Enhancing Graph-Routing Algorithm for Industrial Wireless Sensor Networks

Nouf Helal Alharbi

Submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

December 2023

Abstract

Industrial Wireless Sensor Networks (IWSNs) are gaining increasing traction, especially in domains such as the Industrial Internet of Things (IIoT), and the Fourth Industrial Revolution (Industry 4.0). Devised for industrial automation, they have stringent requirements regarding data packet delivery, energy consumption balance, and End-to-End Transmission (E2ET) time. Achieving effective communication is critical to the fulfilment of these requirements and is significantly facilitated by the implementation of graph-routing – the main routing method in the Wireless Highway Addressable Remote Transducer (WirelessHART), which is the global standard of IWSNs.

However, graph-routing in IWSN creates a hotspot challenge resulting from unbalanced energy consumption. This issue stems from the typical configuration of WirelessHART paths, which transfers data packets from sensor nodes through mesh topology to a central system called the Network Manager (NM), which is connected to a network gateway. Therefore, the overall aim of this research is to improve the performance of IWSNs by implementing a graph-routing algorithm with unequal clustering and optimisation techniques.

In the first part of this thesis, a basic graph-routing algorithm based on unequal clustering topologies is examined with the aim of helping to balance energy consumption, maximise data packet delivery, and reduce the number of hops in the network. To maintain network stability, the creation of static clusters is proposed using the WirelessHART Density-controlled Divide-and-Rule (WDDR) topology. Graph-routing can then be built between Cluster Heads (CHs), which are selected according to the maximum residual energy rate between the sensor nodes in each static cluster. Simulation results indicate that graph-routing with the WDDR topology and probabilistic unequal clustering outperforms mesh topology, even as the network density increased, despite isolated nodes found in the WDDR topology.

The second part of this thesis focuses on using the Covariance-Matrix Adaptation Evolution Strategy (CMA-ES) algorithm. This addresses the three IWSN requirements that form the focus of this research, by proposing three single-objective graph-routing paths: minimum distance (PODis), maximum residual energy (POEng), and minimum end-to-end transmission time (POE2E). The research also adapts the CMA-ES to balance multiple objectives, resulting in the Best Path of Graph-Routing with a CMA-ES (BPGR-ES). Simulation results show that the BPGR-ES effectively balances IWSN requirements, but single-objective paths of graph-routing does not achieve balanced

energy consumption with mesh topology, resulting in a significant reduction in the efficiency of the network.

Therefore, the third part of this thesis focuses on an Improvement of the WDDR (IWDDR) topology to avoid isolated nodes in the static cluster approaches. The IWDDR topology is used to evaluate the performance of the single-objective graph-routing paths (PODis, POEng, and POE2E). The results show that in IWDDR topology, single-objective graph-routing paths result in more balanced energy consumption.

Acknowledgments

AlhamduliLlahi Robil Alamin!

First and foremost, I would like to express my deepest gratitude to Allah for granting me the patience and strength to complete this thesis.

This journey has been a challenging yet rewarding experience, especially during the first year when the COVID-19 pandemic hit the world. Despite the difficulties, this thesis has given me new knowledge and made me more resilient in the face of challenges. This thesis would not have been possible without the following people.

I am thankful for the tremendous support and guidance from my supervisors, Dr Lewis M. Mackenzie and Prof Dimitrios P. Pezaros. Their suggestions, constructive feedback, patience, and encouragement have been invaluable throughout my PhD journey.

I would also like to extend my gratitude to Taibah University, Saudi Arabia, for providing me with a scholarship and financial support during my studies. Furthermore, I wish to acknowledge the support I received from the Saudi Arabian Cultural Bureau in London.

I take this opportunity to express my profound gratitude to my beloved mother, Aishah, for her constant encouragement and prayers. My heartfelt thanks go to my amazing brothers, sisters, family, and friends for their continuous love, encouragement and support. I owe special thanks to my children, Sofana and Battal, who are the greatest joy in my life for their love, patience, and accompanying me throughout my study period.

Finally, I would like to dedicate this thesis to my dear father, Helal Alharbi, who was the first person who encouraged me to seek knowledge. Perhaps your illness and your departure from this world was the most difficult part of this doctoral journey. Your love, encouragement, and support will always be remembered in my heart. May God have mercy on you.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACK	Acknowledgement message
AP	Access Point
ASN	Absolution Slot Number
BFS	Breadth-First Search
BPGR-ES	Best Path of Graph-Routing Evolution Strategy
CCA	Clear Channel Assessment
CDF	Cumulative Distribution Function
CH	Cluster Head
CMA-ES	Covariance-Matrix Adaptation Evolution Strategy
CMs	Cluster Members
DDR	Density controlled Divide-and-Rule
DLPDU	Data-Link Layer Protocol Data Unit
E2ET	End-to-End Transmission time
EA	Evolutionary Algorithm
EBREC	Energy Balancing Routing algorithm based on Energy Consumption
EEUC	Energy-Efficient Unequal Clustering
EIF	Energy Imbalance Factor
ELHFR	Exhenced Least-Hop First Routing
ES	Evolution Strategy

GA	Genetic Algorithm
HART	Highway Addressable Remote Transducer
IIoT	Industrial Internet of Things
Industry 4.0	Fourth Industrial Revolution
IWDDR	Improve WirelessHART Density-controlled Divide-and-Rule
IWSN	Industrial Wireless Sensor Network
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MATLAB	Matrix Laboratory
NM	Network Manager
NPDU	Network layer Protocol Data Unit
OSI	Open Systems Interconnection
PDR	Packet Delivery Ratio
PODis	Path based on Distance
POE2E	Path based on minimum End-to-End transmission time
POEng	Path based on maximum residual Energy
RSL	Received Signal Level
SI	Swarm Intelligence
TCE	Total Consumed Energy
TDMA	Time Division Multiple Access
TTL	Time-To-Live
WDDR	WirelessHART Density-controlled Divide-and-Rule
WirelessHART	Wireless Highway Addressable Remote Transducer
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Chapter 1

Introduction

1.1 Overview

Network monitoring systems have been widely adopted to control and monitor field devices in industrial processes and manufacturing plants, including those producing commodities such as chemicals, electricity, and crude oil. However, as industrial monitoring systems need to check the network continuously, the deployment of multiple field devices is required to maintain process stability. In the early industrial age, field devices and monitoring systems communicated via wired communication protocols such as INTERBUS, WorldFIP, and the Highway Addressable Remote Transducer (HART). Wired communication protocols, despite their proven data packet delivery, are costly, difficult to install, time-consuming, and incapable of meeting the requirements of the Fourth Industrial Revolution (Industry 4.0) [1]. Therefore, wired protocols have upgraded to wireless versions, including the Wireless Highway Addressable Remote Transducer (WirelessHART), the International Society of Automation (ISA 100.11a), and Wireless networks for Industrial Automation-Process Automation (WIA-PA). These comprise the three major communication standards designed specifically for industrial process automation in Industrial Wireless Sensor Networks (IWSNs) [2],[3].

