehensive survey on bio-inspired algorithms: Taxonomy, applications, and future directions,” *ArXiv*, vol. abs/2506.04238, 2025. [Online]. Available: <https://api.semanticscholar.org/CorpusID:279243669>
- [86] J. A. López-Gómez, F. P. Romero, and E. Angulo, “Bio-inspired algorithms for the characterization of excellent performance in handball players: A data-driven methodology,” *Expert Systems with Applications*, vol. 274, p. 126821, 2 2025. [Online]. Available: <https://doi.org/10.1016/j.eswa.2025.126821>
- [87] A. M. Hemeida, S. A. Hassan, A.-A. A. Mohamed, S. Alkhalaf, M. M. Mahmoud, T. Senjyu, and A. B. El-Din, “Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research,” *Ain Shams Engineering Journal*, vol. 11, no. 3, pp. 659–675, 1 2020. [Online]. Available: <https://doi.org/10.1016/j.asej.2020.01.007>
- [88] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.
- [89] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm Optimization in Wireless-Sensor Networks: A Brief survey,” *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262–267, 7 2010. [Online]. Available: <https://doi.org/10.1109/tsmcc.2010.2054080>
- [90] R. V. Kulkarni, G. K. Venayagamoorthy, and M. X. Cheng, “Bio-inspired node localization in wireless sensor networks,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 205–210.

- [91] G. Sharma, M. Kharub, and Bharti, "Particle swarm based node localization in wireless sensor networks," *International Journal of Scientific & Engineering Research*, vol. 8, pp. 100–105, Dec. 2017.
- [92] I. F. M. Zain and S. Y. Shin, "Distributed localization for wireless sensor networks using binary particle swarm optimization (bpso)," in *2014 IEEE 79th Vehicular Technology Conference (VTC Spring)*, 2014, pp. 1–5.
- [93] S. Phoemphon, C. So-In, and N. Leelathakul, "A hybrid localization model using node segmentation and improved particle swarm optimization with obstacle-awareness for wireless sensor networks," *Expert Systems with Applications*, vol. 143, p. 113044, 11 2019. [Online]. Available: <https://doi.org/10.1016/j.eswa.2019.113044>
- [94] V. Kanwar and A. Kumar, "DV-Hop localization methods for displaced sensor nodes in wireless sensor network using PSO," *Wireless Networks*, vol. 27, no. 1, pp. 91–102, 8 2020. [Online]. Available: <https://doi.org/10.1007/s11276-020-02446-5>
- [95] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 5941–5956, 6 2020. [Online]. Available: <https://doi.org/10.1007/s12652-020-02195-5>
- [96] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 1 2014. [Online]. Available: <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [97] Y. Zhang and Y. Liu, "A novel localization algorithm based on grey wolf optimization for wsns," in *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, 2020, pp. 127–130.
- [98] H. Cui, J. Zhao, C. Zhou, and N. Zhang, "Localization for Wireless Sensor Networks Assisted by Two Mobile Anchors with Improved Grey Wolf Optimizer," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–20, 12 2022. [Online]. Available: <https://doi.org/10.1155/2022/6292629>
- [99] H. Y. Abuaddous, G. Kaur, K. Jyoti, N. Mittal, S. Mahajan, A. K. Pandit, A. R. Alsoud, and L. Abualigah, "Repulsion-based grey wolf optimizer with improved exploration and exploitation capabilities to localize sensor nodes in 3D wireless sensor network," *Soft Computing*, vol. 27, no. 7, pp. 3869–3885, 11 2022. [Online]. Available: <https://doi.org/10.1007/s00500-022-07590-y>

