

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 MATHEMATICS AND COMPUTER SCIENCE



DOCTORAL THESIS

Submitted in partial fulfillment of the requirements for the degree of

Doctor in Computer Science

Specialization: Artificial Intelligence

Presented by:

HARROUZ Fatima

**Intelligent Clustering in Wireless Sensor Networks:
Integrating Machine Learning with Metaheuristic Strategies**

Publicly defended on May 14, 2026, before the examination committee:

Name	Academic Rank	Institution	Role
Dr. RABHI Seddik	Associate Professor	University of Adrar, Algeria	Chair
Pr. OMARI Mohammed	Professor	University of Adrar, Algeria	Supervisor
Pr. KADDI Mohammed	Professor	University of Adrar, Algeria	Co-Supervisor
Dr. MAMOUNI El Mamoun	Associate Professor	University of Adrar, Algeria	Examiner
Dr. SABA Djamel	Research Director	URERMS – Adrar, Algeria	Examiner

Academic Year: 2025–2026

Acknowledgements

All praise and gratitude are due to Almighty Allah, whose guidance, strength, and wisdom enabled me to successfully complete this PhD dissertation. The achievement of this academic journey was made possible by Allah's divine guidance, mercy, and countless blessings. I am truly humbled and thankful for His grace and support throughout this endeavor.

My heartfelt appreciation goes to my supervisor, Professor Mohammed Omari, for his unwavering guidance, continuous encouragement, and invaluable insights during the course of this research. His patience, expertise, and dedication have been instrumental in shaping this work, and I am profoundly grateful for his mentorship and support.

I would also like to extend my sincere thanks and gratitude to Professor Mohammed Kaddi for his availability, constructive feedback, and valuable advice, which have greatly contributed to the quality and completion of this work.

To all my teachers who have inspired my passion for learning and contributed to my academic growth, I express my deepest appreciation for their guidance and support throughout my educational journey.

Finally, I would like to express my heartfelt gratitude to my parents and family for their continuous encouragement, patience, and steadfast belief in me throughout every stage of this journey. Their love and support have been the cornerstone of my success.

Dedication

To my father, my mother, my siblings, and my extended family, for their endless love and support.

To me for saving the day

Abstract

Wireless Sensor Networks (WSNs) are limited by battery energy, making energy efficiency and network lifetime critical challenges. Clustering helps reduce communication overhead and balance energy consumption through efficient Cluster Head (CH) management. This work proposes PUMA-GRID, an energy-efficient protocol that combines the Puma Optimization Algorithm (PUMA) for adaptive CH selection with a lightweight grid based multi-hop routing strategy inspired by k-nearest neighbor (k-NN) logic. During clustering, PUMA selects optimal CHs using a weighted fitness function based on residual energy, distance between nodes and CHs, and distance between CHs and the Base Station (BS), ensuring balanced energy usage and stable cluster formation. For routing, the sensing field is divided into grid cells where neighboring CHs are selected as relays using a k-NN-inspired mechanism to construct shorter and more energy efficient communication paths while reducing long range transmissions and communication overhead. The protocol was evaluated under three BS placement scenarios: central, edge, and external. For each scenario, optimal fitness function weights were first selected before conducting MATLAB simulations with randomly deployed nodes in a 200×200 m² sensing field containing up to 600 nodes. PUMA-GRID was then compared with LEACH, Atomic Energy Optimization-based approaches. Simulation results show that PUMA-GRID achieves better network lifetime, residual energy preservation, packet delivery, and communication efficiency, with approximately 35–50% improvement in network lifetime. This improvement is achieved through the combination of adaptive PUMA based CH optimization and efficient k-NN inspired routing, which together reduce transmission cost and balance energy consumption across the network. Finally, the framework is mainly designed for static and homogeneous WSN environments, leaving opportunities for future improvements under more realistic conditions.

Keywords: wireless sensor networks, lifetime, PUMA algorithm, cluster head selection, k-NN, multi-hop communication, base station placement, weighted fitness function.

Arabic Absract

الملخص

تعد شبكات الاستشعار اللاسلكية (Wireless Sensor Networks - WSNs) محدودة من حيث طاقة البطارية، مما يجعل كفاءة الطاقة وإطالة عمر الشبكة من أهم التحديات. ويساهم التجميع (Clustering) في تقليل عبء الاتصالات وتحقيق توازن استهلاك الطاقة من خلال الإدارة الفعالة لرؤوس المجموعات (Cluster Heads - CHs). يقترح هذا العمل بروتوكول PUMA-GRID، وهو بروتوكول موفر للطاقة يجمع بين خوارزمية Puma Algorithm Optimization (PUMA) لاختيار رؤوس المجموعات بشكل تكيفي، وآلية توجيه متعددة القفزات مستوحاة من خوارزمية أقرب الجيران (k-nearest neighbor). - (k-NN) أثناء عملية التجميع، تقوم PUMA باختيار أفضل CHs اعتماداً على دالة ملاءمة موزونة تعتمد على الطاقة المتبقية، والمسافة بين العقد وCHs، والمسافة بين CHs والمحطة الأساسية (Base Station - BS)، مما يضمن توازن استهلاك الطاقة واستقرار تكوين المجموعات. أما في مرحلة التوجيه، فيتم تقسيم منطقة الاستشعار إلى خلايا شبكية، حيث يتم اختيار CHs المجاورة كعقد ترحيل باستخدام آلية مستوحاة من k-NN لبناء مسارات اتصال أقصر وأكثر كفاءة في استهلاك الطاقة مع تقليل الاتصالات بعيدة المدى. تم تقييم البروتوكول تحت ثلاث وضعيات مختلفة لـ BS مركزية، جانبية، وخارجية. وفي كل حالة، تم أولاً اختيار الأوزان المثلى لدالة الملاءمة قبل إجراء محاكاة MATLAB باستخدام عقد موزعة عشوائياً داخل مساحة 200 × 200 م² وبعدها يصل إلى 600 عقدة. بعد ذلك، تمت مقارنة PUMA-GRID مع LEACH وطرق Energy Atomic Optimization، بالإضافة إلى بروتوكولات PUMA السابقة. أظهرت نتائج المحاكاة أن PUMA-GRID يحقق أداءً أفضل من حيث عمر الشبكة، والحفاظ على الطاقة المتبقية، وكفاءة نقل البيانات، مع تحسين يقارب 35%-50% في عمر الشبكة. ويعود هذا التحسن إلى الدمج بين الاختيار التكيفي لـ CHs باستخدام PUMA وآلية التوجيه المستوحاة من k-NN، مما يقلل تكلفة الإرسال ويحقق توازناً أفضل لاستهلاك الطاقة. وأخيراً، فإن البروتوكول مصمم أساساً لبيئات WSNs الثابتة والمتجانسة، مما يفتح المجال لتطويره مستقبلاً ضمن ظروف أكثر واقعية وديناميكية.

الكلمات المفتاحية: شبكات الاستشعار اللاسلكية، عمر الشبكة، خوارزمية PUMA، اختيار رؤوس العناقيد، أقرب الجيران، k-NN، الاتصال متعدد القفزات، موقع محطة القاعدة، دالة ملاءمة موزونة.

Résumé

Les réseaux de capteurs sans fil (Wireless Sensor Networks - WSNs) sont limités par la capacité énergétique des batteries, ce qui rend l'efficacité énergétique et la durée de vie du réseau essentielles. Le clustering permet de réduire la surcharge de communication et d'équilibrer la consommation d'énergie grâce à une gestion efficace des têtes de cluster (Cluster Heads - CHs). Ce travail propose PUMA-GRID, un protocole écoénergétique combinant l'algorithme Puma Optimization Algorithm (PUMA) pour la sélection adaptative des CHs et une stratégie de routage multi-sauts inspirée du k-nearest neighbor (k-NN). PUMA sélectionne les CHs optimaux à l'aide d'une fonction de fitness basée sur l'énergie résiduelle, la distance entre les nœuds et les CHs, ainsi que la distance entre les CHs et la station de base (Base Station - BS). Le protocole a été évalué selon trois scénarios de placement du BS (central, bordure et externe). Des poids optimaux ont d'abord été déterminés avant de réaliser des simulations MATLAB avec des nœuds déployés aléatoirement dans une zone de 200×200 m² contenant jusqu'à 600 nœuds. PUMA-GRID a ensuite été comparé à LEACH, aux approches basées sur Atomic Energy Optimization et aux précédents protocoles PUMA. Les résultats montrent que PUMA-GRID améliore la durée de vie du réseau, la conservation de l'énergie et l'efficacité des communications, avec un gain d'environ 35–50 % grâce à la combinaison de l'optimisation adaptative par PUMA et du routage inspiré du k-NN. Enfin, le protocole est principalement conçu pour des environnements WSNs statiques et homogènes.

Mots-clés : réseaux de capteurs sans fil, durée de vie du réseau, algorithme d'optimisation PUMA, sélection des chefs de cluster, k-plus-proches-voisins (k-NN), communication multi-sauts, déploiement de la station de base, fonction de coût pondéré.

List of Contents

Acknowledgements	I
Dedication	II
Abstract	III
Arabic Absract	IV
Résumé	V
List of Contents	VI
List of Figures	XI
List of Tables	XI
Abbreviations	XII
General Introduction	1
Chapter 1: Wireless Sensor Network: Overview	4
1.1 Introduction	5
1.2 Wireless Sensor Network	5
1.3 Architecture of Wireless Sensor Networks	6
1.3.1 Sensors	6
1.3.2 Base Station	9
1.3.3 WSN application	10
1.4 Communication Architecture	11
1.5 Challenges of WSN	11
1.5.1 Resource Limitations	11
1.5.2 Operational and Design Challenges	12
1.6 Routing	13
1.6.1 Flat Based Routing	13
1.6.2 Hierarchical Based Routing	13
1.7 Clustering	16
1.7.1 Introduction to clustering	16
1.7.2 Cluster Formation Process	17
1.7.3 Advantages of Clustering	19
1.7.4 Challenges of Clustering in WSNs	19
1.8 Conclusion	21
Chapter 2: Literature Review	22

2.1	Introduction	23
2.2	LEACH	23
2.2.1	Multi-Hop Routing LEACH	24
2.2.2	LEACH-C	26
2.2.3	Two-Level Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH)	26
2.2.4	Weighted Low-Energy Adaptive Clustering Hierarchy (W-LEACH)	28
2.2.5	Distance-Based Cluster Head Algorithm (DBCH)	28
2.2.6	LEACH with Vice Cluster Head (LEACH-VH)	29
2.3	Clustering using Fuzzy Logic	30
2.3.1	Cluster Head Election Mechanism Using Fuzzy Logic (CHEF)	30
2.3.2	An Energy-Aware Fuzzy Approach for Uneven Clustering in WSNs (EAUCF)	30
2.3.3	Distributed Load Balancing Unequal Clustering in WSNs using Fuzzy Approach (DUCF)	31
2.3.4	Enhancing Energy Efficiency in WSNs through a Fuzzy Rule-Based Clustering Algorithm (EETPF)	32
2.3.5	Fuzzy Logic Based Distributed Energy Efficient Clustering Algorithm for WSNs (EEDCF)	33
2.3.6	Fuzzy-Based Cluster Routing in WSN (FBCR)	34
2.3.7	Fuzzy-Based Energy Efficient Clustering Protocol for WSN (FEEC)	35
2.3.8	Multi-objective fuzzy clustering algorithm for WSNs (MOFCA)	36
2.4	Clustering Using Machine Learning	37
2.4.1	Optimal Packet Size K-means Clustering (OPSKC)	37
2.4.2	Soft-k-Means Clustering Algorithm	38
2.4.3	K-means and Dijkstra-based Clustering Protocol (KICMH)	39
2.4.4	Improvement of DBSCAN	40
2.4.5	Distributed DBSCAN Protocol (DDBSCAN)	41
2.5	Clustering Using Metaheuristic Algorithms	42
2.5.1	Atomic Energy Optimization-based WSN Clustering (AEOWSNC)	43
2.5.2	Spotted Hyena Optimization-based Clustering (SHO-CH)	44
2.5.3	African Vulture Optimization Algorithm (AVOACS)	44
2.5.4	DEAI-PSO-Based Clustering Algorithm for IoT-Oriented WSNs	45
2.6	Hybridization Protocols	47
2.6.1	An Optimized Fuzzy Clustering Algorithm (OFCA)	47
2.6.2	Fuzzy Based Energy Efficient Cluster Head Selection (FEECS)	49
2.6.3	Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization	50
2.6.4	Particle Swarm Optimization Based K-means	52
2.6.5	hybrid clustering approach integrates the Harmony Search with K-means	52
2.6.6	Cluster Based Energy Efficient Routing Protocol using SA-LEACH to WSNs	54
2.7	Conclusion	57

Chapter 3: Puma Optimizer **59**

3.1	Introduction	60
3.2	Biological Inspiration	60
3.3	Mathematical Model and Search Phases	61
3.3.1	Initialization	61
3.3.2	Unexperienced Phase	61
3.3.3	Experienced Phase	61

3.3.4	Exploration Phase	65
3.3.5	Exploitation Phase	67
3.3.6	Updating and Termination	68
3.4	Computational Complexity	69
3.5	Advantages and Features	69
3.6	Conclusion	70
Chapter 4: Proposed Method		71
4.1	Introduction	72
4.2	Proposed Method: PUMA-GRID – PUMA based Clustering with Grid k-NN Routing	72
4.2.1	Initialization	73
4.2.2	PUMA-Based Clustering and Fitness Evaluation	73
4.2.3	Machine learning inspired grid based multi hop routing via K-NN	75
4.2.4	Adaptive Operation and Steady-State Execution	78
4.2.5	Complexity Analysis of PUMA-GRID	79
4.2.6	Clustering complexity	80
4.2.7	Routing complexity	80
4.2.8	Overall complexity of PUMA-GRID	80
4.3	Conclusion	81
Chapter 5: Simulation Setup, Results, and Discussion		82
5.1	Introduction	83
5.2	Simulation Setup and Energy Consumption Model	83
5.3	Choosing the Optimal Weights for the Fitness Function	84
5.3.1	Impact of Weight Combinations on Different metrics (BS inside the Network)	86
5.3.2	Impact of Weight Combinations on Different metrics (BS outside the Network)	91
5.3.3	Impact of Weight Combinations on Different metrics (BS at the edge the Network)	96
5.3.4	Sensitivity Analysis of the Weighted Fitness Function	100
5.3.5	Objective Function for Optimal Weight Selection	101
5.4	Comparison of Different Routing Protocols	104
5.4.1	Comparison of Different Routing Protocols when BS is inside the network	105
5.4.2	Comparison of Different Routing Protocols when BS is outside the network	111
5.4.3	Comparison of Different Routing Protocols when BS is at the edge of the network	116
5.4.4	Runtime Evaluation of Clustering and Routing Mechanisms	120
5.4.5	Impact of BS placement on multi-hop grid k-NN routing	121
5.5	Discussion	122
5.6	Scalability Analysis of the Protocols	124
5.7	Global Discussion	126
5.8	Limitations	128
5.9	Conclusion	129
General Conclusion		131
References		134

List of Figures

Figure 1.1:	Light Dependent Resistor	5
Figure 1.2:	WSN Architecture	6
Figure 1.3:	Sensors Components	7
Figure 1.4:	Flat Routing	14
Figure 1.5:	Tree Routing	14
Figure 1.6:	Chain based routing architecture in WSNs	15
Figure 1.7:	Clustering based routing architecture in WSNs	16
Figure 2.1:	Multi-Hop Routing LEACH	25
Figure 2.2:	Two-Level LEACH	27
Figure 2.3:	Fuzzy inference system of DUCF	31
Figure 2.4:	EETPF Flow Chart	33
Figure 2.5:	EEDCF Flow Chart	34
Figure 2.6:	OPSKC Flow Chart	38
Figure 2.7:	Distributed DBSCAN Protocol	42
Figure 2.8:	Phases of the protocol	51
Figure 3.1:	The Puma Optimization Method	69
Figure 4.1:	Grid-based routing.	77
Figure 4.2:	Flowchart of the proposed method.	79
Figure 4.3:	Global view of the PUMA-GRID contributions: metaheuristic clustering, grid structure, and kNN-based routing.	79
Figure 5.1:	Impact of weight configurations on FND, HND, and LND with BS inside the network area.	87
Figure 5.2:	Evaluation of data delivery under various weight schemes with internal BS placement.	88
Figure 5.3:	Effect of weight combinations on CFI with BS inside the network.	89
Figure 5.4:	Effect of weight combinations on live nodes with BS inside the network.	90

Figure 5.5:	Impact of weight configurations on FND, HND, and LND with BS outside the network area.	91
Figure 5.6:	Evaluation of data delivery under various weight schemes with external BS placement.	92
Figure 5.7:	Effect of weight combinations on CFI with BS outside the network.	93
Figure 5.8:	Effect of weight combinations live nodes with BS outside the network.	95
Figure 5.9:	Impact of weight configurations on FND, HND, and LND with BS at the edge the network.	96
Figure 5.10:	Evaluation of data delivery under various weight schemes with Edge BS placement.	97
Figure 5.11:	Effect of weight combinations on CFI with BS at the edge the network.	98
Figure 5.12:	Effect of weight combinations live nodes with BS at the edge the network.	99
Figure 5.13:	Comparison of FND, HND, and LND across different routing protocols with BS inside.	105
Figure 5.14:	Comparison of the number of packets delivered to the BS across different routing protocols with BS inside.	107
Figure 5.15:	Comparison of the number of live nodes across different routing protocols with BS inside.	108
Figure 5.16:	Residual energy comparison across different routing protocols with BS inside	109
Figure 5.17:	CFI comparison across different routing protocols with BS inside	110
Figure 5.18:	Comparison of FND, HND, and LND across different routing protocols with BS outside the network.	111
Figure 5.19:	Comparison of the number of packets delivered to the BS across different routing protocols with BS outside the network	112
Figure 5.20:	Comparison of the number of live nodes across different routing protocols with BS outside the network.	113
Figure 5.21:	Residual energy comparison across different routing protocols with BS outside the network	115
Figure 5.22:	CFI comparison across different routing protocols with BS outside of the network	116
Figure 5.23:	Comparison of FND, HND, and LND across different routing protocols with BS at the edge of the network.	117

Figure 5.24: Comparison of the number of packets delivered to the BS across different routing protocols with BS at the edge.	118
Figure 5.25: Comparison of the number of live nodes across different routing protocols with BS at the edge of the network.	119
Figure 5.26: Residual energy comparison across different routing protocols with BS outside the network	119
Figure 5.27: CFI comparison across different routing protocols with BS at the edge of the network	120

List of Tables

Table 1.1: Classification of sensors based on type, measured parameters, and examples.	8
Table 1.2: Classification of sensors based on power source and mode of operation.	8
Table 1.3: Summary of key metrics and parameters influencing clustering and CH selection in WSNs.	18
Table 2.1: Summary of clustering protocols	55
Table 5.1: Simulation Parameters	85
Table 5.2: Objective Function Values for Different Weight Combinations and BS Positions	103
Table 5.3: Simulation parameters for comparing routing protocols.	104
Table 5.4: Runtime for Routing Protocols	121
Table 5.5: Routing behaviour across BS placements	122
Table 5.6: Performance comparison for $N = 600$ nodes with the BS inside the network.	124
Table 5.7: Performance comparison for $N = 600$ nodes with the BS outside the network.	125
Table 5.8: Performance comparison for $N = 600$ nodes with the BS at the edge of the network.	125

Abbreviations

ACO	Ant Colony Optimization
ADC	Analog-to-Digital Converter
AEO	Atomic Energy Optimization
AEO-SH	AEO Single-Hop
BS	Base Station
CDMA	Code Division Multiple Access
CFI	Coverage Fairness Index
CH	Cluster Head
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
FND	First Node Dead
GA	Genetic Algorithm
GPS	Global Positioning System
GWO	Grey Wolf Optimizer
HEED	Hybrid Energy-Efficient Distributed Clustering
HND	Half Nodes Dead
IC	Integrated Circuit
IoT	Internet of Things
k-NN	k-Nearest Neighbors
LEACH	Low-Energy Adaptive Clustering Hierarchy
LND	Last Node Dead

OSI	Open Systems Interconnection
PEGASIS	Power-Efficient GATHERing in Sensor Information Systems
PO	Puma Optimizer
PSO	Particle Swarm Optimization
PUMA-SH	PUMA Single-Hop
QoS	Quality of Service
SA	Simulated Annealing
SN	Sensor Node
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

General Introduction

In a world that is increasingly connected and driven by data, it has become essential to develop efficient technologies capable of interacting seamlessly with the physical environment. Wireless Sensor Networks (WSNs) represent one of the most practical solutions for this purpose. A WSN consists of numerous small, low-cost, and energy-limited sensor nodes that cooperate to observe their surroundings and forward the collected information to a central Base Station (BS), where it can be processed and analyzed. As societies depend more on real-time data for monitoring, automation, and decision-making, WSNs have established themselves as a reliable and versatile technology for environmental observation, industrial operations, precision agriculture, healthcare, and security applications.

A typical WSN is composed of many distributed nodes equipped with sensing, computing, and wireless communication capabilities. Because these nodes are usually battery-powered and frequently deployed in remote or harsh environments, conserving energy becomes a fundamental design requirement. The limited energy supply directly influences the operational lifetime of the network, the reliability of transmitted data, and the overall performance of the system.

To address these limitations, clustering has emerged as one of the most effective strategies for reducing communication load and balancing energy consumption. In clustering, sensor nodes are grouped into clusters, each managed by a Cluster Head (CH). The CH aggregates data from its cluster members and forwards it to the BS. This significantly reduces the number of long-range transmissions and distributes energy consumption more evenly across the network. However, the efficiency of clustering-based protocols depends critically on how the CHs are selected and how data is routed to the BS. Determining the optimal number and placement of CHs is a complex optimization problem influenced by residual energy, communication distance, and network topology.

Over the years, many clustering protocols have been proposed, ranging from classical solutions such as LEACH and HEED to advanced metaheuristic approaches including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Atomic Energy Optimization (AEO). While classical protocols established foundational clustering principles, they rely heavily on probabilistic or random selection, often leading to unstable cluster formation and early node

failure. Furthermore, although advanced metaheuristics improve network organization, they may suffer from high computational complexity in large-scale deployments or fail to maintain long-term energy balance once the first nodes begin to die. These limitations motivate the development of more adaptive and lightweight intelligent clustering mechanisms.

Within this context, the present work introduces PUMA-GRID, a hybrid clustering and routing protocol that integrates the Puma Optimization Algorithm (PUMA) for adaptive CH selection with a grid-based multi-hop routing strategy inspired by k-nearest neighbor (k-NN) principles. During clustering, PUMA simulates the strategic hunting behavior of pumas to locate the most efficient set of CHs while balancing exploration and exploitation. The CH selection process relies on a weighted fitness function based on residual energy, distance between nodes and CHs, and distance between CHs and the BS. The weighting parameters can be adapted according to deployment conditions and network objectives.

PUMA-GRID differentiates itself through a novel architectural synergy that separates the energy problem into high-level organization and lightweight real-time execution. While many hybrid approaches rely on computationally expensive algorithms for both clustering and routing, PUMA-GRID uses PUMA as a “global intelligence” mechanism for adaptive CH selection and combines grid-based routing with k-NN logic as a lightweight routing engine. This integration enables the protocol to achieve strong energy efficiency and scalability while maintaining lower computational overhead, making it more suitable for practical large-scale WSN deployments. For data transmission, the proposed method employs a grid-assisted multi-hop routing mechanism. The sensing field is divided into uniform grid cells to reduce and control the routing search space. Routing decisions rely on neighboring CH distances to construct shorter communication paths, reduce hop lengths, minimize transmission cost, and distribute forwarding load more uniformly across the network. The combined effect of clustering and routing produces a robust and energy-efficient communication structure.

To validate these contributions, the protocol was subjected to extensive MATLAB simulations involving up to 600 nodes deployed randomly within a 200×200 m² sensing field. Performance was evaluated under three distinct BS placement scenarios: central, edge, and external. For each scenario, optimal fitness-function weights were first selected before comparing PUMA-GRID with LEACH, AEO-based approaches, and previous PUMA-based protocols. The evaluation considered metrics such as network lifetime, throughput, residual energy preservation, and communication efficiency.

The results demonstrate that PUMA-GRID achieves superior performance in terms of network lifetime, energy balancing, packet delivery, and communication efficiency. This improvement is

achieved through the synergy between adaptive PUMA-based CH optimization and lightweight grid-assisted routing, which together reduce communication overhead and minimize energy-intensive long-range transmissions.

The current framework operates under the assumption of static and homogeneous sensor nodes and utilizes a simplified free-space propagation model. Although PUMA-GRID demonstrates strong scalability, its routing performance may still be influenced by the fixed virtual grid structure and by highly dynamic deployment conditions.

This thesis is organized to systematically address the energy–complexity trade-off in WSNs. Chapter 1 introduces the fundamentals of WSNs and their energy constraints. Chapter 2 analyzes existing clustering and routing protocols and identifies the research gaps addressed by PUMA-GRID. Chapter 3 presents the PUMA optimizer and its adaptation for CH selection. Chapter 4 details the proposed hybrid clustering and routing methodology. Finally, Chapter 5 presents the simulation setup and provides a comprehensive performance evaluation under different topological scenarios.

Chapter 1

Wireless Sensor Network: Overview

1.1 Introduction

WSN . represents a core technology within the Internet of Things (IoT) ecosystem. It has rapidly evolved and been adopted across diverse application areas. A WSN comprises numerous spatially distributed sensor nodes (Figure 1.1) to observe physical or environmental phenomena and transmit the gathered data wirelessly to a central BS for subsequent processing. Due to their adaptability and broad range of uses, WSNs have attracted considerable research interest, acting as a vital interface between the physical and digital worlds and enabling new opportunities for real time monitoring and data acquisition [1, 2].

This chapter offers a comprehensive overview of WSNs , covering their key components, classifications, applications, and the various challenges encountered in practical deployments[3, 4].

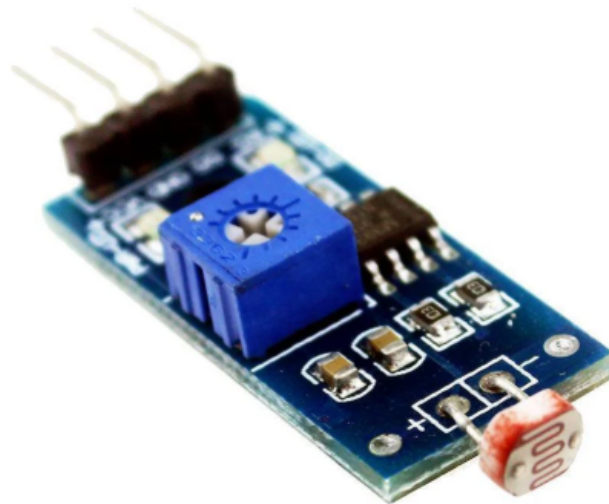


Figure 1.1: Light Dependent Resistor.

1.2 Wireless Sensor Network

A Wireless Sensor Network is a distributed system composed of many low-cost, energy-constrained sensor nodes equipped with sensing, processing, and wireless communication capabilities. These nodes self organize to monitor physical or environmental phenomena such as temperature, pressure, or motion, and collaboratively transmit the collected data, possibly through multi-hop routing, to a central entity for further analysis [5, 6, 7].

1.3 Architecture of Wireless Sensor Networks

A WSN typically comprises three primary components: sensor nodes, base station, and the monitored events. These elements work together to sense, collect, and transmit data, forming the foundation for effective monitoring and communication within the network. The general architecture of a WSN is shown in figure 1.2. Understanding the structure and components of a WSN is essential, as it helps identify key challenges and guides the development of efficient protocols aimed at improving resource management and overall network organization[1].

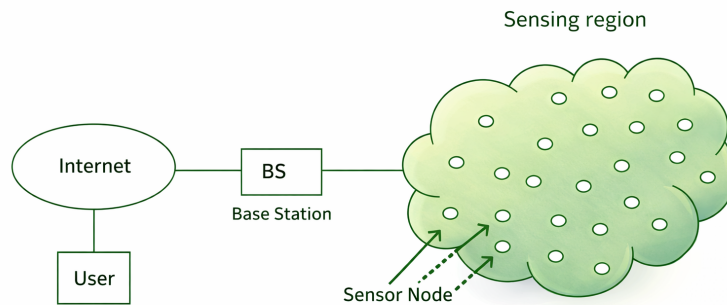


Figure 1.2: WSN Architecture.

1.3.1 Sensors

A sensor is a device that generates an output signal in response to a detected physical phenomenon. Broadly defined, it is a component, machine, or subsystem capable of detecting changes or events in the environment and transmitting the corresponding information to other electronic systems or a computer processor for further processing. Sensors are typically integrated with additional electronic components to enable data acquisition, analysis, and communication. A typical sensor node consists of four basic components[1, 8, 9], which are listed below.

- **Sensing Unit**

The sensing unit is responsible for detecting and measuring physical phenomena in the environment. It typically consists of two subcomponents: the sensor(s) and an Analog to Digital Converter (ADC). The sensor generates analog signals based on environmental conditions such as temperature, light, pressure, or motion. The ADC then converts these analog signals into digital data that can be processed by the node's processor.

- **Processing Unit**

The processing unit controls the operations of the sensor node and manages communication with other nodes. It executes protocols and algorithms that enable efficient sensing tasks. This unit generally includes a microcontroller or microprocessor and a small memory unit for temporary data storage and program execution. It plays a central role in data processing, decision making, and coordination.

- **Transceiver Unit**

The transceiver unit enables wireless communication between the sensor node and other nodes or the BS. It supports both data transmission and reception, allowing the node to send collected information and receive control commands. It typically operates using standards such as IEEE 802.15.4 or ZigBee and is one of the most energy-consuming components due to frequent communication.

- **Power Unit**

The power unit supplies the energy required for the operation of all components in the sensor node. It generally consists of a small capacity battery (e.g., less than 0.5 Ah at 1.2 V). Since replacing or recharging batteries in large scale deployments is impractical, energy conservation is a major design concern in WSNs. In some designs, the power unit may be combined with energy harvesting modules such as solar cells to extend node lifetime. Figure 1.3 presents the functional components of a wireless sensor node.

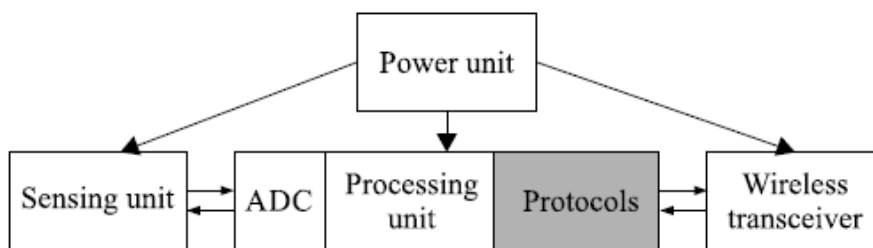


Figure 1.3: Sensors Components [10].

Sensors can be classified according to the physical phenomenon or measured they detect. Table 1.1 presents the main categories of sensors, the parameters they measure, and representative examples used in various WSN applications [11, 12, 13].

Table 1.1: Classification of sensors based on type, measured parameters, and examples.

Sensor Type	Parameters Measured	Examples / Details
Temperature	Temperature, heat	Thermistors, Thermocouples, Resistance Temperature Detectors, IC Sensors, Infrared Sensors, Pyroelectric Sensors, Thermoelectric Sensors
Mechanical / Force	Pressure, position, displacement, strain, vibration, shock, density, weight, load, force, touch	Accelerometers, Strain Gauges, Linear Variable Differential Transformer, Proximity Sensors, Pressure Sensors, Position Sensors
Light / Optical	Light intensity, vision, imaging	Photoelectric Sensors, CMOS Image Sensors, Charge Coupled Device Sensors, Optical Sensors, Photometers, Radiometers, Camera Sensors
Chemical / Gas	Gas and chemical concentration, CO ₂ level, air quality, humidity, moisture, pH level	Electrochemical Sensors, Chemiresistors, Non Dispersive Infrared Sensors, Gas Sensors, Chemical Sensors, Humidity Sensors
Flow / Level	Flow rate, fluid velocity, water level, viscosity	Differential Pressure Sensors, Ultrasonic Sensors, Thermal Displacement Sensors, Level Sensors, Flow Sensors
Acoustic / Sound	Sound, vibration, acoustic energy	Microphones, Sound Sensors
Electromagnetic	Electric current, potential difference, magnetic flux intensity	Magnetometers, Electrical Sensors, Current Transducers, Voltage Transformers, Anisotropic Magnetoresistance Sensors

Sensors can also be classified based on their energy requirements: whether they require an external power source to function or operate by capturing naturally emitted or reflected energy from the environment. Table 1.2 summarizes the main categories and examples [14, 8].

Table 1.2: Classification of sensors based on power source and mode of operation.

Sensor Type	Description	Examples / Details
Active Sensors	Require an external power supply to operate. They emit their own energy signal toward the target and measure the reflected or returned signal.	Light Detection and Ranging, Photoconductive Cells, Radar Sensors, Ultrasonic Sensors
Passive Sensors	Do not require an external power supply. They detect natural radiation or energy emitted or reflected by the observed object.	Radiometers, Charge Coupled Device Sensors, Infrared Sensors, Film Cameras

1.3.2 Base Station

The base station serves as the central component of WSN, responsible for collecting, processing, and storing the data sensed by individual nodes[2]. It acts as an interface between the WSN and the external computational environment, often functioning as a gateway to other networks; such as the IoT, or as an access point for human interaction. Communication within the network typically relies on wireless transmission, where sensed data is forwarded hop by hop through intermediate nodes using low power links until it reaches the BS.

The BS plays a vital role in data aggregation and communication within WSNs . Its placement and mobility significantly influence energy consumption, communication efficiency, and overall network performance[1, 15].

BS is typically positioned to optimize data collection; however, its location relative to the sensor field varies depending on network design and application requirements[16]:

- **Outside the WSN Area:** In several network models, particularly those used in simulations, the BS is positioned outside the sensing field. For instance, in a 100×100 m WSN, it may be placed at coordinates such as (175,50) or (125,50), while larger deployments (e.g., 200×200 m) may position it further away at (275,50). This configuration is often used to evaluate energy consumption in long range communication scenarios.
- **Inside or Centralized within the WSN Area:** The BS can also be placed within the sensor field, often at the center (e.g., at (50,50) in a 100×100 m area), as commonly seen in studies based on the LEACH protocol. A centrally located BS ensures balanced communication distances, minimizing transmission energy and improving network lifetime.
- **As an Interface or Gateway:** The BS functions as an interface between the WSN and external networks (such as IoT), or as an access point for human interaction. It enables the transfer and aggregation of sensed data to centralized storage or cloud platforms for further processing and decision making.

Depending on the network's objectives and scale, the BS may be either stationary or mobile[17]:

- **Stationary (Static BS):** Most conventional WSN architectures employ a fixed BS with ample power and processing capacity. It remains static throughout operation, serving as the final destination for data collected from all nodes.
- **Mobile (Data Sink or Mobile Gateway):** In large-scale or energy-constrained WSNs, the BS can be mobile mounted on vehicles or Unmanned Aerial Vehicles to improve coverage, reduce transmission distance, and enhance energy efficiency. The mobile sink typically follows a predefined trajectory with designated stop points, enabling periodic

data gathering from distributed sensor nodes.

1.3.3 WSN application

The event being monitored often represents the specific application domain of the WSN, as the nature of the monitored phenomenon directly determines the network's purpose and design requirements.

- **Acoustic Monitoring Applications** [18]

Early WSNs, such as the Sound Surveillance System, used acoustic sensing for submarine detection. Modern systems apply similar principles for vehicle detection, classification, and localization based on sound intensity and frequency analysis.

- **Environmental Monitoring** [19]

One of the most widespread WSN applications, involving the measurement of temperature, humidity, air quality, CO₂ levels, water levels, and atmospheric pressure. These systems support pollution control, climate observation, and disaster prediction.

- **Agricultural Monitoring** [20]

WSNs are widely deployed in precision agriculture to optimize crop production and resource usage. They monitor greenhouse variables such as temperature, humidity, soil moisture, CO₂, and soil quality, aiding irrigation management and plant health assessment.

- **Intelligent Transportation Systems** [21]

WSNs improve transportation management through vehicle detection, motion tracking, road condition monitoring, and smart parking. They help reduce congestion, improve traffic flow, and enhance road safety.

- **Healthcare Monitoring** [22]

In Wireless Body Area Networks, WSNs enable real-time health tracking by monitoring heart rate, temperature, SpO₂, blood pressure, and other physiological parameters through wearable or implantable biosensors. These systems are essential for remote healthcare and patient management.

- **Military Surveillance and Security Applications** [23]

WSNs are widely used in military operations for battlefield monitoring, intrusion detection, and target tracking. Networks equipped with magnetometers, acoustic sensors, and passive infrared detectors can detect and track soldiers, vehicles, or enemy activity in real time, enhancing situational awareness and security operations.

1.4 Communication Architecture

The most common WSN architecture follows the Open Systems Interconnection (OSI) model, typically employing five layers and three cross layers[10, 24]. The Five Layers:

1. **Application Layer:** Handles traffic management and provides the software needed to convert sensed data into a usable form for information retrieval.
2. **Transport Layer:** Ensures congestion avoidance and reliability, particularly when the WSN communicates with external networks.
3. **Network Layer:** Its primary function is routing. Due to resource limitations, it focuses on power conservation, managing limited memory and buffers, and supporting self organization since sensor nodes typically lack global identifiers.
4. **Data Link Layer:** Responsible for multiplexing data streams, Media Access Control (MAC), frame detection, and error control.
5. **Physical Layer:** Provides the physical means for transmitting bits, including modulation, signal detection, and frequency selection.

WSN present numerous challenges related to design, resource management, security, and operational constraints, largely due to the limited capabilities of sensor nodes and the harsh environments in which they are typically deployed.

1.5 Challenges of WSN

1.5.1 Resource Limitations

The design and implementation of WSNs are fundamentally limited by the constrained resources of individual sensor nodes[25].

- **Energy Constraints:** Energy consumption represents one of the most critical challenges in WSNs. Since sensor nodes are typically battery powered and often deployed in remote or inaccessible environments, replacing or recharging batteries is usually impractical or impossible. Consequently, efficient energy management is vital to prolong network lifetime [26].

- **Communication Cost:** Communication is the most energy intensive operation in a WSN. Routing protocols must therefore prioritize energy efficient data transmission. Typically, the energy required to transmit a message is almost twice that needed for reception, making communication optimization essential for sustainable operation [27].

- **Processing and Memory Limitations:** Sensor nodes possess limited computational power, memory, and bandwidth, which restricts their ability to execute complex algorithms. Imple-

menting advanced data processing or security mechanisms is often challenging due to these resource constraints [28].

- **Cost of Production:** Although WSNs aim to maintain low production costs, the inclusion of components such as sensors, processing units, localization systems, and mobilizers increases manufacturing expenses. Balancing cost effectiveness with performance and reliability remains an ongoing design challenge [29].

1.5.2 Operational and Design Challenges

WSN face numerous challenges related to deployment planning, operational reliability, and environmental conditions that affect their overall performance and longevity.

- **Deployment and Coverage:** Sensor deployment is often performed in an ad hoc or random manner, making it difficult to ensure optimal area coverage. Poor placement can lead to redundant sensing in some areas and blind spots in others, thereby affecting the quality of service (QoS) and overall monitoring effectiveness [30].

- **Localization:** Identifying the physical positions of sensor nodes after random deployment is a fundamental and challenging task, as nodes are typically unaware of their coordinates. Accurate localization is essential for data interpretation, routing, and efficient network operation [31].

- **Fault Tolerance and Robustness:** Due to deployment in harsh or uncontrolled environments, sensor nodes are prone to failure or malfunction. A fault tolerant network must therefore maintain functionality and data reliability even when individual nodes fail because of environmental factors or physical damage [30].

- **Scalability:** Large scale WSNs, often consisting of hundreds or thousands of nodes, face significant scalability challenges. Traditional communication protocols are unsuitable for such networks because global addressing schemes become inefficient and difficult to manage [32].

- **Real Time Requirements and Latency:** Many WSN applications, particularly those in safety critical domains such as environmental monitoring or contamination detection, require timely data transmission. Excessive latency (typically beyond 200 ms) can compromise early warning capabilities and system responsiveness [3].

- **Synchronization and Calibration:** Time synchronization among sensor nodes is vital for coordinated operations, data fusion, and event detection. Additionally, sensor calibration poses a challenge since low cost devices are susceptible to degradation over time due to factors like temperature fluctuations, humidity, and prolonged environmental exposure [33].

1.6 Routing

Sensor Networks employ different routing topologies, which define how sensor nodes communicate, organize, and forward data toward the BS [34]. The chosen topology determines the logical structure of data transmission paths and has a direct influence on network performance, including energy efficiency, scalability, reliability, and QoS. Selecting an appropriate routing topology is therefore crucial for optimizing resource usage and ensuring effective operation in various application domains [35]. Routing in WSNs can be implemented in different ways, such as flat routing, chain-based routing, cluster-based routing, and hierarchical routing, depending on the network structure and application requirements.

1.6.1 Flat Based Routing

- Structure: All sensor nodes play identical roles and share equal responsibilities for sensing, processing, and communication. There is no predefined hierarchy or clustering among nodes [36, 37].
- Communication: Data is forwarded through multi hop or flooding mechanisms, where each node relays information until it reaches the BS.
- Advantages: Simple to deploy and manage; suitable for small scale networks with uniform node energy levels.
- Disadvantages: Flooding leads to redundant transmissions, packet collisions, and excessive energy consumption; scalability is limited.
- Performance: Offers high reliability for small networks but performs poorly in large-scale WSNs due to rapid energy depletion and network congestion.

Figure 1.4 present how to route data using flat structure.

1.6.2 Hierarchical Based Routing

Hierarchical routing organizes nodes into multiple logical levels to improve energy efficiency and scalability. It includes three main architectures: chain based, tree based, and cluster based [38, 39].

A • Tree Based Routing

- Structure: Nodes are organized hierarchically in a parent–child relationship, forming a spanning tree rooted at the BS [40, 41].
- Communication: Child nodes send sensed data to their parent nodes, which aggregate and

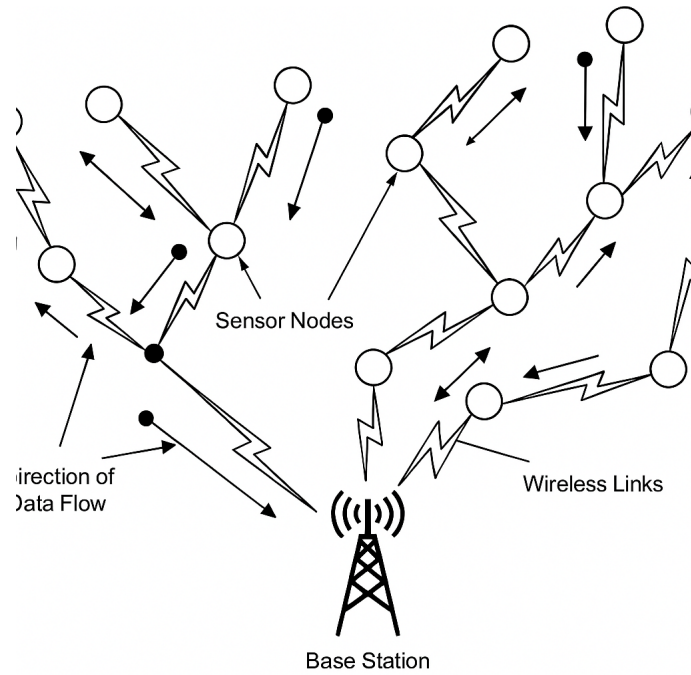


Figure 1.4: Flat Routing.

forward it upward toward the BS.

- Advantages: Ensures organized data aggregation, reduces redundant transmissions, and supports scalable network expansion.
- Disadvantages: Vulnerable to parent node failures that can disconnect entire subtrees; maintenance overhead increases with node mobility.
- Performance: Balances scalability and efficiency but requires fault tolerant mechanisms to handle node or link failures.

The architecture shown in Figure 1.5 demonstrates how tree based routing organizes nodes hierarchically to deliver data.

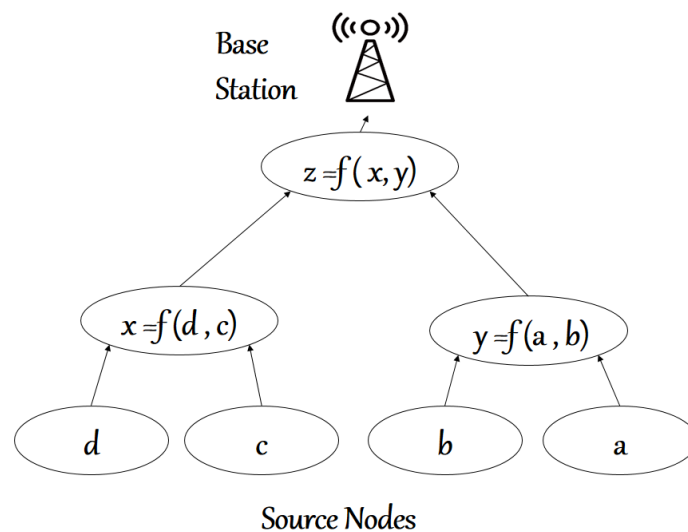


Figure 1.5: Tree Routing [42].