IWSNs are a key part of Industry 4.0 and the Industrial Internet of Things (IIoT). This makes them a promising model for smart industrial automation. The main advantages of IWSNs are their flexibility, low cost of deployment and redeployment, reduced cable infrastructure (eliminating the requirement for regular cable maintenance), and self-organisation in addition to self-healing capabilities (enabling multiple field devices to be supported) [4], [5]. Consequently, IWSNs can be applied to a variety of fields, including environmental monitoring [6], personal health monitoring [7], the automotive industry [8], smart cities [9], agricultural monitoring [10], and smart grids [11]. As a result, the global IWSN market is projected to grow to $8.67 billion by 2025, with a 50% to 90% reduction in infrastructure costs compared to its wired counterpart [12].

The WirelessHART network is the global standard for IWSNs. Its infrastructure is typically based on IEEE 802.15.4, operates in the 2.4 GHz band, and uses a mesh topology involving battery-powered field devices (wireless sensor nodes) connected to a gateway through Access Points (APs). This gateway communicates with the network control system for plant automation, which is referred to as the Network Manager (NM) [13]. The NM is accountable for network configuration, communication scheduling between network devices, routing management, and system health monitoring and reporting. The centralisation of the WirelessHART network facilitates superior control of network operations and reduces the cost of devices [3]. However, routers, handhelds, and adapters, which are considered auxiliary devices, are still needed to extend the range of the network, connect the wireless network to wired devices, and configure wireless sensor nodes.

Each sensor node contains a communication module, a sensing module, and a processing module. The communication module includes a transceiver for exchanging information with other sensor nodes. The sensing module utilises transducers to enable the collection of environmental data. The processing module analyses data packets that are sensed locally; a transmitter then sends the data packet to the gateway for further processing. However, sensor nodes are compromised by limitations of energy, memory, and communication range. The energy levels of each sensor node are quickly consumed during the communication process but, as industrial sites are often inaccessible, it is not always possible to replace or recharge this node. In such circumstances, a sensor node depleted of energy will die, resulting in a decrease in network performance [5].

The IWSN applications of industrial process automation require communication that is reliable, low-latency, and has an energy consumption balance. The centralisation of WirelessHART networks can cause the sensor nodes near the gateway to be overburdened with high traffic loads compared to those located further away. Consider, how all data packets collected across the entire network area are forwarded through these adjacent nodes to reach the gateway. Consequently, these overloaded nodes will expire more rapidly than their counterparts due to an imbalance in energy consumption (referred to as the *hotspot problem*). This could potentially result in partitioning of the network. Due to the diverse characteristics of devices, topologies, and wireless network properties (comprising interference, signal strength, and signal reflections), balancing energy consumption while optimising the performance of the network in terms of data packet delivery and low latency is a complex and challenging process.

Routing is an essential task of the NM and is key to satisfying IWSN requirements. Routes built by the NM are employed by the sensor nodes to send data packets through the network area. The selection of the routes is critical to achieving the desired network performance and meeting the exact requirements of the different IWSN applications. Data packet delivery is augmented through graph-routing and the implementation of path redundancy. Graph-routing is the most common method of routing used in WirelessHART networks and will be the exclusive consideration of this thesis. A *graph* in a WirelessHART network refers to a routing structure that forms a directed end-to-end connection between network devices; each intermediate sensor node on a path to the gateway may have multiple neighbours to which the data packet can be forwarded. If the communication with a neighbour fails, a sensor node can try to send the data packet through another neighbour. Thus, the graph-routing algorithm uses a first-path approach to transmit data packets from a source sensor node to the gateway [2], [14].

Over the past decade, the graph-routing algorithms for centralised management protocols have been developed [15]–[21]. These algorithms aim to reducing energy consumption, transmission errors, delay and resource utilisation while increase data packet delivery via path redundancy. The routing algorithms proposed in the WirelessHART network are augmented through the use of traditional techniques such as Dynamic Programming (DP), which uses the Breadth-First Search (BFS), the Floyd-Warshall, or Dijkstra's algorithm. Although these routing techniques can effectively enhance network performance, they each focus on just one requirement of the IWSN and ignore other requirements. Furthermore, they fail to address the need for balance between the multiple requirements of the IWSNs [22].

1.2 Motivation

Adapting traditional techniques of routing algorithms to balance an IWSN's requirements and improve the network's performance can be difficult. The main goal of these algorithms is to establish routing paths and deliver data packets through intermediate nodes. For example, in the real-world context of an IWSN temperature monitoring system in the furnaces of a nuclear plant, communications adopt a more critical role. This is a potentially hazardous application: if the monitored temperature exceeds a certain level, the alerting system may be required to function as a safety system, placing additional demands on the sensor nodes, particularly those near the gateway [23]. As a result, balancing the energy consumption of sensor nodes, increasing data packet delivery, and reducing delay are essential requirements of this real-life IWSN. However, these requirements can be challenging to achieve as the interference and noise in industrial environments can cause constant redundancy, leading to high latency, and unbalanced energy consumption. Centralised routing algorithms that can optimise the performance of the IWSN are, therefore, a relevant research topic and one that has not been widely explored in current literature.

The topology of the WirelessHART network may be a mesh, cluster, or star configuration [3]. Clustering techniques, which divide sensor nodes in a network area into groups (clusters), can, therefore, be applied to alleviate the hotspot problem and enhance network performance. This may provide a more practical alternative for future Industry 4.0 and IWSN protocols. In clusters, the Cluster Head (CH) acts as an intermediary between member nodes in the cluster and the gateway [24]. The use of clusters could achieve a more balanced consumption of energy: it is the receiving and routing of data packets that is responsible for depleting energy, but in this configuration most sensor nodes would not be required to forward data packets. This would enhance data packet delivery, reduce overheads, and improve network topology. However, the focus of this technique is on improving the topology of the network and not the creation of effective graph paths for graph-routing.

The use of optimisation techniques for creating and selecting paths in a centralised manner may also be beneficial to IWSNs. These techniques consist of a group of mathematical algorithms providing optimal (or best) solutions under specific criteria, typically set as objective function (f). To achieve the desired goals of this thesis, optimisation techniques need to create a well-defined, objective functional design as per the requirements of IWSNs. Optimisation techniques have been used in routing algorithms in ad-hoc networks and in Wireless Sensor Networks (WSNs), of which IWSNs are a special case. Previous research (e.g., [22], [25]) has shown that optimisation techniques are useful for finding best routing in WSNs, as best routing can promote improved data packet delivery, balanced energy consumption, and reduced End-to-End Transmission (E2ET) time.

Covariance Matrix Adaptation-Evolution Strategy (CMA-ES) is one of the foremost state-of-the-art optimisation techniques: a type of Evolutionary Algorithms (EAs) based on population methods. As such, it has been adopted as a standard tool for continuous optimisation in many research laboratories and industrial environments worldwide [26].