- [100] W. S. Jeyaseelan, K. V. Kumar, T. Jayasankar, and R. Ponni, “Improved grey wolf optimization based node localization approach in underwater wireless sensor networks,” *Measurement Science Review*, vol. 24, no. 3, pp. 95–99, 6 2024. [Online]. Available: <https://doi.org/10.2478/msr-2024-0013>
- [101] Z. Xu, “Wireless sensor network localization incorporating gray wolf optimization and dv-hop algorithm,” *IEEE Access*, vol. 12, pp. 168 594–168 606, 2024.
- [102] S. N. Ghorpade, M. Zennaro, and B. S. Chaudhari, “Gwo model for optimal localization of iot-enabled sensor nodes in smart parking systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 1217–1224, 2021.
- [103] P. Kathirolu and K. S., “Localization by Salp Swarm Optimization with Doppler Effect in Wireless Sensor Networks,” *International Journal of Computer Network and Information Security*, vol. 13, no. 6, pp. 26–40, 12 2021. [Online]. Available: <https://doi.org/10.5815/ijcnis.2021.06.03>
- [104] S. Rani, H. Babbar, P. Kaur, and A. A. Khan, “A novel approach of localization with single mobile anchor using quantum-based Salp swarm algorithm in wireless sensor networks,” *Soft Computing*, 9 2023. [Online]. Available: <https://doi.org/10.1007/s00500-023-09261-y>
- [105] Z. Liu, C. Jin, C. Liu, G. Jiang, N. Jia, N. Chen, and H. Peng, “Large scale WSNs node localization based on salp swarm algorithm using density peak clustering strategy,” *Wireless Networks*, 7 2025. [Online]. Available: <https://doi.org/10.1007/s11276-025-04011-4>
- [106] W. Liu, J. Li, A. Zheng, Z. Zheng, X. Jiang, and S. Zhang, “DV-HOP algorithm based on Multi-Objective SALP Swarm Algorithm Optimization,” *Sensors*, vol. 23, no. 7, p. 3698, 4 2023. [Online]. Available: <https://doi.org/10.3390/s23073698>
- [107] X. Liu and H. Xu, “Application on target localization based on salp swarm algorithm,” in *2018 37th Chinese Control Conference (CCC)*, 2018, pp. 4542–4545.
- [108] Y. Liu and J. Chen, “A K-means based firefly algorithm for localization in sensor networks,” *International Journal of Parallel Emergent and Distributed Systems*, vol. 34, no. 4, pp. 364–379, 3 2018. [Online]. Available: <https://doi.org/10.1080/17445760.2018.1446211>
- [109] L. Kaur, A. Seehra, and D. Singh, “Node localization in wireless sensor network using firefly algorithm,” *Advances in Computer Science and Information Technology*, vol. 5, no. 2, 2018.

- [110] A. Kumar and Karampal, "Optimized range-free 3d node localization in wireless sensor networks using firefly algorithm," in *2015 International Conference on Signal Processing and Communication (ICSC)*, 2015, pp. 14–19.
- [111] V.-O. Sai, C.-S. Shieh, T.-T. Nguyen, Y.-C. Lin, M.-F. Horng, and Q.-D. Le, "Parallel firefly algorithm for localization algorithm in wireless sensor network," in *2015 Third International Conference on Robot, Vision and Signal Processing (RVSP)*, 2015, pp. 300–305.
- [112] E. Tuba, M. Tuba, and M. Beko, *Two stage wireless sensor node localization using Firefly algorithm*, 12 2017. [Online]. Available: https://doi.org/10.1007/978-981-10-6916-1_10
- [113] V. Gupta and B. Singh, "Centroid based localization utilizing artificial bee colony algorithm," *International Journal of Computer Networks And Applications*, vol. 6, no. 3, p. 47, 6 2019. [Online]. Available: <https://doi.org/10.22247/ijcna/2019/49655>
- [114] T. Chen and L. Sun, "A connectivity weighting DV-HOP localization algorithm using modified artificial bee colony optimization," *Journal of Sensors*, vol. 2019, pp. 1–14, 6 2019. [Online]. Available: <https://doi.org/10.1155/2019/1464513>
- [115] D. Ekmekci and H. Altinkaya, "Localization of Sensor Nodes in 3D Wireless Sensor Networks with a Single Anchor by an Improved Adaptive Artificial Bee Colony (iaABC) Algorithm," *Applied Sciences*, vol. 15, no. 7, p. 3548, 3 2025. [Online]. Available: <https://doi.org/10.3390/app15073548>
- [116] S. Aslan and S. Demirci, "Solving UAV Localization Problem with Artificial Bee Colony (ABC) Algorithm," *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pp. 735–738, 9 2019. [Online]. Available: <https://doi.org/10.1109/ubmk.2019.8907034>
- [117] R. K. Jena, "Artificial Bee Colony Algorithm based Multi-Objective Node Placement for Wireless Sensor Network," *International Journal of Information Technology and Computer Science*, vol. 6, no. 6, pp. 25–32, 3 2014. [Online]. Available: <https://doi.org/10.5815/ijitcs.2014.06.04>
- [118] R. Masadeh, B. A, and A. Sharieh, "Sea Lion Optimization Algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 1 2019. [Online]. Available: <https://doi.org/10.14569/ijacsa.2019.0100548>

- [119] M. Yazdani and F. Jolai, “Lion optimization algorithm (loa): A nature-inspired metaheuristic algorithm,” *Journal of Computational Design and Engineering*, vol. 3, no. 1, pp. 24–36, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2288430015000524>
- [120] H. Tizhoosh, “Opposition-based learning: A new scheme for machine intelligence,” in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’06)*, vol. 1, 2005, pp. 695–701.
- [121] M. Chawla and M. Duhan, “Levy Flights in Metaheuristics Optimization Algorithms – A review,” *Applied Artificial Intelligence*, vol. 32, no. 9-10, pp. 802–821, 9 2018. [Online]. Available: <https://doi.org/10.1080/08839514.2018.1508807>
- [122] X.-S. Yang and S. Deb, “Cuckoo search via lévy flights,” in *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*, 2009, pp. 210–214.
- [123] Y. Ling, Y. Zhou, and Q. Luo, “Lévy flight trajectory-based whale optimization algorithm for global optimization,” *IEEE Access*, vol. 5, pp. 6168–6186, 2017.