B • Chain Based Routing

- Structure: Sensor nodes form a linear or near linear chain, where each node communicates only with its immediate neighbors [43, 44].
- Communication: Data is aggregated sequentially along the chain, with each node forwarding information to the next until the final node sends the fused data to the BS.
- Advantages: Reduces the number of transmissions and achieves efficient data aggregation, minimizing total energy usage.
- Disadvantages: Increased latency due to sequential forwarding; chain breakage from node failure can disrupt communication.
- Performance: Highly energy efficient for small to medium sized networks but less suitable for real time or large scale applications requiring low latency.

As shown in Figure 1.6, chain based routing enables sequential data transmission between nodes, reducing the number of direct communications with the BS and improving energy efficiency.

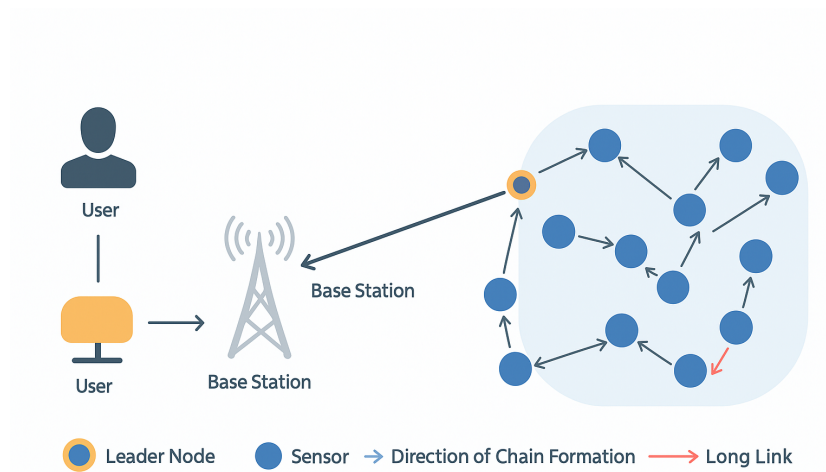


Figure 1.6: Chain based routing architecture in WSNs [45].

C • Cluster Based Routing

- Structure: The network is divided into clusters, each managed by a Cluster Head responsible for coordinating intra cluster communication and data aggregation [7].
- Communication: Member nodes send their data to the CH, which aggregates and transmits the processed information to the BS either directly or via other CHs.
- Advantages: Significantly reduces transmission load and energy consumption; enhances scal-

ability and supports hierarchical data management.

- Disadvantages: CH selection and rotation introduce computational overhead; uneven cluster formation can lead to energy imbalance.
- Performance: Provides a good trade off between energy efficiency, scalability, and network lifetime—making it one of the most widely adopted architectures in WSNs.

Figure 1.7 outlines the clustering based routing mechanism, in which nodes form clusters to minimize long distance transmissions and prolong overall network life

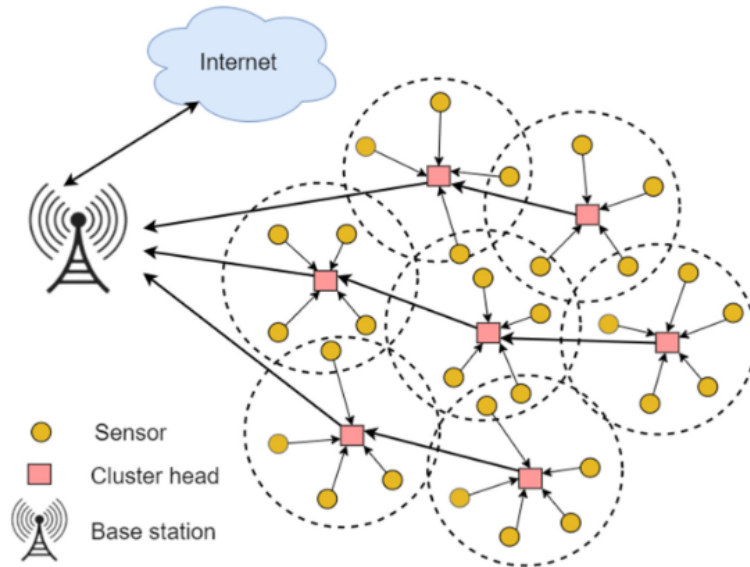


Figure 1.7: Clustering based routing architecture in WSNs [46].

1.7 Clustering

1.7.1 Introduction to clustering

Clustering in WSNs is a fundamental technique designed to enhance the efficiency and organization of network communication[7]. It involves dividing the network into smaller groups, called clusters, each managed by a CH that acts as a coordinator for its member nodes. The CH is responsible for collecting data from the nodes within its cluster, performing data aggregation to eliminate redundancy, and transmitting the aggregated information to the BS [47]. The main purpose of clustering is to reduce energy consumption, minimize redundant transmissions, and extend the overall network lifetime, especially in large-scale WSN deployments [46]. It plays a vital role in addressing the scalability and energy challenges inherent in WSNs. As the number of sensor nodes increases, direct communication between each node and the BS becomes inefficient and energy consuming[48]. Clustering alleviates this issue by enabling localized communication, where nodes transmit data only to their respective CHs,

thereby conserving energy and reducing transmission distances. Additionally, by balancing the workload among the nodes and rotating the CH role periodically, clustering prevents energy imbalance and hotspot formation, ensuring that no node prematurely depletes its battery [49]. This hierarchical organization also improves network scalability, allowing WSNs to efficiently manage thousands of nodes without overwhelming the communication or routing infrastructure [46]. Clustering protocols generally operate through two main phases [50]:

- **Cluster Formation and Cluster Head Selection:** Nodes are grouped into clusters based on criteria such as distance, residual energy, or signal strength. A node in each cluster is elected as the CH using deterministic, probabilistic, or optimization based selection mechanisms.
- **Intra Cluster and Inter Cluster Communication:** Member nodes collect environmental data and send it to the CH using short range communication. The CH aggregates or fuses the received data to reduce redundancy, then transmits the aggregated information to the BS, either directly or through multi hop communication via other CHs. Periodic re clustering or CH rotation ensures adaptability and balanced energy consumption.

1.7.2 Cluster Formation Process

Clustering techniques are generally classified into two categories based on how the clustering process is carried out [51]:

- **Centralized Clustering**

In centralized clustering, the BS is responsible for managing the entire network. It performs key operations such as CH selection, cluster formation, and configuration of communication links. With global knowledge of the network, the BS can make optimized decisions that balance energy consumption and enhance overall performance. However, this approach introduces high communication overhead and a single point of failure, making it less suitable for large-scale or dynamic networks.

- **Distributed Clustering**

In distributed clustering, decision making is performed locally by the sensor nodes without continuous dependence on the BS. Each node uses localized information—such as residual energy, node density, or distance to determine its role and form clusters with nearby nodes. This approach improves scalability and robustness, allowing the network to adapt to topology changes. However, it may result in suboptimal cluster formation due to the absence of global network information.

Table 1.3 presents the key metrics that influence both cluster formation and CH selection in WSNs [52]. Parameters such as energy level, distance to the BS, node density, and cluster size serve as essential indicators for identifying suitable CH candidates and organizing nodes into efficient clusters. By evaluating these metrics, clustering protocols can balance energy consumption, minimize communication cost, and extend the overall network lifetime [53, 54].

Table 1.3: Summary of key metrics and parameters influencing clustering and CH selection in WSNs.

Metric / Parameter	Description / Definition	Role in Clustering / CH Selection
Residual Energy (E_{res})	Remaining energy of a node at a given round.	Identifies nodes capable of supporting aggregation and transmission tasks.
Initial Energy (E_0)	Initial battery capacity of each node.	Helps prioritize stronger nodes for CH selection.
Average Energy (E_{avg})	Mean residual energy across all nodes.	Regulates CH probability and fitness based selection.
Energy Consumption	Energy spent on sensing, processing, and transmitting.	Minimizing this cost improves the longevity of clustering rounds.
Energy Balance	Uniform power usage among nodes.	Avoids early failures and enhances stability.
Distance to BS	Euclidean distance between node and BS.	Closer nodes consume less energy when transmitting.
Distance to CH / Members	Distance between CH and its cluster nodes.	Reducing this distance minimizes intra-cluster cost.
Inter CH Distance	Distance separating neighboring CHs.	Ensures good distribution and improves routing.
Node Centrality	Position relative to neighbors or the network center.	Helps pinpoint optimal CH locations.
Location Awareness	Node coordinate information (X,Y).	Enables distance based clustering and structured topologies.
Node Degree (n_i)	Number of neighbors directly reachable.	Indicates local density and aids in balancing cluster size.
Cluster Density / Radius	Influence radius of a CH.	Prevents cluster overlap and helps maintain energy balance.
Link Quality	Reliability of wireless connections.	Ensures stable intra cluster communication.
Cluster Size	Number of nodes attached to a CH.	Balanced sizes avoid overload and improve energy distribution.
Cluster Count (N_C)	Number of clusters created.	Influences scalability and routing overhead.
Communication Type	Single hop or multi hop.	Determines transmission cost and network reachability.
Mobility	Speed or movement pattern of nodes.	Low mobility increases cluster stability.
Node Heterogeneity	Differences in energy or capability.	High energy nodes make better CHs.
Latency / Delay	Time taken for routing or packet transfer.	Works as a QoS constraint in CH selection.

1.7.3 Advantages of Clustering

The clustering mechanism offers several core benefits[25]:

- **Energy Efficiency and Network Lifetime Extension:** By localizing communication, clustering minimizes long distance transmissions one of the most energy intensive operations in WSNs. CHs perform data aggregation to remove redundancy and compress information before sending it to the BS, conserving both energy and bandwidth. This results in prolonged network lifetime and extended stability periods before the first node failure[52].
- **Load Balancing:** Rotating the CH role periodically distributes energy consumption evenly among nodes, preventing premature energy depletion. Techniques such as unequal clustering help mitigate the hotspot problem, ensuring fair energy use across the network.[50]
- **Scalability and Simplified Network Management:** Clustering efficiently manages large scale WSNs by introducing a hierarchical structure that reduces communication complexity. It enables easier administration and resource allocation, even when thousands of nodes are deployed [7].
- **Quality of Service (QoS) and Reliability:** Through mechanisms like TDMA scheduling, clustering reduces collisions and delays during data transmission. It enhances connectivity, stability, and throughput, while dynamic CH selection improves fault tolerance and network robustness [1].
- **Data Aggregation and Processing:** The CH performs data fusion, combining similar data from member nodes to minimize redundancy. This not only optimizes bandwidth usage but also reduces unnecessary data traffic to the BS[50].

1.7.4 Challenges of Clustering in WSNs

Clustering in WSNs is a complex and still open research problem. Although it significantly improves energy efficiency and prolongs network lifetime, several challenges continue to hinder the design of highly effective clustering protocols [30]. The main issues are summarized as follows:

1. Energy Management and Lifetime Constraints.

Energy is the most critical limitation in WSNs because nodes typically operate on small, non rechargeable batteries.

- **Energy Holes and Uneven Consumption:** Single hop models such as LEACH create “energy holes,” where CH near the BS deplete energy rapidly due to excessive relay tasks. Unequal clustering methods aim to reduce this hotspot issue [3].

- **Suboptimal CH Selection:** Random CH election used in classical LEACH does not consider residual energy or distance, resulting in early node deaths and inefficient energy use.

- **High Energy Burden on CHs:** CHs consume significantly more energy for data aggregation and long range transmission; hence, frequent role rotation is necessary to balance consumption [55].

2. Scalability and Complexity Issues

Large scale WSNs with hundreds or thousands of nodes introduce substantial scalability challenges.

- **Algorithmic Complexity:** Determining optimal CH positions and forming balanced clusters is an NP hard problem requiring heavy computation.

- **Communication Overhead:** Centralized approaches (e.g., LEACH-C) require continuous reporting to the BS, while distributed protocols cause high message exchange during CH elections. Both increase energy consumption.

- **Management Complexity:** As the network expands, synchronizing communication, maintaining stable clusters, and ensuring effective data aggregation become more complicated.

3. Network Stability and Topology Dynamics

The highly dynamic nature of WSNs affects cluster reliability.

- **Cluster Instability:** Protocols with random or inconsistent clustering (e.g., SEP) trigger frequent re-clustering, which wastes energy and reduces network stability.

- **Real Time Adaptation Limits:** In centralized protocols such as LEACH-C, updating CH positions in real time is difficult, especially in mobile or event driven applications.

4. Data Integrity, Security, and Quality of Service (QoS)

Beyond energy efficiency, ensuring secure, reliable, and high quality data transmission is still challenging [56].

- **Data Loss and Low Throughput:** Poor clustering increases packet loss and decreases throughput, especially when physical security is weak [48].

- **Security Vulnerabilities:** Limited computational resources restrict the use of strong cryptography, making WSNs vulnerable to gray hole, black hole, and other routing attacks [56].

- **QoS Constraints:** Meeting requirements for latency, reliability, and fault tolerance can be difficult; multi layer or hierarchical clustering may introduce additional delays.

5. Algorithmic Limitations

Many optimization based clustering methods suffer from built in weaknesses.

- **Parameter Sensitivity:** Performance strongly depends on parameters such as CH percentage and weight coefficients tuning them optimally is NP hard.

- **Local Optima:** Numerous metaheuristics (PSO, Firefly, etc.) may prematurely converge, producing suboptimal CH sets.
- **Integration Complexity:** Hybrid and multi objective algorithms require precise coordination among parameters, increasing algorithmic complexity.

1.8 Conclusion

Clustering plays a central role in improving the performance of WSNs by reducing energy consumption and extending network lifetime. It achieves this by organizing nodes into groups and assigning CHs to manage local communication and data aggregation. However, despite its efficiency, clustering remains a challenging task. Selecting the optimal CHs, forming balanced clusters, and determining effective forwarding paths are complex, NP hard problems that greatly influence the overall network performance.

The next chapter provides a review of several clustering based protocols proposed to address these challenges, focusing on how they aim to minimize energy consumption and enhance the stability and lifetime of the network.

Chapter 2

Literature Review

2.1 Introduction

Clustering is one of the most effective techniques used to enhance the scalability, energy efficiency, and overall lifetime of WSNs. In this method, sensor nodes are organized into clusters, each managed by a CH responsible for collecting, aggregating, and transmitting data to the BS. This hierarchical communication structure minimizes redundant transmissions, reduces long distance data transfers, and distributes energy consumption more evenly across the network. However, the selection of optimal CHs, the formation of well balanced clusters, and the routing of aggregated data toward the BS remain challenging optimization problems due to the limited energy resources and dynamic topology of WSNs. Over the past two decades, numerous clustering based protocols have been proposed to address these issues, each employing different strategies for energy balancing and network stability. These protocols can be broadly classified into classical schemes (e.g., LEACH, Hybrid Energy-Efficient Distributed (HEED), Power-Efficient GATHERING in Sensor Information Systems (PEGASIS)), fuzzy logic based methods, metaheuristic and swarm intelligence based algorithms (such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Grey Wolf Optimizer (GWO)), and machine learning driven techniques including k-means, Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and neural network assisted clustering. More recent studies have introduced hybrid and adaptive models that integrate multiple optimization strategies to achieve a better trade off between energy efficiency, stability, and communication overhead.

The following section reviews and compares representative clustering protocols from these categories, highlighting their approaches to CH selection, cluster formation, and data routing within energy-efficient WSN and IoT environments.

2.2 LEACH

Heinzelman et al. proposed in [57], a protocol called LEACH to reduce energy consumption through cluster based organization. Each cluster is composed of ordinary member nodes and a CH responsible for aggregating data and forwarding it directly to the BS in a single hop. The LEACH protocol functions in periodic rounds, with each round divided into a setup phase followed by a steady-state (data transmission) phase. During the setup phase, CHs are selected autonomously based on a threshold and a random number, ensuring distributed election across

the network, using the following equation:

$$T(n) = \begin{cases} \frac{P}{1 - P \times \left(r \bmod \frac{1}{P}\right)} & \text{if } n \in G, \\ 0 & \text{otherwise.} \end{cases} \quad (2.1)$$

After CHs are chosen, non-CH nodes join the nearest CH based on received signal strength, and the CH creates a TDMA schedule that assigns transmission times to its members. In the steady-state phase, nodes transmit their sensed data to the CH according to the TDMA schedule, then switch to sleep mode to save energy. The CH aggregates the received data and transmits it to the BS. This process repeats in multiple rounds allowing CH rotation. LEACH significantly improves network lifetime compared to direct transmission by reducing communication distance and evenly distributing the energy load among all nodes. Its distributed and probabilistic CH election mechanism eliminates the need for centralized control, which makes it simple and scalable for small to medium-sized networks. However, despite these advantages, LEACH also presents several limitations. The random selection of CHs can lead to an uneven number and spatial distribution of CHs, which may cause energy imbalance and premature death of nodes in certain regions of the network. Furthermore, because CHs transmit aggregated data to BS using a single-hop communication model, the protocol becomes less energy-efficient in large-scale networks or when the BS is located far from the sensing field. Consequently, while LEACH provides better energy efficiency than direct transmission, its performance declines under more complex or large-scale deployment scenarios, motivating the development of subsequent clustering protocols that incorporate residual energy awareness, multi-hop routing, and optimized CH selection strategies to overcome these shortcomings. Several modifications have been introduced to improve the performance of the original LEACH protocol. These enhanced versions aim to optimize CH selection, balance energy use, and extend network lifetime. Some of these improved protocols are discussed in the following sections.

2.2.1 Multi-Hop Routing LEACH

Multi-hop LEACH [58] extends the original LEACH protocol to enhance energy efficiency, particularly in large-scale WSNs. In this protocol, the BS actively participates in the cluster discovery process to identify CHs at different hierarchical levels. Initially, the BS broadcasts a control message to discover CHs located within one hop, forming the first communication layer. The CHs that are farther away respond with lower transmission power, and their messages are relayed through intermediate CHs belonging to the previously formed layers. This procedure

continues iteratively, with the BS updating its list of discovered CHs at each step, until no new CHs are detected thus establishing a multi level hierarchical structure for data forwarding. Following cluster formation, CHs send aggregated data to the BS using optimal multi-hop paths. CHs close to the BS send their data directly, while those located farther away rely on intermediate CHs to relay the information (Figure 2.1). The setup phase in Multi-hop LEACH remains identical to that of the original LEACH, involving cluster formation and CH election, whereas the steady-state phase introduces inter-cluster communication between CHs. This hierarchical, multi-hop routing mechanism reduces long-distance transmissions, minimizes overall energy dissipation, and extends the network's operational lifetime compared to the single-hop communication used in the original LEACH protocol.

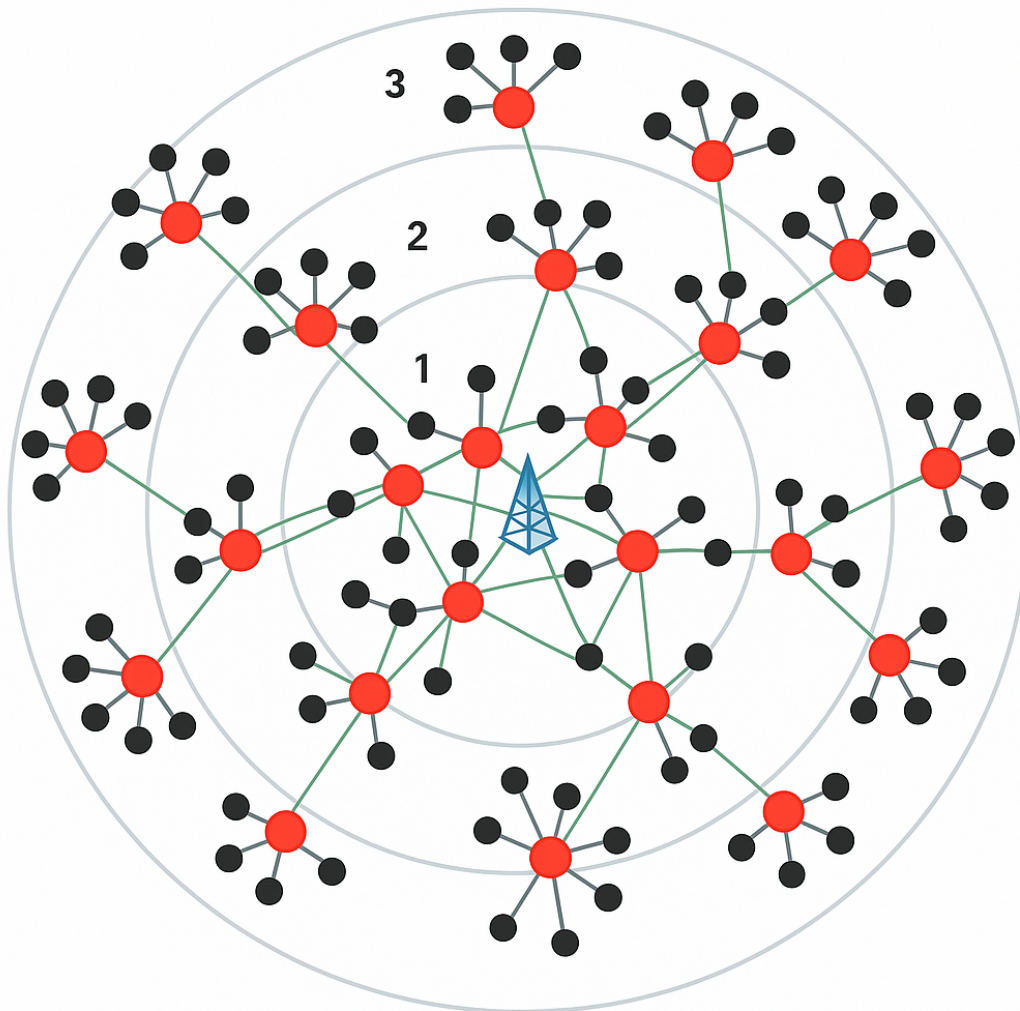


Figure 2.1: Multi-Hop Routing LEACH [58].

2.2.2 LEACH-C

Heinzlam et al proposed a centralized extension of LEACH [59], where cluster heads are selected by BS based on node energy levels using a SA approach. The enhancement primarily focuses on the setup phase. During this phase, each node sends its location and energy level to the BS. The BS eliminates nodes with energy levels below the network's average energy level. Then, using simulated annealing, the BS selects CHs from the remaining nodes and forms clusters. The steady phase of this approach remains the same as in LEACH. Through experiments and tests, LEACH-C has demonstrated improved packet delivery compared to LEACH, along with reduced energy consumption. This improvement is achieved by controlling the number of clusters.

2.2.3 Two-Level Hierarchy for Low-Energy Adaptive Clustering Hierarchy (TL-LEACH)

TL-LEACH [60] is a protocol that extends the original LEACH structure by introducing an additional hierarchical level to enhance energy efficiency and prolong the network lifetime in WSNs. The main operation of TL-LEACH relies on the creation and rotation of a two-level clustering structure that allows localized data aggregation and reduces the average transmission distance between nodes. During the setup phase, TL-LEACH organizes the network into three categories of nodes: primary CHs (CH_i), secondary CHs (CH_{ij}), and simple nodes (SNs). The primary CHs represent the top-level CHs. They are responsible for collecting aggregated data from secondary CHs and transmitting the final, fused data directly to the BS. The secondary CHs form an intermediate layer between the simple nodes and the primary CHs. Each secondary CH collects data from its associated simple nodes, performs partial data aggregation, and forwards the processed data to its corresponding primary CH. The simple nodes are the ordinary sensing nodes that collect environmental data and transmit it to their designated secondary CH. This structure introduces a two-level (Figure 2.2) hierarchy compared to the single-level clustering used in the original LEACH, resulting in improved data management and reduced communication cost. TL-LEACH operates in rounds, similar to LEACH, and each round consists of four main phases: advertisement, cluster setup, schedule creation, and data transmission. In the advertisement phase, nodes independently decide whether to act as CH_i , CH_{ij} , or SNs and broadcast their status using the Carrier Sense Multiple Access mechanism. During the cluster setup phase, each secondary CH associates itself with a primary CH, and each simple node joins the nearest secondary CH based on signal strength, leading to the formation of a

two-level hierarchical structure. The schedule creation phase involves the generation of TDMA schedules and the assignment of CDMA codes by the primary CHs, which are then distributed through the secondary CHs to the simple nodes. Finally, in the data transmission phase, communication occurs in two levels: simple nodes send their data to the associated secondary CH, which aggregates and forwards the data to the primary CH. The primary CH then performs final aggregation and transmits the processed data directly to the BS. The energy efficiency of TL-LEACH is achieved through reduced transmission distances, distributed data fusion, and balanced load distribution. Shorter communication paths reduce the energy required for data transmission, while performing data aggregation at both secondary and primary CH levels minimizes the total amount of transmitted data. In addition, the randomized rotation of both primary and secondary CHs distributes the energy load more evenly among nodes, preventing early node failures. Simulation results show that TL-LEACH achieves smoother energy consumption and extends the network lifetime by approximately 30% compared to the original LEACH protocol, demonstrating its effectiveness in dense and large-scale WSN deployments.

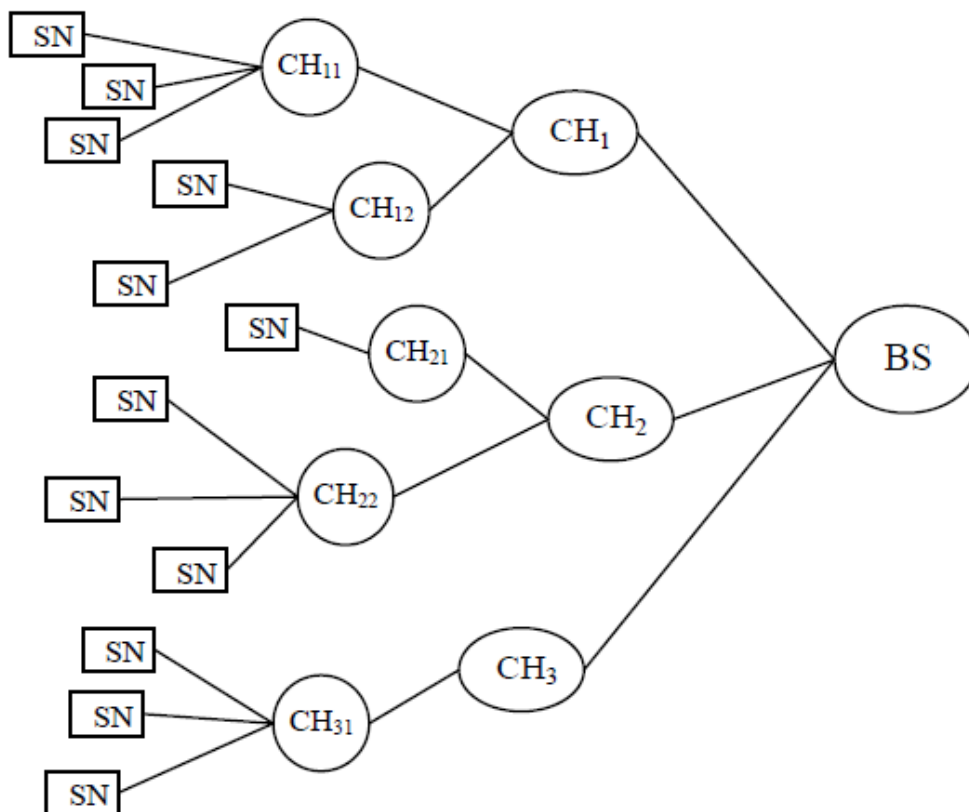


Figure 2.2: Two-Level LEACH [60].

2.2.4 Weighted Low-Energy Adaptive Clustering Hierarchy (W-LEACH)

W-LEACH[61] is a data aggregation protocol for WSNs designed to reduce energy consumption by minimizing redundant transmissions and controlling data flow. It enhances LEACH by using a weighting mechanism that accounts for both residual energy and local node density, enabling more balanced and efficient CH selection. Unlike LEACH, which randomly elects CHs, W-LEACH operates in a centralized manner. In each round, the BS computes a weight w_i for every sensor node s_i using its remaining energy e_i and density d_i , defined as:

$$w_i = \begin{cases} e_i \times d_i, & \text{if } d_i > d_{\text{thresh}} \\ d_i, & \text{otherwise.} \end{cases} \quad (2.2)$$

where d_{thresh} is the density threshold used to identify low-density regions. The node density d_i is given by:

$$d_i = \frac{1 + \text{number of alive sensors within range } r}{n} \quad (2.3)$$

where r is the communication range and n the total number of active nodes. Nodes in denser areas with higher energy are prioritized for CH selection, while those in sparse zones transmit less frequently to conserve energy. W-LEACH operates in two phases. In the *setup phase*, the BS selects up to $p\%$ of nodes with the highest weights as CHs, ensuring no two CHs are within range r . In the *steady-state phase*, only a small fraction ($x\%$) of sensors send data to their CHs, and nodes in low-density areas ($d_i \leq d_{\text{thresh}}$) transmit with a probability p_s . CHs aggregate and forward compressed data to the BS. Simulations demonstrate that W-LEACH enhances energy balance, extends the stability period, and increases network lifetime compared to LEACH, especially in non-uniform and heterogeneous WSNs.

2.2.5 Distance-Based Cluster Head Algorithm (DBCH)

The Distance Based Cluster Head (DBCH)[62] approach, proposed by Sharma et al. in their work entitled “A Proposed Energy Efficient Distance-Based Cluster Head (DBCH) Algorithm: An Improvement over LEACH,” enhances the traditional LEACH protocol by introducing a modified CH election threshold that considers both the residual energy and the distance to the BS. In this approach, each node independently determines whether to become a CH or remain a normal node based on its residual energy, initial energy, and distance to BS. By integrating these factors, DBCH ensures that nodes closer to the BS and with higher remaining energy have a greater probability of being elected as CHs. This modification minimizes long-distance transmissions, balances energy consumption among nodes, and extends the overall network

lifetime. Data transmission from CHs to the BS occurs through a single-hop communication model, which reduces energy loss and improves efficiency compared to the original LEACH protocol. The threshold function used for CH selection in the DBCH algorithm is defined as:

$$T(n) = \frac{p}{1 - p \times \left(r \bmod \frac{1}{p}\right)} + (1 - p) \left(\frac{D_{\max} - D_{i \rightarrow BS}}{D_{\max} - D_{\min}} \right) \left(\frac{E_R}{E_0} \right) \quad (2.4)$$

where:

- p is the desired percentage of CHs,
- r is the current round number,
- $D_{i \rightarrow BS}$ is the distance between node i and the BS,
- D_{\max} and D_{\min} are the maximum and minimum distances from nodes to the BS,
- E_R is the residual energy of node i ,
- E_0 is the initial energy of the node.

This threshold enhances the CH election process by favoring nodes that are both closer to the BS and possess higher remaining energy, thereby achieving better energy balance and improving the overall network performance compared to LEACH.

2.2.6 LEACH with Vice Cluster Head (LEACH-VH)

In the original LEACH protocol, the CH remains active throughout the communication phase. Due to continuous overhearing and idle listening, the CH consumes more energy and may die prematurely. Once the CH fails, the cluster becomes non-functional, and data can no longer reach the BS. To address this limitation, the LEACH-VH[63] protocol was introduced. Each cluster includes a secondary node called the Vice CH, which acts as a backup CH when the original CH fails. Initially, all nodes have equal energy, and the setup phase remains identical to that of LEACH. CHs are randomly selected in the first round, and during subsequent rounds, each CH nominates a node with higher residual energy as its VCH. The VCH remains in a low-power sleep mode to conserve energy and becomes active only when the current CH depletes its energy or fails. This mechanism ensures cluster continuity, as the VCH immediately replaces the failed CH, maintaining uninterrupted data transmission to the BS. Consequently, LEACH-VH enhances network reliability, minimizes data loss, and extends the overall network lifetime compared to the traditional LEACH protocol. According to [64], the introduction of a Vice CH significantly improves performance in terms of stability period and data delivery efficiency. However, this enhancement comes at the cost of slightly increased energy consumption during the setup phase, as additional resources are required to elect and manage the Vice CH within each cluster.

2.3 Clustering using Fuzzy Logic

Clustering based on fuzzy logic has emerged as an effective approach in WSNs to enhance energy efficiency, achieve balanced energy consumption, and apply human like reasoning in decision making processes. By incorporating fuzzy inference systems[65], these protocols handle uncertainties in node parameters such as energy, distance, and connectivity, leading to more adaptive and intelligent CH selection and overall network optimization.

2.3.1 Cluster Head Election Mechanism Using Fuzzy Logic (CHEF)

In their study[66], Kim et al. proposed a clustering method inspired by LEACH, where cluster head (CH) selection is guided by a fuzzy inference system. During the selection process, each node generates a random value and assesses its probability of becoming a CH using fuzzy logic. The system uses two key inputs: the node's remaining energy, categorized by three linguistic levels (low, medium, high), and the distance to neighboring nodes, also expressed in three terms (close, medium, far). These inputs are used to define nine fuzzy rules that evaluate the suitability of each node for CH responsibilities. The defuzzification is performed via the Center-of-Area method, yielding a clear decision on whether a node should function as a CH or remain a regular member. Each node then broadcasts its computed probability, and the node with the highest value among its neighbors assumes the CH role. Nodes with lower probabilities send join requests to the nearest CH. Once clusters are established, CHs aggregate data from their members and forward it to the base station. Simulation results indicate that this fuzzy-based approach enhances network performance by extending network lifetime and ensuring a more uniform cluster distribution. However, the method has a limitation: calculating the sum of distances to all neighboring nodes increases computational load in dense networks, which may reduce efficiency.

2.3.2 An Energy-Aware Fuzzy Approach for Uneven Clustering in WSNs (EAUCF)

EAUCF [67] proposed by Bagci and Yazici, is a clustering algorithm designed to extend network lifetime by balancing the workload among nodes. Similar to LEACH, it divides the network into unequal clusters—smaller clusters near the BS and larger clusters farther away. Initially, the BS broadcasts the maximum competition radius to all nodes. During the CH election, tentative CHs are randomly selected. Each tentative node evaluates a fuzzy IF-THEN rule based on its residual energy (Low, Medium, High) and its distance to the BS (Near,

Medium, Far). The fuzzy inference system uses nine rules, and the Mamdani model with the Center-of-Area method is applied for defuzzification, producing the node's competition radius. Each tentative CH then broadcasts its ID, residual energy, and competition radius. Nodes with higher energy are more likely to become CHs, while others join the nearest cluster. CHs aggregate data from their members and forward it to the BS. While the protocol improves network lifespan, its main inconvenience lies in the added computational complexity due to fuzzy logic calculations and message exchanges during CH election.

2.3.3 Distributed Load Balancing Unequal Clustering in WSNs using Fuzzy Approach (DUCF)

Baranidharan and Santhi proposed the DUCF protocol[68], which employs fuzzy logic for cluster formation and CH election. The protocol proceeds in rounds, each including cluster setup and subsequent data gathering. During the cluster formation phase, each node runs a fuzzy inference system (Figure 2.3) with three inputs: residual energy (High, Average, Low), node degree (number of neighbors within a radius: High, Medium, Low), and distance to the BS (Near, Reachable, Distant). The system uses 27 rules and produces two outputs: the probability of the node becoming a CH and the cluster size it can accommodate (seven linguistic variables). Nodes with higher probabilities within a given radius are selected as CHs, while others join the nearest CH. Each CH broadcasts an advertisement message, and nodes send membership requests. The CH checks its cluster size to accept or reject new members; rejected nodes try the next closest CH, and in the worst case, a node becomes a CH itself. Once clusters are formed, CHs assign a TDMA schedule for their members. During the data collection phase, nodes transmit data according to this schedule, and CHs aggregate the data before forwarding it to the BS either directly or via the nearest CH in a multi-hop path.

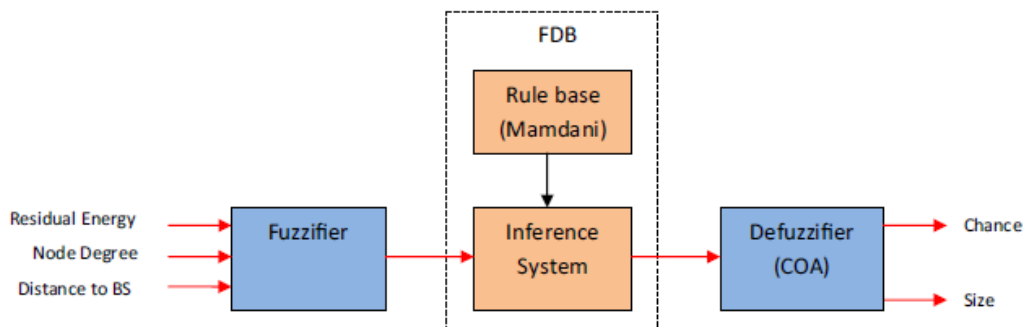


Figure 2.3: Fuzzy inference system of DUCF [68].

Simulations show that DUCF effectively balances energy consumption, extends network

lifetime, and improves message transmission. By limiting cluster size, it ensures better resource utilization. However, as the number of dead nodes grows, orphan nodes may become CHs, causing uneven energy distribution and higher energy consumption in packet forwarding.

2.3.4 Enhancing Energy Efficiency in WSNs through a Fuzzy Rule-Based Clustering Algorithm (EETPF)

Lipare et al proposed a fuzzy rule-based clustering protocol [69] aimed at improving energy efficiency in WSNs. The algorithm forms clusters and efficiently routes data to the BS by operating in rounds, each consisting of two main phases: clustering and routing. The clustering phase is divided into three sub-phases: fuzzy CH competition, CH selection, and cluster formation. During the fuzzy CH competition, nodes apply fuzzy rules using three input parameters: residual energy (low, medium, high), distance to the BS (short, reachable, long), and the number of neighboring nodes (less, medium, large). The outputs determine each node's probability of becoming a CH and its data processing capacity. Next, in the CH selection phase, sensor nodes broadcast messages containing their identifier, probability, and capacity. Nodes compare their probability with neighbors and send join requests to nodes with higher probabilities. If a node receives join requests exceeding its capacity, it assumes the role of CH and broadcasts this status. During cluster formation, CHs send acceptance messages to nearby non-CH nodes based on capacity. If capacity is exceeded, a rejection message is sent. Non-CH nodes receiving multiple acceptances select the nearest CH to join. In rare cases, a node may declare itself a CH. Non-CH nodes then forward sensed data to their CHs according to a TDMA schedule. CHs aggregate this data and transmit it to the BS via single-hop or multi-hop routes depending on distance. The routing phase involves selecting certain CHs as routers to establish an efficient path to the BS. This phase also has three sub-phases: fuzzy router competition, router selection, and routing path formation. In the competition sub-phase, CHs apply fuzzy rules using residual energy, distance to the BS, and the number of connected nodes as inputs, producing probability and capacity as outputs. During router selection, CHs are sorted by distance to the BS, and candidate messages containing probability and capacity are broadcast within range. CHs then compare received messages, sending responses to nodes with lower probabilities but feasible capabilities (Figure 2.4). The process continues until the routing path reaches the BS. If no messages are received, a CH directly transmits data to the BS. This protocol effectively reduces energy consumption in each round and extends network lifetime by optimizing cluster formation and routing paths. However, CHs may experience higher workloads, necessitating additional CHs, and scenarios with many orphan nodes may

lead to increased energy consumption.

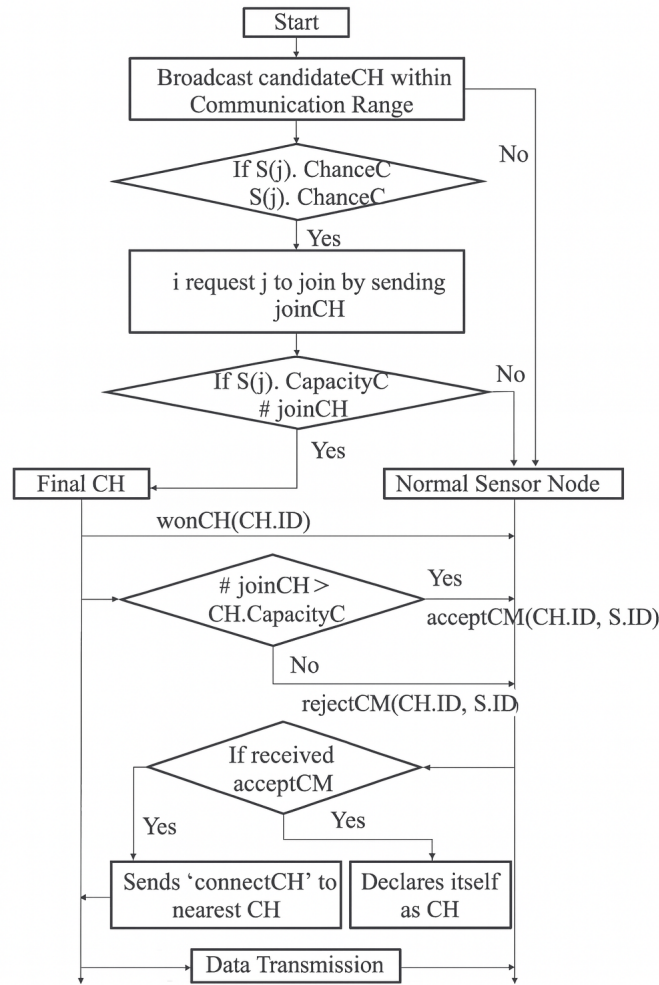


Figure 2.4: EETPF Flow Chart [69].

2.3.5 Fuzzy Logic Based Distributed Energy Efficient Clustering Algorithm for WSNs (EEDCF)

The EEDCF [70] protocol, proposed by Zhang et al., is a decentralized clustering algorithm designed to minimize energy consumption in WSNs and reduce intra cluster communication. It employs fuzzy logic with three input variables: a node's residual energy, its degree (number of neighboring nodes within a certain radius), and the average energy of its neighbors. During cluster formation, each node gathers energy information from its neighbors and calculates the average neighbor energy. Using a fuzzy inference system with three linguistic variables for each input—residual energy (High, Medium, Low), node degree (High, Medium, Low), and average neighbor energy (Weak, Normal, Strong) the node determines its probability of becoming a CH. The Mamdani model is applied for fuzzification, and the Center-of-Area method is used for defuzzification. A node becomes a CH if its probability exceeds that of its neighbors;

otherwise, it joins the nearest CH. In rare cases, a node becomes a CH if no nearby clusters are available. Data transmission follows the same procedure as LEACH (Figure 2.5). While EEDCF improves overall energy efficiency, the protocol introduces extra energy consumption due to frequent exchanges of residual energy information, and neglecting the distance to the BS can result in an excessive number of CHs.

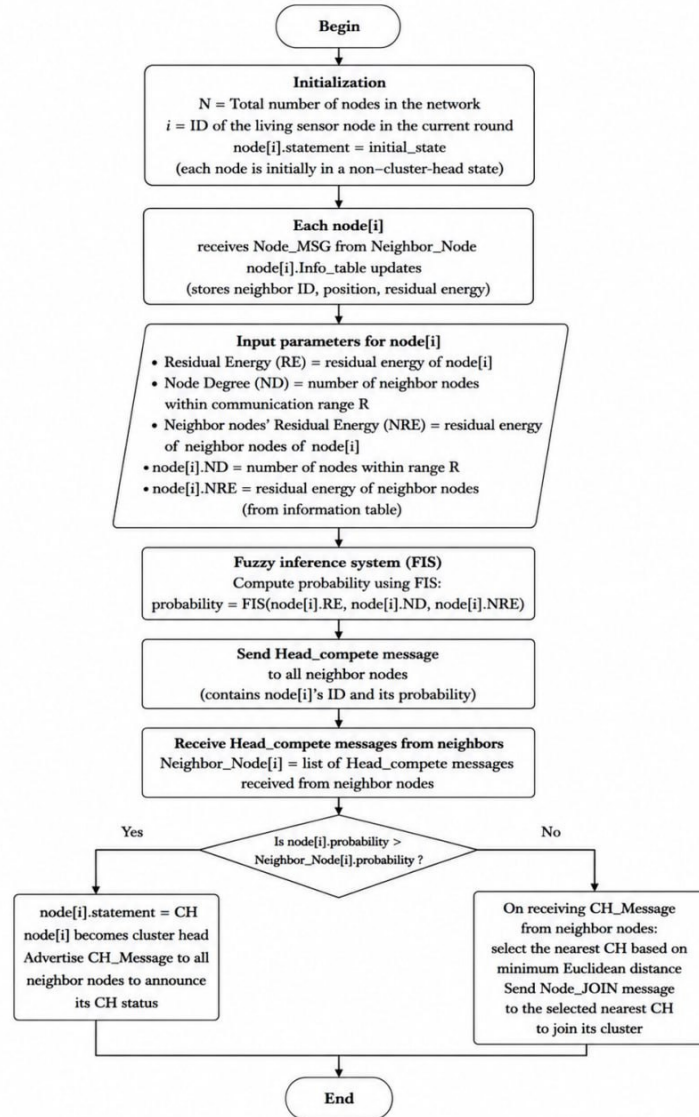


Figure 2.5: EEDCF Flow Chart.

2.3.6 Fuzzy-Based Cluster Routing in WSN (FBCR)

Naderloo et al. introduced in [71], a fuzzy based clustering protocol for WSNs named FBCR. This protocol partitions the network into four regions and selects CHs using fuzzy logic. The operation proceeds in rounds, each consisting of a setup phase followed by a steady phase. Once the sensors are deployed, every node sends its location information to the BS, which then assigns nodes to specific regions and calculates the distances between each node, the BS, and

the center of its region using the Euclidean method. The number of CHs per region depends on the number of nodes within that region. During the setup phase, the BS selects CHs based on a fuzzy logic system that considers three factors: residual energy (classified into very low, low, medium, high, and very high), distance to the BS, and distance to the center of the region (classified as low, medium, and high). The fuzzy inference system uses 45 IF-THEN rules to determine CH selection. After selection, the BS informs each node of its assigned region and the characteristics of the CHs within it. Each node then chooses the nearest CH based on received signal strength and joins the corresponding cluster. In the steady phase, CHs create a TDMA schedule and communicate it to their members. Non-CH nodes transmit their data to their assigned CH during their allocated time slot, while nodes not scheduled remain in sleep mode. The CHs aggregate the received data and forward it to the BS. Additionally, CHs report their residual energy to the BS at the end of each round to inform subsequent CH selection. Simulation results showed that FBCR enhances network lifetime by reducing CH over-selection. However, the protocol may face scalability challenges in networks with a large number of nodes, potentially leading to a significant workload for the BS.