The use of CMA-ES for creating routes and graphs in a centralised way may be useful for IWSN standards and future wireless IIoT. A graph-routing algorithm that can optimise the performance of the WirelessHART network through the adoption of mesh or clustering topologies, and the adaptation of optimisation techniques, is a key understudied research topic. The use of IIoT, Industry 4.0, and the growth of the IWSN market are strong research motivators as these technologies still require considerable improvement for their use to become attractive. Previous studies indicate that this is the first research to specifically adopt evolution strategies to support the selection of optimal paths for IWSNs. The graph-routing algorithm applied in this thesis focuses on the characteristics of WirelessHART networks. Graph-routing creates paths in a mesh or clustering topology, using path redundancy and multi-hop routing to provide additional data packet delivery in industrial environments. These novel graph-routing algorithms are evaluated using different scenarios and protocols.

1.3 Thesis Statement

Graph-routing is the main routing method used in WirelessHART networks, and the paths it builds are key to enhancing data packet delivery, balanced energy consumption, and E2ET time – the objectives of this thesis. Resulting from the centralised management of data transmission, a conventional graph-routing approach is susceptible to the hotspot problem as it applies multi-hop and first-path-available methods to transmit data packets from the source sensor node to the gateway. This approach induces an energy consumption imbalance whereby sensor nodes near the gateway deplete energy faster than other sensor nodes in the network. Furthermore, this approach impacts E2ET time by creating conflicts between transmissions where two paths share a sensor node (sender or receiver).

This thesis asserts that by changing the topology of the WirelessHART network based on the cluster technique, using optimisation techniques to construct more efficient graph-routing paths, and exploiting path redundancy, the performance of graph-routing can be significantly improved in terms of data packet delivery, balanced energy consumption, and transmission time. Furthermore, it can achieve a balance between these requirements.

1.4 Thesis Contributions

The main contributions of this thesis are as follows:

- The design of an energy balance optimised model for the topology of the WirelessHART network. This is called the WirelessHART Density-controlled Divide-and-Rule (WDDR) topology and has been devised to alleviate the hotspot problem in the network area by reducing overload on sensor nodes around the gateway. In this model, the basic graph-routing algorithm of the WirelessHART network is applied between CHs to reduce communication multi-hops, hence improving data packet delivery.
- The design of three, single-objective paths of graph-routing using a CMA-ES algorithm. This is in accordance with the requirements of IWSNs on which this thesis is focused; namely, to minimise energy consumption (POEng), to minimise transmission time (POE2E), and to maximise data packet delivery (PODis).
- The enhancement of the CMA-ES algorithm with multiple objectives (BPGR-ES) to select the best path for graph-routing; these multiple objectives depend on the three single-objective paths outlined above. The key aim is to identify the most effective path that promotes a balance between the strict requirements of the IWSNs whilst also improving energy consumption balance in relation to all the sensor nodes in the network.

- The design of an isolated nodes-optimised model within the WDDR topology to avoid creating isolated sensor nodes in static clusters. This is known as Improve WDDR (IWDDR) and the method reconciles pre-set and dynamic clusters to address isolated nodes, thereby increasing data packet delivery and balancing energy consumption. Furthermore, the CMA-ES algorithm is used to select CHs based on the design of two objective functions: their central location from other sensor nodes in the same cluster; and their proximity to the gateway if the gateway is within their communication range, or their proximity to other CHs if the gateway is outside their communication range.

1.5 List of Publications

During the PhD program, the majority of the outcomes of the research were published in various conference proceedings and a journal. The following is a list of publications:

1. **N. Alharbi**, L. Mackenzie, and D. Pezaros, "Evaluation of Graph Routing Single Objective Paths Using Pre-set Unequal Clustering," in *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*, Nov. 2022, pp. 323–328. doi: 10.1109/ITNAC55475.2022.9998382.
2. **N. Alharbi**, L. Mackenzie, and D. Pezaros, "Enhancing Graph Routing Algorithm of Industrial Wireless Sensor Networks Using the Covariance-Matrix Adaptation Evolution Strategy," *Sensors 2022, Vol. 22*, vol. 22, no. 19, p. 7462, Oct. 2022, doi: 10.3390/S22197462.
3. **N. Alharbi**, L. Mackenzie, and D. Pezaros, "Effect of Unequal Clustering Algorithms in WirelessHART networks," in *3rd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, Jan. 2022, pp. 7–12. doi: 10.1109/MENACOMM50742.2021.9678302.

1.6 Thesis Structure

The structure of this thesis is based on six thematic chapters. Chapter 2 describes the theoretical concepts of the WirelessHART network, and provides a comprehensive literature review of the proposed graph-routing algorithms in the WirelessHART network. Chapter 3 outlines the background information essential to the understanding of clustering and optimisation techniques, the key solution methods applied in this thesis. It also provides the required preliminaries used throughout the thesis, explaining the required concepts of graph-routing, and providing a justification of the methods. Chapter 4 presents a new algorithm using clustering techniques. Chapter 5 describes three new methods utilising optimisation techniques to enhance the graph paths of the graph-routing algorithm towards a single objective. In addition, a new graph-routing algorithm based on multiple-objective functions is outlined. Chapter 6 evaluates the graph-routing of single-objective paths using the clustering technique. Finally, Chapter 7 presents the conclusions and recommendations for future work. This structure is summarised in Figure 1.1.