2.3.7 Fuzzy-Based Energy Efficient Clustering Protocol for WSN (FEEC)

Kumar Rai and Daniel proposed **FEEC** [72], a clustering protocol that employs fuzzy logic for selecting super cluster heads (SCHs). The protocol operates in rounds, each consisting of two phases: CH and SCH selection, followed by data transmission. After node deployment, each node generates a random number between 0 and 1. If this number is below a calculated threshold, the node is chosen as a CH; otherwise, it joins the nearest CH. The threshold considers parameters such as residual energy, average distance, node density, and other criteria to prevent purely random CH selection. The probability of a node i being selected as a CH is given by:

$$P_i^d(r) = \frac{P_{opt} \times E_i(r)}{D_{av}(r) \times E_{av}(r)} \quad (2.5)$$

where $E_i(r)$ is the residual energy of node i , $D_{av}(r)$ is the average node density, $E_{av}(r)$ is the average network energy, and P_{opt} is the optimal CH probability. The threshold function controlling CH election is expressed as:

$$T_{th}^d = \frac{P_i^d(r)}{1 - P_i^d(r) \times \text{mod}\left(\frac{r_1}{P_i^d(r)}\right)} \quad (2.6)$$

Once CHs are selected, each CH runs a fuzzy inference system with four input parameters: residual energy, average node density (low, medium, high, very high), communication quality (poor, medium, good, very good), and average sink distance (reasonable, good, extensive). The system uses 64 rules and 7 output membership functions to determine the likelihood of a CH becoming a SCH. The CH with the highest likelihood and shortest distance to the BS is chosen as the SCH. During the data transmission phase, CHs gather data from their members and forward it to the SCH, which aggregates and sends the data to the sink. Simulations showed that FEEC improves network performance, with SCHs significantly extending network lifetime and enhancing packet delivery rate. However, the SCH bears a high computational and communication load, as it participates in CH election, SCH selection, and long-range data transmission, which accelerates its energy depletion.

2.3.8 Multi-objective fuzzy clustering algorithm for WSNs (MOFCA)

Sert et al proposed a clustering algorithm MOFCA [73], to prolong the lifetime of the WSN and to address the problem of hotspot and the hole energy, that uses a multi-objective fuzzy clustering algorithm to calculate the competition radius for nodes, and select CHs based on the residual energy. The process is divided into rounds, with each round consisting of two phases: CH election and data transmission. During the CH election phase, every node generates a random number between 0 and 1. If this number is below a predefined threshold (representing the desired tentative CHs), the node becomes a tentative CH. Then, it applies a fuzzy logic algorithm using three parameters: remaining energy, distance to the BS, and node density. Node density is determined by calculating the ratio of alive nodes within a certain radius to the total number of alive nodes. Each tentative CH broadcasts a message within its radius, containing its ID, residual energy, density, and competition radius. Within this radius, the node with higher energy becomes the CH, and in the case of a tie, the node with higher density is selected as the CH. Other nodes send join requests to the closest CH. The clusters formed are unequal in size, with clusters closer to the BS having a smaller radius and those farther away having a larger radius. During the data transmission phase, the CHs collect data from nodes, aggregate it, and send it to the BS either directly or through a multi-hop path. While the protocol has shown improvement in prolonging the network lifetime compared to other protocols, it still faces the challenge of having too many members in a cluster, particularly when the CH is close to the BS, resulting in a hotspot problem.

2.4 Clustering Using Machine Learning

Clustering is a fundamental unsupervised learning technique in machine learning, used to group data objects based on similarity without prior class labels. In WSNs, it plays a key role in organizing sensor nodes to optimize energy consumption, improve scalability, and extend network lifetime. Machine learning approaches enable adaptive and data driven clustering by analyzing multiple parameters such as node distance, residual energy, and density. Among these approaches, algorithms such as k-means[74] and DBSCAN have been widely adopted in WSN research for efficient cluster formation and CH selection, contributing to enhanced energy efficiency and overall network stability.

2.4.1 Optimal Packet Size K-means Clustering (OPSKC)

Razzaq et al. [75] proposed the Optimal Packet Size K-means Clustering (OPSKC) protocol to enhance the energy efficiency of WSNs. The algorithm combines the K-means clustering technique with an optimal fixed packet size model for CH selection. The protocol operates in three main phases: initialization, cluster formation, and CH election. During the initialization phase, the BS broadcasts an initialization request message to all sensor nodes. Each node replies with a message containing its residual energy and current location, which the BS uses for subsequent cluster formation. In the cluster formation phase, the BS determines the optimal number of clusters (K) using the following equation:

$$K = \sqrt{\frac{N}{2\pi}} \times \sqrt{\frac{\varepsilon_{fs}}{\varepsilon_{mp}} \times \frac{F}{x_{BS}^2}} \quad (2.7)$$

where N is the total number of nodes in the network, ε_{fs} and ε_{mp} are the energy amplification parameters of the free-space and multipath models, respectively, F is the network field dimension, and x_{BS} represents the distance from the BS to the sensing field. Using this value of K , the BS applies the K-means algorithm to partition the network, assigning each node to the nearest cluster center based on Euclidean distance. The process iterates until cluster assignments stabilize. In the CH election phase, two weight functions are computed for each node to determine the most suitable CH:

$$W_{Ni} = c_1 \times E_{ni} + c_2 \times D_{cci} \quad (2.8)$$

$$W_{std} = c_1 \times E_{total} + c_2 \times \text{Avg}(D_{cc}) \quad (2.9)$$

where W_{Ni} is the weight of node i within a cluster, E_{ni} is its residual energy, and D_{cci} is the distance between node i and its cluster center. W_{std} represents the standard weight corresponding to the ideal CH candidate, E_{total} is the total energy required to transmit and aggregate the

optimal packet size L_{opt} , and $Avg(D_{cc})$ is the average distance of all nodes to their cluster centers. Constants c_1 and c_2 are used to balance the influence of energy and distance factors. The node whose weight W_{Ni} is closest to W_{std} is selected as the CH for that round. After CHs are elected, they aggregate data from their cluster members and forward it directly to the BS (Figure 2.6). Simulation results showed that OPSKC enhances energy efficiency, increases throughput, and prolongs network lifetime compared to conventional K-means-based protocols. However, since all nodes transmit reply messages to the BS in each round, a slight increase in energy consumption is observed.

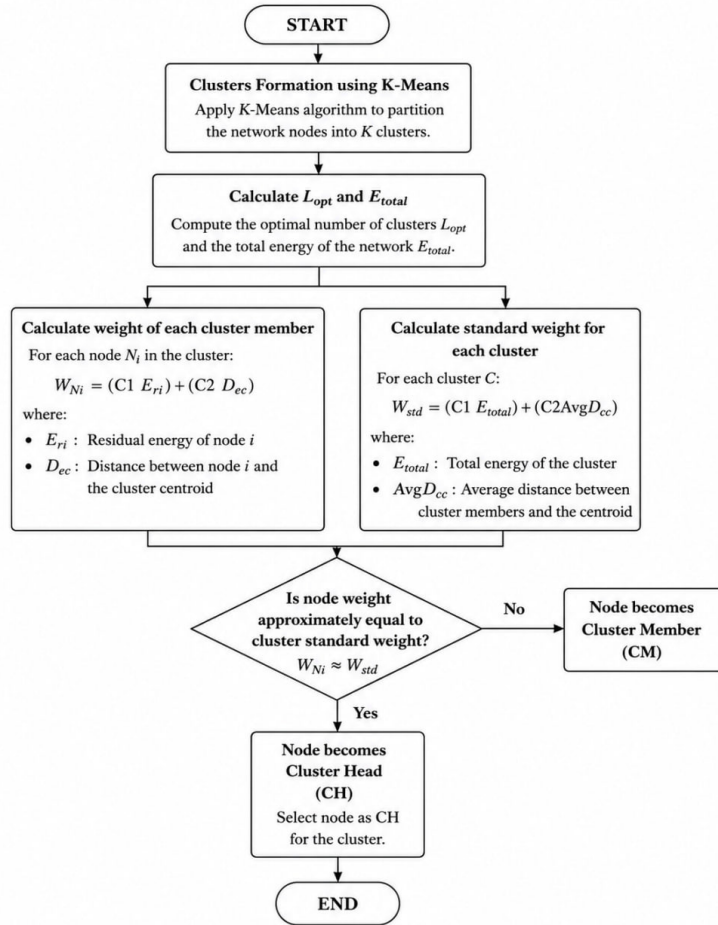


Figure 2.6: OPSKC Flow Chart [75].

2.4.2 Soft-k-Means Clustering Algorithm

Zhu et al. proposed a clustering protocol named soft K-means [76], which combines the Clustering by Fast Search and Find of Density Peaks (CFSFDP) algorithm and Kernel Density Estimation (KDE) for cluster center identification, along with soft K-means for cluster formation. Unlike traditional methods that assign each node to a single cluster, this approach maintains a list of candidate CHs. The process begins with the BS forming initial clusters, while

re-clustering is triggered only when necessary. Each sensor node first broadcasts a “HELLO” message containing its ID and location to collect neighborhood information, which is then sent to the BS via multi-hop routing. The BS selects optimal cluster centers based on node density and distance — nodes with higher density and greater distance are preferred. Here, density represents the number of active nodes within a given range, and distance denotes either the maximum distance to neighboring nodes for dense nodes or the minimum distance to the nearest dense node otherwise. Once cluster centers are determined, the BS applies soft K-means to form clusters. Each node is assigned a membership probability (ranging from 0 to 1) for belonging to each cluster, ensuring that the sum of these probabilities across clusters equals 1 and that every cluster contains at least one node with non-zero membership. Nodes initially join the CH with the highest membership probability, but edge nodes may be reassigned to a secondary CH to achieve balanced energy distribution. After clusters are established, the BS identifies CH candidates within each cluster based on residual energy and proximity to cluster centers, broadcasting this information to all nodes. During the steady phase, member nodes transmit data to their CHs, which then aggregate and forward it to the BS. If a CH’s energy falls below a predefined threshold, it issues a “SWITCH” message to activate the next CH candidate and inform its members. Once all CHs in a cluster are exhausted, the final active CH sends a “RESTART” message to the sink to initiate a new clustering round.

2.4.3 K-means and Dijkstra-based Clustering Protocol (KICMH)

Yang et al. [77] introduced a clustering protocol that integrates the K-means algorithm for cluster formation and Dijkstra’s algorithm for multi-hop data transmission. The number of clusters is calculated in the same manner as in OPSKC., considering factors such as the number of nodes, network size, and communication range. The BS initially gathers the coordinates of all nodes and randomly selects initial cluster centroids. Each node is associated with its nearest centroid, and new centroids are iteratively recalculated based on the mean position of the assigned nodes until convergence is achieved. CH election is divided into two stages: network initialization and CH update. During network initialization, the BS selects the CH of each cluster based on the node nearest to the cluster’s central point, which is computed using the distance formula applied by the BS. When the residual energy of a CH drops below a predefined threshold, the CH broadcasts a competitive CH message to all cluster members. Each cluster member then computes its CH competition factor according to its residual energy, distance to

the BS, and distance to the cluster center using the following equation:

$$CH_{\text{compete}} = \alpha E_{\text{residual}} + (1 - \alpha) \frac{d(k_i, M_k)}{d(k_i, BS)}, \quad \alpha \in [0, 1] \quad (2.10)$$

where α is a system-defined weight coefficient between 0 and 1, E_{residual} represents the residual energy of node i , $d(k_i, M_k)$ is the distance between the node and its cluster center, and $d(k_i, BS)$ is the distance from the node to the BS. The computed competition factors are transmitted to the CH, which selects the node with the highest value as the next CH and informs the BS. Upon receiving this information, the Dijkstra algorithm is employed by the BS to determine the shortest energy-efficient path between CHs and the BS. Instead of using geometric distance alone, the algorithm calculates energy consumption between each pair of CHs by combining the free-space and multipath attenuation models. These values are represented in a weighted undirected graph matrix, where each edge weight corresponds to the transmission energy between two CHs. Using this energy-based matrix, the BS runs the Dijkstra algorithm, taking itself as the root node, to identify the minimum-energy multi-hop path for data forwarding from CHs to the BS. During the data transmission phase, each node sends its sensed data to the CH, which aggregates and forwards it through the Dijkstra-defined route toward the BS. Simulation results demonstrate that this protocol effectively prolongs network lifetime and enhances routing efficiency. However, scalability issues may arise in large-scale deployments, and selecting nodes near the BS as CHs can lead to uneven energy consumption among clusters.

2.4.4 Improvement of DBSCAN

In the work titled "Improvement of WSN Lifetime via Intelligent Clustering Under Uncertainty," Sahoo et al. [78] proposed a clustering protocol designed to enhance the lifetime of WSNs. The approach begins by applying the DBSCAN algorithm to partition the network into clusters, with the optimal number of clusters validated using the Silhouette index and the elbow method. Nodes identified as outliers are treated as individual clusters that directly transmit their sensed data to the sink. For other clusters, the initial CH is selected based on four parameters: residual energy, distance to the BS, average intra-cluster distance, and reliability. The reliability of a node is defined as the ratio of its number of neighboring nodes within the cluster to the maximum number of nodes in the same cluster. Since these parameters contribute differently to CH selection, their relative importance is determined using the entropy-weighted method. The probability of each node becoming a CH is then computed using a weighted combination of residual energy, distance to the BS, average intra-cluster distance, and reliability. When two or more nodes within a cluster exhibit identical probabilities, the

Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is applied to rank the candidates, and the node with the highest ranking is chosen as the CH. In subsequent rounds, CH selection is guided by a counter mechanism that is incremented when a node satisfies favorable conditions such as possessing the highest residual energy, being closest to the sink, or having the largest average intra-cluster distance. During the communication phase, member nodes transmit their sensed data to their respective CHs, which aggregate and forward the data to the sink. Comparative evaluations demonstrated that the proposed DBSCAN-based clustering approach outperformed k-means clustering in terms of network lifetime, scalability, and resilience. However, the study did not fully specify the CH selection mechanism in later rounds, nor did it elaborate on the resolution process for ties between nodes with equal selection probabilities.

2.4.5 Distributed DBSCAN Protocol (DDBSCAN)

Kadhum Hameed and Idrees proposed a distributed clustering protocol for IoT networks [79] that optimizes energy consumption using the DBSCAN algorithm. The protocol operates in rounds comprising setup and steady-state phases (Figure 2.7). During the setup phase, each sensor node determines whether it can act as a core point by scanning its neighborhood to count how many nodes fall within its sensing radius (ε), which defines the density threshold for cluster formation. The parameters ε and MinPts (minimum number of nodes required to form a cluster) are user provided. If the number of neighbors of a node exceeds MinPts, the node becomes a core point and either joins an existing cluster or initiates a new one. After cluster formation, core nodes within each cluster exchange information such as residual energy, status, location, and number of members. CH election is then carried out in a distributed and periodic manner using a fitness function that evaluates the suitability of each core node based on its remaining energy, number of neighbors, and position relative to other core nodes in the cluster. The fitness function is expressed as

$$F_j = \frac{E_{remaining}}{E_{initial}} + \left(1 - \sum_{j \in N} \|S_j(x, y) - S_i(x, y)\|\right) + \frac{S_j(\text{Members})}{\text{Cluster}(\text{Members})} \quad (2.11)$$

where $E_{remaining}$ and $E_{initial}$ denote the residual and initial energies of node j , respectively, N represents the number of core nodes in the cluster, and $S_j(x, y)$ and $S_i(x, y)$ are the coordinates of nodes j and i . The node achieving the highest F_j value is selected as the CH for that round. In the steady-state phase, the CH allocates TDMA time slots for its members to transmit data efficiently and enter sleep mode when idle. The CH aggregates and forwards the collected data to the sink, thereby minimizing redundant transmissions and conserving energy. Simulation

results demonstrate that the proposed protocol enhances network lifetime and energy balance compared to LEACH and I-LEACH. However, the protocol depends heavily on core nodes, and once all core points fail, the network becomes inoperative even if some non-core nodes remain functional.

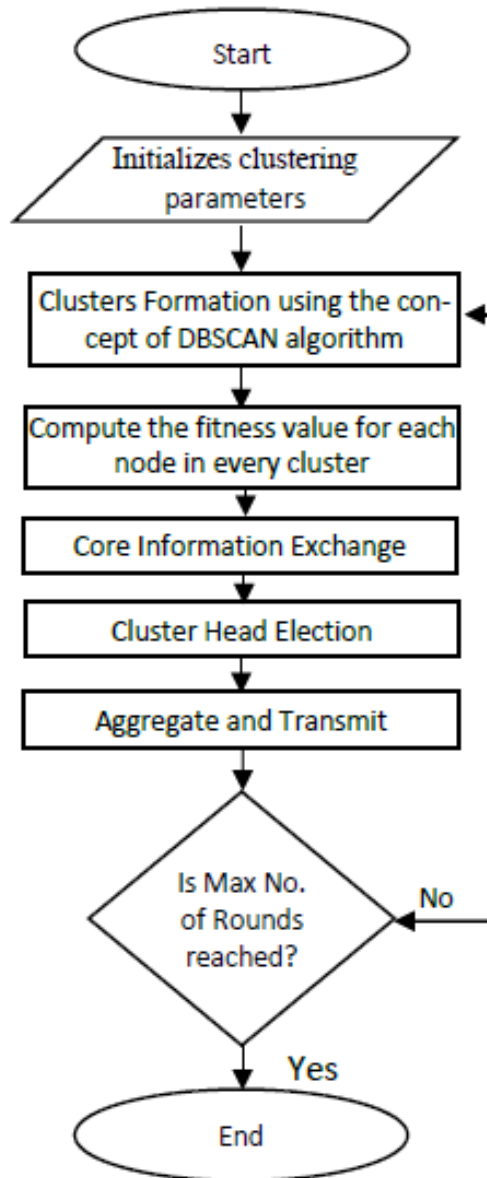


Figure 2.7: Distributed DBSCAN Protocol [79].

2.5 Clustering Using Metaheuristic Algorithms

Since their emergence in the early 1980s, metaheuristic algorithms have evolved significantly, offering powerful strategies to enhance computational efficiency, solve complex large-scale optimization problems, and generate robust solutions. They have achieved remarkable success in

addressing various combinatorial optimization tasks [80, 81], with well-known examples including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), and Grey Wolf Optimizer (GWO). In the context of WSNs, metaheuristic algorithms have been extensively applied to optimize clustering, particularly in the selection of CHs and the formation of balanced clusters. These algorithms are capable of exploring large solution spaces and escaping local optima, making them well suited for minimizing energy consumption and extending network lifetime. Each node or solution is typically represented as a candidate CH configuration, and the optimization process iteratively improves these configurations based on objective functions that consider parameters such as residual energy, node distance, and communication cost. By balancing exploration and exploitation, metaheuristic-based clustering approaches provide adaptive and energy-efficient solutions compared to traditional deterministic methods.

2.5.1 Atomic Energy Optimization-based WSN Clustering (AEOWSNC)

In AEOWSNC [82], a clustering protocol based on the Atomic Energy Optimization (AEO) algorithm [83] was proposed to extend the operational lifetime of WSNs by selecting optimal CHs. In this approach, each atom in the AEO model represents a candidate CH set, which is initialized randomly with a predefined number of CHs and assigned an energy level that reflects its effectiveness in clustering. Through iterative processes such as energy transfer and dissipation, the atoms evolve toward better solutions that reduce overall communication costs. The objective function evaluates each solution by considering the total distance from sensor nodes to their assigned CHs and from CHs to the BS. Solutions that achieve lower total distances are regarded as stronger, retaining their energy, while weaker solutions gradually lose energy and are replaced. This mechanism ensures a balance between exploration of new candidate solutions and exploitation of the best ones. The protocol operates in a centralized manner, where CHs aggregate data from their member nodes and transmit it directly to the BS. Simulation results demonstrated that AEOWSNC outperforms conventional clustering protocols in terms of both energy efficiency and network lifetime. Nevertheless, a notable limitation is that the objective function only accounts for distance and neglects the residual energy of CHs. As a result, some CHs remain active until their energy is fully depleted, which causes uneven energy consumption among nodes and may reduce network coverage over time. This observation emphasizes the importance of incorporating energy aware optimization criteria in future protocol enhancements to achieve more balanced energy utilization and longer network sustainability.

2.5.2 Spotted Hyena Optimization-based Clustering (SHO-CH)

The SHO-CH protocol [84], proposed by Sharma et al., is an energy efficient, cluster-based routing scheme designed for heterogeneous WSNs. Its primary goal is to extend network lifetime while maintaining balanced energy consumption among sensor nodes. The protocol is inspired by the cooperative hunting behavior of spotted hyenas, emulating their strategies of encircling, attacking, and tracking prey to achieve a balance between exploration and exploitation during the CH selection process. CH selection is guided by a fitness function that combines several key parameters, including residual energy, the distance between nodes and their respective CHs, and the distance from CHs to the BS. Through iterative updates, the algorithm identifies CHs that optimize the trade off between energy efficiency and spatial distribution. Once CHs are established, they aggregate data from their member nodes and transmit it either directly to the BS or via intermediate CHs located closer to the BS, enabling efficient multi-hop communication. Simulation results demonstrate that SHO-CH significantly enhances network lifetime and achieves more balanced energy utilization compared to conventional clustering approaches. These results highlight the effectiveness of nature-inspired metaheuristic strategies in optimizing WSN operations, particularly in heterogeneous network environments.

2.5.3 African Vulture Optimization Algorithm (AVOACS)

AVOACS [85], proposed by Kumar et al., leverages the scavenging and foraging behaviors of vultures to optimize CH selection in WSNs. In this approach, each vulture represents a candidate CH configuration, which is evaluated using a fitness function that incorporates residual energy, distance to the sink, intra-cluster distance, and a communication mode decider (CMD). The four objective functions corresponding to these parameters are defined as:

$$Obj_1 = \frac{1}{N} \sum_{i=1}^N E_i \quad (2.12)$$

$$Obj_2 = \sum_{i=1}^N \frac{D(N_i, s)}{D_{avg}(N, s)} \quad (2.13)$$

$$Obj_3 = \sum_{i=1}^N \left[\frac{D(C_i, GH)}{D(N_i, s)} + D(N_i, s) \right] \quad (2.14)$$

$$Obj_4 = \sum_{i=1}^N \frac{D(N_i, s)}{D_{avg}(N, s)} \times \frac{1}{0.1M} \quad (2.15)$$

The overall fitness function is then calculated as:

$$F = \frac{1}{\alpha Obj_1 + \beta Obj_2 + \gamma Obj_3 + \delta Obj_4} \quad (2.16)$$

N Total number of nodes

E_i Residual energy of node i

$D(N_i, s)$ Distance between node i and sink s

$D_{avg}(N, s)$ Average distance from nodes to sink

After evaluation, the two fittest vul-

C_i Communication node of i

GH Gateway or CH

M Clustering parameter

$\alpha, \beta, \gamma, \delta$ Weight coefficients ($\alpha + \beta + \gamma + \delta = 1$)

tures act as leaders, guiding the others, which update their positions relative to these leaders. A dynamic hunger rate controls the balance between exploration and exploitation: initially favoring wide exploration and later promoting intensive exploitation. Two exploitation strategies are applied: refining searches via siege-fighting and spiral flight, or intensifying them by averaging around leaders and performing aggressive jumps. This adaptive mechanism ensures a smooth transition from global search to local refinement, reducing the risk of premature convergence. Simulation results show that AVOACS achieves more balanced energy distribution, improved network stability, and longer network lifetime compared to conventional clustering protocols.

2.5.4 DEAI-PSO-Based Clustering Algorithm for IoT-Oriented WSNs

Haris and Nam proposed a PSO framework enhanced with a Double-Exponential Adaptive Inertia (DEAI) mechanism [86] to optimize CH selection in IoT based WSNs. Unlike conventional PSO-based clustering methods that use fixed or linearly varying inertia weights, the DEAI-PSO model dynamically regulates the balance between exploration and exploitation through a double-exponential adaptive scheme. This mechanism improves convergence stability toward energy-efficient configurations and mitigates premature stagnation common in standard PSO variants. The main objective of this method is to reduce energy depletion, particularly for nodes located near the BS, which are often burdened with heavy relay loads in multi-hop communication. The network model considers randomly deployed IoT devices with a centrally

located BS having an unlimited power supply. Each node can act as a Cluster Member (CM), CH, or free node, depending on the optimization outcome. In the PSO framework, each particle represents a potential CH configuration. The inertia term (ω_k) maintains previous momentum, the cognitive component (c_p) drives the particle toward its personal best position, and the social component (c_g) encourages movement toward the globally best solution. These components interact dynamically throughout the optimization process to achieve an optimal balance between exploration and exploitation. The DEAI mechanism introduces a dual dependency on temporal and spatial factors, defined as

$$\omega_k = \omega_{\min} + (\omega_{\max} - \omega_{\min}) \exp\left(-\alpha \frac{k}{T}\right) \exp(-\beta d(X_i^k, Pbest_i)) \quad (2.17)$$

where k is the current iteration, T the total number of iterations, and $d(X_i^k, Pbest_i)$ the Euclidean distance between a particle's current position and its personal best. The parameters α and β control temporal and spatial adaptation, respectively. Early in the optimization, a small ω_k promotes wide exploration for diverse CH configurations. As iterations progress, ω_k increases gradually, emphasizing local refinement near promising solutions. This double-exponential design ensures a smooth transition between exploration and exploitation phases, maintaining swarm diversity and preventing abrupt behavioral changes. The DEAI-PSO algorithm applies a multi-objective fitness function that evaluates each CH configuration based on minimizing energy consumption and balancing communication distances between nodes and CHs. The number of clusters is determined adaptively, taking into account both free-space and multipath fading energy models. Consequently, nodes with higher residual energy and optimal spatial positioning are chosen as CHs, leading to better energy distribution and an extended network lifetime. An additional feature of this approach is the use of a distance-based free-node strategy. Nodes located within a given threshold distance from the BS are considered free nodes and send their data directly to the BS, avoiding redundant cluster formation nearby. This strategy reduces transmission overhead and contributes to overall energy efficiency. The resulting communication model operates in three tiers: free nodes communicating directly with the BS, cluster members sending data to CHs, and CHs forwarding aggregated data to the BS using single-hop or multi-hop links. Compared with traditional PSO or single-exponential models, the DEAI-PSO approach provides a two-dimensional adaptive control that integrates both temporal progression and spatial awareness, preventing premature convergence and maintaining solution diversity. Moreover, by embedding the free-node classification directly within the optimization process, it supports dynamic, context-aware communication. However, this

method introduces a moderate computational overhead due to the need to calculate spatial dependencies and double-exponential inertia in each iteration. Its performance is also sensitive to the selection of control parameters α and β ; poor tuning may cause unbalanced exploration and exploitation or slower convergence. Despite this inconvenience, the DEAI-PSO model remains a strong and scalable approach that significantly improves energy efficiency and network lifetime in IoT-oriented WSNs.

2.6 Hybridization Protocols

In recent years, researchers have increasingly turned to hybrid clustering approaches in WSNs to overcome the limitations of single-technique protocols. Instead of relying solely on one method, hybrid models combine the strengths of machine learning, metaheuristic optimization, fuzzy logic, and LEACH-based mechanisms to achieve better energy efficiency and network longevity [87, 88, 89]. Machine learning contributes predictive and adaptive decision-making capabilities, enabling the system to respond intelligently to dynamic changes in network conditions. Metaheuristic algorithms such as PSO, GA, and ACO provide efficient global search strategies for selecting optimal CHs. Fuzzy logic, in turn, refines the decision process by handling uncertainty in parameters such as residual energy, distance, and node density. LEACH serves as the foundational protocol upon which these hybrid methods are built, ensuring simple cluster formation and communication management. By integrating these diverse paradigms, hybrid protocols achieve a more balanced trade-off between exploration and exploitation, enhance the adaptability of CH selection, and significantly reduce communication and energy overhead compared to conventional clustering schemes.

2.6.1 An Optimized Fuzzy Clustering Algorithm (OFCA)

OFCA[90] is an energy-efficient clustering and routing protocol for WSNs that integrates Fuzzy Logic for cluster formation and PSO for multi-hop routing. The main objective of OFCA is to prolong network lifetime by balancing energy dissipation across the network, particularly addressing the “hot spot problem” that occurs near the BS. The operation of OFCA is divided into rounds, each consisting of two main phases: the Setup Phase (clustering) and the Steady State Phase (routing). The setup phase is executed centrally at the BS and involves the selection of CHs and the determination of cluster sizes to ensure balanced energy consumption. Fuzzy logic is applied to handle the uncertainty inherent in CH election. The Type-1 Sugeno Fuzzy Inference System (FIS) computes a Confidence Factor (CF) for each node, representing

its suitability to become a CH. The FIS uses three dynamically varying input parameters to compute the CF: (1) the residual energy of the node, where higher energy increases the likelihood of becoming a CH; (2) the distance from the BS, where nodes closer to the sink have a higher chance of selection; and (3) the concentration, defined as the number of nodes within the communication range of the candidate node. Based on these inputs, each with three membership levels (low, medium, and high), the FIS applies 27 fuzzy rules to calculate the CF. The node with the highest CF value is chosen as the CH. Although CHs near the BS may be initially preferred, the algorithm ensures that in subsequent rounds, CHs located farther away are selected when they retain higher residual energy, thus balancing energy consumption and mitigating the energy-hole problem near the BS. After CH selection, unequal clustering is performed to distribute energy consumption more effectively. The communication radius of each CH is determined according to its distance from the BS. CHs closer to the BS are assigned smaller communication radii than those located farther away, ensuring unequal and balanced clustering. This design reduces the workload on CHs near the BS, which already handle more inter-cluster relay traffic. Once the CHs and their radii are established, the BS broadcasts a CH message. Nodes within the communication range respond with a JOIN-REQ message to join a cluster, becoming Cluster Members. During the steady state phase, data is collected, aggregated, and transmitted to the BS through an optimized multi-hop route determined using PSO. CHs send a DATA-REQ message to their member nodes, which reply with sensed data according to a TDMA (Time Division Multiple Access) schedule defined by their respective CHs. Each CH then performs local data aggregation to minimize redundancy and reduce transmission overhead. For inter-cluster communication, the aggregated data is forwarded from CHs to the BS using a multi-hop path via other CHs. This routing process is formulated as an optimization problem solved by PSO. PSO is chosen for its simplicity and fast convergence toward near-optimal solutions. Its objective is to determine the optimal set of relay CHs that maximize the total residual energy and node concentration while minimizing the average distance to the BS. The optimization problem is expressed as a linear programming model with three objectives combined into a single fitness function F , which must be minimized:

$$F = w_1 \frac{1}{f_1} + w_2 f_2 + w_3 \frac{1}{f_3} \quad (2.18)$$

where f_1 represents the inverse of total residual energy (to maximize it), f_2 is the average distance between selected CHs and the BS (to minimize it), and f_3 denotes the inverse of total node concentration (to maximize connectivity). The weighting coefficients w_1 , w_2 , and w_3 control the influence of each factor. For agricultural applications, the optimal weights are

$w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$, giving higher priority to energy conservation. The optimal multi-hop path obtained from the PSO process is then used to forward the aggregated packets from the CHs to the BS. Inter-cluster packets are transmitted without further aggregation to ensure data consistency and timely delivery. After one round of operation, which may include several TDMA schedules, CHs are re-elected and the clustering process restarts. This continuous re-clustering and optimization process ensures a balanced energy distribution among all nodes, thereby extending the overall network lifetime. However, a limitation of OFCA lies in its computational overhead. The combination of fuzzy logic and PSO introduces additional processing complexity, which may not be ideal for networks with resource-constrained nodes or in scenarios requiring real-time decisions.

2.6.2 Fuzzy Based Energy Efficient Cluster Head Selection (FEECS)

FEECS [91] algorithm is a hybrid protocol designed to extend the lifetime of WSNs by enhancing energy efficiency during both cluster formation and CH selection. It integrates the k-means clustering algorithm with a fuzzy logic system (FLS) to achieve balanced energy consumption and improve network stability. FEECS begins by organizing sensor nodes into clusters using the k-means algorithm, which efficiently groups nodes based on proximity. Each sensor node calculates its distance to a set of randomly initialized cluster centers using Euclidean distance and joins the cluster with the nearest center. The cluster centers are iteratively updated until stable clusters are formed. This process minimizes intra-cluster distances and reduces communication overhead within clusters. Once clusters are established, the fuzzy logic system is applied to select the most suitable node in each cluster to act as the CH. The selection process is based on four input parameters: Residual Energy (RE), Distance to BS (D2BS), Node Density (Den), and Cluster Status (CS). These parameters are converted into linguistic values (such as low, medium, high, near, far, yes, and no) and processed through fuzzy inference rules to calculate each node's chance of becoming a CH. Nodes with higher residual energy, closer proximity to the BS, higher local density, and that have not previously served as CHs are given a higher selection chance. The node with the maximum fuzzy output value in each cluster is elected as the CH. After CH selection, intra-cluster communication is managed efficiently using a TDMA scheduling scheme, where each node transmits data during its assigned time slot, reducing collisions and unnecessary retransmissions. The CH aggregates the collected data and transmits it to the BS either directly or via multi-hop communication, depending on network distance. By combining the structural clustering capability of k-means with the adaptive decision-making power of fuzzy logic, FEECS effectively balances energy consumption,

reduces communication cost, and significantly prolongs the network lifetime.

2.6.3 Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization

The system described is the Fuzzy Logic (FL) LEACH Technique-Based PSO [92] protocol proposed by M. Gamal et al. in their work “Enhancing the Lifetime of WSNs Using Fuzzy Logic LEACH Technique-Based Particle Swarm Optimization” [19]. This hybrid protocol aims to extend the lifetime and improve the throughput of WSNs by integrating K-means clustering, PSO optimization, and fuzzy logic-based CH selection. The method operates in two interconnected phases (Figure 2.8): cluster formation and steady-state communication. In the setup phase, the BS is responsible for forming clusters at the beginning of the network operation, assuming static sensor node positions. The optimal number of clusters is determined using the gap statistic evaluation method, which identifies the configuration that minimizes intra-cluster variation. Once the number of clusters is set, K-means clustering is applied to partition the sensor nodes using Euclidean distance as the similarity measure. The output of K-means, representing the initial cluster centers, is treated as one of the particles in the PSO population, while the remaining particles are initialized randomly. PSO then refines the cluster formation by iteratively updating the cluster centers based on a following fitness function designed to minimize quantization error and balance the energy distribution across nodes. T

$$J = \frac{\sum_{j=1}^{N_C} \left(\sum_{s_p \in C_{ij}} \frac{d(s_p, c_j)}{|C_{ij}|} \right)}{N_C} \quad (2.19)$$

the particle achieving the best fitness defines the final cluster configuration and clustering is done only once at the beginning of the network. **Where:**

- J : Objective function value (average intra-cluster distance).
- N_C : Total number of clusters in the network.
- C_{ij} : The j -th cluster (group of sensor nodes).
- $|C_{ij}|$: Number of sensor nodes (cardinality) in cluster C_{ij} .
- s_p : Any sensor node that belongs to cluster C_{ij} .
- c_j : CH or centroid of cluster j .
- $d(s_p, c_j)$: Distance between sensor node s_p and its CH c_j .

After cluster formation, the network operates in rounds consisting of CH selection and data transmission. During each round, two types of CHs are selected: the Primary CH (PCH) and the Secondary Cluster Head (SCH). The PCH is responsible for transmitting aggregated data to the BS, while the SCH collects and aggregates data from cluster members before forwarding

it to the PCH. Both selections are performed using fuzzy logic inference systems executed at the BS. For PCH selection, the fuzzy system uses three input parameters: residual energy, distance to the cluster center, and distance to the BS. The system applies twenty-seven fuzzy rules defined using the Mamdani model and defuzzifies the result using the centroid of area method to determine each node's chance of becoming a PCH. The node with the highest chance is elected. The SCH selection uses two fuzzy inputs—residual energy and distance to the PCH—and applies eight rules to determine the node most suitable for this role. This dual-level CH selection effectively balances energy consumption between nodes near the cluster center and those handling data aggregation or transmission. In the steady-state phase, data transmission proceeds in three stages: sensor nodes send their data to the SCH according to a TDMA schedule, the SCH aggregates the data and forwards it to the PCH, and finally, the PCH transmits the aggregated data to the BS through single-hop communication. This layered communication reduces the load on individual CHs and prevents premature energy depletion in nodes close to the BS. Reported findings indicate that the protocol enhances network lifetime and energy efficiency compared to conventional clustering methods. However, the authors did not clearly articulate the specific contribution of integrating K-means with PSO during cluster formation, nor did they clarify whether the PSO particles were derived from K-means-generated cluster centers or initialized randomly. Despite this ambiguity, the combination of clustering techniques, optimization, and fuzzy decision-making offers a promising direction for improving energy balance and prolonging the operational stability of WSNs.

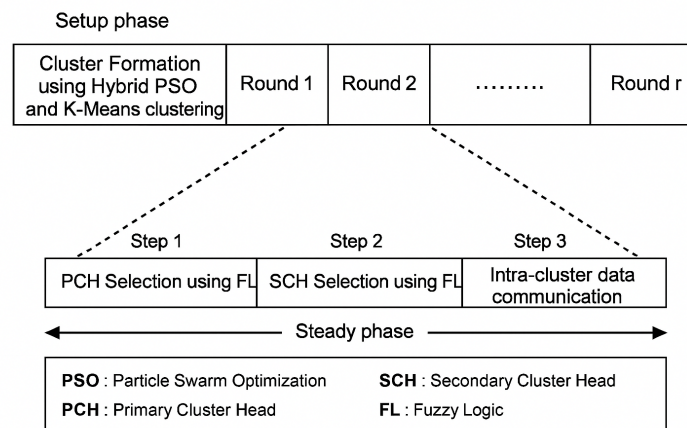


Figure 2.8: Phases of the protocol [92].

2.6.4 Particle Swarm Optimization Based K-means

In [93], Sun et al. proposed a centralized clustering protocol for WSNs that aims to balance energy consumption and select optimal CHs to extend network lifetime. The approach combines PSO with K-means clustering, where the number of clusters k is fixed and determined during network initialization. After node deployment, each sensor sends its ID and location to the BS, which then applies PSO to optimize the positions of cluster centers. In this method, PSO initializes a population of particles, each representing a candidate set of k cluster centers. Particles are iteratively updated based on a fitness function that measures the total distance between all nodes and their corresponding cluster centers. Each particle tracks both its personal best and the global best, with velocity and position updates calculated using Euclidean distance. Once PSO converges, the global best particle is used as input to the K-means algorithm, which refines the cluster centers and assigns nodes to their nearest cluster. CH election occurs in every round. In the first round, the node closest to each cluster center is selected as CH. In subsequent rounds, CH selection is guided by a fitness function that accounts for both residual energy and distance from the cluster center:

$$f(i) = a(i)f_1(i) + b(i)f_2(i), \quad (2.20)$$

$$f_1(i) = \frac{E_{\text{res}}(i)}{\frac{1}{n} \sum_{j=1}^n E_{\text{res}}(j)}, \quad (2.21)$$

$$f_2(i) = \frac{d_{\text{max}} - d(x_i, \text{cen}_j)}{d_{\text{max}} - d_{\text{min}}}, \quad (2.22)$$

where $E_{\text{res}}(i)$ is the residual energy of node i , $d(x_i, \text{cen}_j)$ is the distance between node i and its cluster center, and $a(i)$, $b(i)$ are adaptive weights that depend on the node's residual energy. After CH selection, member nodes transmit data to their CHs according to a TDMA schedule and enter sleep mode. CHs aggregate the collected data and forward it directly to the sink. However, this study raises a question regarding the necessity of applying K-means after PSO, as both algorithms perform similar clustering tasks, potentially introducing redundant computation.

2.6.5 hybrid clustering approach integrates the Harmony Search with K-means

The hybrid clustering approach integrates the Harmony Search (HS) meta-heuristic method with the conventional K-means algorithm[94]. This combination is designed to overcome the

fundamental limitation of the K-means algorithm, which often converges to a locally optimal solution due to the random selection of initial centroids. By using Harmony Search to generate optimized initial CH positions, the algorithm enhances the accuracy and efficiency of K-means, leading to improved energy conservation in WSNs. In the first phase, Harmony Search is applied to determine the approximate positions of CHs. Harmony Search is a population-based optimization algorithm inspired by musical improvisation, where musicians adjust pitches to achieve better harmony. In this context, each harmony vector in the Harmony Memory (HM) represents a potential set of CH locations. The Harmony Memory is initialized with randomly generated solutions, and algorithm parameters such as the Harmony Memory Consideration Rate (HMCR) and the Pitch Adjustment Rate (PAR) are defined. During improvisation, a new harmony vector is generated either by selecting values from existing solutions in the Harmony Memory with probability HMCR, or by generating random values with probability $(1 - \text{HMCR})$. If a value is chosen from the Harmony Memory, it may be further modified with probability PAR. The newly improvised solution is then evaluated using the objective function, and if it has a lower energy cost than the worst solution in the Harmony Memory, it replaces it. This process is repeated for a predefined number of iterations. The final output of the Harmony Search phase is a set of optimized CH positions, which are then used as the initial centroids for K-means. The fitness or objective function minimized by the Harmony Search algorithm represents the total communication energy in the network and is expressed as:

$$f = \sum_{i=1}^k \sum_{j=1}^{n_i} [E_{elec} \cdot l + \epsilon_{fs} \cdot l \cdot d_{ij}^2] + \sum_{i=1}^k [E_{elec} \cdot l + \epsilon_{mp} \cdot l \cdot d_{iBS}^4] \quad (2.23)$$

where k is the number of clusters, n_i is the number of member nodes in cluster i , E_{elec} is the energy required for electronic circuitry, l is the size of the transmitted data in bits, ϵ_{fs} and ϵ_{mp} denote the amplifier energy for free-space and multipath models, d_{ij} is the distance between node j and its CH, and d_{iBS} is the distance between CH i and the BS. This function accounts for both intra-cluster energy consumption and the energy used by CHs to transmit aggregated data to the BS. In the second phase, the K-means algorithm refines the CH positions obtained from Harmony Search. Sensor nodes are assigned to the nearest CH based on Euclidean distance. New centroids are then recalculated as the mean position of all nodes within each cluster. This process is repeated until the centroids no longer change, indicating convergence. By combining the global search capability of Harmony Search with the fast convergence of K-means, the hybrid approach achieves energy-efficient clustering and prolongs the network lifetime.

2.6.6 Cluster Based Energy Efficient Routing Protocol using SA-LEACH to WSNs

The SA-LEACH protocol presented in [95], enhances the conventional LEACH algorithm by introducing SA to refine the CH selection process. LEACH initially selects CHs randomly, often resulting in nodes with low residual energy being chosen, which accelerates energy depletion and degrades overall network performance. SA-LEACH first forms clusters using the LEACH mechanism, then applies SA to explore neighboring nodes and identify a better CH candidate by minimizing the total energy consumed during each transmission cycle. The optimization is based on the first-order radio model, and the remaining network energy after each cycle is evaluated using the objective function:

$$\begin{aligned}
 f(x_i) = f(x_{i-1}) &- \sum_{i=1}^{CH_n} E_{TX}(S_i \rightarrow CH_n) + \sum_{i=1}^{CH_n} E_{CH_i} \\
 &+ E_{TX}(CH_n \rightarrow SN) + \sum_{i=1}^c E_{TX}(S_i \rightarrow SN).
 \end{aligned} \tag{2.24}$$

which accounts for the transmission energy of cluster members, the reception energy of CHs, and the energy of isolated nodes communicating directly with the sink. By probabilistically accepting better or slightly worse solutions, the SA process avoids local minima and converges toward a CH configuration that yields lower overall energy consumption.

Table 2.1: Summary of clustering protocols

Protocol	CH Selection	Routing	Parameters	Notes (Advantages / Limitations)
LEACH	Probabilistic; threshold-based	Single-hop	R, P	Simple and distributed; random CHs; weak in large networks.
LEACH-C	BS computes CHs via energy/positions	Single-hop	Energy, coordinates	Balanced CH count; Stable then LEACH; Centralized cost.
MR LEACH	As LEACH	Multi-hop CH relays	Hop levels, CH distance	Good for large fields; bottleneck near BS.
TL-LEACH	Primary + secondary CHs	Two-level routing	Signal strength, TDMA/CDMA	Reduced distance; more overhead.
W-LEACH	Weighted (energy+density)	Single-hop	Energy, density	Balanced CHs; centralized work.
DBCH	Threshold uses energy + BS distance	Single-hop	Energy, $D_{i \rightarrow BS}$	Better balance; still limited by single-hop.
LEACH-VH	CH + vice-CH backup	Single-hop by backup	Residual energy	More reliable; extra VCH cost.
CHEF	Fuzzy: energy + local distance	Single-hop	Energy, distance, fuzzy rules	Uniform clustering; high computation.
EAUCF	Fuzzy competition radius	Single-hop	Energy, BS distance	Reduces hotspot; extra messages.
DUCF	Fuzzy (energy, degree, BS dist.)	Single-/multi-hop	Energy, degree	Good balancing; orphan CHs possible.
EEDCF	Distributed fuzzy	Single-hop	Energy, degree, neighbor energy	Efficient clusters; ignores BS distance.

Continued on next page

Protocol	CH Selection	Routing	Parameters	Notes
EETPF	Fuzzy CH + fuzzy routers	Multi-hop	Energy, BS dist., neighbors	Reduced cost; CH load increases.
FBCR	BS fuzzy CH selection	Single-hop	Energy, BS/regional distance	CH stability; BS workload high.
FBECS	Decentralized fuzzy logic	Single-hop	Energy, density, BS dist.	Improved lifetime; more control traffic.
FEEC	Threshold + fuzzy super-CH	Two-level routing	Energy, density, link quality	Strong performance; SCH overload.
MOFCA	Fuzzy multi-objective radius	Single-/multi-hop	Energy, density, BS dist.	Longer lifetime; cluster overflow near BS.
Soft K-means	Density peaks + KDE + soft membership	Single-hop	Density, membership	Balanced clusters; heavy BS computation.
OPSKC	K-means + weighted CH score	Single-hop	Energy, center distance	Good throughput; reply overhead.
KICMH	K-means + energy-based CH update	Multi-hop (Dijkstra)	Energy, $d(\cdot)$	Energy-aware routing; scalability limits.
DDBSCAN	DBSCAN + core-node fitness	Single-hop	Energy, core density	Adaptive; fails if core nodes die.
Improved DBSCAN	Entropy-weighted + TOPSIS	Single-hop	Energy, BS distance, reliability	Better CH selection; later-rounds unclear.
AEOWSNC	AEO global optimization	Single-hop	Node-CH, CH-BS distances	Efficient; ignores residual energy.
SHO-CH	Hyena-based optimization	Multi-hop	Energy, distances	Robust; tuning required.

Continued on next page

Protocol	CH Selection	Routing	Parameters	Notes
AVOACS	Vulture multi-objective fitness	Single-/multi-hop	Energy, distance, CMD	Energy balanced; heavy computation.
DEAI-PSO	Adaptive inertia PSO	3-tier routing	Energy, distance, inertia parameters	No early convergence; parameter sensitive.
OFCA	Fuzzy CH + PSO routing	Multi-hop	Energy, BS dist., density	Mitigates hotspot; computational cost.
FEECS	K-means + fuzzy logic	Single-/multi-hop	Energy, density, BS dist.	Balanced clusters; more overhead.
FL-LEACH-PSO	K-means + PSO + fuzzy PCH/SCH	Two-level routing	Energy, distances, fuzzy rules	High lifetime; complex.
PSO-K-means	PSO centers + K-means refinement	Single-hop	Energy, centroid distance	Stable clusters; redundant steps.
HS + K-means	Harmony Search initialization	Single-hop	Energy (E_{elec} , ϵ_{fs})	Better initial centers; HS tuning needed.
SA-LEACH	Random + SA	Single-hop routing	Energy	Better energy balance

2.7 Conclusion

This chapter presented a comprehensive survey of the clustering techniques and routing strategies that served as the foundation for this study. The reviewed works covered a wide range of approaches, including classical LEACH-based protocols, fuzzy-logic methods, machine learning techniques such as K-means and DBSCAN, as well as metaheuristic and hybrid optimization models. Each category introduced different mechanisms for CH selection, cluster formation, and data forwarding, showing how energy efficiency, stability, and scalability are addressed using various design principles.

The analysis highlighted how different protocols attempt to balance residual energy, communication distance, node density, and routing structure. Some methods favored simplicity and distributed operation, while others relied on centralized decision making or optimization driven

selection processes. Machine learning and metaheuristic algorithms brought adaptiveness and stronger optimization capabilities, whereas fuzzy logic improved decision quality in uncertain environments. Hybrid approaches integrated multiple techniques to enhance clustering accuracy and data transmission efficiency, especially in large-scale or heterogeneous deployments.

The insights obtained from this review guided the motivations and design choices of the proposed methodology in the next chapter. By examining the advantages and limitations of existing solutions, we identified key factors; such as balanced cluster formation, energy aware CH selection, and efficient multi hop routing that are essential for improving overall network performance.

The following chapter builds upon this foundation and presents the proposed protocol, developed to address the gaps observed in the literature and achieve more sustained and energy-efficient operation in WSNs.

Chapter 3

Puma Optimizer

3.1 Introduction

The Puma Optimizer (PO) [96] is a recently developed metaheuristic algorithm proposed by Abdollahzadeh et al. (2024). It is inspired by the intelligent hunting strategies and adaptive behavior of pumas, also known as cougars or mountain lions. PO models the natural balance between two critical phases of optimization: exploration (searching new areas) and exploitation (refining promising areas). Unlike traditional optimizers with fixed transition rules, PO introduces an intelligent phase-changing mechanism that allows it to automatically switch between exploration and exploitation based on past performance and learning experience. This mechanism enables the algorithm to escape local minima, preserve diversity, and converge effectively toward the global optimum. Each individual in the population represents a puma that searches within a multidimensional space for prey (optimal solution). The best performing individual at each iteration is called the male puma and serves as the leader, while the rest are female pumas that explore or exploit according to the phase control mechanism.

In the following chapter, we will explore the detailed functioning of the PO, examining how its biologically inspired strategies translate into a powerful optimization algorithm for solving complex problems. This chapter explains the algorithm's key components, including population structure, phase-changing mechanism, and the balance between exploration and exploitation, highlighting how the adaptive behavior of pumas is modeled to achieve efficient and robust convergence.

3.2 Biological Inspiration

In nature, pumas are solitary, intelligent predators capable of both long-distance roaming and short, powerful attacks. They rely on memory to revisit successful hunting grounds while also exploring new regions when prey availability changes. These behavioral traits inspired the PO's design, where:

- Exploration simulates wide territorial roaming to discover new prey regions (global search).
- Exploitation simulates ambush and sprint hunting to capture prey in promising zones (local refinement).
- Memory and experience correspond to adaptive phase switching between exploration and exploitation.

This dynamic and self-learning behavior is the foundation of the PO algorithm.

3.3 Mathematical Model and Search Phases

3.3.1 Initialization

At the start, the algorithm generates a population of N_{pop} pumas (solutions) randomly distributed within the search bounds:

$$X_i = L_b + rand(0, 1) \times (U_b - L_b) \quad (3.1)$$

where L_b and U_b are the lower and upper limits of the search space. Each puma's fitness value is computed using the problem's objective function (for example, energy-based fitness in WSNs).

The best puma, with the lowest cost or highest fitness, is assigned as the male puma.

3.3.2 Unexperienced Phase

At the beginning of the optimization process, pumas are considered unexperienced because they have no prior knowledge about the search space. During the first few iterations (typically three), the algorithm performs both exploration and exploitation simultaneously to gather experience. Two scoring functions are defined for each phase based on the cost improvements obtained at each iteration. These scores quantify how effective each phase has been in improving fitness. During these early iterations, PO alternates between both phases to collect comparative data on their effectiveness. Mathematically, the unexperienced phase calculates phase performance indicators (for example, $f_1^{explore}$, $f_1^{exploit}$, $f_2^{explore}$, $f_2^{exploit}$) using the improvement in fitness values between consecutive iterations. After the third iteration, the phase with the higher accumulated score is favored in subsequent iterations, marking the transition to the experienced phase. This learning strategy ensures that PO begins without bias and gradually learns which phase contributes more effectively to progress in the current problem landscape.

3.3.3 Experienced Phase

After the initial learning period, pumas gain experience, and the algorithm intelligently decides at each iteration whether to explore or exploit using a phase change mechanism. This mechanism employs three control functions to evaluate and score the two phases:

1. Improvement function (intensification) measures the magnitude of fitness improvement between successive iterations.
2. Consistency function (resonance) evaluates performance stability over several past iterations.
3. Diversity function rewards phases that have been inactive for several iterations to maintain exploration diversity.

At every iteration, these three functions are combined to compute an overall exploration score and exploitation score. If $Score_{exploit} > Score_{explore}$, the algorithm enters the exploitation phase; otherwise, it continues exploring. The scores are dynamically updated, ensuring the algorithm remains adaptive throughout the optimization process. This mechanism represents the intelligence of the puma, enabling it to balance memory-based revisiting (exploitation) and curiosity driven searching (exploration). The PO adaptively updates the exploration and exploitation scores according to the quality of the generated solutions and the improvement rate achieved during the search process.

The exploration and exploitation scores are computed as:

$$Score_{Explore} = (PF1 \times f1_{Explore}) + (PF2 \times f2_{Explore}) \quad (3.2)$$

$$Score_{Exploit} = (PF1 \times f1_{Exploit}) + (PF2 \times f2_{Exploit}) \quad (3.3)$$

The cost-based improvement functions are defined as:

$$f1_t^{exploit} = PF1 \left| \frac{Cost_{exploit}^{old} - Cost_{exploit}^{new}}{T_t^{exploit}} \right| \quad (3.4)$$

$$f1_t^{explore} = PF1 \left| \frac{Cost_{explore}^{old} - Cost_{explore}^{new}}{T_t^{explore}} \right| \quad (3.5)$$

To avoid premature convergence, a penalty term is introduced for non-selected phases:

$$f3_t = f3_t + PF3 \quad (3.6)$$

The final adaptive exploration and exploitation scores are then updated as:

$$F_t^{explore} = (\alpha_t \times f1_t^{explore}) + (\alpha_t \times f2_t^{explore}) + (\delta_t \times f3_t^{explore}) \quad (3.7)$$

$$F_t^{exploit} = (\alpha_t \times f1_t^{exploit}) + (\alpha_t \times f2_t^{exploit}) + (\delta_t \times f3_t^{exploit}) \quad (3.8)$$

where:

$$\delta_t = 1 - \alpha_t \quad (3.9)$$

This adaptive scoring mechanism enables the Puma Optimizer to dynamically balance global exploration and local exploitation during the optimization process.

Algorithm 1 illustrate how puma operates and algorithm 2 indicates how scores are computed.

Algorithm 1 Puma Optimizer (PO)

Require: Population size N , maximum iterations T_{max} , parameter settings

Ensure: Best solution X^{Best} and fitness value

```

1: Initialize a population of  $N$  pumas  $X_i$  within  $[LB, UB]$ 
2: Evaluate the fitness of all pumas
3: Identify the best solution  $X^{Best}$ 
   /* Unexperienced Phase */
4: for  $t = 1$  to 3 do
5:   Apply Exploration Phase
6:   Apply Exploitation Phase
7: end for
   /* Experienced Phase */
8: for  $t = 4$  to  $T_{max}$  do
9:   if  $Score_{Explore} > Score_{Exploit}$  then
10:    Apply Exploration Phase
11:    if new solution improves  $X^{Best}$  then
12:      Update  $X^{Best}$ 
13:    end if
14:   else
15:    Apply Exploitation Phase
16:    if new solution improves  $X^{Best}$  then
17:      Update  $X^{Best}$ 
18:    end if
19:   end if
20:   Update control parameters  $(f_1^{explore}, f_2^{explore}, f_3^{explore}, f_1^{exploit}, f_2^{exploit}, f_3^{exploit})$ 
21:   Recompute  $Score_{Explore}$  and  $Score_{Exploit}$ 
22: end for
23: return  $X^{Best}$ 

```

Algorithm 2 Adaptive Phase Scoring Mechanism in the Puma Optimizer (PO)**Require:** Current iteration t , maximum iterations $MaxIter$ **Ensure:** Selected phase for the next iteration/* **Exploration Function Updates** */

1: Update immediate exploration improvement:

$$f1_t^{explore} = PF1 \left| \frac{Cost_{explore}^{old} - Cost_{explore}^{new}}{T_t^{explore}} \right|$$

2: Update cumulative exploration improvement:

$$f2_t^{explore} = PF2 \left| \frac{\sum_{i=1}^3 (Cost_{explore,i}^{old} - Cost_{explore,i}^{new})}{\sum_{i=1}^3 T_{t,i}^{explore}} \right|$$

3: Update exploration penalty function:

$$f3_t^{explore} = \begin{cases} 0, & \text{if exploration is selected} \\ f3_t^{explore} + PF3, & \text{otherwise} \end{cases}$$

/* **Exploitation Function Updates** */

4: Update immediate exploitation improvement:

$$f1_t^{exploit} = PF1 \left| \frac{Cost_{exploit}^{old} - Cost_{exploit}^{new}}{T_t^{exploit}} \right|$$

5: Update cumulative exploitation improvement:

$$f2_t^{exploit} = PF2 \left| \frac{\sum_{i=1}^3 (Cost_{exploit,i}^{old} - Cost_{exploit,i}^{new})}{\sum_{i=1}^3 T_{t,i}^{exploit}} \right|$$

6: Update exploitation penalty function:

$$f3_t^{exploit} = \begin{cases} 0, & \text{if exploitation is selected} \\ f3_t^{exploit} + PF3, & \text{otherwise} \end{cases}$$

7: Update adaptive parameters (α_t, δ_t)

8: Compute exploration score:

$$F_t^{explore} = (\alpha_t f1_t^{explore}) + (\alpha_t f2_t^{explore}) + (\delta_t f3_t^{explore})$$

9: Compute exploitation score:

$$F_t^{exploit} = (\alpha_t f1_t^{exploit}) + (\alpha_t f2_t^{exploit}) + (\delta_t f3_t^{exploit})$$

10: **if** $F_t^{explore} > F_t^{exploit}$ **then**11: Select **Exploration Phase**12: **else**13: Select **Exploitation Phase**14: **end if**

3.3.4 Exploration Phase

The exploration phase is modeled after the roaming behavior of pumas as they traverse vast territories in search of prey. Its primary purpose is to investigate new and unexplored regions of the search space, increasing solution diversity and reducing the risk of premature convergence to local optima.

In this phase, each search agent acts as a puma that moves either randomly or by interacting with other agents. This dual exploration strategy helps maintain diversity within the population and ensures a balanced search across the optimization landscape.

Each puma either performs:

- **Random Jump** ($rand_1 \geq 0.5$)

A random global move, representing long-range territorial search.

- **Searching Between Pumas (otherwise)**

A directed move influenced by other pumas' positions, representing learning from the community and a gain factor G .

A simplified model of exploration movement is:

$$Z_i = \begin{cases} Rand_{dim} \times (U_b - L_b) + L_b, & \text{if } rand_1 > 0.5, \\ X_a + G \cdot (X_a - X_b) + G \cdot ((X_a - X_b) - (X_c - X_d) + ((X_c - X_d)(X_e - X_f))), & \text{otherwise} \end{cases} \quad (3.10)$$

$$G = 2 \times rand_2 - 1 \quad (3.11)$$

where $X_a, X_b, X_c, X_d, X_e, X_f$ are random pumas, G controls the degree of exploration, and $Rand_{dim}$ is a random vector generated with the same dimension as the Puma solution vector.. After new positions are generated, fitness is evaluated, and improved solutions replace their predecessors. This dual mechanism provides a good balance between random discovery and guided exploration.

Adaptive Solution Update

The generated position Z_i is used to produce a candidate solution X_{new} as follows:

$$X_{new} = \begin{cases} Z_i, & \text{if } j = j_{rand} \text{ or } rand_3 \leq U, \\ X_a, & \text{otherwise.} \end{cases} \quad (3.12)$$

Here, j_{rand} represents a randomly selected dimension, $rand_3$ is a random number, and U is an adaptive control parameter within $[0, 1]$. The parameter U increases dynamically when

an improved solution is found, allowing greater exploration as the optimization progresses. Initially, U is small, meaning high-quality solutions undergo minor changes, while lower-quality ones experience larger updates.

If the new candidate X_{new} has a better fitness value than the current solution X_i , it replaces it in the population. The operational steps of the exploration phase are detailed in Algorithm 3.

Exploration Control and Phase Switching

The decision to execute the exploration phase is determined by a phase-switching mechanism inspired by the puma's intelligence and memory. This mechanism compares two scores: $ScoreExplore$ and $ScoreExploit$, computed using functions (f_1, f_2, f_3) that evaluate performance escalation, consistency, and diversity.

If $ScoreExplore > ScoreExploit$, the algorithm prioritizes exploration, ensuring sufficient diversity and reducing the risk of local entrapment. This adaptive mechanism enables PO to dynamically balance between exploration and exploitation, achieving efficient global optimization performance, especially in multimodal problems.

Algorithm 3 Exploration Phase of the Puma Optimizer (PO)

Require: Population size N_{pop} , lower and upper bounds (L_b, U_b)

Ensure: Updated population X

```

1: /* Exploration Phase */
2: for each puma  $i = 1$  to  $N_{pop}$  do
3:   Generate a random number  $rand_1 \in [0, 1]$ 
4:   if  $rand_1 > 0.5$  then
5:     /* Random exploration within the search space */
6:      $Z_i \leftarrow Rand_{dim} \times (U_b - L_b) + L_b$ 
7:   else
8:     /* Guided exploration using peer information */
9:     Select random pumas  $X_a, X_b, X_c, X_d, X_e, X_f$ 
10:     $Z_i \leftarrow X_a + G \cdot (X_a - X_b) + G \cdot ((X_a - X_b) - (X_c - X_d) + ((X_c - X_d)(X_e - X_f)))$ 
11:   end if
12:   Update the current solutions
13:   /* Evaluate the fitness of the new candidate */
14:   Evaluate  $f(Z_i)$ 
15:   if  $f(Z_i) < f(X_i)$  then
16:     /* Update the position if the new candidate is better */
17:      $X_i \leftarrow Z_i$ 
18:   end if
19: end for
20: return Updated population  $X$ 

```

3.3.5 Exploitation Phase

In the exploitation stage, the pumas refine the best solutions using two hunting-inspired operators: The exploitation phase represents one of its two core mechanisms, working alongside the exploration phase to achieve superior optimization performance. This stage focuses on intensification, refining the best solutions discovered so far and accelerating convergence toward global optima. It is inspired by the intelligence and memory of pumas, which revisit previously successful hunting areas to maximize their chances of success.

The exploitation phase is guided by two primary behavioral models derived from puma hunting strategies ambush and sprinting. Within this framework, the algorithm introduces adaptive mechanisms to enhance local search efficiency. The best-performing solution in the population is referred to as the *Pumamale*, symbolizing the leader in the group.

Each puma either performs:

1. **Fast Running / Ambush** ($rand_4 \leq 0.5$)

This mode simulates a puma's rapid chase toward prey or a sudden ambush. The ambush includes short leaps toward nearby prey (neighboring solutions) and long jumps toward the best puma's prey (*Pumamale*).

2. **Secondary Ambush / Sprinting Mode** ($rand_4 > 0.5$)

This strategy is divided into two subcases based on a random number $rand_6$ and a static parameter L :

- **Sub-case 2a** ($rand_6 \leq L$): The update process involves the *Pumamale* and incorporates exponential functions to perform a fine-tuned local search around promising regions.
- **Sub-case 2b (Otherwise)**: A more complex search mechanism is applied, combining multiple adaptive parameters to balance intensification around the best solution and exploration of new regions.

The general form of exploitation movement is:

$$X_{\text{new}} = \begin{cases} \text{mean}(Sol_{\text{total}}) - X_r(1 - (-1)^b) - \frac{X_i}{1 + a \times rand_5}, & \text{if } rand_4 < 0.5, \\ X_{\text{male}} + 2 rand_7 e^{rand_{n1}} (X_r - X_i), & \text{if } rand_6 < L, \\ 2 rand_8 (F_1 - R X_i + F_2(1 - R) X_{\text{male}}), & \text{otherwise.} \end{cases} \quad (3.13)$$

where X_{male} is the best solution guiding the others, $\text{mean}(Sol_{\text{total}})$ represents the mean value of all solutions in the current population, X_r denotes a randomly selected solution from the population, F_1 is an adaptive exploitation coefficient controlling the local refinement intensity,

and R is a randomization factor within the range $[-1, 1]$ used to balance stochastic movements during the search process. These operators ensure both local refinement and controlled random perturbation, preventing premature convergence.

After computing X_{new} , it replaces the current solution X_i only if it yields a lower fitness value, ensuring that only improved solutions are retained. The sequence of operations carried out during the exploration phase is illustrated in Algorithm 4.

Algorithm 4 Exploitation Phase of the Puma Optimizer (PO)

Require: Population size N_{pop} , control parameters a, b, L, F_1, F_2 , and R

Ensure: Updated population X

```

  /* Exploitation Phase */
1: for each puma  $i = 1$  to  $N_{pop}$  do
  /* Select the reference pumas */
2:   Choose a random puma  $X_r$  and the best puma  $X_{male}$ 
  /* Case 1: Local refinement using average-based movement */
3:   if  $rand_4 < 0.5$  then
4:      $X_{new} \leftarrow mean(Sol_{total}) - X_r(1 - (-1)^b) - \frac{X_i}{1 + a \times rand_5}$ 
5:   else
  /* Case 2: Controlled movement toward best or random solution */
6:     if  $rand_6 < L$  then
7:        $X_{new} \leftarrow X_{male} + 2 \times rand_7 \times e^{rand_{n_1}}(X_r - X_i)$ 
8:     else
9:        $X_{new} \leftarrow 2 \times rand_8(F_1 - R \times X_i + F_2(1 - R) \times X_{male})$ 
10:    end if
11:  end if
  /* Evaluate and update if improvement is found */
12:  Evaluate fitness  $f(X_{new})$ 
13:  if  $f(X_{new}) < f(X_i)$  then
14:     $X_i \leftarrow X_{new}$ 
15:  end if
16: end for
17: return Updated population  $X$ 

```

3.3.6 Updating and Termination

After completing the operations of the selected phase, each puma's new position is evaluated. The male puma is updated if a better fitness value is found. The process continues until the maximum iteration number $MaxIter$ is reached or another termination criterion (such as convergence tolerance) is met. Algorithm 2 describes the update process used to determine whether the algorithm performs exploration or exploitation. Figure 3.1 provides an overview of the Puma Optimizer (PO) methodology, highlighting its main operational phases.

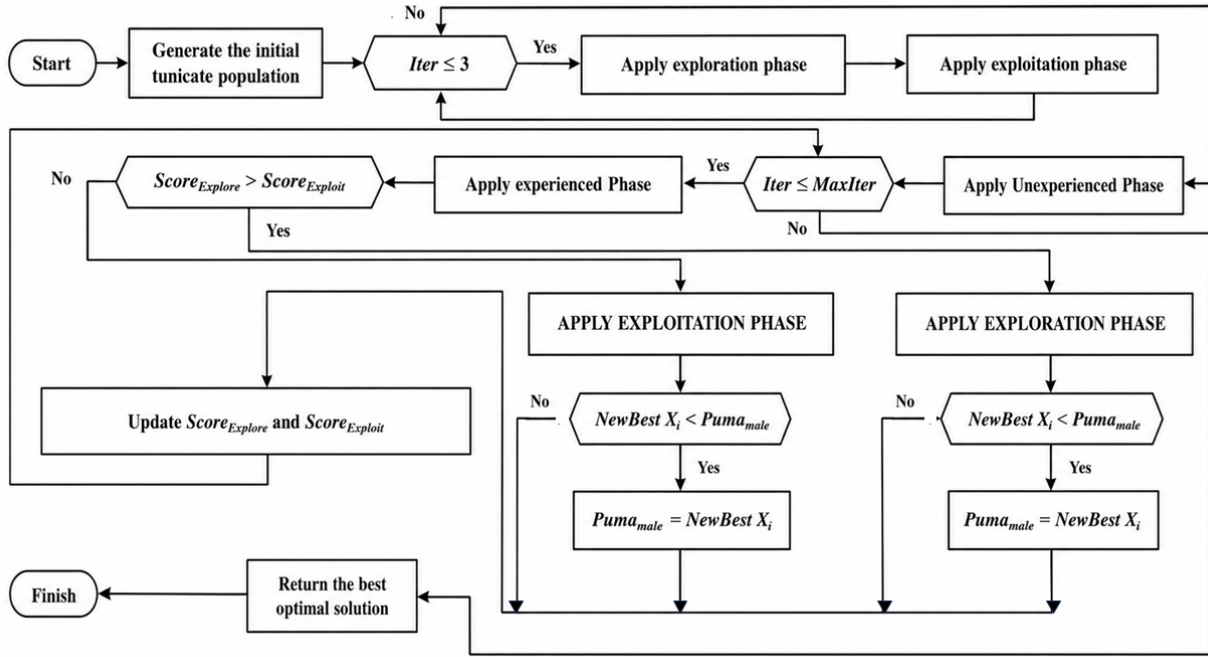


Figure 3.1: The Puma Optimization Method [96]

3.4 Computational Complexity

Although the algorithm includes multiple sub-mechanisms (exploration, exploitation, and phase switching), its total computational complexity is comparable to other swarm-based methods:

$$O(T \times N \times D)$$

where T is the maximum number of iterations, N the population size, and D the problem dimension. The adaptive phase mechanism adds only minor overhead while significantly improving convergence speed.

3.5 Advantages and Features

The main advantages of the Puma Optimizer are:

- Adaptive phase control that automatically balances exploration and exploitation.
- Learning-based evolution using past performance to guide future behavior.
- High convergence accuracy due to efficient exploitation.
- Strong global search ability preventing premature convergence.
- Ease of adaptation to both continuous and discrete optimization problems.

3.6 Conclusion

The PO demonstrates strong optimization capability through its adaptive balance between exploration and exploitation. Its dynamic search mechanism enables efficient global exploration while maintaining accurate local refinement around promising solutions. By alternating between experienced and unexperienced phases, the optimizer improves convergence speed, preserves population diversity, and reduces the risk of premature convergence toward local optima. In addition, PO provides stable performance and strong adaptability in complex and dynamic optimization environments. These characteristics make it a reliable and intelligent metaheuristic optimizer for solving challenging optimization problems with high efficiency and solution quality. Building upon these advantages, the next chapter presents the integration of the Puma Optimizer into Wireless Sensor Network clustering. The proposed method exploits the optimizer's adaptive search behavior to enhance Cluster Head selection and cluster formation by considering important network metrics such as residual energy, communication distance, and load balancing. Through the combination of exploration and exploitation mechanisms, the proposed approach aims to achieve energy-efficient clustering, prolonged network lifetime, and improved overall network stability.

The next chapter presents the adaptation of the Puma Optimizer for CH election in WSNs, detailing its implementation, parameter tuning, and performance evaluation.

Chapter 4

Proposed Method

4.1 Introduction

WSNs play a critical role in applications ranging from environmental monitoring to smart cities. However, they face significant challenges, including limited energy resources, dynamic topologies, and the need for efficient data delivery. Clustering strategies are essential to address these issues, as they can improve network lifetime, reduce communication overhead, and enhance overall performance.

Despite numerous protocols proposed in the literature, no approach guarantees an optimal clustering solution, and mathematical proof of global optimality is absent, highlighting the need for adaptive, energy-aware, and intelligent methods.

This chapter presents the adaptation of the Puma Optimizer for WSNs, focusing on CH selection, cluster formation, and energy-efficient routing. By leveraging PUMA's exploration and exploitation capabilities, the proposed method dynamically selects CHs to balance energy consumption. Subsequently, clusters are formed to optimize intra-cluster communication, and routing strategies are designed to minimize transmission distances and prolong network lifetime.

4.2 Proposed Method: PUMA-GRID – PUMA based Clustering with Grid k-NN Routing

The proposed protocol, PUMA-GRID, integrates two complementary intelligent components: a metaheuristic optimizer for CHs selection and a machine learning inspired routing strategy for energy efficient data forwarding. PUMA serves as the metaheuristic engine responsible for exploring and refining CH configurations, while a k -Nearest Neighbors (k-NN) mechanism provides a learning based routing strategy that selects forwarding CHs based on spatial similarity and proximity patterns. This dual integration aligns with the central theme of the thesis combining machine learning with metaheuristics to enhance energy efficiency in WSNs.

In this study, PUMA-GRID assumes a single BS located either inside or at the edge of the monitored area, which serves as the central data collection point. This assumption aligns with most benchmark WSN configurations and facilitates consistent performance comparison. Although deploying multiple BSs could further reduce communication distances and balance network load, the proposed framework was designed and evaluated under a single-BS scenario. To complement clustering, the approach incorporates a grid based, machine learning inspired multi-hop routing mechanism. Grid based routing divides the WSN into uniform virtual cells, facilitating energy efficient data forwarding through structured multi hop paths and localized

packet transmission. As high-lighted in [38], this approach enhances scalability and ensures predictable communication costs, although its performance depends on appropriate grid configuration to prevent problems such as empty cells and uneven node distribution. This design explicitly combines a metaheuristic component (PUMA for CH selection) with a machine learning inspired component (grid k-NN routing).

4.2.1 Initialization

In the initialization phase, the clustering process is prepared by setting up candidate solutions for CH selection. In the proposed protocol, the CH selection process is not predetermined within each grid cell but dynamically optimized through the PUMA algorithm. All sensor nodes are initially eligible to become CHs, and their selection depends on the fitness function that considers residual energy, distance to the BS, and intra-cluster communication distance. After random deployment of nodes in the target area, each node transmits its position information to the BS, which then begins executing the PUMA algorithm. A population of m individuals (candidate solutions) is generated, where each individual is represented as an n -length binary vector. In this encoding, a value of 1 denotes that the node is selected as a CH, while 0 indicates a regular sensor node. The desired number of CHs is specified as a user defined percentage of the total nodes. This binary representation, consistent with classical metaheuristic clustering approaches, enables flexible exploration of CH configurations and establishes a solid foundation for the optimization process.

4.2.2 PUMA-Based Clustering and Fitness Evaluation

PUMA balances exploration and exploitation through adaptive control mechanisms embedded in its search dynamics. During the exploration phase, candidate solutions undergo wide, randomized position adjustments that preserve diversity and help the algorithm avoid premature convergence. As optimization progresses, PUMA transitions into the exploitation phase, where updates become more focused, favoring local improvements around the current best solution. The number of CHs is not strictly enforced during this process, allowing the search to flexibly explore a broader range of configurations. This hyper heuristic switching mechanism, as demonstrated in recent applications of the Puma Optimizer, dynamically adjusts the exploration–exploitation ratio according to the optimization context, enabling progressive refinement of clustering results while avoiding local optima. The fitness function used in PUMA-GRID integrates three key metrics:

1. The total distance between each regular node and its nearest CH.

2. The distance from each CH to the BS.
3. The residual energy of the selected CHs.

These components are combined using a set of weighted coefficients w_1 , and w_2 , whose values are specified in the experimental setup. The objective function is formulated as follows:

$$Fitness = w_1 \cdot f_1 + w_2 \cdot f_2 \quad (4.1)$$

The sub-objectives are defined as follows:

$$f_1 = \frac{1}{N \times d_{\max}} \left[\sum_{j=1}^{N_{CH}} \sum_{i=1}^N \begin{pmatrix} \text{dist}(CH_j, S_i), & \text{if } S_i \text{ belongs to } CH_j \\ 0, & \text{otherwise} \end{pmatrix} + \sum_{j=1}^{N_{CH}} \text{dist}(CH_j, BS) \right] \quad (4.2)$$

$$f_2 = \frac{\sum_{j=1}^{N_{CH}} E_{\text{res},j}}{N_{CH} \cdot E_{\text{init}}} \quad (4.3)$$

noindent where:

- N is the total number of nodes in the network,
- N_{CH} is the total number of cluster heads,
- S_i denotes sensor node i ,
- CH_j denotes cluster head j ,
- $\text{dist}(CH_j, S_i)$ is the Euclidean distance between node S_i and its corresponding cluster head CH_j (intra-cluster communication),
- $\text{dist}(CH_j, BS)$ is the Euclidean distance between cluster head CH_j and the BS (inter-cluster communication),
- d_{\max} is the maximum distance between any two nodes in the network, used for normalization,
- $E_{\text{res},j}$ is the residual energy of cluster head j ,
- E_{init} is the initial energy of each node,
- w_1 and w_2 are weighting coefficients satisfying $w_1 + w_2 = 1$.

The PO Algorithm operates in a continuous domain, while the problem of CH selection is inherently binary: a sensor node i is either a CH (1) or a non-CH (0). To bridge this gap, a simple two-step conversion mechanism is applied after each continuous position update of the Pumas.

Although the initial bounds for the Puma positions are set to $LB = 0$ and $UB = 1$, the mathematical operations within the PUMA updates (such as the movement towards the best solution) can result in continuous position values (X_i) that fall outside this $[0, 1]$ range (i.e., negative values or values greater than 1). To ensure the search space remains valid for binariza-

tion, a min–max normalization is first applied to the entire position vector X . The normalized position value X_i^{norm} is then converted into a binary decision b_i using a fixed threshold of 0.5:

$$X_i^{\text{norm}} = \frac{X_i - \min(X)}{\max(X) - \min(X)},$$

$$b_i = \begin{cases} 1, & \text{if } X_i^{\text{norm}} \geq 0.5 \\ 0, & \text{if } X_i^{\text{norm}} < 0.5 \end{cases} \quad (4.4)$$

After applying normalization and threshold based binarization, the updated binary vectors represent the next generation of Pumas. These newly generated solutions are then evaluated through the objective function to determine their fitness and guide the subsequent search steps. After evaluating all candidate solutions in the PUMA population using the objective function, the individual with the minimum cost value is chosen as the best solution. This puma represents the most energy efficient clustering configuration for the current round, achieving the optimal trade off among intra-cluster communication, CH to BS transmission, residual energy, and the cluster count penalty. In this context, a round refers to a complete operational cycle consisting of cluster formation, data sensing, aggregation by CHs, and data transmission to the BS. Algorithm 5 illustrates how PUMA operates in selecting CHs.

4.2.3 Machine learning inspired grid based multi hop routing via K-NN

The routing component of PUMA-GRID employs k-NN[97, 98] technique as a lightweight, non-parametric learning mechanism. Instead of relying on rigid geometric rules, k-NN[99] enables each CH to select a next hop relay based on local spatial similarity specifically, proximity and directional relevance toward the BS. This preserves the essence of k-NN: decisions are made by examining the closest neighbors in feature space and choosing the one that best advances the objective (minimizing transmission cost).

Motivation for integrating k-NN into routing: since metaheuristics are already used for CH selection, the next relay CH must be chosen using a simple and low cost method to avoid adding extra computation time. For this reason, k-NN is integrated as a straightforward decision mechanism that can quickly select the next relay CH based on distance and proximity information without requiring heavy processing. It is:

- **local and distributed:** only nearby CHs are considered,
- **memory efficient:** requires storing only a small neighbor list,
- **distance driven:** decisions reflect spatial similarity and energy cost,
- **adaptive:** CH selection changes every round, so the neighbor set adapts automatically.

Algorithm 5 Binary Puma Optimization Algorithm for WSN Clustering

```

1: Inputs: Number of sensors  $N$ , sensor positions  $(X_i, Y_i)$ , residual energy, BS position  $(BS_x, BS_y)$ , maximum iterations  $T_{\max}$ , weighted coefficients  $w_1, w_2$ 
2: Output: Optimal binary vector of CHs best fitness value
3: Fitness Function: Weighted sum of residual energy, distance to cluster center, and distance to BS
4: Initialize a random population of pumas  $Z_i$ , where  $i = 1, 2, \dots, N$ 
5: Evaluate fitness of each puma based on clustering quality and energy metrics
6: Identify the best solution as the male puma (leader)
7: Apply Unexperienced Phase
8: for iteration = 1 to 3 do
9:   Apply Exploration Phase
10:  Apply Exploitation Phase
11: end for
12: Apply Experienced Phase
13: for iteration = 4 to  $T_{\max}$  do
14:   Apply Experienced Phase
15:   if  $Score_{Explore} > Score_{Exploit}$  then
16:     Apply Exploration Phase
17:     if New best  $X_i$  (from exploration) is better than current male puma then
18:       Update best puma
19:     end if
20:   else
21:     Apply Exploitation Phase
22:     if New best  $X_i$  (from exploitation) is better than current male puma then
23:       Update male puma
24:     end if
25:   end if
26:   Update adaptive parameters of PO
27:   Adjust exploration and exploitation rates accordingly
28: end for
29: return Best CH selection (male puma) and corresponding fitness value

```

Thus, k-NN allows CHs to behave as “local learners,” where each forwarding step selects the neighbor that most closely matches the desired direction toward the BS under minimum expected energy cost.

k-NN based multi hop forwarding rule: routing begins with a grid-based search for possible relay CHs. For each CH_i :

1. If the BS lies in an adjacent grid cell, CH_i sends directly to the BS.
2. Otherwise, the BS inspects the adjacent grid cells that lie closer to the BS (Manhattan distance = 1). If one or more CHs exist in these cells, they form the candidate set.
3. If no CH is found, the search is expanded outward in a Manhattan-like pattern:

$$d_{\text{grid}} = 2, 3, 4, \dots$$

until at least one candidate CH is discovered. This guarantees that routing continues

even when some grid cells are empty.

4. Once the candidate CHs are collected, distances are computed:

$$C_j = \text{Dist}(CH_i, CH_j) + \text{Dist}(CH_j, BS) \quad (4.5)$$

where Euclidean distance is used for both terms.

5. The k-NN rule is then applied by selecting the k candidates with the smallest values of C_j . The neighbor with the minimum distance sum becomes the next relay.

This two stage mechanism (Manhattan grid expansion + k-NN distance evaluation) ensures relay selection is both geometrically efficient and learning inspired, while remaining compatible with the grid structure. Figure 4.1 illustrates the operation of the multi-hop routing mechanism.

Practical choice of k : in the proposed implementation, $k = 1$ was found to provide the most stable and energy efficient performance. This value minimizes computation and communication overhead while still preserving the k-NN logic, because:

- candidates are first filtered through the grid expansion mechanism,
- then the closest relay is selected based on the minimized distance sum C_j .

Thus, even with $k = 1$, the routing process follows the k-NN principle by always choosing the relay that is most spatially relevant and energetically efficient.

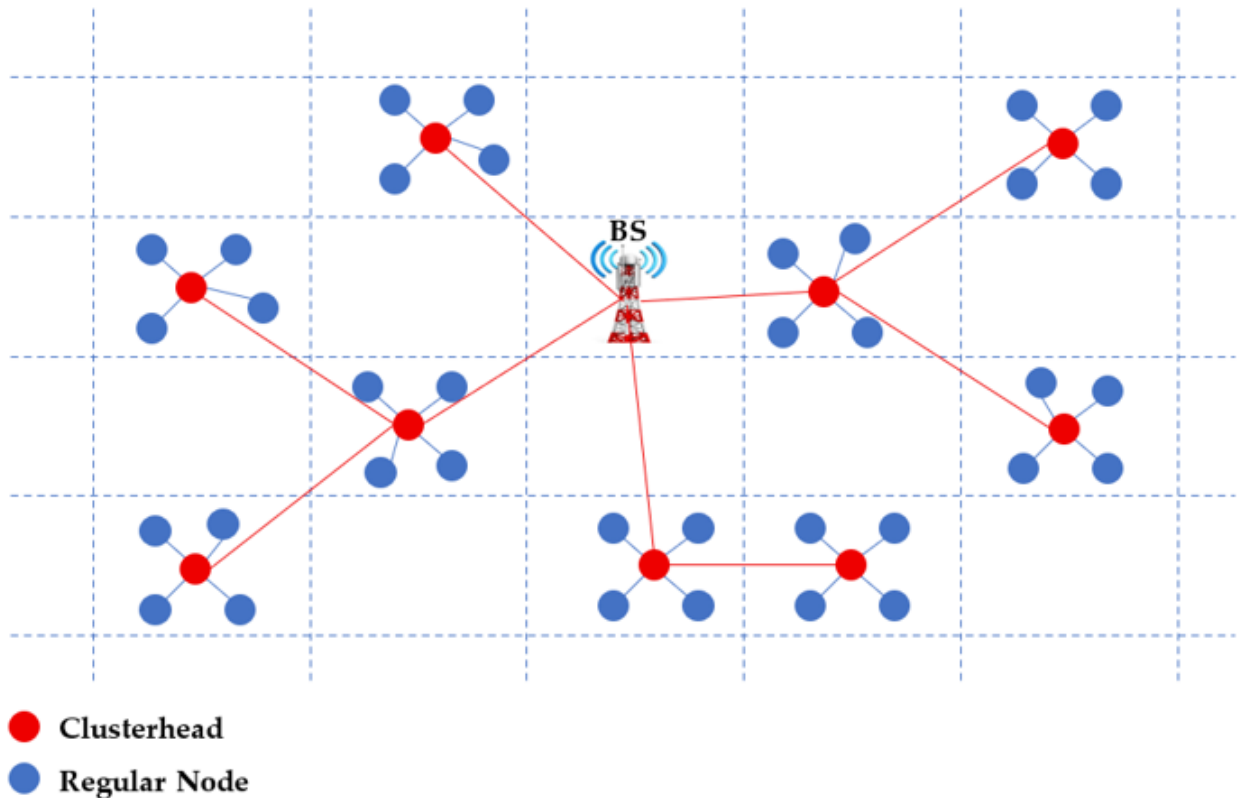


Figure 4.1: Grid-based routing.

Algorithm 6 Grid-Based Multi-Hop Routing Algorithm

```

1: Input: CH positions, BS position, grid size  $G$ , network dimensions, parameter  $k$ 
2: Output: Next relay for each CH (either BS or another CH)
3: Divide the network area into grid cells of size  $G \times G$ 
4: Assign each CH to its corresponding grid cell
5: Determine the grid cell containing the BS
6: for each  $CH_i$  not located in the BS grid cell do
7:   Let  $cell_i$  be the grid cell of  $CH_i$  and  $cell_{BS}$  the grid cell of the BS
8:   if ManhattanDistance( $cell_i, cell_{BS}$ ) = 1 then
9:     Set next relay of  $CH_i$  to BS (direct transmission)
10:  else
11:    Initialize search radius  $R \leftarrow 1$ 
12:    Initialize candidate set  $\mathcal{C} \leftarrow \emptyset$ 
13:    while  $\mathcal{C} = \emptyset$  do
14:      Identify all grid cells  $cell$  such that:
      ManhattanDistance( $cell_i, cell$ ) =  $R$  and  $cell$  is closer to the BS than  $cell_i$ 
15:      Collect all CHs located in these cells and add them to  $\mathcal{C}$ 
16:      Increase search radius  $R \leftarrow R + 1$ 
17:    end while
18:    for each candidate  $CH_j \in \mathcal{C}$  do
19:      Compute Euclidean distance  $d_1 = \text{Dist}(CH_i, CH_j)$ 
20:      Compute Euclidean distance  $d_2 = \text{Dist}(CH_j, BS)$ 
21:      Compute total cost
      
$$C_j = d_1 + d_2$$

22:    end for
23:    Select the  $k$  candidates with smallest  $C_j$  values
24:    In the proposed implementation, set  $k = 1$  and choose
      
$$CH_{\text{next}} = \arg \min_{CH_j \in \mathcal{C}} C_j$$

25:      Set next relay of  $CH_i$  to  $CH_{\text{next}}$ 
26:    end if
27:  end for
28: Each CH forwards its aggregated data hop by hop using the same rule until the BS is
    reached

```

4.2.4 Adaptive Operation and Steady-State Execution

Upon determining the best individual (lowest-cost solution), PUMA-GRID organizes clusters by enabling CHs to broadcast advertisements. Ordinary nodes join their closest CH, and a time-division schedule is established. In the steady-state phase, regular nodes sense data and transmit to their CH, which then aggregates data and forwards it through the grid-based multi-hop path toward the BS. Re-clustering is triggered when CH residual energy falls below thresholds or when load imbalance arises, ensuring sustained energy-aware operations. Figure 4.2 depicts the flowchart of the proposed method, while Figure 4.3 offers a general overview

of PUMA-GRID.

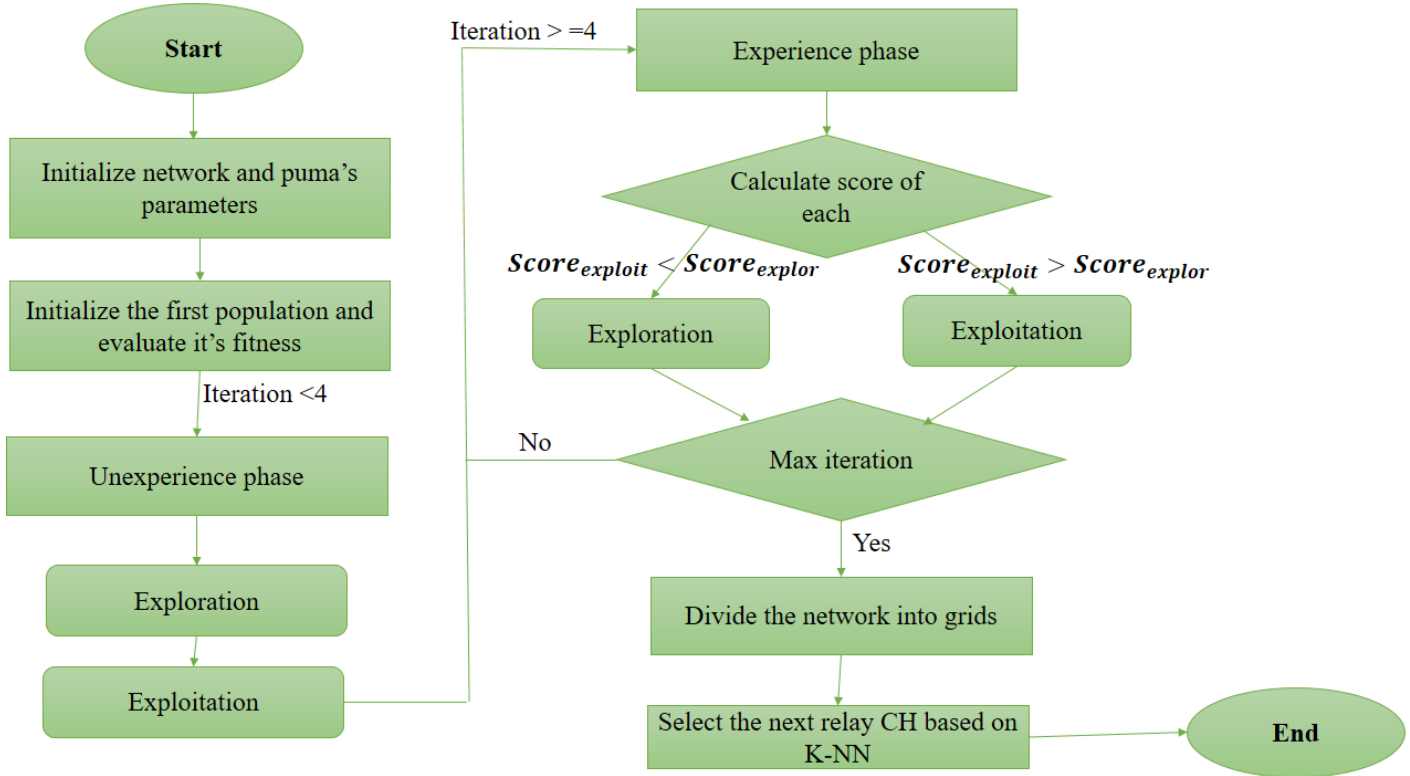


Figure 4.2: Flowchart of the proposed method.