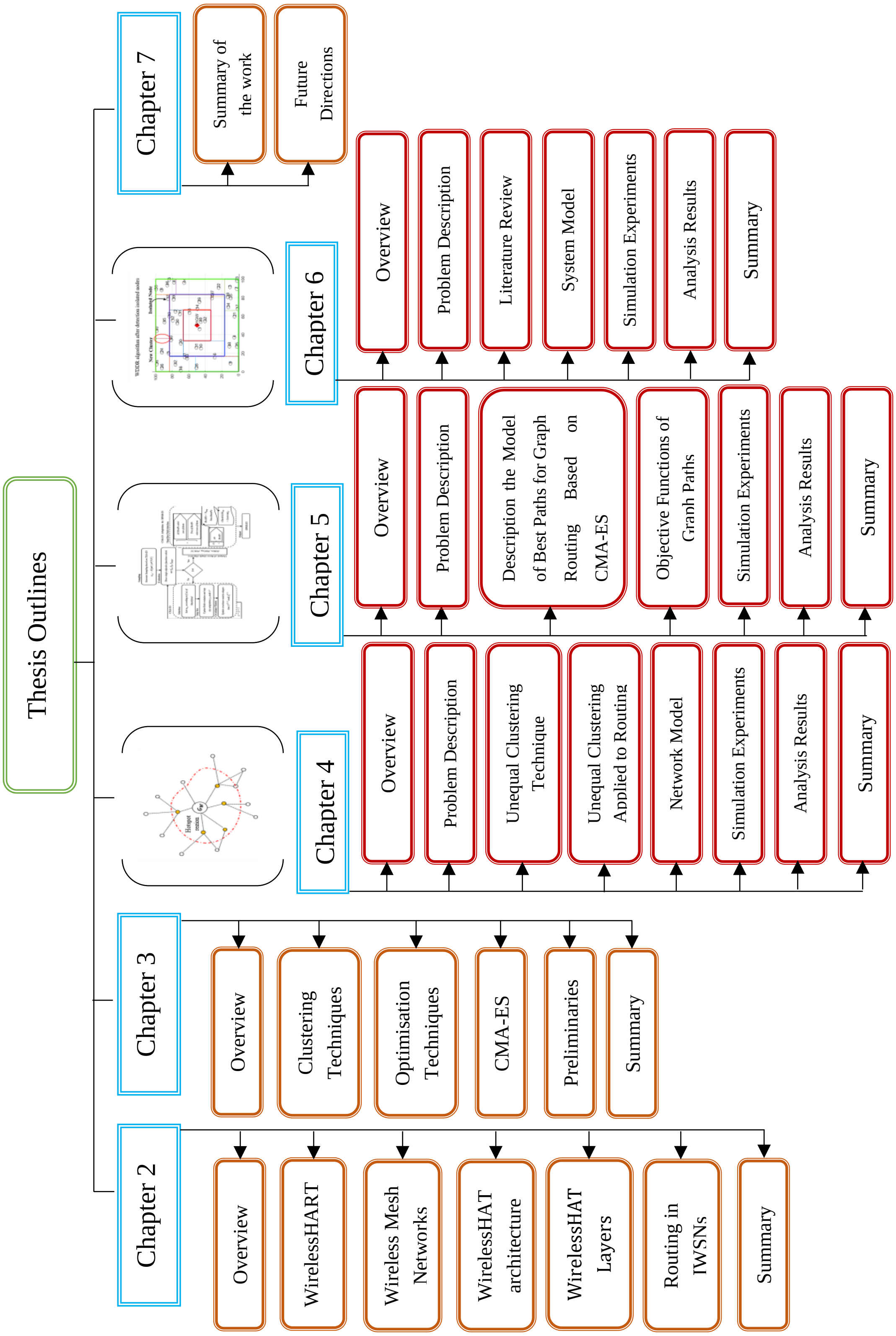


Figure 1.1 Thesis Outline.

Chapter 2

WirelessHART Networks

2.1 Overview

The most notable wireless communication standards designed specifically for IWSNs include WirelessHART, ISA 100.11a, and WIA-PA. WirelessHART is a widely adopted standard for IWSNs; it was first released in 2007 and provides high data packet delivery, security and compatibility with existing wired HART devices. As it eliminates the need for additional hardware and infrastructure WirelessHART enables the easy and cost-effective deployment of wireless networks. This makes it an ideal choice for industrial automation applications. ISA 100.11a is a comprehensive standard proposed by the International Society of Automation in 2009 which uses IEEE 802.15.4 at the physical layer, similar to WirelessHART. However, due to being relatively new and with fewer robust security features compared to WirelessHART, ISA 100.11a is less widely used, which could make it less suitable for some industrial applications [4]. WIA-PA is a proprietary standard developed by the Fieldbus Foundation which gained popularity in China in 2011. However, it uses only static routing, and field devices in the WIA-PA network lack routability, which means that the network must have routers to enable communication between network devices [27]. As a result, it has limited compatibility with other wireless communication protocols, making it less suitable for applications that require more flexible and interoperable solutions [23]. Therefore, this research focuses on the characteristics of the WirelessHART network because it is most widely used and provides reliable, secure, and easy-to-use industrial automation and control systems with numerous advantages for industrial applications, as Chapter 2 will explain.

This chapter provides a detailed description of the WirelessHART network. Section 2.2 discusses the evolution of the HART standard, followed by an overview of Wireless Mesh Networks (WMNs) in Section 2.3, and the pertinent details of WirelessHART architecture in Sections 2.4 and 2.5. Moreover, WirelessHART layers are described in detail in Section 2.6. Due to the focus of this thesis on graph-routing algorithms, a comprehensive literature review of the WirelessHART network's graph-routing algorithms is presented in the final section of this chapter.

2.2 WirelessHART

The HART Communication Foundation (HCF), known as the HART protocol, is a communication service that enables communication with smart process devices and controls [28]. It's been around since the late 1980s. Figure 2.1 illustrates how the HART protocol has developed over time. When the HART protocol was first introduced, it was a simple 4-20mA cable-based protocol to enable two-way communication with support for only 4 million field-wired devices [29]. By 2002, HART6-compatible devices, including controllers and digital control valves, were included in the HART protocol. The IEEE 802.15.4-based protocol (HART7) using a 2.4 GHz frequency channel was finalised on September 7, 2007 [15]. It is the first global Industrial Wireless Sensor Networks (IWSNs) protocol specifically designed to fulfil the requirements of industrial environments. It was ratified as the WirelessHART network by the International Electrotechnical Commission (IEC 62591) in early 2010, and over 30 million HART devices have been installed worldwide [5], providing it with a clear advantage in the industry. These are the reasons that drove the focus on WirelessHART in this research.

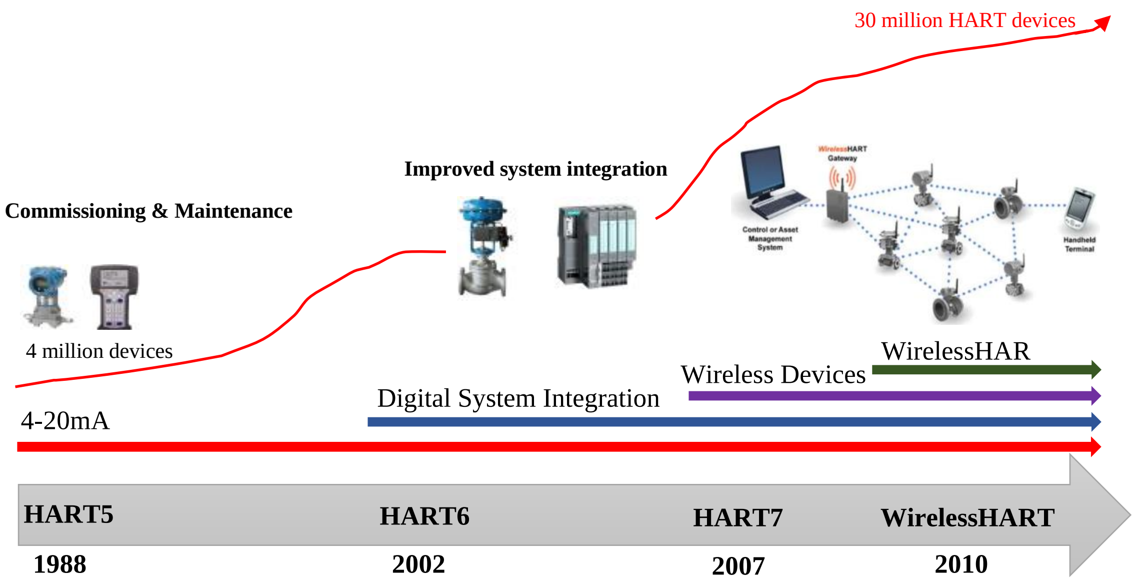


Figure 2.1 Stages of development of the HART protocol.