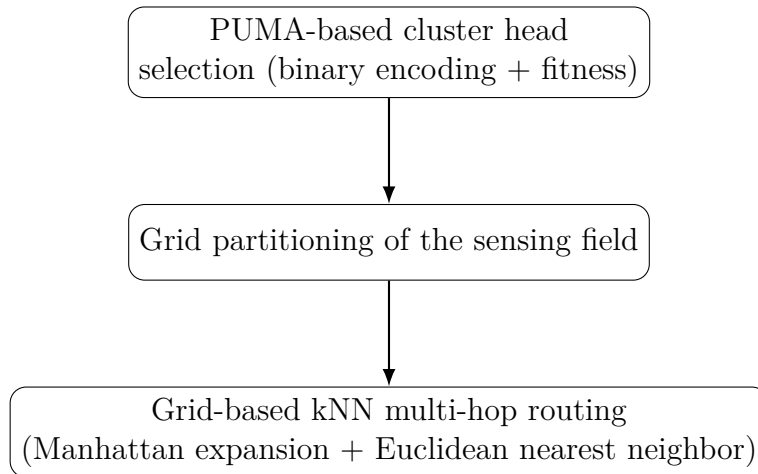


Figure 4.3: Global view of the PUMA-GRID contributions: metaheuristic clustering, grid structure, and kNN-based routing.

4.2.5 Complexity Analysis of PUMA-GRID

The computational behavior of PUMA-GRID can be understood by analyzing its two main components separately: the PUMA based CH selection process and the grid assisted k-NN routing mechanism. Evaluating these stages independently clarifies how the protocol maintains global optimization capability while keeping routing overhead low and compatible with WSN

constraints.

4.2.6 Clustering complexity

In the clustering phase, the Puma Optimizer evaluates a population of P candidate solutions, each representing one possible CH configuration over N sensor nodes. Computing the fitness of a single candidate requires evaluating intra cluster distances, CH to BS distances, and CH residual energy values; this results in a cost of $O(N)$ per candidate. Across I optimization iterations, the overall clustering complexity becomes $O(P \times N \times I)$, which is typical for metaheuristic clustering methods. This cost remains practical for WSNs because PUMA converges rapidly and its operations can be parallelized conceptually. The clustering stage therefore dominates the computational burden of the protocol.

4.2.7 Routing complexity

The routing stage is intentionally designed to remain lightweight. After clustering, the monitored field is divided into G grid cells and each CH is assigned to the cell it occupies. When selecting a relay, a CH does not inspect all other CHs. Instead, it searches only the grid cells located in the direction of the BS. This spatial filtering restricts the search to a very small, fixed set of neighboring cells. Because the Manhattan expansion examines only adjacent or second layer grids until at least one candidate relay is found, the cost of discovering relay candidates is effectively $O(G_{\text{local}}) \approx O(1)$, where G_{local} denotes the number of locally inspected grid cells. Once the candidate CHs are found, the k-NN mechanism computes Euclidean distances for at most K possible relays. The k-NN evaluation step therefore has complexity $O(K)$, which remains small because the grid structure ensures that the candidate set is always limited. Thus, one forwarding decision incurs routing overhead proportional only to the number of filtered candidate CHs.

4.2.8 Overall complexity of PUMA-GRID

When the clustering and routing components are considered together, the total computational effort per operational round is dominated by the PUMA based CH selection, while routing contributes only a minor additive cost. In simplified form, the overall complexity can be expressed as the clustering term $P \times N \times I$ plus a small routing term proportional to K . Because route construction can also be performed centrally at the BS using CH coordinates already known after clustering the energy burden on CHs remains minimal. This extended analysis shows that PUMA-GRID integrates global optimization with lightweight routing. The

clustering stage performs the heavy computation, while the routing stage relies on grid filtering and nearest neighbor evaluation to maintain low complexity and strong scalability. Together, these properties confirm that PUMA-GRID remains compatible with WSN resource constraints while still delivering optimized clustering and energy aware multi hop forwarding.

4.3 Conclusion

This chapter presented the methodological framework used to adapt both the PUMA optimization process and the grid KNN routing strategy to the specific requirements of WSNs. The adaptation was carried out in a centralized manner, ensuring that CH selection and routing decisions are optimized globally while respecting WSN constraints such as limited energy, communication range, and the spatial distribution of nodes. The chapter described how the solution representation, the objective function, and the grid structure were redesigned so that PUMA and KNN can operate efficiently and coherently within the same architecture. Beyond the individual components, the methodology highlights how the two elements complement each other. PUMA provides an intelligent clustering mechanism that distributes energy consumption by selecting suitable cluster heads, while the KNN routing layer introduces a simple learning inspired decision rule that guides data forwarding through locally optimal paths. Although routing must remain lightweight due to the limited capabilities of sensor nodes, the use of KNN allows each relay decision to mimic a small classification step based on spatial similarity and direction toward the BS. In this way, the routing phase builds on the structure formed during clustering and further strengthens the link between metaheuristic optimization and machine learning concepts. The entire framework demonstrates how clustering and routing can be integrated into a unified design that takes advantage of adaptive search, neighborhood learning, grid organization, and spatial filtering. This integration ensures predictable communication patterns, reduces energy waste, and preserves network lifetime under different deployment conditions.

The next chapter will evaluate the performance of the proposed approach. It will examine the influence of different weight combinations in the objective function under three BS placement scenarios (inside, at the edge, and outside the network). The obtained results will then be compared with several well known clustering and routing protocols to assess the improvements introduced by the proposed method.

Chapter 5

Simulation Setup, Results, and Discussion

5.1 Introduction

This chapter presents a comprehensive simulation study and performance evaluation of the proposed PUMA-Grid protocol. The objective is to examine its energy efficiency, clustering effectiveness, routing stability, and overall operational robustness across a variety of network conditions.

The first phase of the study involves an intensive analysis to determine the optimal weights for the multi objective fitness function used in PUMA. Since the fitness function integrates two key factors distance (intra and inter cluster) and residual energy a wide range of weighted combinations was tested. These evaluations were conducted under three different BS locations (inside, at the edge, and outside the sensing field) to ensure that the selected weights remain effective across diverse topologies and communication constraints.

After identifying the optimal weight configuration, it was integrated into the complete PUMA-GRID protocol, combining the optimized clustering process with the proposed grid k-NN multi-hop routing strategy. The enhanced protocol was then benchmarked against several state of the art clustering and routing schemes to evaluate performance in terms of network lifetime, residual energy consumption, packet delivery efficiency, and coverage preservation.

In addition to routing performance comparison, the study also includes an analysis of execution time for each protocol to assess computational cost and practical feasibility. Finally, a scalability evaluation was carried out using large-scale networks with higher node densities to examine how well the proposed approach adapts to more demanding deployment scenarios.

5.2 Simulation Setup and Energy Consumption Model

In real world WSN deployments, sensor nodes determine their geographic locations using a variety of localization methods, chosen based on application requirements and cost considerations. Outdoor networks often employ GPS modules, whereas signal-based techniques such as Received Signal Strength Indication, Time of Arrival, or Time Difference of Arrival are suitable when GPS is unavailable. Alternatively, anchor based or centroid localization approaches can be used, leveraging a few reference nodes with known positions. Typically, the BS location is predefined and communicated to all nodes during network setup, allowing each node to store the information locally for use in clustering and routing decisions.

For energy modeling, this study uses the first-order radio model, summarized in Equation 5.1. Here, transmitting a packet of L bits over a distance d consumes $E_{TX}(L, d)$ energy, while receiving the same packet requires $E_{RX}(L)$ energy.

The transmission energy is determined according to the channel model:

$$E_{TX}(L, d) = \begin{cases} L \cdot E_{elec} + L \cdot \varepsilon_{fs} \cdot d^2, & d < d_0 \\ L \cdot E_{elec} + L \cdot \varepsilon_{mp} \cdot d^4, & d \geq d_0 \end{cases} \quad (5.1)$$

where E_{elec} is the per-bit energy required for transmission or reception. The critical distance d_0 that separates free-space and multi-path propagation is:

$$d_0 = \sqrt[4]{\frac{\varepsilon_{fs}}{\varepsilon_{mp}}} \quad (5.2)$$

Receiving an L -bit message consumes:

$$E_{RX}(L) = L \cdot E_{elec} \quad (5.3)$$

The simulations were carried out in MATLAB to assess node level and network wide energy performance. A set of n nodes was randomly deployed within the monitoring area. Nodes collected data continuously, aggregated it at CHs, and then forwarded it to the BS either directly or via multi-hop communication through other CHs. Table 5.1 presents the simulation parameters and the various BS positions considered.

5.3 Choosing the Optimal Weights for the Fitness Function

To enhance the energy efficiency of the proposed PUMA-WSNC protocol, a simplified multi-objective fitness function was utilized, combining two key factors: distance and residual energy. The fitness function considers:

- **Distance factor:** accounts for both intra-cluster distances (sensor nodes to their respective CHs) and inter-cluster distances (CHs to the BS),
- **Residual energy factor:** reflects the remaining energy of the cluster heads.

Each factor is associated with a weighted coefficient, w_1 for distance and w_2 for residual energy. The fitness function is designed to be minimized, and the PUMA solution with the lowest cost is considered the most optimal CH configuration for that iteration. To determine the most suitable weight combinations, comprehensive simulations were conducted under three deployment scenarios of the Base Station:

- Located inside the sensor field (center position),
- Placed at the edge of the sensor field,
- Situated outside the network boundary.

A full factorial exploration would involve multiple combinations of the two weighted coefficients. Nonetheless, all combinations were simulated and analyzed. The simulation parameters listed in Table 5.1 were selected based on widely adopted configurations in WSN studies to ensure fair comparison and reproducibility. The evaluation of each configuration was based on the following performance indicators:

- FND, HND, LND: Respectively, the rounds when the first, half, and last nodes die, used to estimate network lifetime and stability.
- Live Nodes per Round: Tracks the network’s vitality across simulation time.
- Number of Packets Sent to the BS: Indicates the data delivery capability.
- Coverage Fairness Index (CFI): In many real-world WSN applications (e.g., environmental monitoring, area surveillance, disaster detection), maintaining even spatial distribution of live nodes is just as important as node count. A network where 20 live nodes are spread across the entire area provides much better sensing coverage and reliability than one where 30 live nodes are clustered in just one corner. How CFI Works:

1. Divide the network into equal-sized grid cells (e.g., 10×10 m, 20×20 m). We choose
Grid Size = 20 m
2. At each round, count:
 - *OccupiedCells* = number of cells with at least one live node.
 - *TotalCells* = total number of grid cells.
3. Compute:

$$CFI = \frac{OccupiedCells}{TotalCells} \quad (5.4)$$

4. Interpretation: $CFI = 1$ (perfect spatial fairness), $CFI \approx 0$ (poor distribution).

Table 5.1: Simulation Parameters

Simulation and Parameters	Parameter Value
Network size	100×100 m ² , 200×200 m ²
Initial energy of each node	0.1 J
Number of nodes	200
CH	5%
Packet size	500 bytes
E_{elec}	50 nJ/bit
E_{fs}	10 pJ/bit/m ²
E_{mp}	0.0013 pJ/bit/m ⁴
d_0	10 m
E_{DA}	5 pJ/bit/signal
<i>BaseStationPosition</i>	(50,50), (150,50), (0,0).

The simulation parameters presented in Table 5.1 were chosen according to commonly adopted settings in WSN research to facilitate reproducibility and fair comparison. The network area of $100 \times 100 \text{ m}^2$ with 200 sensor nodes represents a moderate-density deployment suitable for evaluating scalability. The proportion of CHs was set to 5%, consistent with protocols like LEACH and its variants, providing a balance between energy efficiency and communication overhead. Standard benchmarks were also followed for the initial node energy (0.1 J) and packet size (500 bytes). The energy model parameters (E_{elec} , ε_{fs} , and ε_{mp}) correspond to the first-order radio model, while the threshold distance $d_0 = 10 \text{ m}$ separates free-space and multi-path propagation regions. Prior to simulation, each node is initialized with essential network information, including total node count, grid dimensions, and the BS location, which is broadcast once at setup. These parameters enable accurate computation of distances and support cluster formation and routing processes. Although fixed weights are used in the fitness function for this study, the framework can be extended to support adaptive weight adjustment. In such a case, weights could be updated dynamically in real time based on network conditions, including average residual energy, local node density, or communication cost, to enhance performance under varying scenarios.

5.3.1 Impact of Weight Combinations on Different metrics (BS inside the Network)

The results in figure 5.1 illustrate how the FND, HND, and LND metrics vary across different combinations of the weighting coefficients. Higher values of w_1 , which emphasize intra and inter cluster distance minimization, consistently produce higher FND values. By prioritizing shorter distances, early communication becomes more energy efficient and delays the first node death. However, when w_1 dominates excessively, CH selection becomes biased toward spatially favorable nodes, causing these nodes to be used more frequently and deplete their energy faster, which explains the weaker HND performance.

In contrast, higher values of w_2 , which prioritize residual energy, significantly improve HND. By selecting CHs with higher remaining energy, the protocol distributes the energy consumption more evenly across the network, thereby delaying the point at which half of the nodes die. This comes at the cost of slightly lower FND values, since the energy rich candidates are not always the closest in terms of distance.

The most favorable LND values are obtained when both weights take moderate values. In this balanced configuration, the protocol benefits simultaneously from distance efficient communication and energy-aware CH selection. Communication distances remain low, while the

residual energy is preserved across nodes. This balance leads to more uniform energy consumption and extends the overall network lifetime.

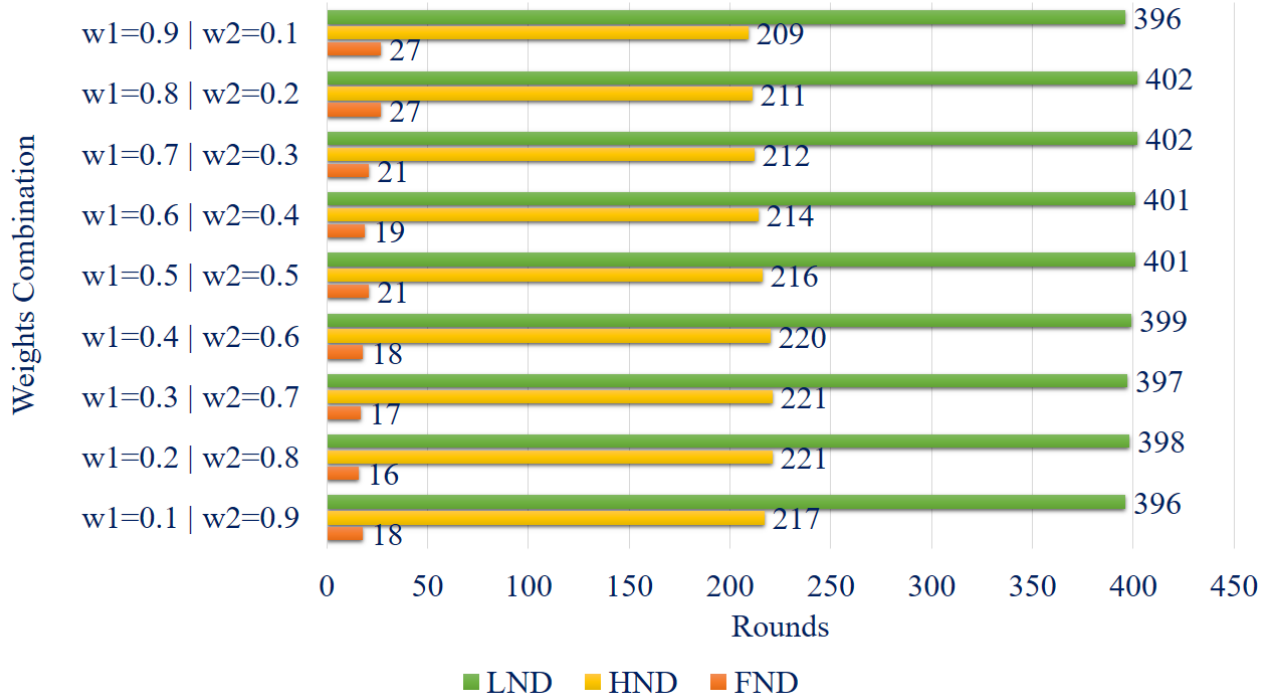


Figure 5.1: Impact of weight configurations on FND, HND, and LND with BS inside the network area.

Overall, when the BS is located inside the monitored region, a balanced weighting such as $w_1 = 0.6$ and $w_2 = 0.4$ offers the best compromise. It improves both the early- and mid-life stability while preserving the already strong LND performance characteristic of PUMA.

The results clearly show that increasing the distance weight w_1 leads to a substantial increase in the number of delivered packets. A higher emphasis on distance optimization encourages the selection of CHs that are not only closer to surrounding nodes but also better positioned with respect to the BS. This spatial advantage shortens communication paths and reduces transmission energy, enabling nodes to participate in a larger number of communication rounds before their energy is depleted.

Figure 5.2 illustrates the total number of packets successfully transmitted to the BS for different weight combinations (w_1, w_2) when the BS is positioned inside the network field. As w_1 decreases and more importance is assigned to the residual energy factor w_2 , the total number of transmitted packets gradually declines. Although prioritizing residual energy helps balance energy consumption across nodes, it reduces the spatial optimality of CH selection. Energy rich nodes may be chosen as CHs even when they occupy less favorable geographic positions, which increases communication distances and reduces overall forwarding efficiency.

Furthermore, when w_2 dominates, CH selection becomes strongly energy dependent, resulting in a smaller number of CHs being elected in several rounds. Fewer CHs directly translate into fewer aggregated data packets, further contributing to the reduction in total packet delivery.

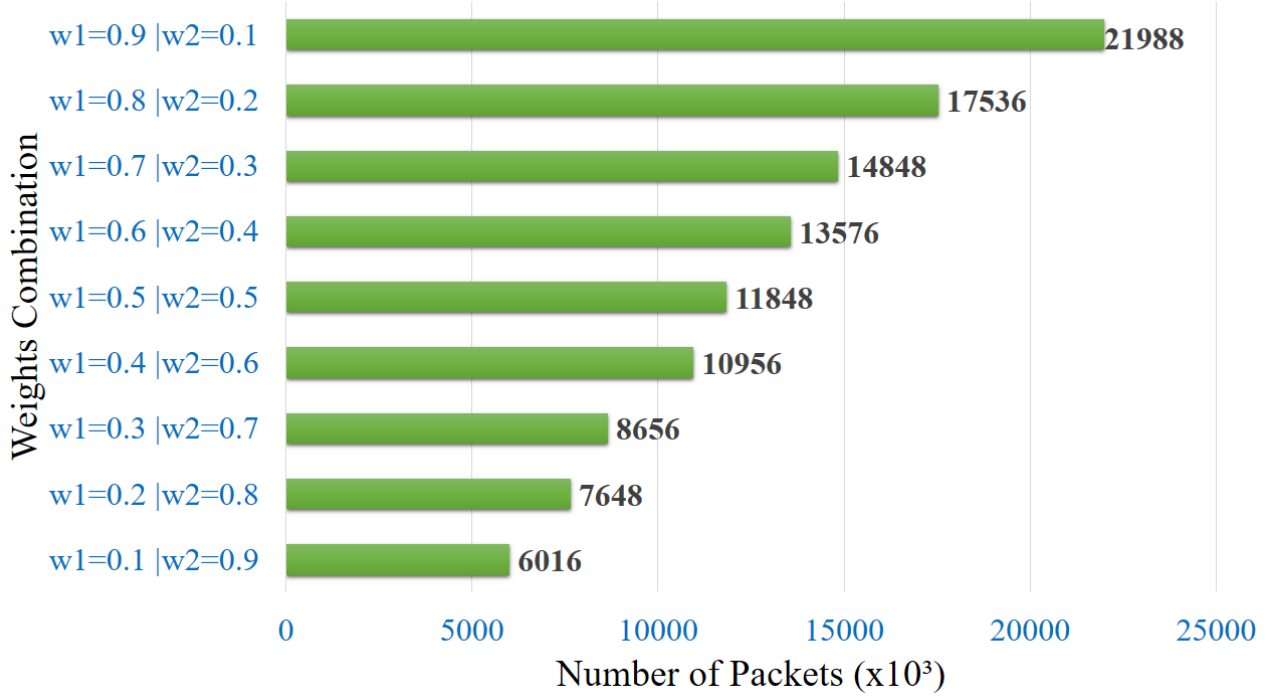


Figure 5.2: Evaluation of data delivery under various weight schemes with internal BS placement.

Despite these variations, the results indicate that both distance and energy parameters play essential yet complementary roles in sustaining communication efficiency. Distance optimization enhances throughput by forming short, efficient communication links and maintaining an appropriate number of CHs, while the energy component ensures fair participation and avoids premature node depletion.

In conclusion, when the BS is located inside the network, assigning greater weight to the distance factor (w_1) increases throughput due to shorter transmission paths and more effective CH placement. At the same time, incorporating the energy factor (w_2) maintains long term balance among nodes. Therefore, a moderate and well balanced configuration between the two objectives yields the most favorable performance in terms of data delivery and energy sustainability.

Figure 5.3 illustrates the variation of CFI with respect to the number of rounds for different weighting combinations (w_1, w_2) when the BS is positioned at the center of the sensing area. The CFI metric evaluates the uniformity of node deaths across the monitored region, where the field is divided into subregions of size $10\text{ m} \times 10\text{ m}$.

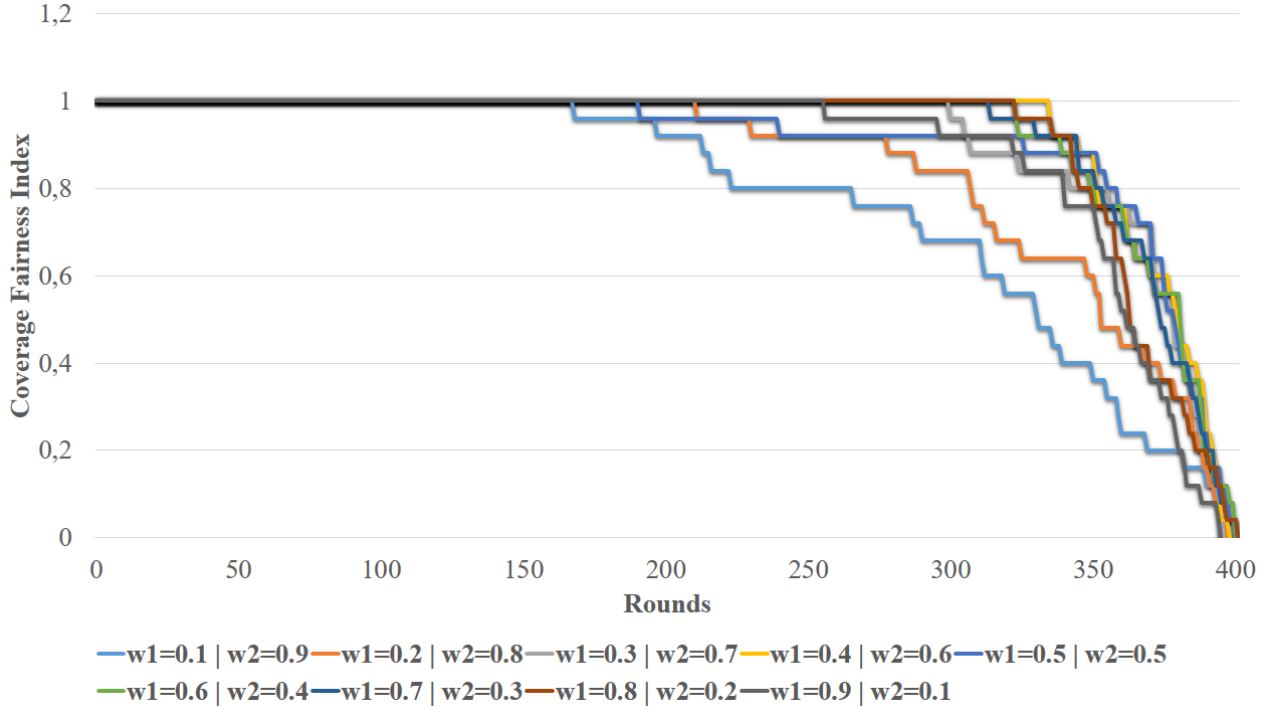


Figure 5.3: Effect of weight combinations on CFI with BS inside the network.

Initially, all configurations start with a CFI equal to 1, as all nodes are alive and uniformly distributed. As the number of rounds increases, the CFI gradually decreases, reflecting the progressive death of nodes in specific regions. The results show that configurations emphasizing the distance factor (e.g., $(w_1 = 0.9, w_2 = 0.1)$ and $(w_1 = 0.8, w_2 = 0.2)$) maintain a higher CFI over a longer duration. This behavior confirms that minimizing intra and inter cluster distances helps distribute the communication load more evenly, reducing early energy depletion in isolated areas.

In contrast, when the energy weighting factor w_2 dominates, the CFI declines more rapidly after approximately 150 rounds. Although these configurations extend the survival of certain high energy nodes, they create spatial energy imbalance, as distant or low energy nodes deplete earlier. This leads to localized node failures and coverage gaps in regions farther from the BS.

The balanced configuration $(w_1 = 0.6, w_2 = 0.4)$ achieves the best compromise, maintaining high fairness in node distribution until around 350 rounds, after which the CFI drops more gradually. This confirms that combining both distance and residual energy metrics in the clustering process leads to better spatial energy distribution and more homogeneous coverage degradation across the network.

In summary, when the BS is located inside the network, distance-aware configurations (high w_1 values) ensure smoother coverage decay, while energy-aware configurations (high w_2) delay individual node deaths but cause uneven coverage. A balanced weighting strategy main-

tains uniform node death progression and reduces the risk of early coverage holes, reinforcing the efficiency of the proposed PUMA-based clustering mechanism.

Since the FND, HND, and LND metrics were already examined earlier, figure 5.4 presents only the initial segment of the live-nodes evolution, where the slight variations between the weight combinations are most visible. The overall decrease in the number of active nodes is highly similar across all nine configurations, which indicates that the general energy consumption pattern remains almost identical regardless of the chosen (w_1, w_2) pair. Because the full curves follow nearly the same trajectory, only the first rounds are plotted to highlight the minor early differences.

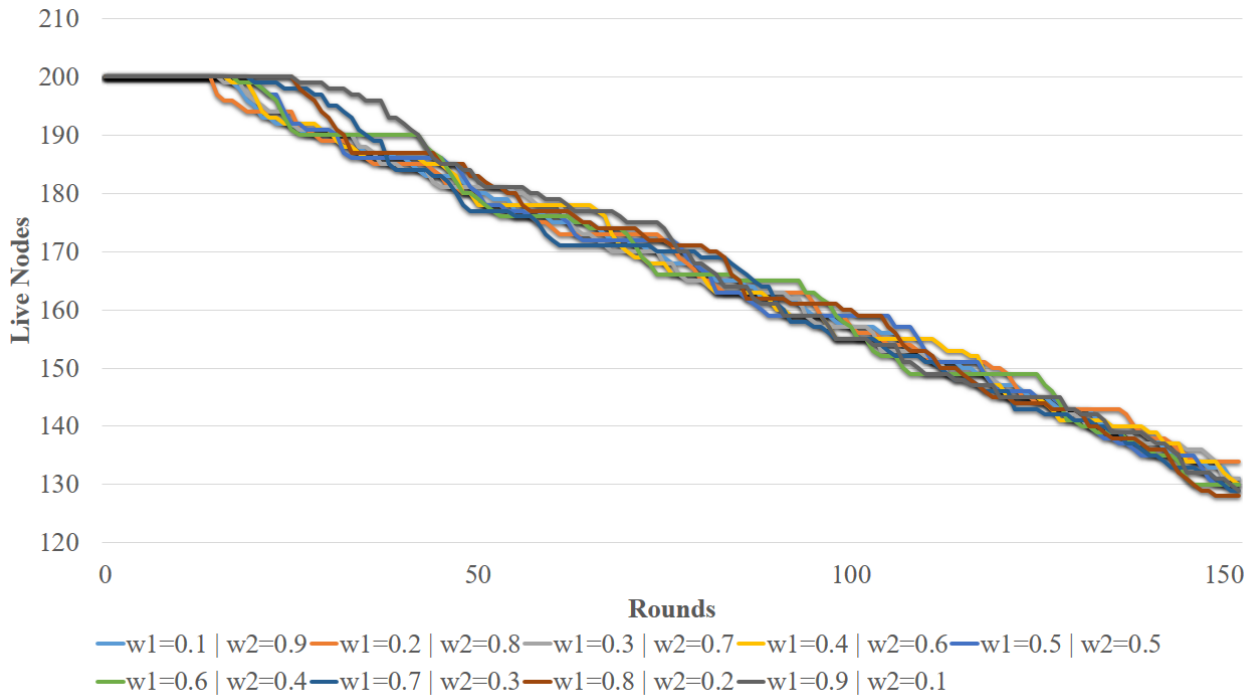


Figure 5.4: Effect of weight combinations on live nodes with BS inside the network.

A closer look at the curves shows that configurations with higher distance weight (w_1) experience a slightly slower decline in the early phase, consistent with their improved FND performance. In contrast, configurations with higher energy weight (w_2) exhibit a marginally faster reduction in the first rounds, reflecting their earlier FND values. Despite these small deviations, the live node profiles remain very close to one another, confirming that PUMA maintains balanced and uniform energy depletion for all tested weight combinations. This reinforces the earlier conclusion that PUMA is robust inside the network field and performs consistently well across all distance energy trade offs. Despite these small differences, the overall decrease remains almost identical for all nine weight configurations, indicating that the PUMA-based clustering mechanism achieves highly uniform energy consumption across the network.

The near parallel slopes of the curves confirm that the weighting strategy affects only the early round behavior, while the general rate of node depletion remains stable and consistent across all scenarios.

5.3.2 Impact of Weight Combinations on Different metrics (BS outside the Network)

Figure 5.5 illustrates the effect of the weighting parameters (w_1, w_2) on the FND, HND, and LND metrics when the BS is placed outside the sensing area.

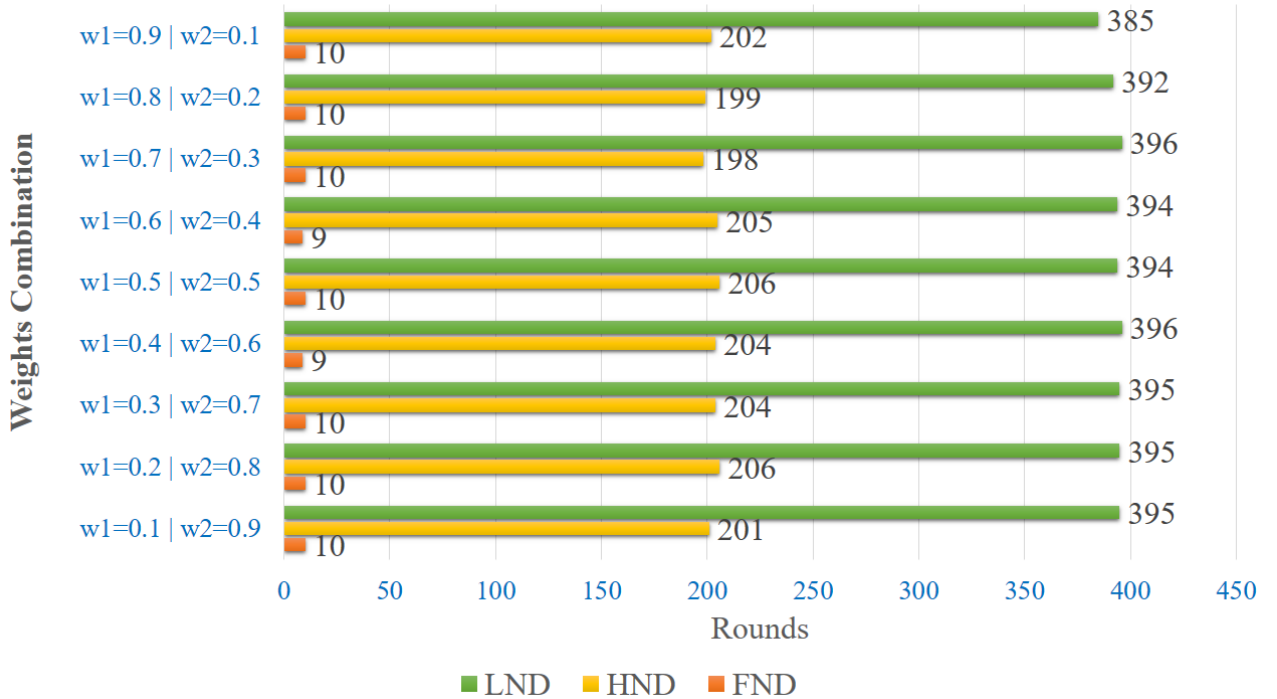


Figure 5.5: Impact of weight configurations on FND, HND, and LND with BS outside the network area.

In this scenario, CH must transmit over longer distances, which increases energy consumption and accelerates battery depletion. As a result, the relative influence of distance and energy weighting becomes more critical for achieving balanced performance. When the distance weight w_1 is dominant (e.g., $w_1 = 0.8-0.9$), the network demonstrates better early round stability, with slightly delayed FND values compared to energy focused configurations. This behavior arises because minimizing inter cluster and CH to BS distances reduces the transmission energy per round, helping postpone the first node death. However, placing excessive emphasis on w_1 creates uneven energy consumption among CHs, particularly those positioned farther from the BS, which must handle long range transmissions. As a result, the network deteriorates more rapidly in the later stages, and the HND and LND values remain nearly constant across all

distance heavy configurations. As the energy weight w_2 increases (e.g., $w_2 = 0.6$ – 0.9), CH selection shifts toward nodes with higher residual energy. This produces a more balanced energy distribution and improves fairness across the network. Nevertheless, this strategy does not extend the stability period, because many CHs still face the high energy cost of transmitting over long distances to reach the BS. Consequently, both FND and HND occur slightly earlier than in distance focused cases, as long range communication remains the dominant source of energy depletion. Despite these differences, the total network lifetime (LND) stays relatively stable, fluctuating around 395–400 rounds for all configurations. A moderate combination of distance and energy weights achieves the most balanced trade off, preserving reasonable stability (FND and HND) while maintaining fair energy consumption across nodes, even under the challenging condition of an external BS. In summary, when the BS is located outside the network, a moderate emphasis on distance-related parameters (w_1) is essential to mitigate the energy cost of long-range transmissions, while incorporating the energy factor (w_2) ensures fairness and prevents premature CH exhaustion.

Figure 5.6 presents the total number of packets successfully transmitted to the BS under different weight combinations (w_1, w_2) when the BS is positioned outside the network area.

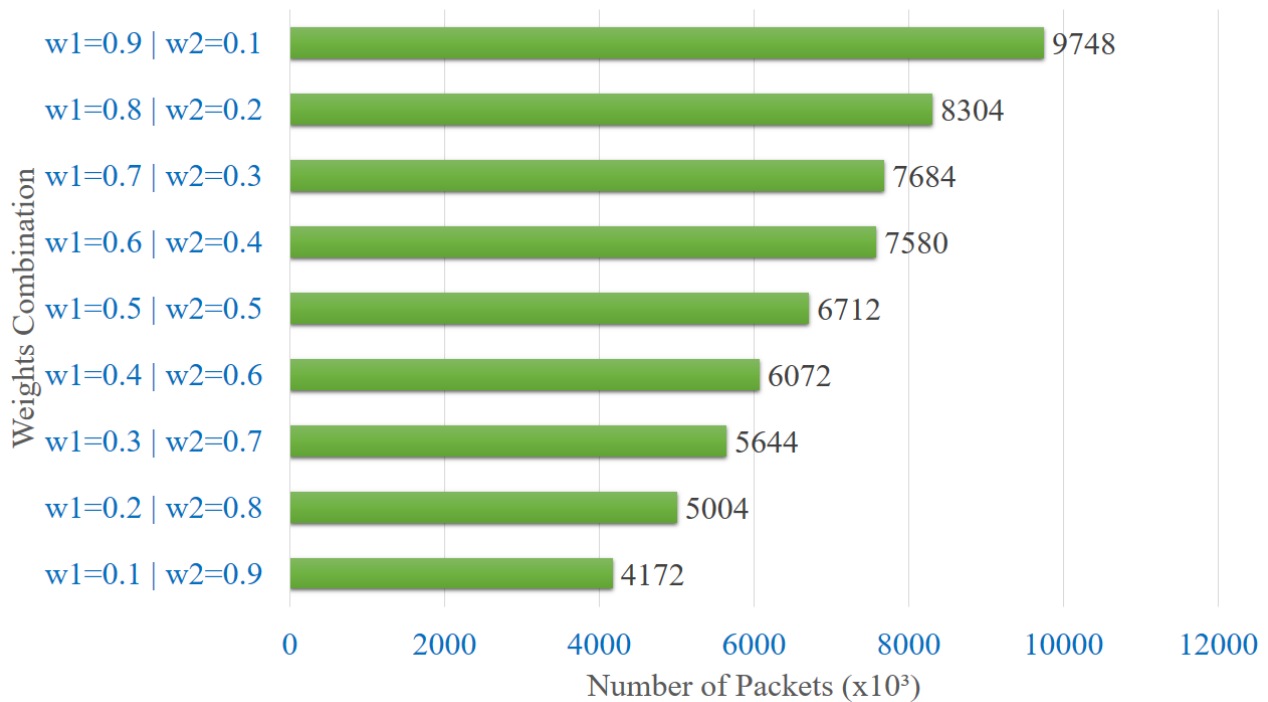


Figure 5.6: Evaluation of data delivery under various weight schemes with external BS placement.

The results show that increasing the importance of the distance factor (w_1) leads to a noticeable improvement in the number of packets delivered to the BS. The highest throughput

is achieved when $w_1 = 0.9$ and $w_2 = 0.1$. This occurs because a strong emphasis on minimizing transmission distances reduces the energy cost per packet and allows CHs to forward data more efficiently.

When the energy component w_2 is increased (e.g., $w_2 = 0.6$ – 0.9), the protocol prioritizes the selection of nodes with higher residual energy during CH formation. Although this approach improves energy fairness across the network, it reduces the total number of transmitted packets because some energy rich nodes may be positioned far from the BS or cluster members. These nodes require longer transmission distances, which increase communication cost and limit the total data forwarded to the BS. The lowest throughput is observed at $(w_1 = 0.1, w_2 = 0.9)$.

Overall, the results demonstrate that both parameters have complementary effects on throughput performance. The distance related factor (w_1) enhances network throughput by promoting shorter transmission paths and higher communication efficiency, whereas the energy factor (w_2) ensures a more balanced energy depletion pattern but slightly reduces packet forwarding intensity. When the BS is located outside the sensing area, higher emphasis on the distance factor is necessary to reduce transmission costs, but integrating the energy factor remains essential to preserve long term network balance.

Figure 5.7 shows the evolution of CFI over the simulation rounds for different weighting combinations (w_1, w_2) when the BS is positioned outside the sensing field.

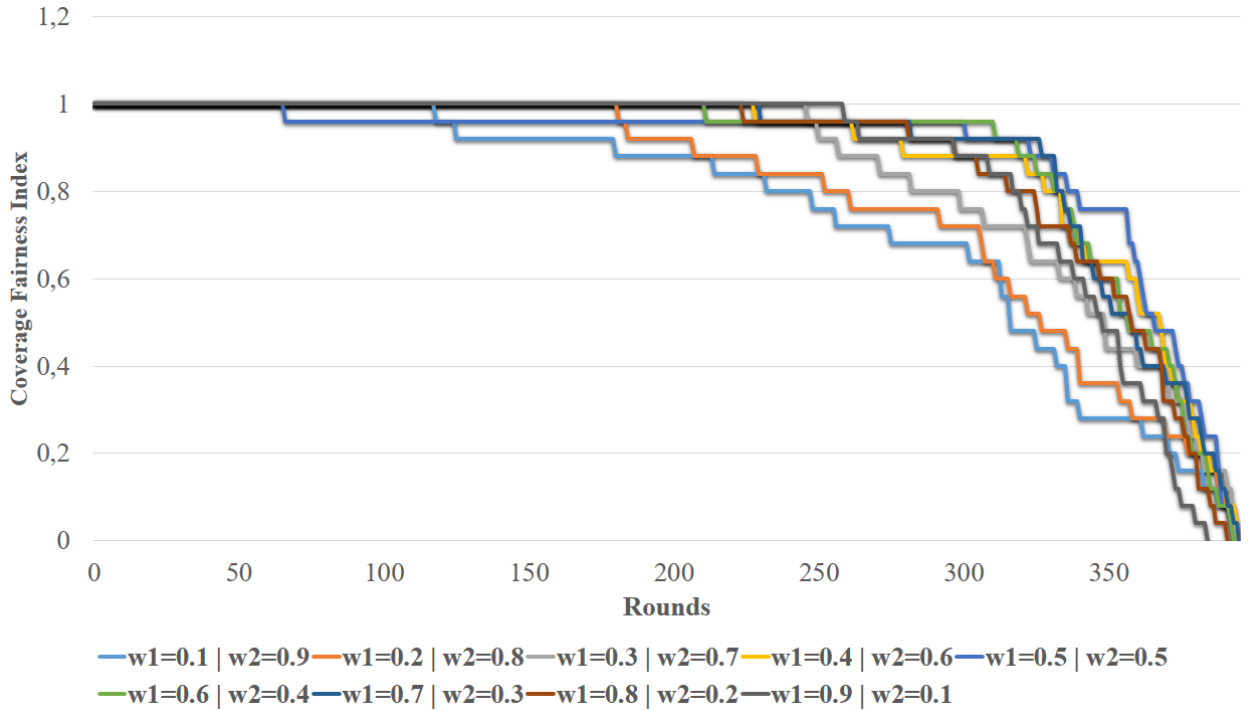


Figure 5.7: Effect of weight combinations on CFI with BS outside the network.

At the start of the simulation, all configurations maintain a CFI of 1, indicating equal

energy distribution across the network. However, as rounds progress, the CFI declines more rapidly. This faster decline results from the higher communication distances required for data transmission to the BS, which intensifies the energy disparity between near and far nodes.

Configurations that emphasize the distance component (e.g., $(w_1 = 0.9, w_2 = 0.1)$ and $(w_1 = 0.8, w_2 = 0.2)$) achieve relatively smoother CFI decline patterns, as minimizing intra and inter cluster distances mitigates excessive communication overhead. Nevertheless, the overall coverage fairness still deteriorates sooner than in previous BS placements due to unavoidable long range transmissions toward the BS.

In contrast, energy oriented combinations such as $(w_1 = 0.2, w_2 = 0.8)$ and $(w_1 = 0.1, w_2 = 0.9)$ exhibit a steeper CFI decline beginning around 150 rounds. These configurations tend to preserve high energy nodes, but the increased distance penalties associated with their spatial positions lead to uneven node depletion and region specific coverage gaps.

Among all configurations, the balanced weighting $(w_1 = 0.6, w_2 = 0.4)$ again provides the most stable CFI trajectory. It achieves gradual coverage degradation and maintains spatial fairness longer, effectively balancing distance minimization and energy preservation. This confirms that under harsh communication conditions, such as when the BS is located outside the network, balanced weighting enables the PUMA-based clustering scheme to sustain fairness and reduce premature regional coverage loss.

In summary, the outside BS scenario demonstrates that the increase in communication range accelerates energy heterogeneity and uneven node deaths. Although distance dominant configurations delay the first coverage holes, balanced weight settings remain the most effective strategy for achieving prolonged spatial uniformity and minimizing the rate of coverage decay.

Figure 5.8 illustrates the variation in the number of live sensor nodes over time for different weighting combinations of w_1 and w_2 when the BS is positioned outside the sensing field. As the simulation progresses, the number of active nodes steadily decreases due to the higher energy expenditure required for long distance transmissions to the external BS. Compared to the scenario where the BS is located inside the network, node deaths begin earlier, and the decline is noticeably steeper, indicating accelerated energy depletion. This behavior is expected, as the increased CH to BS communication distance significantly amplifies transmission energy losses. Although all weighting configurations follow a similar, nearly linear downward trend, slight differences are evident. Configurations that prioritize residual energy (e.g., $w_1 = 0.1, w_2 = 0.9$) retain a marginally higher number of live nodes during the early rounds. This indicates that energy aware CH selection partially mitigates the negative impact of the external BS by favoring nodes with sufficient energy reserves for demanding transmission tasks.

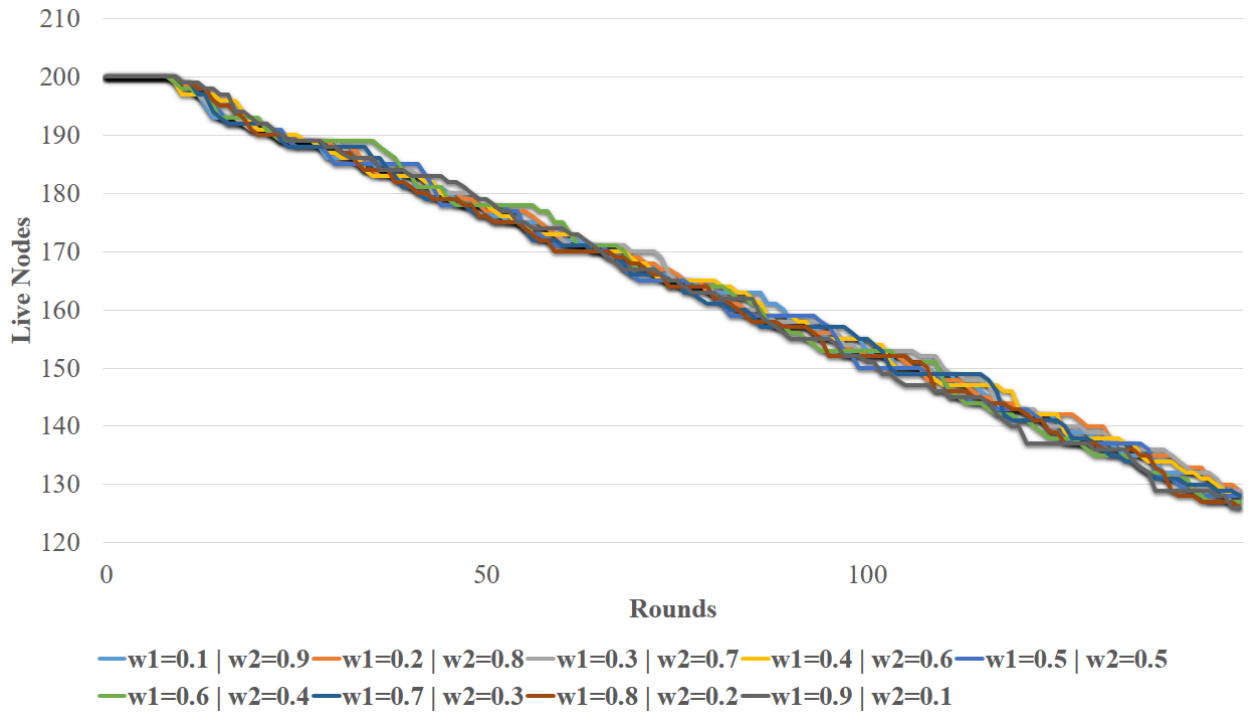


Figure 5.8: Effect of weight combinations live nodes with BS outside the network.

In contrast, distance-dominant configurations (e.g., $w_1 = 0.8$ or $w_1 = 0.9$) experience a faster reduction in active nodes, as many CHs are forced to transmit over long distances to reach the BS. The balanced case ($w_1 = w_2 = 0.5$) exhibits an intermediate behavior, offering a compromise between spatial optimization and energy preservation. Overall, assigning a higher weight to residual energy (w_2) helps counteract the increased communication distances, postpones early node deaths, and promotes more uniform energy consumption across the network. This highlights the importance of energy-aware clustering strategies in challenging deployment environments where the BS lies outside the monitored region.

5.3.3 Impact of Weight Combinations on Different metrics (BS at the edge the Network)

Figure 5.9 presents the impact of the weighting factors (w_1, w_2) on the FND, HND, and LND metrics when the BS is positioned at the edge of the sensing field.

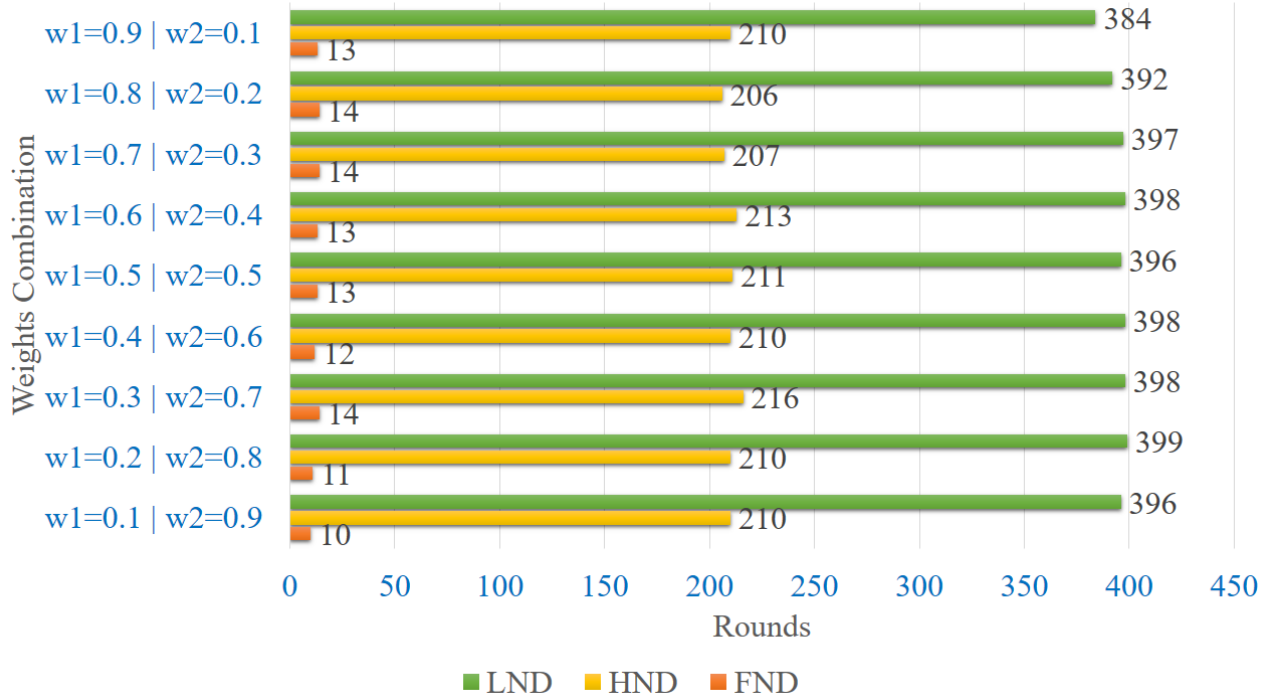


Figure 5.9: Impact of weight configurations on FND, HND, and LND with BS at the edge the network.

Placing the BS at the boundary creates a moderately challenging communication environment: some CHs are located close enough to transmit with low energy cost, while others remain relatively distant and must expend more energy to reach the BS. This configuration therefore represents an intermediate case between the BS inside and BS outside scenarios.

The results indicate that when the distance weight w_1 dominates—most notably for the configuration ($w_1 = 0.9, w_2 = 0.1$)—the network achieves a longer overall lifetime but exhibits a slightly shorter FND compared to energy focused configurations. Prioritizing distance minimization reduces the transmission cost for many CHs, allowing them to operate efficiently over time, though it does not fully prevent early individual node deaths.

As the weighting factors become more balanced, particularly around ($w_1 = 0.6, w_2 = 0.4$) and ($w_1 = 0.5, w_2 = 0.5$), both FND and HND improve. These balanced weight pairs provide an effective compromise between spatial efficiency and energy awareness, enabling CHs to conserve energy while maintaining reasonable transmission distances. This leads to more uniform energy depletion across the network. For energy dominant configurations such as ($w_1 = 0.2, w_2 = 0.8$)

and $(w_1 = 0.1, w_2 = 0.9)$, FND decreases slightly, while LND remains comparable to the other weight combinations. This shows that emphasizing residual energy alone does not significantly extend the total network lifetime in edge deployments, as distant CHs still incur higher transmission costs regardless of their initial energy level. Overall, the network maintains a stable lifetime across all weight configurations, with only minor variations in LND, demonstrating that the edge based BS placement produces consistent performance under a wide range of distance energy weighting strategies.

Figure 5.10 shows the variation in the total number of packets transmitted to the BS for different weight combinations (w_1, w_2) when the BS is positioned at the edge of the sensing area.

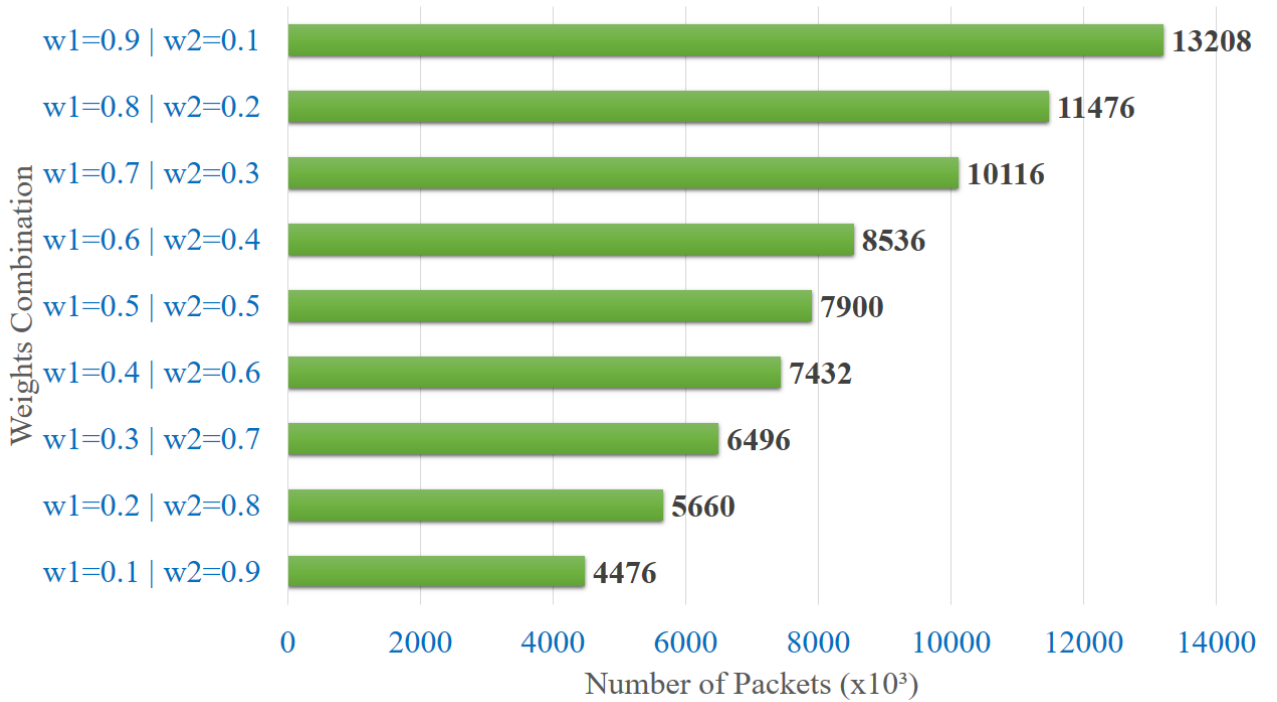


Figure 5.10: Evaluation of data delivery under various weight schemes with Edge BS placement.

In this topology, communication distances are moderate: CHs located near the BS benefit from short transmission paths, while those farther away must expend more energy. As a result, the energy consumption pattern falls between that of the “inside” and “outside” BS scenarios. The highest throughput is obtained for the configuration $(w_1 = 0.9, w_2 = 0.1)$. With a strong emphasis on distance minimization, the protocol selects CHs that are both spatially advantageous and produced in an appropriate number, allowing more packets to be successfully delivered before nodes deplete their energy. This occurs because communication overhead per transmission is reduced, enabling more rounds of stable data delivery.

When w_1 is slightly reduced and the energy term w_2 becomes more influential—such as in $(w_1 = 0.7, w_2 = 0.3)$ and $(w_1 = 0.6, w_2 = 0.4)$ —the number of transmitted packets remains relatively high while energy consumption becomes more balanced across nodes. These intermediate weight combinations achieve a favorable trade off between transmission efficiency and fairness, preventing excessive dependence on CHs located closer to the BS.

For higher w_2 values (e.g., $(w_1 = 0.3, w_2 = 0.7)$ and below), the total packet delivery decreases more sharply. In these energy dominant configurations, CHs are chosen primarily based on residual energy rather than spatial suitability. This may result in selecting CHs that are farther from the BS, increasing communication distances and per transmission energy costs. Additionally, the number of CHs selected becomes smaller and less optimal, further reducing data forwarding capacity and overall throughput.

Overall, placing the BS at the edge yields better throughput than the outside scenario, but results remain lower compared to the case where the BS is located inside the network.

Figure 5.11 depicts the evolution of the CFI for different weight combinations (w_1, w_2) when the BS is positioned at the edge of the sensing field.

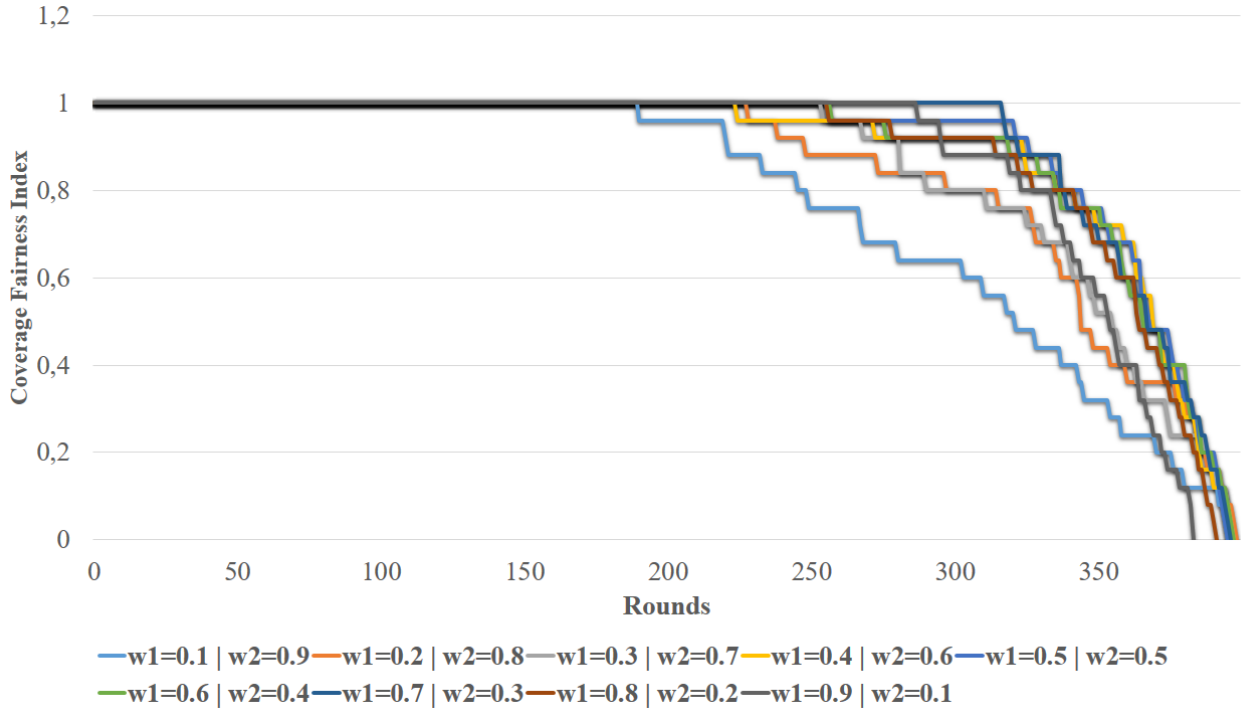


Figure 5.11: Effect of weight combinations on CFI with BS at the edge the network.

At the beginning of the simulation, all weight configurations maintain a CFI of 1, confirming complete coverage uniformity. As the rounds progress, differences between configurations emerge. The edge placement of the BS introduces moderate communication costs, where nodes and CHs closer to the BS consume less energy than those farther away. This spatial imbalance

leads to gradual declines in the CFI as coverage fairness deteriorates.

For distance-dominant configurations, such as $(w_1 = 0.9, w_2 = 0.1)$ and $(w_1 = 0.8, w_2 = 0.2)$, the CFI remains higher over more rounds compared to other combinations. This indicates that when intra and inter cluster distances are heavily prioritized, the clustering structure reduces excessive communication distances, allowing a smoother and more balanced depletion of energy among regions. These configurations therefore preserve fair coverage for a longer time before the first regional coverage holes appear.

In contrast, energy dominant combinations, such as $(w_1 = 0.2, w_2 = 0.8)$ and $(w_1 = 0.1, w_2 = 0.9)$, exhibit an earlier decline in CFI after approximately 150–200 rounds. This faster drop indicates that although nodes with higher residual energy are selected more often as CHs, distant nodes must still expend more energy to transmit data to the BS, leading to uneven depletion rates and spatial imbalance in node deaths.

The figure 5.12 illustrates the variation in the number of live sensor nodes with respect to the number of rounds for different combinations of weighting coefficients (w_1, w_2) , when the BS is placed at the edge of the sensing field.

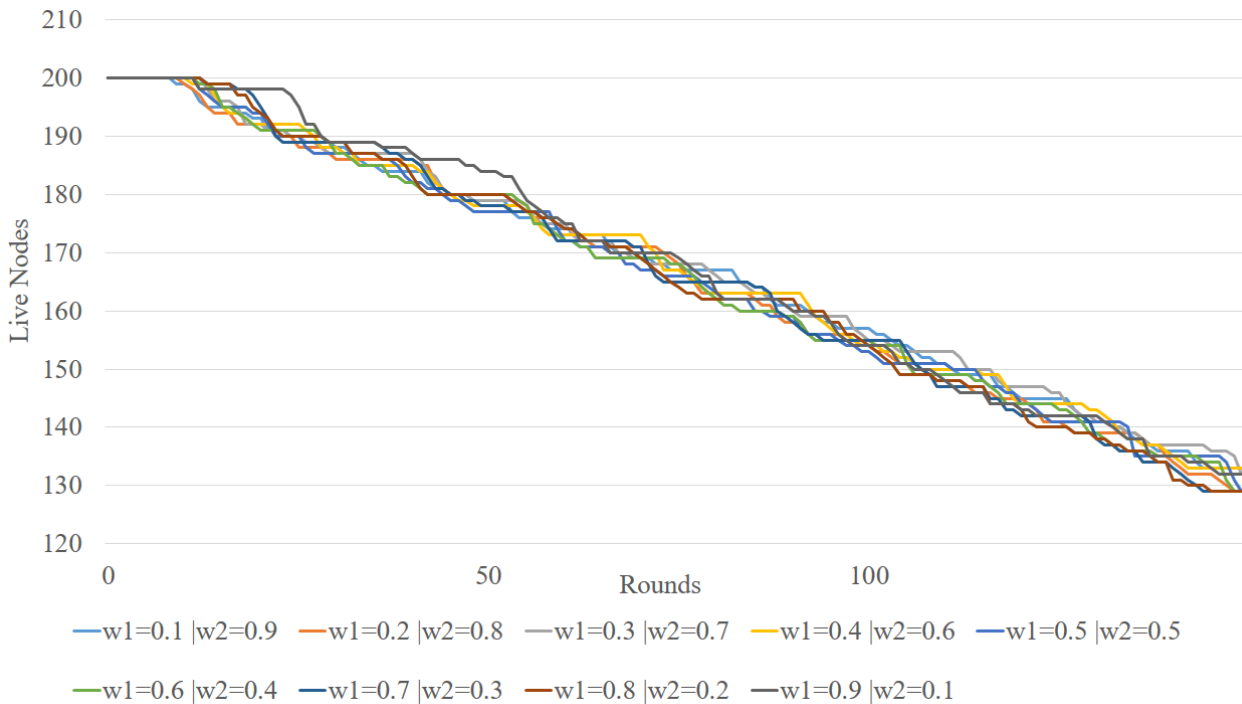


Figure 5.12: Effect of weight combinations live nodes with BS at the edge the network.

When the BS is positioned at the edge of the network, the influence of the weighting coefficients w_1 and w_2 on node survival becomes evident. Larger distance weights ($w_1 = 0.8$ – 0.9) produce a slower initial decline in live nodes and maintain a more stable reduction during the first 70–100 rounds. This is because prioritizing shorter intra cluster and CH–BS distances

lowers communication energy consumption and distributes the workload more evenly across the network. Increasing the energy weight ($w_2 = 0.9$) makes the CH selection process highly energy oriented, preventing low energy nodes from becoming CHs and slightly postponing the first node death while promoting balanced energy usage early on. However, deprioritizing distance occasionally results in CHs forming farther from the BS, leading to costly long range transmissions that accelerate energy depletion in later rounds. Thus, although high w_2 improves early stage stability, its advantage diminishes over time. Overall, in edge based deployments, giving more emphasis to distance (w_1) yields more sustained network stability, whereas energy focused weighting is beneficial mainly at the beginning but becomes less effective as rounds progress.

5.3.4 Sensitivity Analysis of the Weighted Fitness Function

To assess the robustness of the fitness formulation used in PUMA, a qualitative sensitivity analysis was conducted. The objective is to examine how small variations in the weighting coefficients w_1 (distance component) and w_2 (residual-energy component) influence network behavior across the three base-station placements. This complements the previous detailed evaluations by providing a global interpretation of how each metric reacts to incremental changes in the weights.

Sensitivity to the Distance Weight w_1 :

Increasing w_1 in small steps produces a noticeable improvement in metrics related to spatial efficiency, particularly the First Node Dies (FND) and the total number of packets delivered to the BS. Even slight increments such as $w_1 = 0.6 \rightarrow 0.7$ or $w_1 = 0.7 \rightarrow 0.8$ reduce average intra-cluster and CH-BS distances, lowering transmission energy per round. As a result, early-stage stability increases and throughput rises. However, when w_1 becomes too dominant, the effect becomes non-linear: spatially advantageous CH candidates are selected repeatedly, causing them to deplete their energy faster in the later rounds. This limits improvements in HND and LND and sometimes even degrades them. Overall, FND and packet delivery are the most sensitive metrics with respect to variations in w_1 .

Sensitivity to the Energy Weight w_2 :

Incremental increases in w_2 shift the CH selection toward nodes with higher residual energy. This adjustment immediately enhances mid life stability, reflected in higher HND values and a smoother decline in the number of live nodes. The energy component therefore supports more balanced usage of CHs and reduces early energy imbalance. However, when w_2 becomes too large (typically above 0.6), packet delivery decreases due to the selection of energy-rich nodes

located in spatially suboptimal regions. This increases communication distances and reduces throughput. HND and the CFI are the most sensitive metrics with respect to variations in w_2 .

Sensitivity Across BS Placements:

The three base-station placements respond differently to weight variations. When the BS is located inside the network, performance is most sensitive to changes in w_1 , since even small spatial improvements immediately reduce communication costs. In contrast, when the BS is located outside the sensing field, performance becomes more sensitive to changes in w_2 , as long-distance CH transmissions require nodes with sufficient energy reserves. The edge-based BS configuration exhibits intermediate sensitivity: both weights influence performance moderately, reflecting the mixed communication environment.

Global interpretation:

Across all scenarios, the fitness function remains stable for moderate weight values but becomes highly sensitive when either w_1 or w_2 dominates the decision process. The analysis confirms that:

- FND and throughput are most sensitive to spatial efficiency (w_1),
- HND and LND are most sensitive to residual energy (w_2),
- CFI is sensitive to both components, especially when one weight becomes extreme.

These observations justify the optimal moderate combinations identified earlier, such as $(w_1 = 0.6, w_2 = 0.4)$ for the BS inside the network, $(w_1 = 0.3, w_2 = 0.7)$ for the BS outside, and $(w_1 = 0.7, w_2 = 0.3)$ for the BS at the edge. Such balanced configurations maximize the complementary contributions of distance efficiency and residual-energy preservation.

5.3.5 Objective Function for Optimal Weight Selection

All three deployment scenarios, the weighting factors w_1 and w_2 influenced the evaluated metrics differently. Selecting the optimal combination is therefore objective dependent. For instance, if the goal is to maximize the number of packets successfully delivered to the BS, the combination $w_1 = 0.9$ and $w_2 = 0.1$ consistently provides the best performance in every scenario. In contrast, maximizing the Last Node Dies (LND) metric is more scenario dependent, as the optimal weights vary with the BS position and the resulting communication pattern. The same applies to maintaining high coverage, which is strongly shaped by network geometry and topology.

In this study, the overarching objective is to achieve strong performance across all metrics simultaneously: longer FND, HND, and LND values, a higher number of delivered packets, and sustained coverage throughout the network lifetime. To accomplish this, an objective

function was designed to combine all these metrics into a unified performance measure. Each metric was first normalized, and then the optimal weights were determined for each scenario to balance their contributions. The resulting unified objective function enables a comprehensive evaluation, aiming to maximize network lifetime indicators while also increasing data delivery and preserving coverage as nodes gradually deplete their energy. Each network lifetime metric is normalized using min–max normalization:

$$F\tilde{N}D_i = \frac{FND_i - \min(FND)}{\max(FND) - \min(FND)} \quad (5.5)$$

$$H\tilde{N}D_i = \frac{HND_i - \min(HND)}{\max(HND) - \min(HND)} \quad (5.6)$$

$$L\tilde{N}D_i = \frac{LND_i - \min(LND)}{\max(LND) - \min(LND)} \quad (5.7)$$

The number of packets sent and the CFI are also normalized using min–max normalization:

$$\tilde{P}_i = \frac{Packets_i - \min(Packets)}{\max(Packets) - \min(Packets)} \quad (5.8)$$

$$C\tilde{F}I_i = \frac{CFI_i - \min(CFI)}{\max(CFI) - \min(CFI)} \quad (5.9)$$

For the CFI metric, we consider the first round in which the cCFI drops below 1, representing the point where coverage begins to degrade due to node deaths. This ensures that the evaluation focuses on the onset of coverage imbalance rather than later rounds where collapse is expected. The overall objective function, combining all normalized metrics with weighted contributions, is expressed as:

$$Obj_i = 0.5 \cdot \text{avg}(F\tilde{N}D_i, H\tilde{N}D_i, L\tilde{N}D_i) + 0.25 \cdot \tilde{P}_i + 0.25 \cdot C\tilde{F}I_i \quad (5.10)$$

The weighting coefficients for the unified objective function were selected to ensure a balanced contribution between network lifetime indicators and communication reliability metrics. A higher weight (0.5) was assigned to the average of FND , HND , and LND because network lifetime represents the primary optimization objective in WSN routing and clustering problems. These metrics collectively reflect the stability, durability, and energy sustainability of the network. The remaining weights were equally distributed between packet-related performance (\tilde{P}_i) and the Cluster Fitness Indicator ($C\tilde{F}I_i$), with coefficients of 0.25 each, to maintain a compromise between data delivery efficiency and cluster quality without dominating the lifetime objective. The adopted weighting strategy follows the common practice in multi-objective

optimization for WSNs, where the principal objective receives a dominant coefficient while secondary objectives are incorporated with moderate influence to avoid bias toward a single performance criterion. Furthermore, preliminary empirical experiments showed that assigning a larger weight to the lifetime component improves the global stability of the network while preserving acceptable packet transmission performance and clustering efficiency.

The optimal weight set is determined as the configuration that maximizes the objective function. The results presented in Table 5.2 show how different combinations of the weighting coefficients w_1 (intra/inter-cluster distance) and w_2 (residual energy) affect the overall objective function under three distinct BS placements: inside, outside, and at the edge of the sensing area.

Table 5.2: Objective Function Values for Different Weight Combinations and BS Positions

Weights Combination	BS Inside	BS Outside	BS at the Edge
$w_1 = 0.1, w_2 = 0.9$	0.1414	0.5154	0.2
$w_1 = 0.2, w_2 = 0.8$	0.3517	0.7828	0.4246
$w_1 = 0.3, w_2 = 0.7$	0.6072	0.9095	0.7369
$w_1 = 0.4, w_2 = 0.6$	0.7664	0.7114	0.4394
$w_1 = 0.5, w_2 = 0.5$	0.3811	0.4697	0.6015
$w_1 = 0.6, w_2 = 0.4$	0.7207	0.6578	0.6610
$w_1 = 0.7, w_2 = 0.3$	0.7238	0.7582	0.8278
$w_1 = 0.8, w_2 = 0.2$	0.8250	0.7029	0.5154
$w_1 = 0.9, w_2 = 0.1$	0.4317	0.7500	0.5736

When the BS is located inside the network, the objective function increases steadily with higher values of w_1 , reaching its maximum of 0.825 at $(w_1 = 0.8, w_2 = 0.2)$. This indicates that, in this configuration, the intra- and inter-cluster distances play a more dominant role in improving performance, as communication costs are relatively low and balancing energy consumption becomes less critical.

For the BS placed outside the network, the optimal result of 0.9095 occurs at $(w_1 = 0.3, w_2 = 0.7)$. This highlights the importance of emphasizing residual energy in the fitness function when communication distances are longer. A higher weight on residual energy ensures that energy-rich nodes are more frequently selected as CH, thereby extending network lifetime.

In the case where the BS is positioned at the edge of the monitored region, the highest objective value (0.8278) is observed for $(w_1 = 0.7, w_2 = 0.3)$. This configuration represents a trade-off between minimizing transmission distance and maintaining sufficient residual energy across CHs. It suggests that when the BS is not centrally located, the clustering process benefits

from giving slightly more priority to distance while still considering energy balance.

Overall, these results demonstrate that the optimal weighting factors depend strongly on BS placement. Networks with centrally located BSs benefit from distance-focused optimization, whereas networks with peripheral or external BSs require greater emphasis on energy awareness to mitigate the effects of increased transmission range. The identified optimal weights were subsequently used in the final evaluation phase, where the PUMA-GRID protocol was compared against other clustering and routing schemes.

5.4 Comparison of Different Routing Protocols

To assess the effectiveness of the proposed PUMA-GRID protocol, its performance was evaluated against several widely used clustering and routing schemes, including LEACH, MR-LEACH, SA-LEACH, AEO-SH, AEO-GRID, and both single hop and grid-based implementations of PUMA. The evaluation relied on a comprehensive set of performance metrics that jointly describe network longevity and operational efficiency: FND, HND, and LND; the total number of packets successfully delivered to the BS; the evolution of the number of live nodes over time; the residual energy progression; and the CFI, which captures the spatial uniformity of active nodes throughout the simulation. All protocols were tested under three deployment scenarios, with the BS located inside, at the edge of, or outside the sensor field, allowing a clear analysis of their behavior under different communication constraints. The simulation parameters are presented in 5.3.

Table 5.3: Simulation parameters for comparing routing protocols.

Simulation Parameters	Values / Ranges
Network Size	$200 \times 200 \text{ m}^2$
BS Position	(0, 0), (100, 100) (250, 100)
Number of Nodes	300
Node's Initial Energy	0.5 J
Percentage of Clusterheads	5%
Packet Size	500 Bytes
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m ²
ϵ_{mp}	0.0013 pJ/bit/m ⁴
d_0	10 m
Grid Size	40 m

We selected a grid size of 40 m based on both empirical evaluation and structural consid-

erations of the 200×200 m sensing field. Experimental tests with grid dimensions of 20 m, 30 m, and 40 m resulted in maximum network lifetimes of approximately 355, 367, and 367 rounds, respectively. Since both 30 m and 40 m achieved the best performance, the 40 m grid was chosen for the comparative analysis because it offers the largest dimension that still maintains optimal stability while minimizing grid management overhead. This configuration keeps communication distances between adjacent CHs moderate preventing overly long transmissions while also supporting efficient and structured multi-hop routing. Consequently, the 40 m grid size provides a balanced trade off between spatial resolution, energy consumption, and overall routing performance.

5.4.1 Comparison of Different Routing Protocols when BS is inside the network

A comparison of FND, HND, and LND values across all routing protocols is presented in figure 5.13, highlighting how each method manages energy consumption and extends network lifetime when the BS is located inside the sensing field.



Figure 5.13: Comparison of FND, HND, and LND across different routing protocols with BS inside.

Traditional LEACH performs poorly in terms of FND, as its probabilistic and fully random CH selection leads to unstable cluster formation and early node deaths. MR-LEACH slightly improves this behavior: although it uses the same random CH selection, its multi-hop CH to

CH relaying reduces long range transmissions to the BS, resulting in marginally better FND and LND values. Nevertheless, both LEACH and MR-LEACH maintain only moderate HND and weak LND performance, indicating that routing enhancements alone cannot compensate for poor CH selection. SA-LEACH achieves the highest FND of all protocols, demonstrating excellent early round stability. This improvement stems from the application of SA, which replaces poorly chosen LEACH CHs with better positioned alternatives, resulting in a more energy balanced initial configuration. However, despite its strong beginning, SA-LEACH exhibits weaker HND and LND results. Once the first nodes die, the protocol struggles to maintain uniform energy distribution, leading to a rapid decline in network stability and premature lifetime termination. AEO-based protocols show a different behavior pattern. In AEO-SH, FND remains moderate because the algorithm tends to maintain the initially elected CH as long as it remains alive. However, AEO achieves very high HND performance, reflecting balanced mid stage energy usage, and its LND surpasses all LEACH-based approaches. AEO-GRID further amplifies these strengths: by enabling CHs to relay data through the nearest neighboring CHs, it shortens communication distances and preserves CH energy more effectively. As a result, AEO-GRID achieves some of the strongest HND and LND values in the BS-inside scenario. PUMA-SH and PUMA-GRID deliver the best overall results. PUMA-SH provides a balanced performance with moderate FND and strong HND, while its LND is significantly higher than that of traditional or AEO-SH methods. PUMA-GRID, strengthened by distance aware grid routing and PUMA's fitness based CH selection, achieves the longest LND among all protocols. The structured multi-hop routing grid k-NN reduces communication overhead and prevents energy hotspots, allowing the network to sustain connectivity far longer than competing approaches. Overall, the results show that while SA-LEACH excels in initial stability, it cannot maintain long-term energy balance. LEACH and MR-LEACH suffer from fundamental limitations due to random CH selection. AEO and PUMA protocols clearly outperform classical approaches, with grid based k-NN variants providing the most substantial improvements. Among all evaluated schemes, PUMA-GRID achieves the longest network lifetime, followed closely by AEO-GRID, confirming them as the most energy efficient and robust choices for BS inside deployments.

In Figure 5.14, where the BS is positioned inside the network, both LEACH and MR-LEACH exhibit low packet delivery performance, highlighting their inability to balance energy consumption and sustain long term communication. LEACH's fully probabilistic CH election and MR-LEACH's simple multi-hop extension lead to premature energy depletion among CHs, which limits overall throughput. SA-LEACH delivers the fewest packets of all protocols; al-

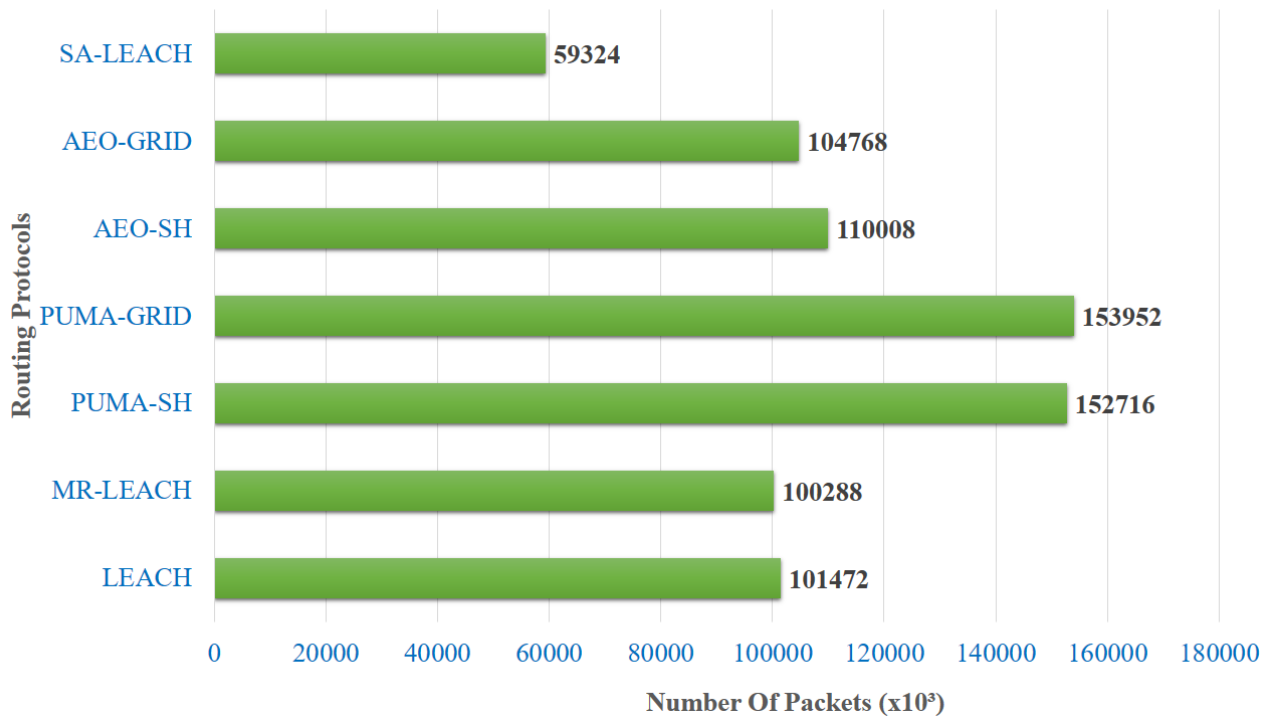


Figure 5.14: Comparison of the number of packets delivered to the BS across different routing protocols with BS inside.

though it achieves the highest FND and maintains excellent early round stability, the network deteriorates rapidly after the first node deaths. This abrupt collapse significantly shortens the effective operational lifetime, resulting in very limited packet delivery to the BS. AEO-SH and AEO-GRID perform better, with noticeable gains in packet delivery compared to LEACH, but their performance remains moderate and unable to match the more advanced designs. In contrast, the PUMA based approaches clearly dominate. Both PUMA-SH and PUMA-GRID deliver more than the number of packets compared to AEO and LEACH, with PUMA-GRID producing the highest values among all protocols. This emphasizes the advantage of combining PUMA's adaptive CH election with grid based K-NN multi-hop routing, which reduces energy consumption and ensures more balanced utilization of resources.

Figure 5.15, which shows the results with the BS located inside the network.

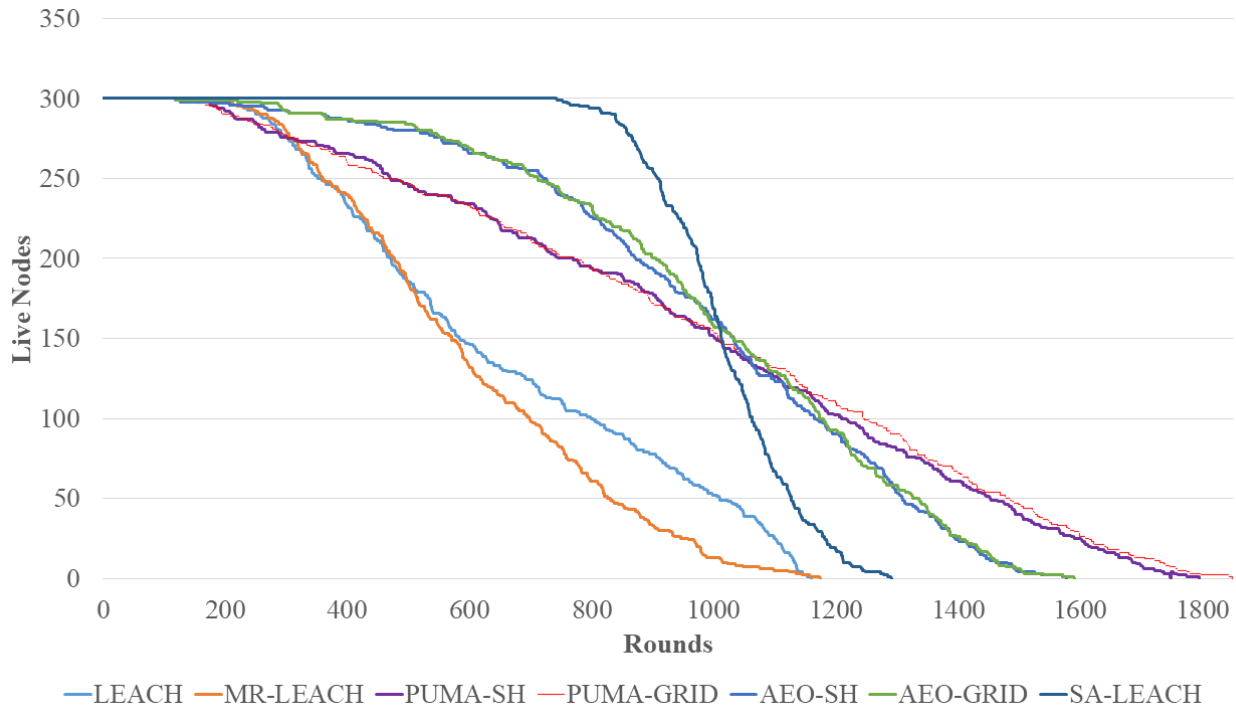


Figure 5.15: Comparison of the number of live nodes across different routing protocols with BS inside.

LEACH and MR-LEACH protocols exhibit very short lifetimes, with both the first and last nodes dying much earlier than in other protocols. This outcome is consistent with their limited energy awareness and reliance on probabilistic CH selection. SA-LEACH keeps all nodes alive the longest at the start thanks to improved CH selection, but once deaths begin, it collapses rapidly, giving it excellent early stability but poor mid- and late-stage lifetime. In contrast, the AEO protocols (both single hop and grid-based) extend the network life-time considerably, with the last node surviving much longer than in LEACH and MR-LEACH. However, while AEO demonstrates strong stability and balanced performance, the PUMA-based protocols, particularly PUMA-GRID, show the best performance overall. PUMA-GRID maintains live nodes for the longest duration, indicating that the combination of adaptive CH selection and grid-based k-NN routing significantly reduces energy imbalance and delays node deaths. PUMA-SH also performs strongly, maintaining a higher number of live nodes than AEO protocols, though it falls slightly behind PUMA-GRID in sustaining the final rounds of operation.

In Figure 5.16, where the BS is located inside the network, the residual energy trends highlight clear differences between the protocols.

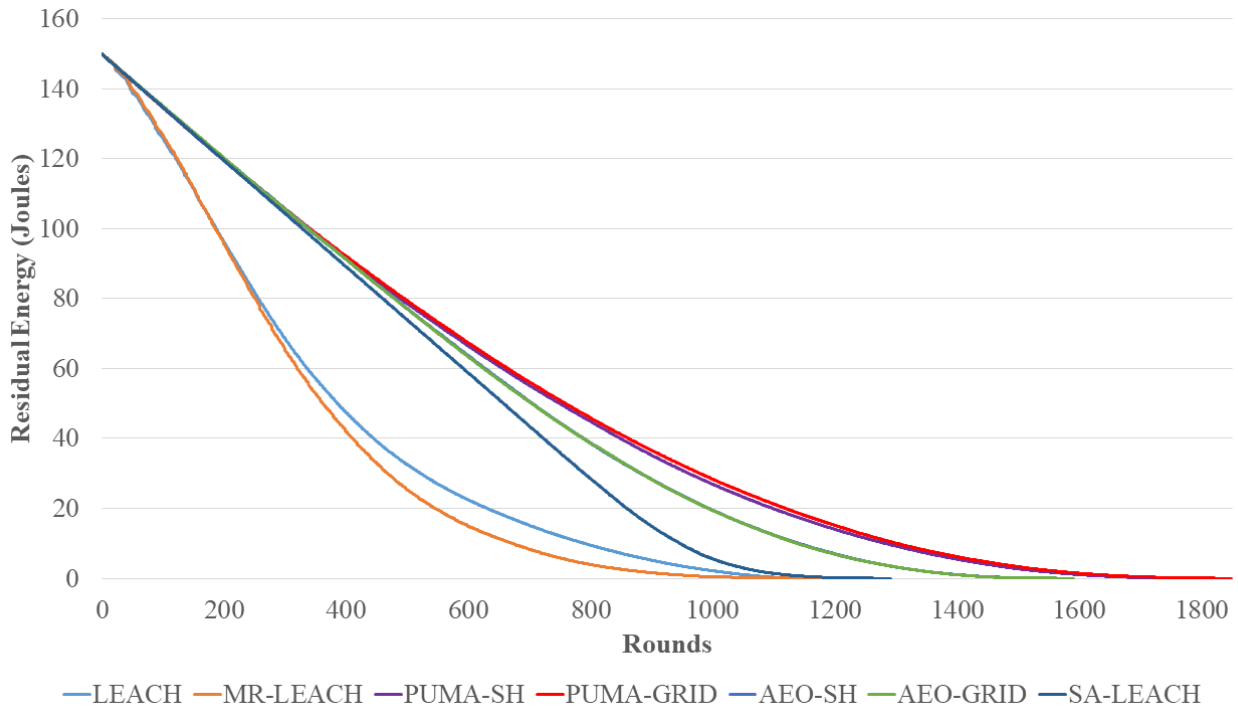


Figure 5.16: Residual energy comparison across different routing protocols with BS inside .