The WirelessHART network includes several capabilities that improve the performance and maintenance of wireless communications, including [5], [29]:

- Wireless Mesh Networking (WMNs),
- Time stamping and synchronisation of data,

- Network and transport layers,
- Enhancing publish/subscribe messaging,
- Including security encryption and decryption.

WirelessHART uses mesh networking technology and thus each device can act as a router for data packets from other devices. Instead of communicating directly with the gateway, a device can simply forward its data packet to the next closest device. This increases network range and utilises redundant communication paths to improve data packet delivery, especially in challenging industrial environments.

2.3 Wireless Mesh Networks (WMNs)

WMNs are communication networks that comprise network devices arranged in a mesh topology [30]. These networks include devices such as sensor nodes, adapters, routers, and gateways. Figure 2.2 shows a basic WirelessHART network with network devices deployed in a mesh topology [31]. All communication occurs, for example, by moving data packets from the network device, through intermediate devices, to the packet's destination. Each movement of a data packet from one field device to another along the path to the final destination (gateway) is called a *hop*.

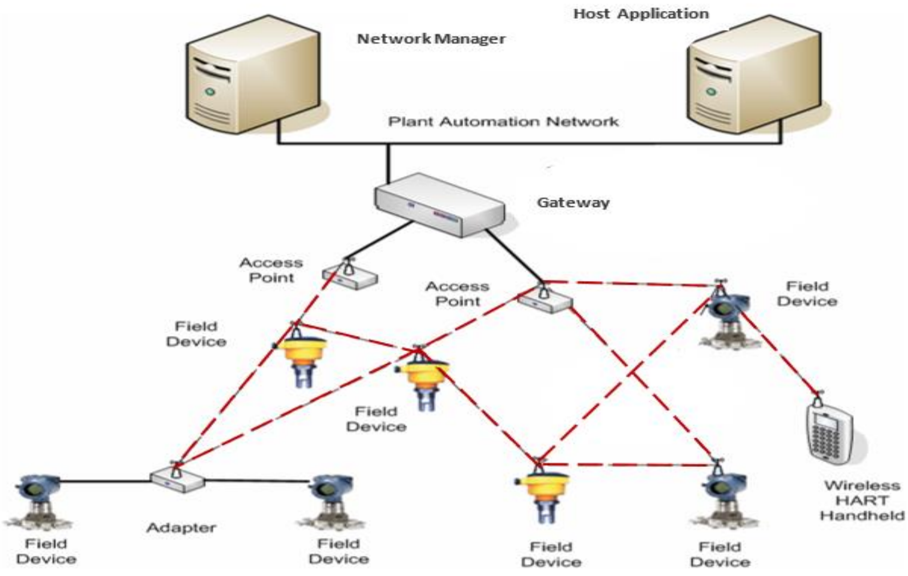


Figure 2.2 Architecture of WirelessHART [29].

In WMNs, all field devices must be able to forward data packets on behalf of other devices. Thus, field devices act as routers. The routing of a data packet from the source device, which has a data packet it needs to send, to the gateway may, in some cases, take several hops; this is called *multi-hop communication* [30].

WMNs have the following advantages [31]:

- Dynamic self-organisation and self-healing, allow for rapid and easy deployment;
- Adaptation, as WMNs can adapt to changes in the environment and reroute data if any link failure occurs;
- Some WMNs use a gateway, base station, or sink for centralised management. This management improves the network's data packet delivery and stability;
- WMNs enhance the performance of networks due to their easy maintenance and configuration, fault tolerance and robustness.

2.4 WirelessHART Architecture

This section describes the basic network device types of the WirelessHART network. The devices and the typical connection of a WirelessHART network with plant automation are shown in Figure 2.2.

2.4.1 Field Devices

Field devices are the small sensor nodes distributed in industrial environments, which make installation and configuration easy. They have limited resources in terms of battery power, processing, and storage space. All field devices must be capable of routing and forwarding data packets. They are connected to the gateway to collect and process data.

2.4.2 Adapter

The WirelessHART adapter is a bridging device that connects traditional, wired HART devices to the WirelessHART network. Thus, the adaptor must support both wired HART communication and WirelessHART, translating signals between the two of them.

2.4.3 Host Application

User applications connected to the plant automation network, the backbone network which controls the fetch process and data from field devices.

2.4.4 Handheld

A Handheld is a WirelessHART-enabled, held-in-the-hand computer that includes a host application. It is used for diagnostics, device configuration, and network information management within each device. Handhelds can gain direct access to the gateway via a Wi-Fi infrastructure [32].

2.4.5 Gateway

The gateway is responsible for query processing and data caching and is composed of a virtual gateway and one or more Access Points (Aps). It connects the host applications and the Network Manager (NM) to the WirelessHART network, allowing data to flow between these two networks. The data collected by the gateway is communicated to the plant automation network using its protocols and interfaces. This communication includes, for example:

- Routine communication of data and events. This communication occurrence is cyclical,
- Communication related to field device maintenance or failure, or as a result of abnormal process conditions,
- Communications related to the configuration of the network.

2.4.6 Access Point (AP)

An AP is a device that connects network devices to the gateway to improve network throughput (rate of data packet delivery).

2.4.7 Network Manager (NM)

The NM is an application that manages the WirelessHART network and its network devices. Each WirelessHART network has only one active NM. Multiple commands are exchanged with network devices. Its main functions are to:

- Communicate with all other network devices by the NM being wired directly to the gateway;
- Form the WirelessHART network, and provide mechanisms for devices to join and leave the network;
- Be responsible for monitoring and maintaining the health of the WirelessHART network;
- Contain a complete list of network devices used for network functions such as routing and scheduling;
- Collect diagnostic and performance information. This information is accessible during runtime, allowing network behaviour analysis;
- Provide security keys to encrypt data between the NM, gateway, and network devices to establish secure communications between network devices through the Security Manager.

2.5 Gateway, APs, and NM Architecture

This section describes the architecture between the gateway, APs, and NM, as illustrated in Figure 2.3. The gateway is functionally divided into:

- A virtual gateway to provide a single-entry point into the WirelessHART network;
- One or more APs via which data packets from the WirelessHART network are sent and received;
- Direct connection to the NM via which the NM exchanges commands with the wireless network;
- One or more host interfaces connecting the gateway to backbone networks (e.g., the plant automation network).