LEACH, MR-LEACH and SA-LEACH deplete their energy rapidly, confirming their limited capacity to distribute communication loads evenly across the network. Both protocols reach near zero energy in significantly fewer rounds, reflecting their vulnerability to hotspot issues and lack of energy aware clustering. With SA-LEACH surpassing them slightly in the beginning, it maintains full stability a bit longer before its performance drops rapidly afterward. In contrast, AEO-SH and AEO-GRID extend energy sustainability further, with nodes maintaining moderate reserves across more rounds. This outcome is consistent with their energy oriented cluster formation, which postpones full depletion. However, the best performance is observed in PUMA based protocols, especially PUMA-GRID, which conserve energy most effectively. The balanced incorporation of residual energy, intra cluster distance, and grid based K-NN routing mechanisms enables slower depletion, maintaining higher energy levels through later rounds. This indicates that PUMA's design succeeds in spreading energy consumption evenly while preventing premature exhaustion of CHs.

Figure 5.17 depicts the evolution of CFI for different Routing Protocols.

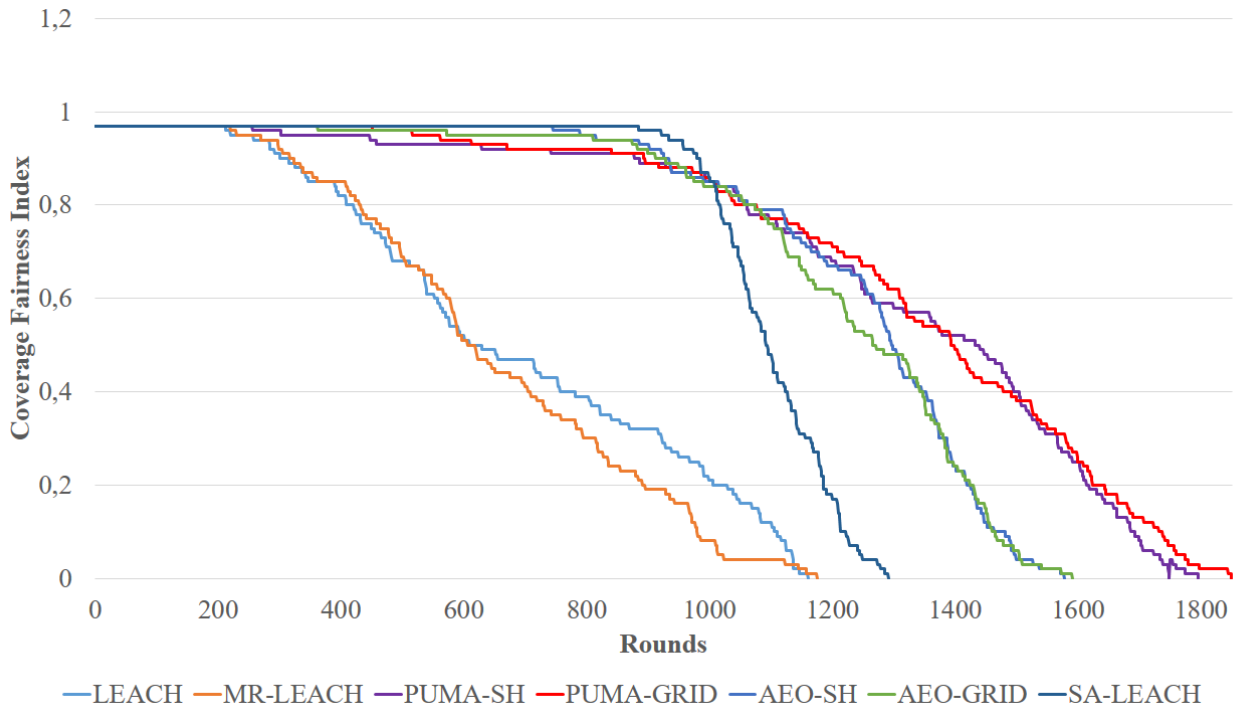


Figure 5.17: CFI comparison across different routing protocols with BS inside .

LEACH and MR-LEACH form the weakest group in terms of preserving coverage fairness. Their CFI drops early and rapidly, reflecting the instability of random CH selection. As nodes die unpredictably, spatial gaps appear quickly across the field, resulting in uneven coverage. Although MR-LEACH slows this decline slightly through multi-hop communication, it still inherits the fundamental randomness of LEACH.

SA-LEACH performs noticeably better during the initial rounds. By refining LEACH’s early CH selection using SA, it maintains full sensing coverage longer than both LEACH and MR-LEACH. However, this improvement is short lived. Once the earliest nodes begin to die, SA-LEACH’s CFI falls sharply. Its optimization enhances early performance but fails to sustain long term spatial balance, placing it between the basic LEACH type protocols and more advanced approaches.

AEO-SH and AEO-GRID form the next stronger tier. Their optimization-based CH selection distributes the sensing and communication load more evenly, helping maintain uniform coverage through the middle rounds. The grid based version extends this stability further; by routing through the nearest relay CH, it reduces local overload and postpones the appearance of coverage holes. Consequently, their CFI declines more gradually compared to SA-LEACH and other LEACH based variants.

The PUMA protocols, particularly PUMA-GRID, achieve the highest and most stable CFI values. Their fitness driven CH selection and structured grid k-NN routing ensure balanced

energy consumption across the network, preventing premature node depletion in any specific area. As a result, both PUMA versions sustain near perfect coverage for the longest duration, with PUMA-GRID providing the best spatial fairness among all evaluated methods.

5.4.2 Comparison of Different Routing Protocols when BS is outside the network

Figure 5.18 presents the FND, HND, and LND performance of all routing protocols when the BS is located outside the sensing field.

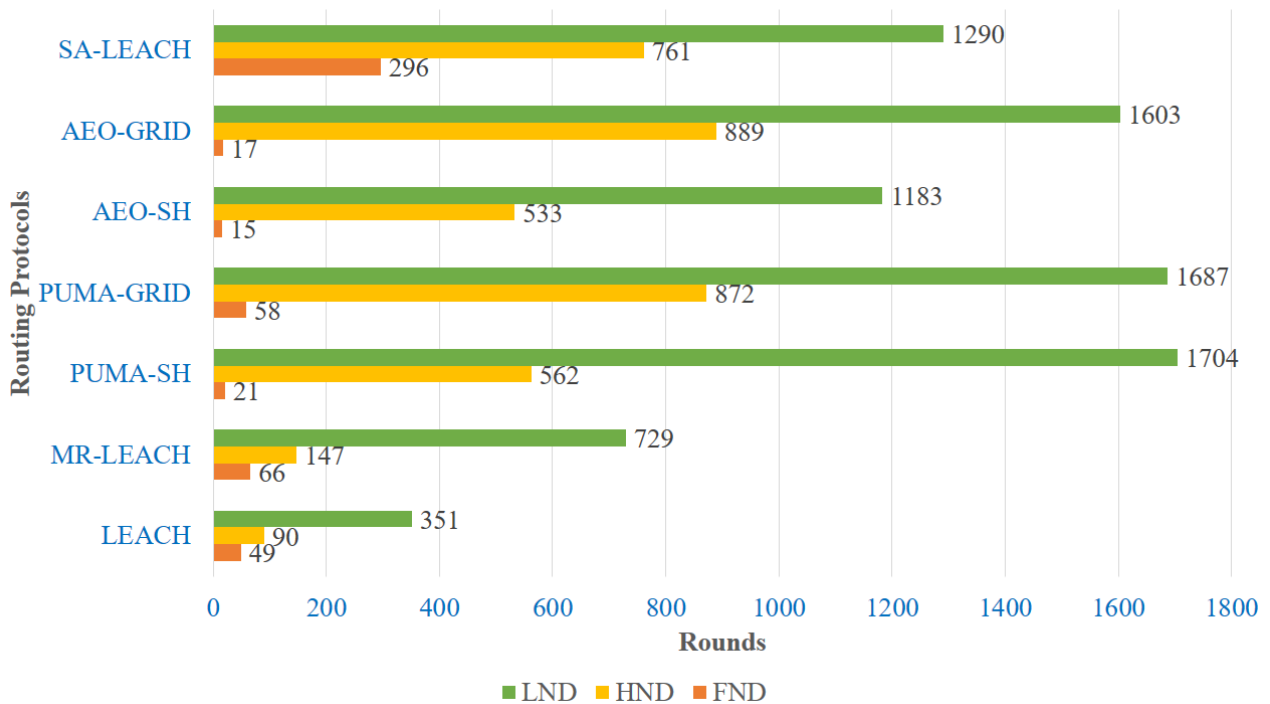


Figure 5.18: Comparison of FND, HND, and LND across different routing protocols with BS outside the network.

In this scenario, communication distances become significantly larger, making efficient clustering and energy aware routing essential for sustaining network lifetime. LEACH shows the weakest results across all metrics, mainly because its random CH election leads to poorly positioned CHs that must transmit over very long distances, rapidly exhausting their energy. MR-LEACH improves slightly over LEACH, as its multi-hop forwarding reduces transmission distances for CHs, allowing them to survive a bit longer, although its gains remain limited due to the inherited randomness in CH selection. SA-LEACH performs better than both LEACH based schemes; however, its FND improvement is not as pronounced as when the BS is at the center. The larger communication distances reduce the benefit of its early round optimization, but its moderate HND and LND values indicate that SA still manages to refine LEACH's CH

choices and distribute energy more evenly than the baseline methods.

AEO-SH performs well in this scenario, as the optimization mechanism behind AEO ensures more balanced CH placement and effective cluster formation, enhancing network stability. AEO-GRID further strengthens these advantages: by using grid based multi-hop routing, it significantly reduces transmission distances and energy consumption, resulting in HND and LND values up to 50% higher than those achieved by the single hop version.

PUMA-based protocols deliver the strongest overall performance. The weighted objective function and the PUMA metaheuristic produce well distributed CHs and highly efficient clustering, enabling substantial extensions of FND, HND, and LND. The grid based k-NN variant provides even greater improvements: grid routing nearly doubles the FND compared to PUMA-SH and increases HND by approximately 50%. However, its LND gain is less pronounced, as CHs located closer to the BS tend to become overloaded with forwarded traffic, creating hotspot regions that reduce late stage survivability.

Figure 5.19 shows the total number of packets delivered to the BS when the BS is located outside the sensing field.

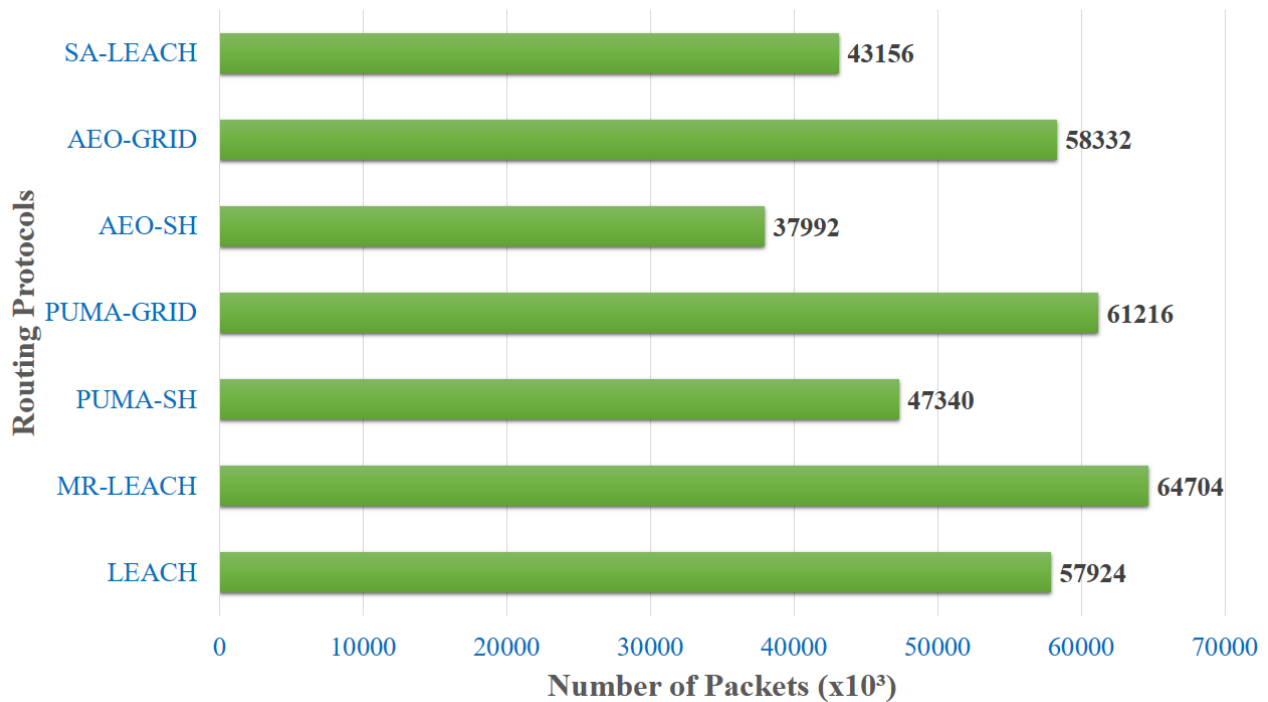


Figure 5.19: Comparison of the number of packets delivered to the BS across different routing protocols with BS outside the network .

In this scenario, packet delivery decreases for all protocols because transmitting over long distances requires significantly higher energy, causing nodes—especially CHs to deplete their batteries more quickly. LEACH and MR-LEACH still achieve relatively high throughput in the

early rounds, primarily because their CH selection mechanism produces many CHs regardless of their position. This results in a large number of packets during the initial phase, but at the cost of a much shorter network lifetime. MR-LEACH performs slightly better than LEACH, as its multi-hop forwarding reduces the transmission burden on CHs.

SA-LEACH also delivers a good number of packets: the SA step improves CH selection and reduces the inefficiencies inherited from LEACH, enabling the protocol to send more data while slightly extending network stability. In contrast, AEO-SH and PUMA-SH record the lowest number of packets. Although these protocols achieve long lifetimes, their strict optimization based on distance and residual energy results in fewer overall transmissions, since they prioritize energy preservation and balanced CH distribution over high packet generation.

Grid based versions of AEO and PUMA significantly increase packet delivery, improving performance by more than 50% compared to their single hop counterparts. By reducing long distance transmissions through structured multi-hop routing, the grid k-NN mechanism enables nodes to preserve energy while still sending large amounts of data, demonstrating the effectiveness of grid assisted routing in BS outside deployments.

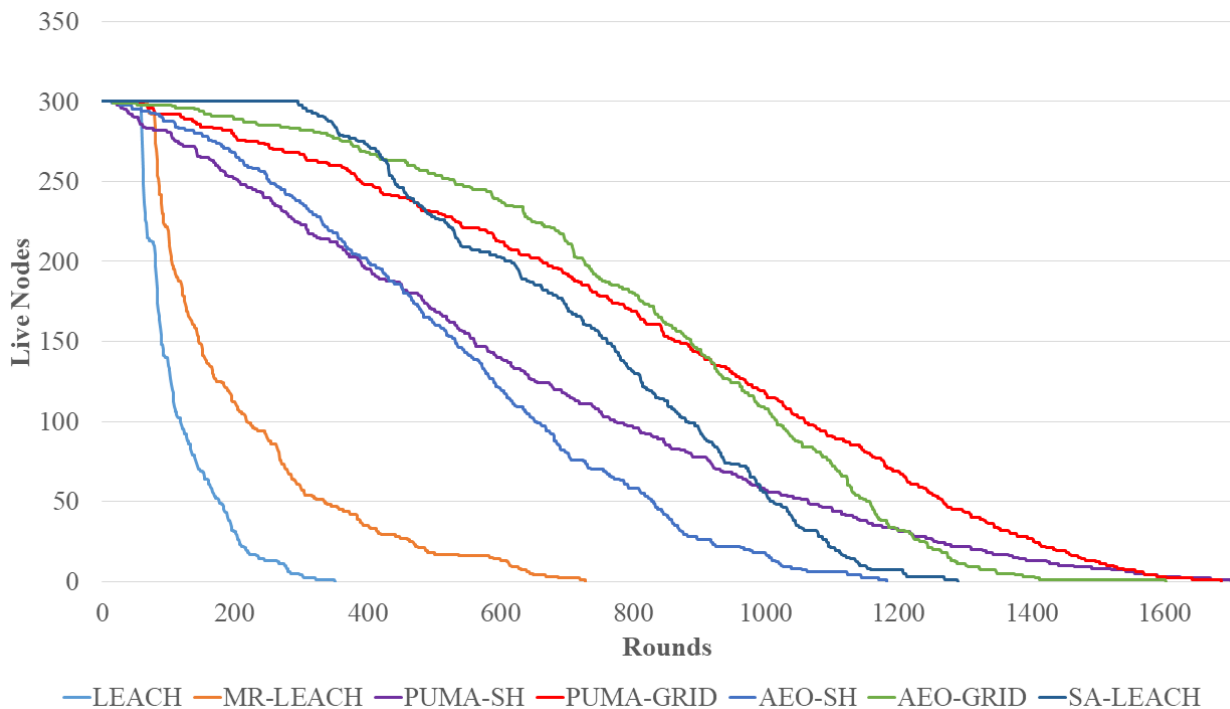


Figure 5.20: Comparison of the number of live nodes across different routing protocols with BS outside the network.

When the BS is located outside the sensing field, the long communication distances required for data transmission accelerate energy depletion dramatically, particularly for CHs operating under random or single hop schemes. Consequently, all protocols experience faster node death

compared to BS inside or BS edge settings. Figure 5.20 illustrates these trends clearly, showing the rapid decline in live nodes across the different routing strategies.

LEACH exhibits the earliest and steepest degradation, with the first nodes dying extremely early and the entire network collapsing soon afterward. Its lack of intelligent CH selection, combined with long single hop transmissions to an external BS, makes it the weakest performer in this scenario. MR-LEACH also performs poorly: although it introduces multi-hop routing, its CH selection remains random and produces highly unbalanced communication loads, resulting in a similarly steep decline. PUMA-SH and AEO-SH offer moderate improvements. Their optimized CH selection delays the first node deaths relative to classical protocols; however, because they rely on single hop transmission to an external BS, CHs located far from the BS are forced to expend very high energy to transmit data over long distances. This heavy transmission cost accelerates their depletion, causing a steady decline and earlier network termination compared to their grid based counterparts. PUMA-GRID achieves the best performance of all protocols in this deployment. Its decline is slow and uniform, maintaining a large number of live nodes for the longest duration. The combination of PUMA's balanced CH selection and grid based k-NN multi-hop routing enables efficient, distributed energy usage, even under the strain of long distance communication. SA-LEACH displays a distinctive behavior: it maintains excellent early round stability, delaying the first node death more effectively than most protocols. However, its advantage diminishes rapidly as the rounds progress. Although it initially surpasses both PUMA-SH and AEO-SH, it cannot match the robustness of grid based methods, which are better suited for handling external BS communication.

Figure 5.21 shows the residual energy decay of all routing protocols when the BS is placed outside the sensing field. In this configuration, long distance transmissions dominate energy consumption, especially for CHs forced to communicate directly with the external BS. LEACH exhibits the steepest and fastest energy decline. Its randomly selected CHs often end up far from the BS and must transmit over long distances, consuming large amounts of energy per round. This leads to an early and abrupt depletion of the total energy of the network. MR-LEACH performs only slightly better: although multi-hop forwarding reduces the CH to BS distance, the randomness in CH selection still causes unbalanced cluster formation. Energy drains rapidly in overloaded regions, resulting in one of the shortest lifetimes among all protocols. SA-LEACH initially conserves energy better than LEACH and MR-LEACH because of improved CH selection through SA. This produces a smoother, slower drop in residual energy during the early rounds. However, once long range CH transmissions accumulate, the advantage of the protocol fades away. Its energy curve drops sharply in the mid phase, showing that early

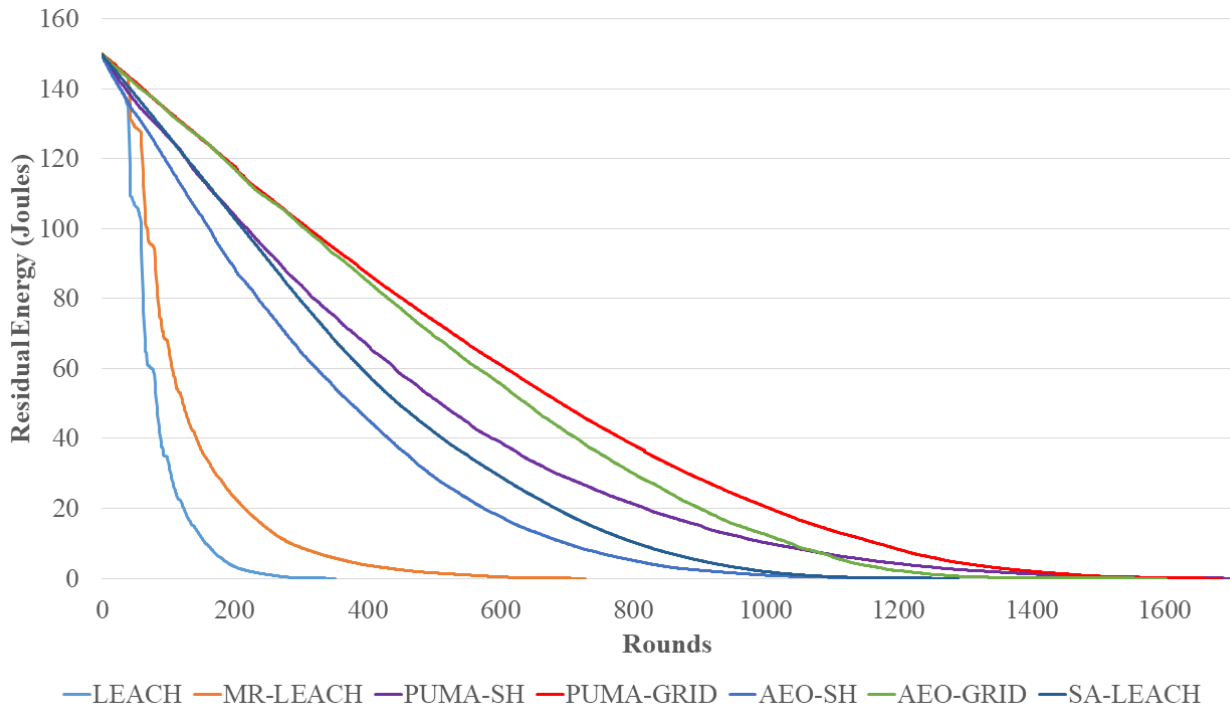


Figure 5.21: Residual energy comparison across different routing protocols with BS outside the network .

optimization does not guarantee long term balance when the BS is far away. AEO-SH maintains energy more efficiently than all LEACH based methods thanks to AEO’s optimization mechanism, which selects balanced CH positions that minimize unnecessary long transmissions. Still, the single hop nature of AEO-SH forces far CHs to consume significant energy, creating a noticeable decline in later rounds. AEO-GRID delivers a stronger performance: multi-hop grid routing distributes the transmission load, significantly reducing the energy required for long distance communication. This results in a more gradual decay and noticeably extended energy availability compared to the single hop version. PUMA-SH performs well, showing slower energy depletion than AEO-SH due to PUMA’s fitness based CH selection, which effectively balances energy and distance. However, it still suffers from the limitations of single hop transmission toward an external BS. PUMA-GRID achieves the best residual energy performance of all protocols. Its energy curve declines the slowest, remaining consistently above all other approaches. The combination of balanced CH selection and grid based k-NN multi-hop routing minimizes long range transmission costs and evenly distributes the energy load across the network. This allows nodes to retain energy for far longer, even under the demanding external BS setting.

Figure 5.22 illustrates the variation of CFI with respect to the number of rounds for different routing protocols when the BS is placed outside the sensing field.

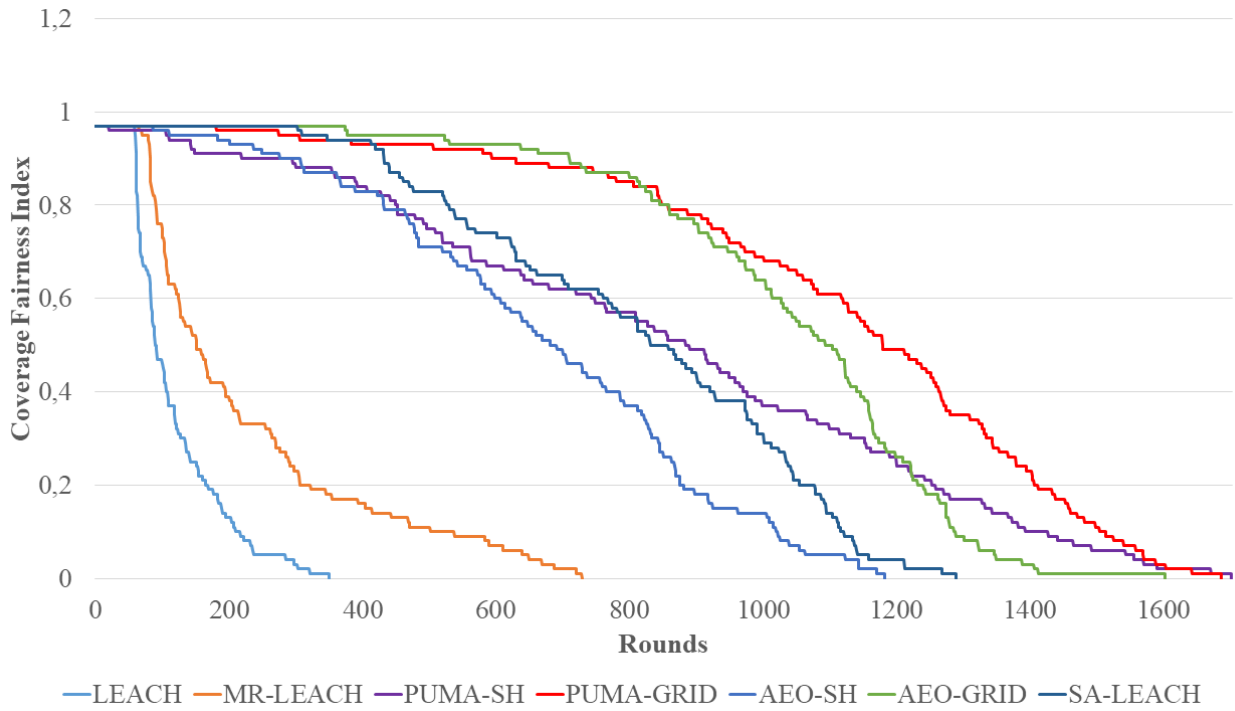


Figure 5.22: CFI comparison across different routing protocols with BS outside of the network .

LEACH and MR-LEACH lose coverage the earliest. Their CH selection causes nodes in distant regions to die quickly, creating coverage gaps almost immediately. MR-LEACH slows this drop slightly, but the improvement is limited. SA-LEACH maintains full coverage a bit longer due to improved early CH selection, but once node deaths begin, its CFI declines sharply. Its early advantage does not translate into sustained coverage under long-distance communication. AEO-SH and AEO-GRID provide more stable coverage. AEO-SH delays the decline, while AEO-GRID goes further by reducing long transmission distances through structured multi-hop routing, resulting in a smoother, slower CFI fall. PUMA-SH and especially PUMA-GRID maintain the most uniform coverage. PUMA's balanced CH selection distributes the load effectively, and grid k-NN routing minimizes long distance transmissions, allowing PUMA-GRID to preserve coverage fairness longer than all other protocols.

5.4.3 Comparison of Different Routing Protocols when BS is at the edge of the network

When the BS is placed at the edge of the sensing field, all protocols experience moderate communication costs higher than the BS inside case but not as severe as when the BS is completely outside. Figure 5.23 illustrates the FND, HND, and LND values across the different routing protocols.

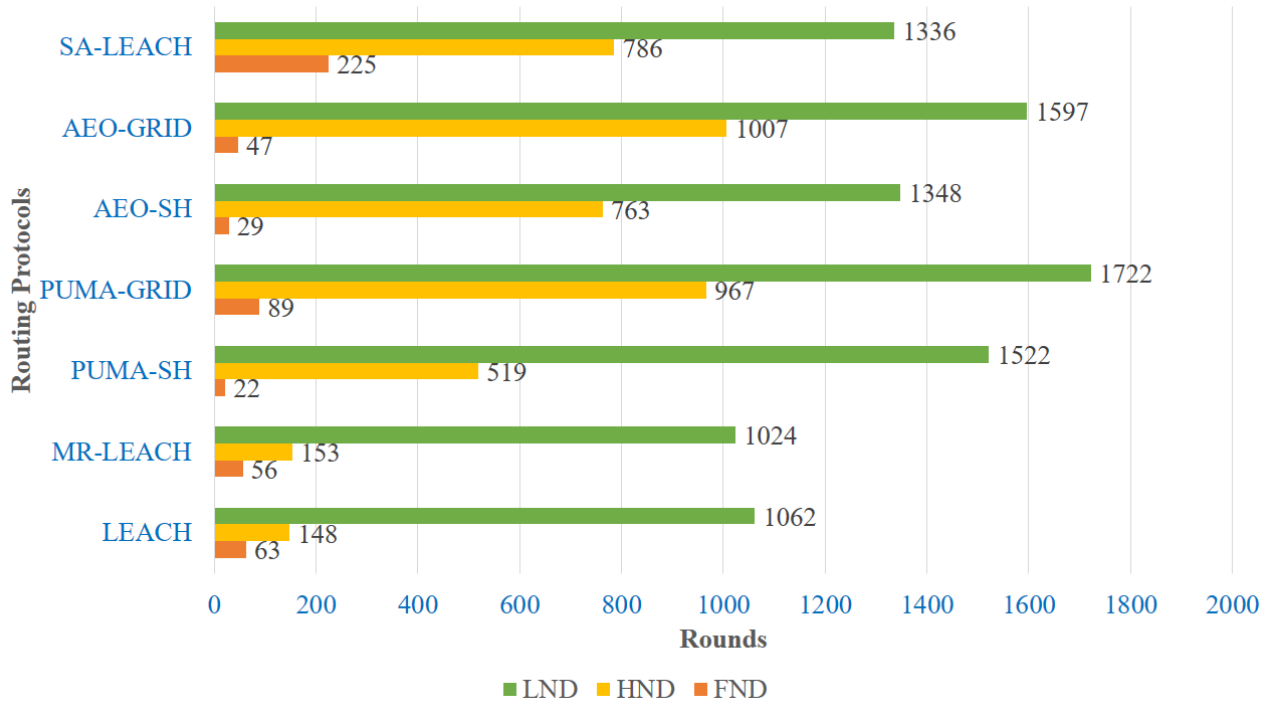


Figure 5.23: Comparison of FND, HND, and LND across different routing protocols with BS at the edge of the network.

LEACH and MR-LEACH show limited performance, with early FND and relatively weak HND/LND due to unbalanced CH selection. SA-LEACH achieves a very strong FND but its improvement does not extend fully into HND and LND. AEO-SH and AEO-GRID perform noticeably better, with AEO-GRID benefiting from shorter multi-hop transmissions, resulting in higher HND and LND. PUMA-SH and PUMA-GRID achieve the best overall lifetimes. PUMA-SH shows strong FND and HND, while PUMA-GRID delivers the highest LND among all protocols, confirming that combining PUMA’s optimized CH selection with grid based k-NN routing provides the most stable and long lasting operation near the network edge.

Figure 5.24 shows the total number of packets delivered to the BS when it is placed at the edge of the sensing field. In this configuration, protocols with more balanced clustering and shorter routing paths naturally achieve higher throughput.

LEACH and MR-LEACH deliver a moderate number of packets, limited by their random CH selection and uneven load distribution. SA-LEACH improves slightly over them but remains constrained by its rapid decline after the early stable phase. AEO-SH and AEO-GRID achieve significantly higher packet delivery, with the grid version benefiting from shorter multi-hop transmissions that reduce CH energy consumption. PUMA-SH and PUMA-GRID provide the highest throughput overall, with PUMA-GRID delivering the maximum number of packets. This highlights the effectiveness of combining optimized CH selection with grid based k-NN

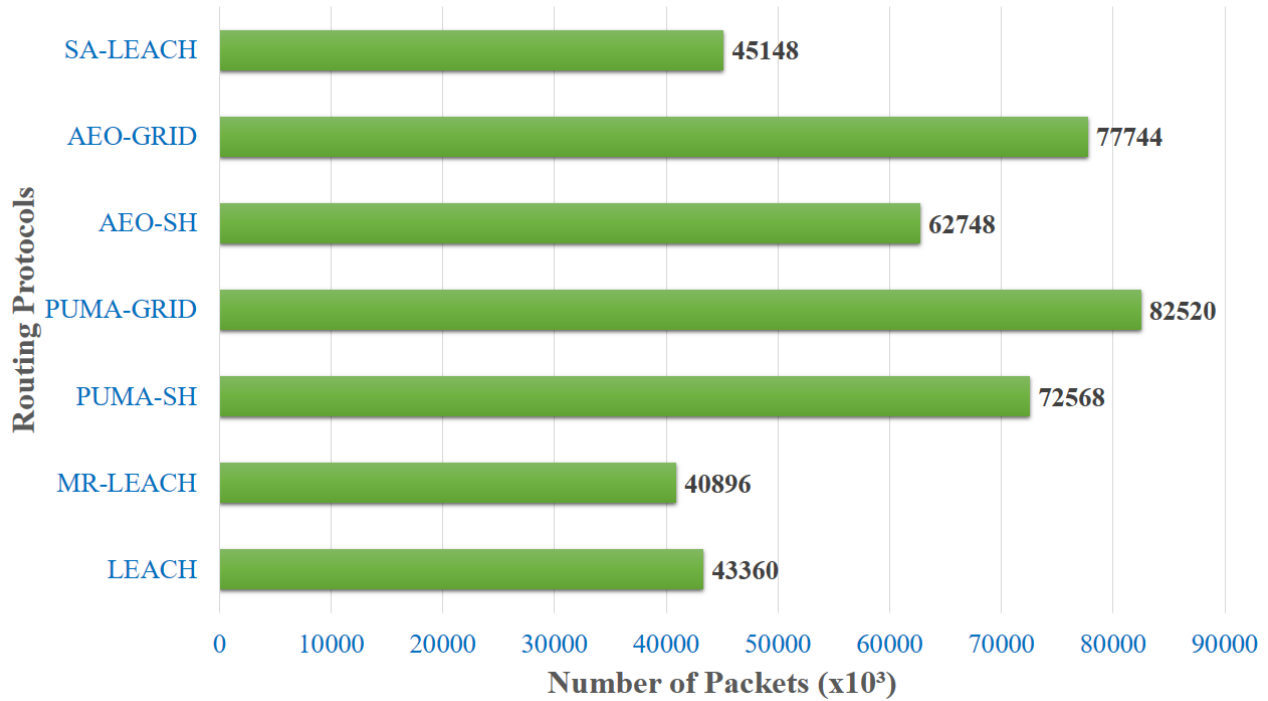


Figure 5.24: Comparison of the number of packets delivered to the BS across different routing protocols with BS at the edge.

routing for improved communication efficiency near the network edge.

Figure 5.12 shows how the number of live nodes decreases over time when the BS is placed at the edge of the sensing field. In this setting, node death occurs faster than in the BS inside case but remains more gradual than when the BS is fully outside. LEACH and MR-LEACH experience the steepest decline, with nodes dying rapidly due to random CH placement and unbalanced energy load. SA-LEACH delays the first few deaths but then drops sharply, showing that its early optimization does not sustain long term balance. AEO-SH and AEO-GRID exhibit smoother decay curves; the grid version extends node survival longer thanks to shorter multi-hop transmissions. PUMA-SH maintains a more gradual decline than AEO-SH, while PUMA-GRID achieves the slowest and most uniform decrease, keeping nodes alive significantly longer than all other protocols. Overall, the comparison shows that the grid based k-NN versions decline slower, optimization based protocols decline moderately, and LEACH type protocols decline the fastest under edge based communication.

Figure 5.26 shows the residual energy evolution when the BS is positioned at the edge of the field.

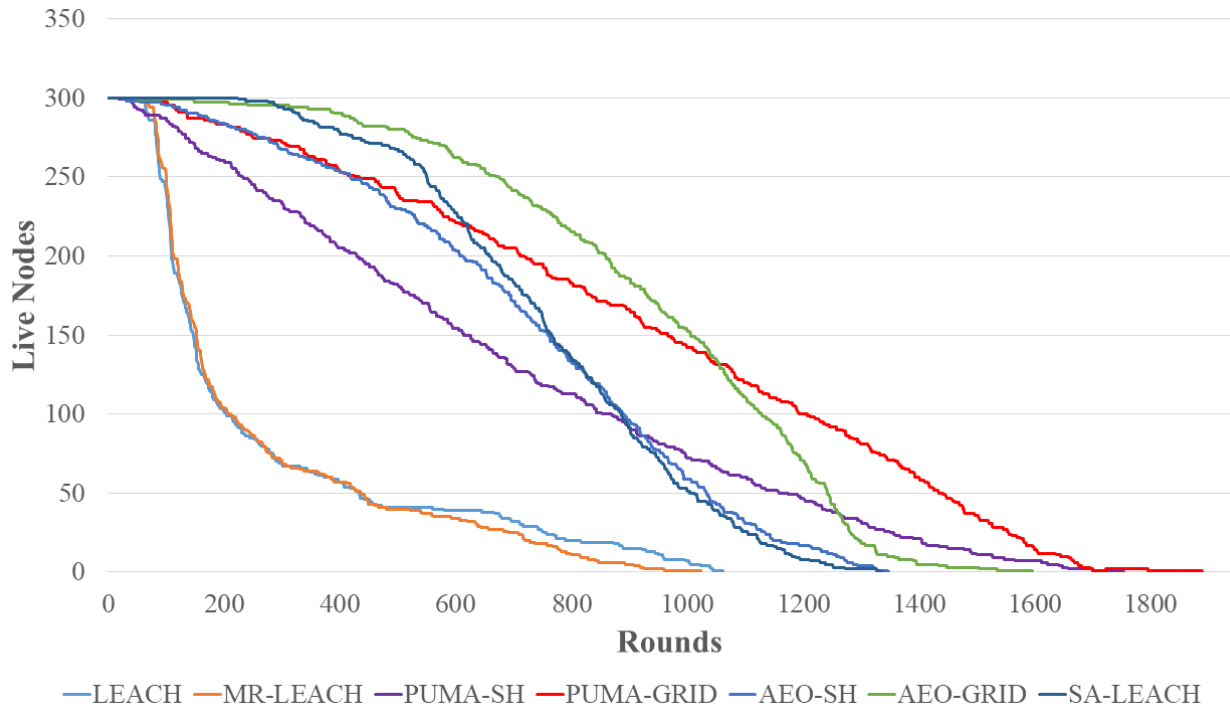


Figure 5.25: Comparison of the number of live nodes across different routing protocols with BS at the edge of the network.

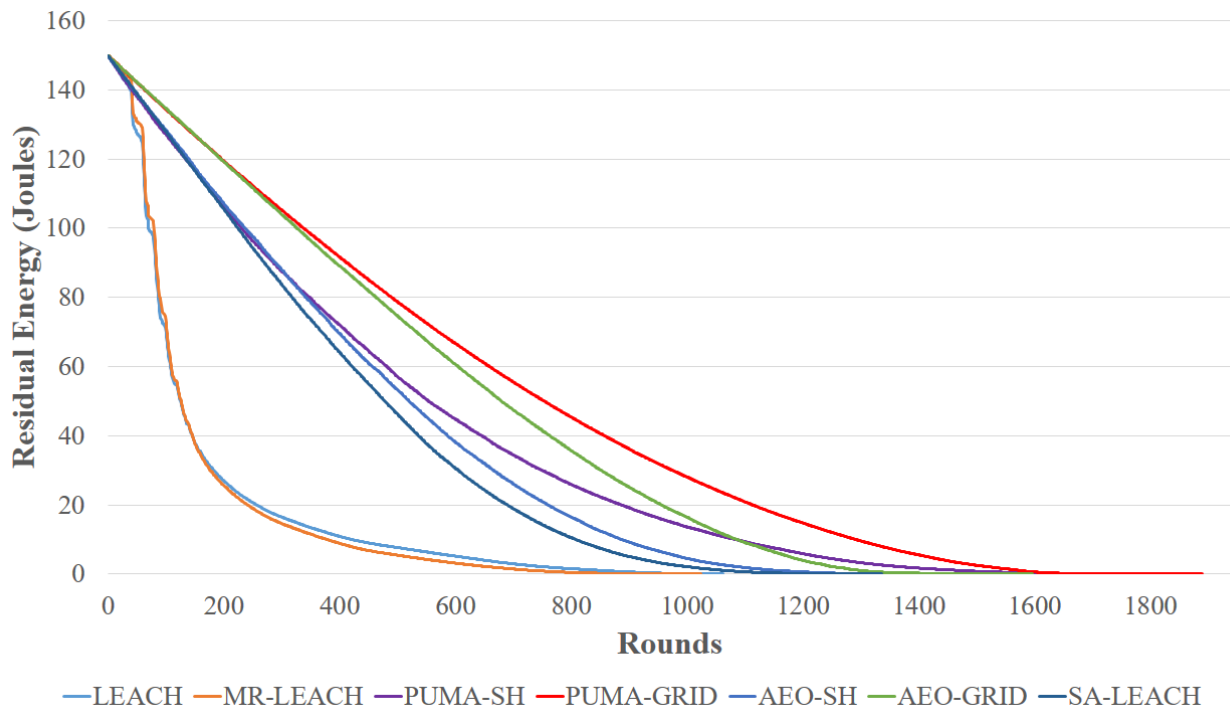


Figure 5.26: Residual energy comparison across different routing protocols with BS outside the network .

In this scenario, energy consumption increases moderately compared to the BS inside case. LEACH and MR-LEACH lose energy the fastest, with their curves dropping almost immediately. SA-LEACH lasts slightly longer but still converges quickly in the mid rounds. AEO-SH and AEO-GRID exhibit smoother and more controlled energy decay, with the grid version main-

taining higher energy levels for a longer period. PUMA-SH shows an even slower decline, while PUMA-GRID retains the highest residual energy throughout most of the simulation. PUMA-GRID provides the best energy preservation, followed by AEO-GRID, while LEACH type protocols deplete energy the fastest under edge based communication.

Figure 5.27 shows how coverage fairness degrades across protocols when the BS is placed at the network edge.

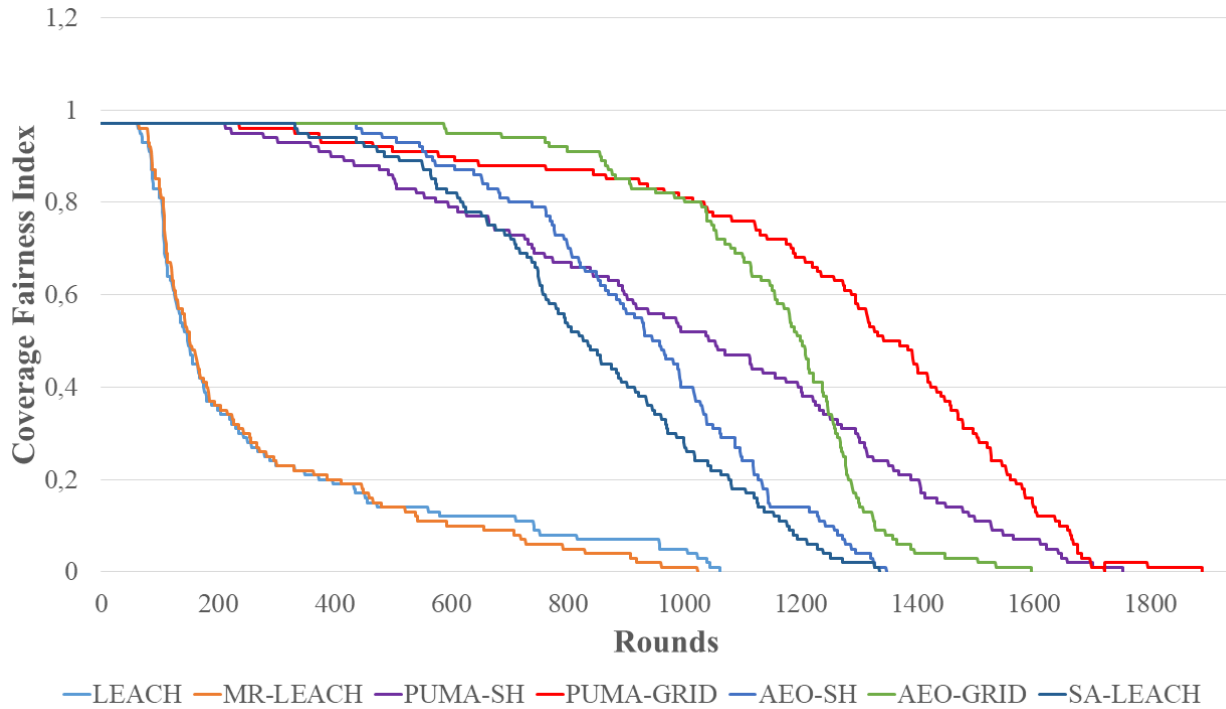


Figure 5.27: CFI comparison across different routing protocols with BS at the edge of the network .

LEACH and MR-LEACH lose coverage almost immediately, indicating very weak spatial balance. SA-LEACH performs slightly better but still drops early due to limited sustainability after the first failures. AEO-SH and PUMA-SH maintain fairness longer thanks to more structured CH selection, though the single hop constraint accelerates decline in later rounds. The grid based versions of AEO and PUMA clearly outperform all others. AEO-GRID prolongs fairness substantially, while PUMA-GRID maintains the highest CFI for the longest period, showing the best ability to preserve uniform coverage even under edge based communication.