The gateway uses standard HART commands or Extensible Markup Language (XML) to communicate with host applications and network devices [28]. The gateway serves as a server and is responsible for collecting and maintaining cached data and command responses from all network devices.

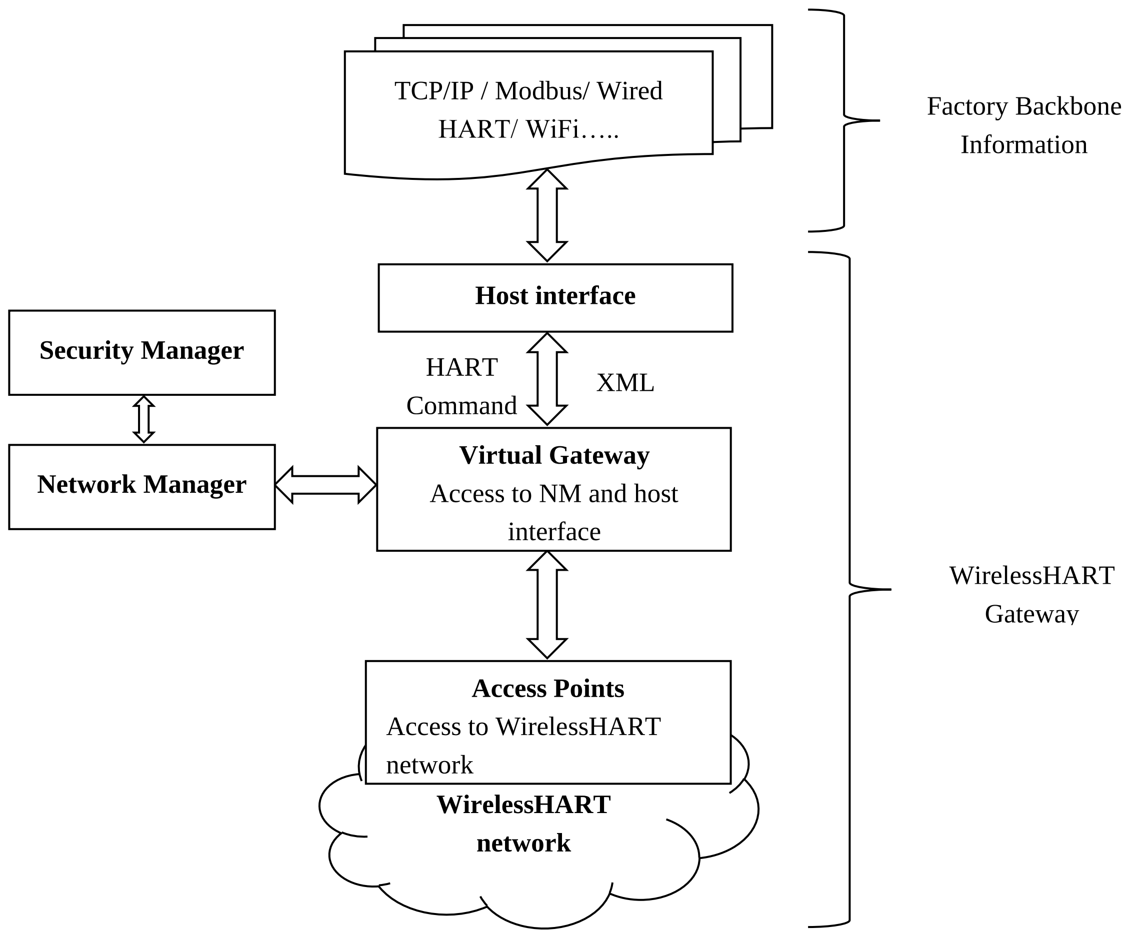


Figure 2.3 Gateway, APs, and NM Architecture.

2.6 WirelessHART Layers

The WirelessHART network's architecture is shown in Figure 2.4 according to the Open Systems Interconnection (OSI) model. The physical, data-link, network, transport, and application layers are the five layers of the standard model [2].

OSI Layer	Function	WirelessHART
Application	Provides the user with network capable applications	Command oriented, Predefine data types and application procedures
Transport	Provides network independent, transparent message transfer	Auto-segmented transfer of large data sets, reliable stream transport, negotiated
Network	End-to-end routing of packets. Resolving network addresses	Power-optimised, redundant path, self-healing wireless mesh network
Data-Link	Establishes data packet structure, framing, error detection, bus arbitration	Secure & reliable, time-synchronised TDMA, frequency agile with ARQ
Physical	Mechanical/Electrical Connection Transmits Raw Bit Stream	2,4 GHz wireless, 802.15.4 based radios, 10 dBm Tx power

Figure 2.4 OSI of WirelessHART Layers [33].

2.6.1 Physical Layer

The lowest layer of the OSI model is the physical one that is responsible for the signalling method, signal strength, device sensitivity, and environment for sending and receiving data across network media. The WirelessHART network is based on the IEEE 802.15.4 standard and operates in the license-free Industrial, Scientific, and Medical (ISM) frequency band in the range of 2400–2483.5 MHz with a 2 MHz bandwidth on each of the 16 channels. The channels as shown in Figure 2.5 are numbered from 11 to 26, with a 5 MHz gap between two adjacent channels and a data rate of up to 250 Kbps. Channel 26, which is not legal in many locales, is not supported.

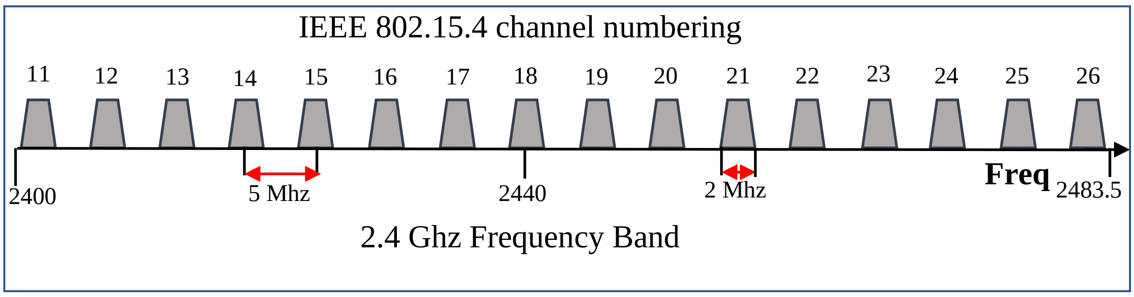


Figure 2.5 Channel numbering of 802.15.4.

The noticeable items in the WirelessHART physical layer are:

- The Time Division Multiple Access (TDMA) technique is combined with channel hopping, allowing many devices to transmit data packets along different channels at the same time.
- Transmit power: a device parameter that affects communication distance. All devices that provide transmit power are programmable from 0 dBm to 10 dBm. Figure 2.6 shows the expected communication distances for indoor (line of sight) and outdoor (non-line of sight) environments [34]. These estimates assume a unity gain omni-directional antenna; the packet error rate should be less than or equal to 1%, with a device receive input level of -82 dBm without interference.