5.4.4 Runtime Evaluation of Clustering and Routing Mechanisms

Table 5.4 presents the computation time required by each protocol to complete CH selection, cluster formation, and routing path determination in the early round where all nodes are still alive.

Table 5.4: Runtime for Routing Protocols

Routing Protocols	LEACH	MR-LEACH	PUMA-SH	PUMA-GRID	AEO-SH	AEO-GRID	SA-LEACH
Runtime (seconds)	0.022537	0.016993	1.432768	1.543625	16.323490	16.522147	0.023767

Evaluating runtime at this stage ensures that no protocol is penalized by reduced node availability or topology changes. These values therefore represent the intrinsic computational complexity of each algorithm and highlight their suitability for real-time or large-scale WSN deployments. LEACH and MR-LEACH achieve the lowest execution times (0.0225 s and 0.0170 s), owing to their simple, probabilistic CH selection that requires no iterative search or optimization. SA-LEACH remains similarly lightweight (0.0238 s), as SA operates on a limited neighborhood and adds only minimal overhead to the basic LEACH structure. PUMA-SH and PUMA-GRID show moderate execution times (1.4328 s and 1.5436 s). The main computational complexity of PUMA actually comes from generating new Pumas from the previous population, recalculating the fitness value for each updated solution, evaluating the exploration–exploitation functions, and updating the internal PUMA parameters at every iteration. Importantly, PUMA does not rely on complex mathematical operators, derivative computations, or heavy metaheuristic structures, which keeps the algorithm efficient. Although the grid based K-NN approach requires slightly more time than the direct version, this additional cost remains minimal because forming the grid and applying K-NN searches are simple and lightweight operations.

AEO-SH and AEO-GRID exhibit the highest execution times (16.323 s and 16.522 s). This is because AEO performs multiple iterative updates, computes attraction–repulsion forces, evaluates full populations of candidate solutions, and requires derivative based calculations within its equations. These numerous mathematical operations significantly increase computational cost. The grid based k-NN version adds only minor overhead, but the overall complexity remains substantially higher than all other protocols.

5.4.5 Impact of BS placement on multi-hop grid k-NN routing

To further examine the behavior of the proposed protocol and specifically evaluate the impact of the grid k-NN routing mechanism, we conducted an additional analysis of routing performance under the three BS placement scenarios. This evaluation focuses on routing structure and efficiency by measuring metrics such as average hop count, average hop distance, and load distribution among CHs. These indicators provide deeper insight into how grid adjacency and nearest neighbor selection influence path length, forwarding balance, and overall energy expen-

diture. As illustrated in Table 5.5, routing behavior is strongly dependent on BS placement. When the BS is located inside the sensing field, most CHs require only two or three hops to reach the destination, and hop distances remain short due to the symmetrical grid structure. This results in low per CH forwarding load and high routing efficiency. In edge deployments, path lengths increase and hop distances become moderately longer, yet the k-NN mechanism significantly reduces detours by prioritizing forward direction neighbors. This produces a more balanced routing load compared to traditional geometric forwarding. In outside placements, routes become naturally longer due to the greater separation between CHs and the BS. However, the Manhattan grid expansion combined with k-NN selection helps distribute the load across multiple relay candidates, preventing premature depletion of specific CHs. Overall, the table highlights the robustness of grid assisted k-NN routing under diverse topological scenarios.

Table 5.5: Routing behaviour across BS placements

BS Place- ment	Avg. Hop Count	Avg. Hop Distance (m)	Max Load per CH	Routing Efficiency and Behavior
Inside	1–2 hops	20–35 m	Low– Moderate	Very high efficiency. Grid adjacency keeps forwarding strictly directional, producing the shortest paths. k-NN rarely needs expansion because almost all CHs fall in adjacent cells. Load remains well balanced.
Edge	2–3 hops	30–55 m	Moderate	Routing remains efficient. k-NN significantly reduces detours by selecting the closest forward CHs. Some CHs at the far side handle slightly more load, but grid filtering prevents overload.
Outside	3–5 hops	45–75 m	Moderate– High	Paths become longer because all data must move outward first. k-NN with grid expansion distributes forwarding more evenly across multiple CH corridors, lowering congestion but increasing hop count.

5.5 Discussion

This section provides an integrated discussion of the performance of all evaluated routing and clustering protocols: LEACH, MR-LEACH, SA-LEACH, AEO-SH, AEO-GRID, PUMA-SH, and PUMA-GRID under the three base-station (BS) placement scenarios: inside the network, at the edge, and outside the sensing field. The analysis synthesizes results from all performance

metrics, including FND, HND, LND, residual energy, number of packets delivered, and the CFI.

LEACH consistently shows the weakest results in all scenarios. Its random CH selection produces unbalanced clusters and forces many nodes; especially those far from the BS to transmit over long distances. As a result, LEACH experiences early instability, fast energy depletion, and poor coverage preservation regardless of BS placement. MR-LEACH offers only a small improvement. Although multi-hop forwarding slightly reduces the transmission load on distant CHs, the protocol still inherits LEACH's fundamental limitation: random CH placement. This leads to uneven cluster sizes, hotspot formation, and rapid decline once nodes near the BS become overloaded. SA-LEACH performs noticeably better in the early stages. SA corrects the weak CH choices made by LEACH and produces well-positioned, energy efficient clusters during initial rounds. However, its improvements are short lived. After the first node dies, SA-LEACH deteriorates rapidly because it optimizes CHs locally per round without maintaining long term balance. Thus, its early stability does not translate into extended lifetime.

AEO-SH achieves a substantial increase in lifetime compared to LEACH-based protocols. Its fitness function based on distances and residual energy—generates more compact, balanced clusters and strategically positions CHs. This reduces transmission cost, improves early and mid stage stability, and enhances energy distribution. However, AEO-SH still relies on single hop communication, so when the BS is positioned at the edge or outside the field, CHs far from the BS consume energy quickly and shorten the late stage lifetime. Integrating grid-based routing changes this behavior significantly. AEO-GRID benefits from structured, short range forwarding between grid cells, preventing CHs near the BS from being overloaded and equalizing transmission cost across the network. This upgrade boosts HND, LND, CFI, and packet delivery—especially in edge/outside scenarios—showing how routing structure is as important as CH selection quality.

PUMA-SH delivers consistently strong performance thanks to its meta-heuristic design. It selects CHs intelligently without complex mathematical operations, derivatives, or heavy computation. This makes PUMA efficient, fast, and able to form balanced clusters with minimal overhead. As a result, PUMA-SH achieves excellent FND, HND, and energy distribution, outperforming both AEO-SH and SA-LEACH across most metrics.

PUMA-GRID further enhances performance by integrating optimized CH selection with an energy aware, grid based k-NN multi-hop routing scheme. The use of k-NN within the grid structure provides several advantages: it ensures that each CH selects its next hop among the closest and suitable neighbors, reduces the probability of long distance transmissions, and

naturally balances traffic across the network. By constraining routing decisions to local grid cells, k-NN also prevents routing bottlenecks and spreads forwarding load more uniformly, which helps delay early node depletion around the BS. This combination enables the protocol to achieve the longest overall lifetime particularly the highest LND when the BS is located inside the network; while also providing superior HND and sustained network stability. Furthermore, PUMA-GRID consistently delivers the highest number of packets to the BS across nearly all deployment scenarios and maintains the best CFI, preserving uniform sensing coverage for a significantly longer duration compared to all other evaluated protocols. The synergy between structured grid partitioning, local K-NN decisions, and optimized CH selection is ultimately what allows PUMA-GRID to outperform all other approaches in both energy efficiency and data delivery reliability. In scenarios where the BS is at the edge or outside the sensing field, PUMA-GRID still remains the best performing protocol overall. However, its LND becomes shorter than PUMA-SH. This is expected: nodes near the BS shoulder most of the forwarding load in multi-hop routing, causing them to drain faster than in the single hop version. Even with this limitation, PUMA-GRID continues to outperform all other protocols in coverage, throughput, stability, and long term operation.

5.6 Scalability Analysis of the Protocols

To investigate the scalability of the proposed routing protocols, the previously used network configurations were repeated with a higher node density. Each scenario was expanded to 600 sensor nodes, and the initial energy of every node was set to 0.1 joules. Since the purpose of this section is to provide a global view of scalability rather than a full round by round analysis, the results are presented directly in a summary table that highlights the key performance indicators under the enlarged network size. Tables 5.6, 5.7, and 5.8 assess the scalability of all

Table 5.6: Performance comparison for $N = 600$ nodes with the BS inside the network.

Metric	LEACH	MR-LEACH	PUMA-SH	PUMA-GRID	AEO-SH	AEO-GRID	SA-LEACH
FND	20	3	18	20	12	21	28
HND	64	24	200	203	240	260	160
LND	192	272	384	391	380	398	348
Packets to BS	5.66×10^7	5.03×10^7	5.43×10^7	4.67×10^8	1.9×10^7	3.3×10^7	1.86×10^7
CFI < 1	39	5	107	190	55	90	72
CFI < 0.5	67	236	320	333	225	240	224
Time need (s)	0.028	0.010	3.19	3.21	210.38	210.4	0.03

protocols under a higher node density of 600 sensors and three BS placement scenarios. Several

Table 5.7: Performance comparison for $N = 600$ nodes with the BS outside the network.

Metric	LEACH	MR-LEACH	PUMA-SH	PUMA-GRID	AEO-SH	AEO-GRID	SA-LEACH
FND	20	3	4	4	8	5	16
HND	64	27	124	178	190	170	157
LND	205	224	376	381	360	380	343
Packets to BS	4.81×10^7	3.36×10^7	2.85×10^7	3.45×10^7	2.15×10^7	2.16×10^7	1.78×10^7
CFI < 1	40	4	55	59	40	60	71
CFI < 0.5	179	179	239	307	280	266	210
Time Need (s)	0.042	0.010	6.94	7.28	130.89	134.46	0.04

Table 5.8: Performance comparison for $N = 600$ nodes with the BS at the edge of the network.

Metric	LEACH	MR-LEACH	PUMA-SH	PUMA-GRID	AEO-SH	AEO-GRID	SA-LEACH
FND	20	2	3	10	20	21	27
HND	64	24	142	192	180	198	160
LND	192	272	375	382	350	366	348
Packets to BS	5.66×10^6	5.03×10^6	2.74×10^6	3.27×10^6	1.26×10^6	1.60×10^6	1.86×10^6
CFI < 1	40	4	71	143	95	100	71
CFI < 0.5	67	29	268	319	220	240	224
Time Need (s)	0.008	0.004	3.78	4.02	128.31	129.04	0.037

performance metrics are used to capture how each protocol behaves as the network grows. The FND, HND, and LND indicators describe the stability period, mid lifetime, and overall lifetime of the network, revealing how energy depletion patterns change when the number of nodes increases. The total number of packets sent to the BS reflects the throughput capacity of each protocol under heavy communication load. CFI is monitored until it drops below 1 and then below 0.5, showing how uniformly the sensing area remains covered as energy becomes scarce. Finally, the time required for CH selection, cluster formation, and routing path construction quantifies the computational scalability of each algorithm in a dense 600 node environment.

Classical clustering methods such as LEACH, MR-LEACH, and SA-LEACH do not scale effectively. As network density increases, these protocols suffer from short stability periods, accelerated energy depletion, and inconsistent packet delivery due to their heuristic or random CH selection strategies. Their routing becomes increasingly inefficient, leading to early node deaths and uneven load distribution, especially under edge and outside BS placements.

In contrast, metaheuristic based protocols; PUMA and AEO demonstrate far stronger scalability because they adapt CH selection to energy and distance conditions. However, the two families differ significantly in computational behavior. In their single hop versions, PUMA-SH maintains a strong balance between lifetime, packet delivery, and execution time: it offers competitive stability while keeping computation within only a few seconds. AEO-SH achieves longer lifetimes, but at the cost of extremely high execution times exceeding 200 seconds, making it

unsuitable for large or dynamic deployments where re-clustering must occur frequently.

When grid based multi-hop routing is introduced, both optimization families benefit from more balanced energy consumption. Yet the scalability trade offs become even more pronounced. PUMA-GRID achieves the most practical large scale performance: it extends lifetime, improves fairness, and increases throughput while keeping execution time low and stable, maintaining the same order of magnitude as PUMA-SH. AEO-GRID provides the longest lifetime overall, but again its computational overhead exceeds 100 seconds per round, rendering it impractical for real time or energy constrained applications despite its endurance advantage.

Overall, the scalability analysis demonstrates that PUMA-GRID is the most balanced and operationally viable solution. It consistently provides strong lifetime, high packet delivery, stable fairness, and acceptable execution time across all BS placements. By comparison, AEO-GRID excels in lifetime but becomes computationally prohibitive, while classical LEACH based methods fail to scale effectively at high node densities. These findings confirm that combining lightweight k-NN routing with PUMA-based CH selection provides an efficient, scalable approach suitable for dense and dynamic WSN environments.

5.7 Global Discussion

The results presented throughout this chapter highlight a clear and consistent relationship between the clustering mechanism, the underlying routing structure, and the overall energy balance of the network. Although each set of experiments was examined individually, weight analysis, protocol comparison, and scalability evaluation, their combined interpretation reveals several unifying principles that explain why certain strategies are more effective than others.

A first observation concerns the fundamental role of the clustering process in determining the early and mid stage performance of the network. The fitness based formulation of PUMA, which integrates spatial efficiency and residual energy, produces well distributed CHs with balanced workloads. As shown in the weight combination analysis, small variations in the distance or energy weights significantly influence FND, HND, and LND. High w_1 values improve early stability by minimizing transmission distances, while high w_2 values promote fairness and prolong mid life operation. These trends demonstrate that clustering is not merely a structural step but a primary driver of energy consumption patterns.

However, clustering alone is not sufficient to guarantee long term performance. The routing strategy plays an equally decisive role, especially in later rounds when energy becomes scarce and communication distances increase. Random single hop protocols such as LEACH and SA-

LEACH illustrate this limitation: even when a round begins with a favorable CH configuration, the lack of routing structure leads to energy hotspots, premature CH exhaustion, and rapid network collapse once the first deaths occur. In contrast, grid k-NN based multi-hop routing provides a stable and efficient relay architecture that prevents overloaded regions and distributes forwarding tasks across multiple CHs. This structural advantage explains why both AEO-GRID and PUMA-GRID consistently outperform their single-hop counterparts across all BS placements.

A third insight emerges when considering the influence of BS location. When the BS is centrally positioned, distance minimization enhances throughput and delays early node deaths, making the clustering process highly sensitive to w_1 . When the BS is placed outside the sensing field, long communication distances amplify the importance of energy aware selection, and routing becomes the dominant source of energy expenditure. In this scenario, protocols with structured multi-hop routing (AEO-GRID and PUMA-GRID) maintain balanced energy usage and achieve the longest lifetimes, while single hop protocols deteriorate rapidly. The edge based configuration naturally exhibits intermediate behavior, showing that clustering and routing contribute jointly to energy preservation.

Across all experiments, a unifying conclusion becomes evident: energy efficiency in WSNs cannot be achieved by optimizing clustering or routing in isolation. The earliest phase of network operation is governed mainly by clustering decisions, but as rounds progress, routing structure increasingly determines whether energy is consumed uniformly or concentrated in a few critical nodes. Protocols such as PUMA-GRID succeed because they harmonize both aspects: the fitness based clustering provides spatially efficient and energy aware CHs, while the grid k-NN based routing ensures that data forwarding remains balanced, predictable, and insensitive to network irregularities.

Finally, the scalability experiments demonstrate that the advantages of structured clustering and routing become even more pronounced as node density increases. Although higher density intensifies competition for CH roles and increases communication activity, grid based routing maintains its robustness by shortening hop distances and distributing relaying tasks. This behavior confirms that the proposed approach is suitable not only for moderate size networks but also for larger deployments where uncontrolled energy imbalance typically emerges more quickly.

These results show that the interplay between clustering and routing forms the core of energy management in WSNs. Approaches that coordinate both mechanisms; particularly combining weighted fitness functions with structured routing, achieve superior longevity, higher

throughput, and more uniform coverage, regardless of deployment topology or BS placement.

5.8 Limitations

While the proposed PUMA-GRID protocol demonstrates significant improvements in energy efficiency, stability, and coverage fairness compared to existing methods, several limitations should be noted.

First, the evaluation was conducted in an idealized simulation environment, where effects such as interference, packet loss, retransmissions, and signal fading were not modeled in detail. The use of a free space propagation model provided a simplified framework to assess the optimization behavior of PUMA, but it does not capture the full complexity of real world wireless channels. Future work should incorporate more realistic communication stacks and physical channel models to validate performance under practical conditions.

Second, the grid based routing framework was primarily designed to complement the PUMA optimization mechanism rather than serve as a fully dynamic routing method. Although the grid k-NN approach effectively reduces long distance transmissions and balances the load among CHs, it does not adapt to variations in node density, node failures, or irregular spatial deployments. Empty grid cells or uneven node distributions may lead to temporary disconnections or routing inefficiencies. Nevertheless, the simplicity and scalability of the grid structure make it suitable for evaluating the energy optimization capability of PUMA based clustering. Future enhancements could include adaptive grid resizing and density aware routing to improve resilience in heterogeneous networks.

Third, the current design assumes all nodes are static and homogeneous in terms of initial energy and communication capability. While the residual energy metric allows PUMA-GRID to adapt to energy decay, the assumption of uniform initial energy does not always hold in practical deployments, and the static grid structure may not handle highly variable node densities or irregular topologies effectively. Extending the protocol to support mobile or heterogeneous nodes through adaptive mechanisms; such as node specific weight tuning or grid resizing would increase its applicability.

Finally, the fitness function weights are determined through simulation rather than through a real time adaptive process. Although effective weight combinations were identified for different BS placements, these values are dependent on the specific network scale and density considered. Real time adaptation based on current network conditions such as remaining energy, node distribution, or traffic patterns could further enhance performance and reliability, particularly

for networks with varying size or density.

Regarding the k-NN component of the routing mechanism, several limitations remain. First, forwarding decisions are primarily based on distance, without explicitly considering residual energy of relay CHs or link quality, which may reduce reliability in dynamic environments. Second, while grid expansion ensures relay discovery, it may occasionally select sub optimal hops when the nearest CH is not the most energy efficient choice. Finally, centralized computation at the BS simplifies node operations but relies on node positions remaining relatively stable between rounds.

Future improvements could include extending the k-NN scoring function to incorporate residual energy and link quality metrics, adapting the value of k according to local network density, or integrating lightweight prediction models to anticipate relay availability. Such enhancements would maintain the simplicity and low overhead of k-NN while improving routing robustness and adaptability in larger or more dynamic WSN deployments.

5.9 Conclusion

This work introduced PUMA-GRID, an enhanced clustering and routing protocol that integrates the Puma Optimization Algorithm with a grid based k-NN multi-hop communication structure. The proposed framework was designed to tackle two fundamental challenges in WSNs: extending network lifetime and maintaining balanced energy consumption across sensor nodes. By leveraging the adaptive exploration exploitation balance of the Puma Optimizer, the protocol achieved more effective and stable CH selection than traditional and contemporary approaches. Simultaneously, the structured grid based k-NN routing minimized long distance transmissions by intelligently selecting intermediate relays, thereby reducing communication overhead and alleviating the hotspot problem.

Extensive simulations conducted under multiple BS placements inside, at the edge, and outside the sensing field demonstrated the robustness of PUMA-GRID across all major performance metrics. The protocol consistently delayed the first, half, and last node deaths, ensured higher packet delivery, maintained more live nodes throughout execution, preserved residual energy more effectively, and maintained superior coverage fairness compared to LEACH variants and AEO-based protocols. The study of weight combinations further confirmed that the balance between intra cluster distance, distance to the BS, and residual energy must be adapted to deployment conditions for optimal performance. Comparative analysis also highlighted the importance of routing structure: while AEO based protocols benefit from grid k-NN routing,

PUMA remained the most stable and well balanced solution across all scenarios. PUMA-SH achieved efficient CH selection with minimal computational cost, whereas PUMA-GRID delivered the highest overall performance, achieving longer lifetimes, superior throughput, and the best coverage fairness. Even under the challenging edge and outside placements; where routing load increases and relay distances grow PUMA-GRID remained the most energy efficient and resilient option, although its LND decreased slightly compared to the single hop version due to additional relay burdens near the BS.

Despite these strong results, several limitations remain. The current framework assumes a static deployment and fixed grid size, which may not be optimal for heterogeneous or dynamic networks. Hotspot formation near the BS persists in distant configurations, indicating the need for improved load balancing strategies. Future research may explore mobility support for both nodes and the BS, dynamic weight learning using reinforcement or deep learning, hotspot mitigation techniques, and real hardware implementation to further validate, optimize, and expand the applicability of PUMA-GRID in large scale and real world WSN environments.

General Conclusion

WSNs represent an essential technological foundation for modern sensing and intelligent monitoring systems. Despite their broad applicability, these networks remain constrained by limited energy resources, which directly impacts operational lifetime, data reliability, and overall performance. Improving energy efficiency therefore remains a central research priority. In this thesis, the focus was placed on designing and optimizing clustering and routing strategies that reduce energy consumption while maintaining robust network performance.

The proposed method, PUMA GRID, integrates the Puma Optimization Algorithm for adaptive CH selection with a grid based routing strategy inspired by a k nearest neighbor decision rule. Through the intelligent search behavior modeled by the PUMA algorithm, CHs are selected in a manner that balances exploration and exploitation. This selection process uses a fitness function that incorporates residual energy, intra cluster distance, and distance to the BS. The use of adjustable weighting parameters allows the protocol to adapt to different deployment scenarios and network topologies, resulting in improved fairness and more balanced energy use.

The routing phase complements the clustering mechanism by constructing multi hop communication paths across a grid partition of the sensing field. Instead of relying on complex or computationally heavy routing models, the protocol applies a k nearest neighbor decision rule to select efficient relay nodes among neighboring grid cells. This simple learning inspired process enables each CH to forward data through shorter paths while avoiding unnecessary long transmissions. As CH positions change each round, routing decisions naturally adapt without introducing additional burden on sensor nodes.

A comprehensive simulation study was conducted to evaluate the performance of the proposed protocol under different BS placements and combinations of weighting factors. The results confirmed that PUMA GRID achieves consistent improvements in energy efficiency and network lifetime compared to conventional protocols such as LEACH, AEO based schemes, and other PUMA variants. The protocol demonstrated longer stability periods, higher packet de-

livery, reduced communication overhead, more uniform coverage, and improved residual energy preservation. These outcomes underline the effectiveness of combining metaheuristic optimization with a lightweight k nearest neighbor routing rule in energy constrained environments.

Beyond the numerical results, this work highlights the importance of adaptive optimization and topology aware routing in the design of efficient WSNs. The framework introduced in this thesis is flexible and can be extended to different architectures or integrated with machine learning models that support local decision making or global optimization. Several future directions can be explored, including extension to heterogeneous networks, incorporation of reinforcement learning or graph based neural models for routing and clustering, and real world deployment in application domains such as smart agriculture, environmental monitoring, or industrial supervision.

In conclusion, PUMA GRID offers an effective and scalable approach to improving energy efficiency and extending the operational lifetime of WSNs. The combination of adaptive CH selection and grid assisted k nearest neighbor routing provides a practical and conceptually sound contribution to the development of sustainable and autonomous communication systems.

Limitations

Despite its advantages, the proposed PUMA GRID protocol presents certain limitations. The optimization process increases computational complexity compared to purely distributed schemes. The use of fixed weighting parameters may not guarantee optimal performance under highly dynamic network conditions. Additionally, the protocol has been validated through simulation only, and real-world deployment constraints such as hardware limitations and environmental interference were not considered.

Future Work

Several promising research directions can further enhance the proposed PUMA GRID framework. First, integrating advanced machine learning techniques such as graph neural networks or reinforcement learning could enable fully adaptive and context-aware cluster head selection and routing decisions in dynamic environments.

Second, extending the protocol to heterogeneous wireless sensor networks, where nodes possess different energy levels and capabilities, would improve its applicability to real-world deployments. Incorporating mobility-aware mechanisms to support mobile base stations or sensor nodes also represents an important extension.

Additionally, dynamic adjustment of the weighting parameters during network operation may further improve energy balance under varying densities and traffic conditions. Hybrid optimization models that combine PUMA with other metaheuristic algorithms could be explored to enhance convergence speed and global search performance. Finally, implementing and validating the proposed protocol on real sensor platforms would provide valuable insights into its scalability, robustness, and practical feasibility.

References

- [1] M. Bhattacharya, “A survey on importance of routing protocol in wsn,” *Journal of Contemporary Issues in Business and Government*, vol. 26, no. 2, pp. 1–8, 2020.
- [2] M. Micheletti, “Energy efficient clustering and routing protocols for wireless sensor networks: A simulation approach,” Jan. 2020.
- [3] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, “A comprehensive survey on real-time applications of wsn,” *Future Internet*, vol. 9, no. 4, p. 77, 2017.
- [4] Z. J. Han and W. Ren, “A novel wireless sensor networks structure based on the sdn,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, p. 874047, 2014.
- [5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [6] K. Römer and F. Mattern, “The design space of wireless sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.
- [7] X. Liu, “A survey on clustering routing protocols in wireless sensor networks,” *Sensors*, vol. 12, no. 8, pp. 11113–11153, 2012.
- [8] F. Tossa, Y. Faga, W. Abdou, E. C. Ezin, and P. Gouton, “Wireless sensor network deployment: Architecture, objectives, and methodologies,” *Sensors*, vol. 25, no. 11, p. 3442, 2025.
- [9] F. Karray, M. W. Jmal, A. Garcia-Ortiz, M. Abid, and A. M. Obeid, “A comprehensive survey on wireless sensor node hardware platforms,” *Computer Networks*, vol. 144, pp. 89–110, 2018.
- [10] A. Kochhar, P. Kaur, P. Singh, and S. Sharma, “Protocols for wireless sensor networks: A survey,” *Journal of Telecommunications and Information Technology*, no. 1, pp. 77–90, 2018.

- [11] K. V. R. Pranav and K. J. Sarma, “An overview of various sensors and their uses,” *International Journal of Multidisciplinary Research*, vol. 4, pp. 27–35, 2022.
- [12] S. Li, Y. Zhang, Y. Wang, K. Xia, Z. Yin, H. Wang, and Y. Zhang, “Physical sensors for skin-inspired electronics,” *InfoMat*, vol. 2, no. 1, pp. 184–211, 2020.
- [13] M. I. Ghozali, A. C. Murti, W. H. Sugiharto, and K. Roder, “Performance comparison of wsn topologies in iot-based water quality monitoring systems,” *ZERO: Jurnal Sains, Matematika dan Terapan*, vol. 9, no. 2, pp. 396–406, 2025.
- [14] S. H. A. et al., “A comprehensive review of sensor technologies in iot,” *Computers*, vol. 14, no. 8, p. 342, 2025.
- [15] H. E. Kiziloz, “On base station localization in wireless sensor networks,” *Balkan Journal of Electrical and Computer Engineering*, vol. 8, no. 1, pp. 57–61, 2020.
- [16] L. Shi, Y. Xu, and D. Wu, “Optimal base station placement for wireless sensor networks with successive interference cancellation,” *Sensors*, vol. 15, no. 1, p. 1676, 2015.
- [17] G. M. A. E. ur Rahman, *An Internet of Things (IoT) Based Wide-Area Wireless Sensor Network (WSN) Platform with Mobility Support*. Doctoral dissertation, University of Saskatchewan, Saskatoon, Canada, 2022.
- [18] A. Bertrand, “Applications and trends in wireless acoustic sensor networks: A signal processing perspective,” in *2011 18th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, pp. 1–6, IEEE, Nov 2011.
- [19] A. Lanzolla and M. Spadavecchia, “Wireless sensor networks for environmental monitoring,” *Sensors*, vol. 21, no. 4, p. 1172, 2021.
- [20] L. Ruiz-Garcia, L. Lunadei, P. Barreiro, and J. I. Robla, “A review of wireless sensor technologies and applications in agriculture and the food industry: State of the art and current trends,” *Sensors*, vol. 9, no. 6, pp. 4728–4750, 2009.
- [21] M. Tubaishat, P. Zhuang, Q. Qi, and Y. Shang, “Wireless sensor networks in intelligent transportation systems,” *Wireless Communications and Mobile Computing*, vol. 9, no. 3, pp. 287–302, 2009.
- [22] T. Jabeen, I. Jabeen, H. Ashraf, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, “An intelligent healthcare system using iot in wireless sensor network,” *Sensors*, vol. 23, p. 5055, 2023.

- [23] C. Singla, “Analysis of military security & surveillance applications using wireless sensor networks,” *International Journal of Sensor Networks and Applications*, vol. 12, no. 1, pp. 45–57, 2024.
- [24] A. Jangra, G. Singh, and H. Arora, “Wireless sensor network: Architecture, design issues and applications,” *International Journal of Scientific & Engineering Research (IJSER)*, vol. 2, no. 11, pp. 6–10, 2014.
- [25] S. I. Hamim and A. B. Ab Rahman, “Optimizing wireless sensor networks: A survey of clustering strategies and algorithms,” *International Journal of Computer Networks and Applications (IJCNA)*, vol. 11, no. 5, pp. 673–689, 2024.
- [26] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [27] M. S. U. Din, M. A. U. Rehman, R. Ullah, C.-W. Park, and B. S. Kim, “Towards network lifetime enhancement of resource-constrained iot devices in heterogeneous wireless sensor networks,” *Sensors*, vol. 20, no. 15, p. 4156, 2020.
- [28] S. Hudda and K. Haribabu, “A review on wsn based resource constrained smart iot systems,” *Discov. Internet Things*, vol. 5, p. 56, 2025.
- [29] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
- [30] R. Priyadarshi, B. Gupta, and A. Anurag, “Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues,” *J. Supercomput.*, vol. 76, pp. 7333–7373, 2020.
- [31] A. K. Paul *et al.*, “Localization in wireless sensor networks: A survey on measurement techniques and strategies,” *Sensors*, vol. 6, no. 4, p. 24, 2017.
- [32] C. Li, H. Zhang, B. Hao, and J. Li, “A survey on routing protocols for large-scale wireless sensor networks,” *Sensors*, vol. 11, pp. 3498–3526, 2011.
- [33] K. Römer, P. Blum, and L. Meier, “Time synchronization and calibration in wireless sensor networks,” *Proceedings of the International Workshop on Wireless Sensor Networks (WSN) / Lecture Notes in Computer Science*, vol. 2920, pp. 1–10, 2004.

- [34] J. Kumari, “A comprehensive survey of routing protocols in wireless sensor networks,” in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 325–330, IEEE, Mar 2015.
- [35] N. Rathi, J. Saraswat, and P. P. Bhattacharya, “A review on routing protocols for application in wireless sensor networks.” arXiv preprint arXiv:1210.2940, 2012. <https://arxiv.org/abs/1210.2940>.
- [36] N. Swarna, A. H. Srinivasa, H. C. Harishkumar, and P. Arathi, “Flat based network routing protocol in wireless sensor network,” *International Journal of Engineering Research & Technology (IJERT)*, *NCAISE – 2015*, vol. 3, no. 21, 2015.
- [37] H. Echoukairi, K. Bourgba, and M. Ouzzif, “A survey on flat routing protocols in wireless sensor networks,” in *Proceedings of UNet’15*, Sep 2015.
- [38] N. Sabor, S. Sasaki, M. Abo-Zahhad, and S. M. Ahmed, “A comprehensive survey on hierarchical-based routing protocols for mobile wireless sensor networks: Review, taxonomy, and future directions,” *Sci. World J.*, vol. 2017, p. 2818542, 2017.
- [39] M. K. Khan, M. Shiraz, Q. Shaheen, S. A. Butt, R. Akhtar, M. A. Khan, and W. Changda, “Hierarchical routing protocols for wireless sensor networks: Functional and performance analysis,” *Sci. World J.*, vol. 2021, p. 7459368, 2021.
- [40] H. . Tan and I. Körpeoğlu, “Tree-based routing algorithms on wireless sensor networks (survey),” *Journal of Service Science and Information Technology*, vol. 16, no. 2, pp. 113–138, 2014.
- [41] D. N. Vashi and U. Desai, “Tree-based power efficient routing techniques in wireless sensor network: A survey,” *International Journal of Computer Applications*, vol. 120, no. 7, pp. 1–9, 2015.
- [42] R. Rajagopalan and P. Varshney, “Data-aggregation techniques in sensor networks: A survey,” *Communications Surveys and Tutorials, IEEE*, vol. 8, pp. 48–63, 02 2007.
- [43] R. K. Verma, S. Jain, and A. Kaushik, “A comparative study and survey of chain-based routing protocols in wireless sensor networks,” *J. Supercomput.*, vol. 81, p. 1076, 2025.
- [44] Neeraj and V. Sahni, “Review on state-of-the-art of pegasis protocol in wsns,” *International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC)*, vol. 5, no. 7, pp. 803–807, 2017.

- [45] S. Lindsey and C. S. Raghavendra, "Pegasis: Power-efficient gathering in sensor information systems," 2002.
- [46] A. M. Jubair, R. Hassan, A. H. M. Aman, H. Sallehudin, Z. G. Al-Mekhlafi, B. A. Mohammed, and M. S. Alsaffar, "Optimization of clustering in wireless sensor networks: Techniques and protocols," *Applied Sciences*, vol. 11, no. 23, p. 11448, 2021.
- [47] D. Wohwe Sambo, B. O. Yenke, A. Förster, and P. Dayang, "Optimized clustering algorithms for large wireless sensor networks: A review," *Sensors*, vol. 19, no. 2, p. 322, 2019.
- [48] M. F. Alomari, M. A. Mahmoud, and R. Ramli, "A systematic review on the energy efficiency of dynamic clustering in a heterogeneous environment of wireless sensor networks (wsns)," *Electronics*, vol. 11, no. 18, p. 2837, 2022.
- [49] S. Misra and R. Kumar, "A literature survey on various clustering approaches in wireless sensor network," in *2016 2nd International Conference on Communication Control and Intelligent Systems (CCIS)*, (Mathura, India), pp. 18–22, IEEE, 2016.
- [50] O. Boyinbode, H. Le, and M. Takizawa, "A survey on clustering algorithms for wireless sensor networks," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBIS 2010)*, (Takayama, Japan), pp. 358–364, 2010.
- [51] H. Xin, "Introduction of centralized and distributed routing protocols," in *2011 International Conference on Electronics, Communications and Control (ICECC)*, (Ningbo, China), pp. 2698–2701, IEEE, 2011.
- [52] S. Dhiviya, A. Sariga, and P. Sujatha, "Survey on wsn using clustering," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*, (Tindivanam, India), pp. 121–125, IEEE, 2017.
- [53] M. N. Riaz, "Clustering algorithms of wireless sensor networks: A survey," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 4, pp. 803–807, 2018.
- [54] S. Naeimi, H. Ghafghazi, C.-O. Chow, and H. Ishii, "A survey on the taxonomy of cluster-based routing protocols for homogeneous wireless sensor networks," *Sensors*, vol. 12, no. 6, pp. 7350–7409, 2012.

- [55] M. Gheisari, A. A. Abbasi, Z. Sayari, Q. Rizvi, A. Asheralieva, and S. e. a. Banu, “A survey on clustering algorithms in wireless sensor networks: Challenges, research, and trends,” in *2020 International Computer Symposium (ICS)*, (Tainan, Taiwan), pp. 294–299, 2020.
- [56] T. Kavitha and D. Sridharan, “Security vulnerabilities in wireless sensor networks: A survey,” *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 31–44, 2010.
- [57] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, (Maui, HI, USA), p. 10 pp. vol.2, IEEE, 2000.
- [58] M. O. Farooq, A. B. Dogar, and G. A. Shah, “Mr-leach: Multi-hop routing with low energy adaptive clustering hierarchy,” in *2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pp. 262–268, IEEE, 2010.
- [59] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [60] V. Loscri, G. Morabito, and S. Marano, “A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leach),” in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, vol. 62, (Stockholm, Sweden), p. 1809, IEEE, Sept. 2005.
- [61] H. M. Abdulsalam and L. K. Kamel, “W-leach: Weighted low energy adaptive clustering hierarchy aggregation algorithm for data streams in wireless sensor networks,” in *Proceedings of the 2010 IEEE International Conference on Data Mining Workshops (ICDMW)*, (Sydney, Australia), pp. 1–8, IEEE, Dec. 2010.
- [62] R. Sharma, N. Mishra, and S. Srivastava, “A proposed energy efficient distance based cluster head (dbch) algorithm: An improvement over leach,” *Procedia Computer Science*, vol. 57, pp. 807–814, 2015.
- [63] A. Mehmood, J. L. Mauri, M. Noman, and H. Song, “Improvement of the wireless sensor network lifetime using leach with vice-cluster head,” *Ad Hoc & Sensor Wireless Networks*, vol. 28, no. 1-2, pp. 1–17, 2015.
- [64] A. Yousaf, F. Ahmad, S. Hamid, and F. Khan, “Performance comparison of various leach protocols in wireless sensor networks,” in *Proceedings of the 2019 IEEE 15th International*

Colloquium on Signal Processing & Its Applications (CSPA), (Penang, Malaysia), pp. 108–113, IEEE, Mar. 2019.

- [65] F. Harrouz, M. Omari, and M. Kaddi, “Clustering in wireless sensor networks using fuzzy logic for clusterhead election: Review,” in *Big Data Analytics and Data Science* (V. Bhateja, H. Lin, M. Simic, J. Tang, and V. Sivakumar Reddy, eds.), vol. 1106 of *Lecture Notes in Networks and Systems*, Springer, Singapore, 2024.
- [66] J.-M. Kim, S.-H. Park, Y.-J. Han, and T.-M. Chung, “Chef: Cluster head election mechanism using fuzzy logic in wireless sensor networks,” in *Proceedings of the 2008 10th International Conference on Advanced Communication Technology (ICACT)*, (Gangwon, Korea (South)), pp. 654–659, IEEE, 2008.
- [67] H. Bagci and A. Yazici, “An energy aware fuzzy approach to unequal clustering in wireless sensor networks,” *Applied Soft Computing*, vol. 13, no. 4, pp. 1741–1749, 2013.
- [68] F. Baranidharan and F. Santhi, “Distributed load balancing unequal clustering in wireless sensor networks using fuzzy approach,” *Journal Name*, vol. Volume, no. Issue, p. 495–506, 2016.
- [69] A. Lipare, D. Edla, and S. Parne, “Fuzzy rule-based system for energy efficiency in wireless sensor networks,” *The Journal of Supercomputing*, vol. 77, pp. 1–24, 2021.
- [70] Y. Zhang, J. Wang, D. Han, H. Wu, and R. Zhou, “Fuzzy-logic based distributed energy-efficient clustering algorithm for wireless sensor networks,” *Sensors*, vol. 17, no. 7, p. 1554, 2017.
- [71] A. Naderloo, S. F. Aghda, and M. Mirfakhraei, “Fuzzy-based cluster routing in wireless sensor network,” *Soft Computing*, vol. 27, pp. 6151–6158, 2023.
- [72] A. Rai and A. Daniel, “Feec: Fuzzy based energy efficient clustering protocol for wsn,” *International Journal of System Assurance Engineering and Management*, vol. 14, pp. 297–307, 2023.
- [73] S. A. Sert, H. Bağcı, and A. Yazıcı, “Mofca: Multi-objective fuzzy clustering algorithm for wireless sensor networks,” *Applied Soft Computing*, vol. 30, pp. 151–165, 2015.
- [74] F. Harrouz, M. Omari, and M. Kaddi, “Clustering in wireless sensor networks using k-means for cluster formation: Review,” in *Innovations in Communication Networks: Sustainability for Societal and Industrial Impact* (V. Bhateja, V. Abdul Hameed, S. Udgate,

and A. Azar, eds.), vol. 1365 of *Lecture Notes in Networks and Systems*, Springer, Singapore, 2025.

- [75] M. Razzaq, D. D. Ningombam, and S. Shin, “Energy efficient k-means clustering-based routing protocol for wsn using optimal packet size,” in *2018 International Conference on Information Networking (ICOIN)*, (Chiang Mai, Thailand), pp. 632–635, 2018.
- [76] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, “Improved soft-k-means clustering algorithm for balancing energy consumption in wireless sensor networks,” *IEEE Internet of Things Journal*, vol. 8, pp. 4868–4881, March 15 2021.
- [77] X. Yang, T. Liu, and D. Deng, “Inter-cluster multi-hop routing algorithm based on K-means,” in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, (Chongqing, China), pp. 1296–1301, IEEE, 2018.
- [78] L. Sahoo, S. S. Sen, K. S. Tiwary, S. Moslem, and T. Senapati, “Improvement of wireless sensor network lifetime via intelligent clustering under uncertainty,” *IEEE Access*, vol. 12, pp. 25018–25033, 2024.
- [79] M. K. Hameed and A. K. Idrees, “Distributed dbscan protocol for energy saving in iot networks,” in *Proceedings of the International Conference on Communication, Computing and Electronics Systems (ICCCES)*, Springer, 2021.
- [80] M. Kmich, N. E. Ghouate, A. Bencharqui, H. Karmouni, M. Sayyouri, S. S. Askar, and M. Abouhawwash, “Chaotic puma optimizer algorithm for controlling wheeled mobile robots,” *Engineering Science and Technology, an International Journal*, vol. 63, p. 101982, 2025.
- [81] K. Chaudhari and A. Thakkar, “Travelling salesman problem: An empirical comparison between aco, pso, abc, fa and ga,” in *Emerging Research in Computing, Information, Communication and Applications (ERCICA 2018), Volume 2*, pp. 397–405, Singapore: Springer Singapore, 2019.
- [82] M. Benhadji, M. Kaddi, and M. Omari, “Atomic energy optimization for wireless sensor network clustering (aeowsnc) protocol for energy-efficient wireless sensor networks,” *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 22802–22810, 2025.

- [83] M. Omari, M. Kaddi, K. Salameh, A. Alnoman, and M. Benhadji, "Atomic energy optimization: A novel meta-heuristic inspired by energy dynamics and dissipation," *IEEE Access*, vol. 13, pp. 2801–2828, 2024.
- [84] N. Sharma, V. Gupta, P. Johri, and A. A. Elngar, "Sho-ch: Spotted hyena optimization for cluster head selection to optimize energy in wireless sensor network," *Peer-to-Peer Networking and Applications*, vol. 18, 2025.
- [85] M. Kumar, A. Kumar, S. Kumar, P. Chauhan, and S. Selvarajan, "An african vulture optimization algorithm based energy efficient clustering scheme in wireless sensor networks," *Scientific Reports*, vol. 14, no. 1, p. 31412, 2024.
- [86] M. Haris and H. Nam, "Enhancing energy efficiency in iot-wsns through optimized pso cluster head selection," *IEEE Access*, vol. 13, pp. 126496–126512, 2025.
- [87] T. A. Chit, N. A. M. Maung, and K. T. Zar, "Modified leach and fuzzy c-means based clustering protocol for wireless sensor networks," *International Journal of Advances in Scientific Research and Engineering (IJASRE)*, vol. 7, no. 3, pp. 1–8, 2021.
- [88] L. K. Tyagi and A. Kumar, "A hybrid trust based wsn protocol to enhance network performance using fuzzy enabled machine learning technique," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 5, pp. 1–9, 2021.
- [89] N. Sikarwar and R. S. Tomar, "A new approach for wireless sensor networks based on tree-based routing using hybrid fuzzy c-means with genetic algorithm," *Engineering, Technology & Applied Science Research*, vol. 14, no. 3, pp. 14141–14147, 2024.
- [90] A. Giri, S. Dutta, and S. Neogy, "An optimized fuzzy clustering algorithm for wireless sensor networks," *Wireless Personal Communications*, vol. 126, pp. 2731–2751, 2022.
- [91] G. Jayaraman and V. R. S. Dhulipala, "Feecs: Fuzzy-based energy-efficient cluster head selection algorithm for lifetime enhancement of wireless sensor networks," *Arabian Journal for Science and Engineering*, vol. 47, pp. 1631–1641, 2022.
- [92] M. Gamal, N. E. Mekky, H. H. Soliman, and N. A. Hikal, "Enhancing the lifetime of wireless sensor networks using fuzzy logic leach technique-based particle swarm optimization," *IEEE Access*, vol. 10, pp. 36935–36948, 2022.

- [93] A. Sun, K. Zhu, J. Du, and H. Cao, "Pso-based k-means algorithm for clustering routing in 5g wsn networks," in *Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps)*, (Madrid, Spain), pp. 1–6, 2021.
- [94] K. Srinivas, S. Baskar, and K. Manikandan, "Optimization of clustering process for wsn with hybrid harmony search and k-means algorithm," in *Proceedings of the Fifth International Conference on Recent Trends in Information Technology (ICRTIT)*, (Anna University, India), 2016.
- [95] S. Chekuri and B. N. Bhandari, "Cluster based energy efficient routing protocol using sa-leach to wireless sensor networks," *Indian Journal of Science and Technology*, vol. 16, no. 7, pp. 492–500, 2023.
- [96] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, *et al.*, "Puma optimizer (po): a novel metaheuristic optimization algorithm and its application in machine learning," *Cluster Computing*, vol. 27, pp. 5235–5283, 2024.
- [97] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.
- [98] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.
- [99] M. Ahmed, A. Taha, A. E. Hassanien, and E. Hassanein, "An optimized k-nearest neighbor algorithm for extending wireless sensor network lifetime," in *Advances in Intelligent Systems and Computing*, vol. 689, pp. 587–597, Springer, 2018.