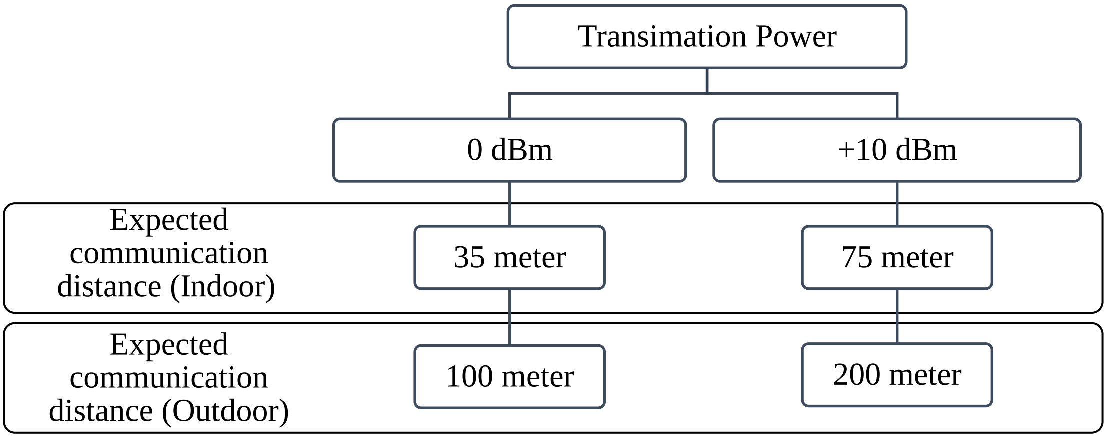


Figure 2.6 Communication distance.

2.6.2 Data-Link Layer

The data-link layer is responsible for preparing reliable data packets for transmission and managing time slots [35]. It is divided into two sublayers:

1. Logical Link Control (LLC) manages the format of frames, message integrity security services, and error detection codes.
2. Medium Access Control (MAC) determines when a network device is allowed to transmit a message.

2.6.2.1 Superframe

The data-link layer uses a TDMA technique in the MAC layer to enable collision-free and deterministic communications. It defines a strict 10 millisecond (ms) timeslot for communication between network devices where each channel is separated into timeslots. A collection of timeslots repeating at a constant rate is called a *superframe* [35].

Superframes are created and maintained by the NM, and each superframe has the properties shown in Table 2.1.

Table 2.1 Properties of a superframe.

Item	Definition
Superframe Number	The NM is given a unique identifier (ID) for each superframe.
NumSlots	Number of timeslots in the superframe.
Active Flag	Flag determines whether the superframe is active or not.
Links	Determines the list of links in the superframe.
Execution Time	The ASN at which the superframe must become active.

All channels in the WirelessHART network must support multiple superframes, from the absolution slot number (ASN) 0, which is the time when the network is first created. A superframe has a periodic structure, and its duration is equivalent to the sum of total individual durations. Figure 2.7 depicts an example of a superframe with a length of 15 timeslots. The cycle period has, therefore, 150 ms.

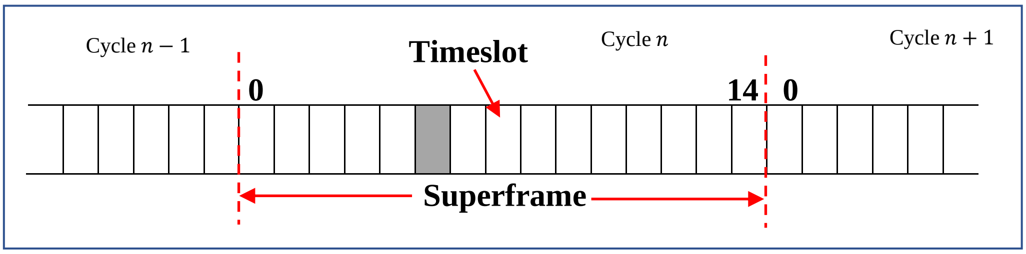


Figure 2.7 Superframe and timeslot.

2.6.2.2 Communication Links

Once a superframe is created, the NM adds, deletes, and modifies communications links within the superframe. Each link defines a communications opportunity between network devices. Therefore, the link contains a reference to a neighbour who is allowed to communicate with the device. According to the routes configured by the NM, the NM may specify multiple links in the same timeslot. The NM may allocate a link as shared to reduce communication resources where multiple source devices compete for transmission within the same timeslot [36]. Each link has the properties as shown in Table 2.2.

Table 2.2 Properties of link.

Item	Definition
LinkId	Unique identifier (ID) of the Link.
NeighborId	the ID of the source and destination device.
LinkType	Defines the link type: normal, broadcast, join, discovery.
LinkOptions	Determines the type of link: **TxLink** denotes a transmit link (source device), **RxLink** denotes a receive link (destination device), **SharedLink** denotes the link is shared by multiple devices.
Timeslot	Determines the number of timeslots in the superframe.
ChannelOffset	Specifies the channel number used for the communication link.

2.6.2.3 Timeslots Structure

All communications occur in timeslots as shown in Figure 2.8. This demonstrates the structure of a timeslot. Each 10 ms time interval is subdivided into multiple subtime intervals. The upper frame represents the source device, which is transmitting a data packet, while the lower frame represents the destination device, which is receiving a data packet. In Figure 2.8, the destination's perception of the timeslot start time is slightly delayed when compared to that of the source. Each timeslot begins by allowing a time interval to prepare the data packet being conveyed for transmission. The WirelessHART Data-Link Layer Protocol Data Unit (DLPDU) establishes the mechanisms for reliable and secure communication within a timeslot from a source device to transmit the data packet. The destination device transmits an acknowledgement message (ACK) to confirm it received the data packet successfully. The DLPDU is the data packet being transmitted. Five DLPDU types are used in the data-link layer:

a) Data DLPDUs: contain network and device data in transit to their destination device;
b) Keep-alive DLPDUs: used to maintain communication between adjacent devices;
c) Advertise DLPDUs: provide information to devices wanting to join the network;
d) ACK DLPDUs: a response message from the destination device to the source device;
e) Disconnect DLPDUs: used when a device leaves the network.

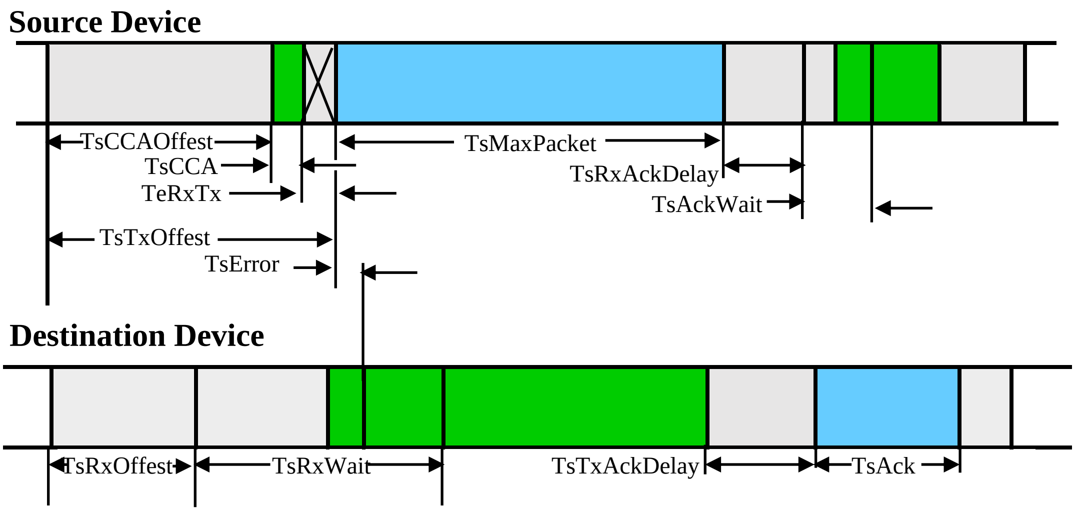


Figure 2.8 The structure of the timeslot [35].

In Table 2.3, the aim of each symbol of the timeslot structure is defined.

Table 2.3 Symbols of the timeslot structure.

Item	Definition
	Source Device
TsCCAOffest	Time from timeslot start to Clear Channel Assessment (CCA) start
TsCCA	The CCA determines whether the channel is available.
TsRxTx	The time it takes to switch from receive to transmit.
TsTxOffset	Time from the start of the timeslot to the start of the preamble transmission.
TsError	The time to which the receiving device perceives the transmitting device is out of synchronisation.
TsMaxPacket	The time required to transmit the longest message possible (133 bytes).
TsRxAckDelay	Time it takes for source device to receive an ACK from destination devices.
TsAckWait	Minimum ACK start time.
	Destination Device
TsRxOffset	Time from the start of the timeslot when the transceiver must start listening.
TsRxWait	Minimum time to wait for message to start.
TsTxAckDelay	Time between message reception and before sending the ACK.
TsAck	Time to transmit ACK.

2.6.2.4 Major Modules in the Data-Link Layer

Figure 2.9 illustrates the overarching structure of the data-link layer that is used in WirelessHART networks [29].

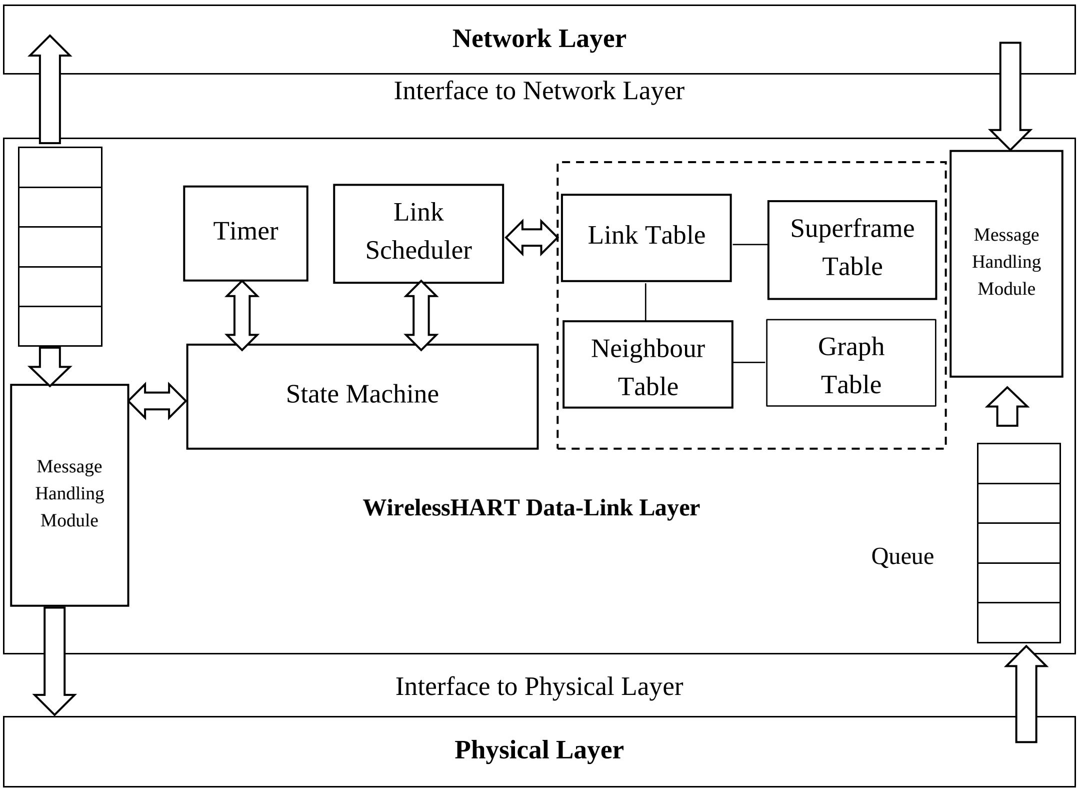


Figure 2.9 Overall data-link layer design [29].

The WirelessHART data-link layer comprises six main modules.

1. **Interfaces.** There are two interfaces, one with the physical layer, and the other with the network layer.
2. **Tables.** Each network device has a series of tables in the data-link layer that control the communications performed by the device. These tables include:

   a) The *superframe table* and *link table*, that store communication links in timeslots in each superframe;
   b) The *neighbour table* which contains a list of all the potential neighbours with which the current device has the potential to communicate directly;
   c) The *graph table* which defines the routing information of different paths created by the NM.

3. **Link scheduler.** This defines the next timeslot that needs to be served based on the schedule of the communication in the link table and the superframe table.
4. **Timer.** This module ensures accurate system timing.
5. **State Machine.** This has three components: a TDMA state machine executes transactions in the timeslot and adjusts timer clocks, while XMIT and RECV engines send and receive packets over the transceiver directly.
6. **Message Handling Module.** This module buffers the data packets of the network and physical layer separately.

2.6.3 Network Layer

The core of the WirelessHART network is the network layer, which is responsible for routing and receiving data packets from the data-link layer. It checks if data packets need to be communicated to the application layer for free-error routing or re-sent to the data-link layer to be forwarded to the next device. Figure 2.10 describes the overall design of the network layer. It has been observed that a security sublayer is implemented in the network layer itself [37]. As the WirelessHART stack does not incorporate a session layer, a security key is defined within the network layer to ensure secure communication.

2.6.3.1 Major Modules in the Network Layer

The network layer is composed of four main modules.

1. **Interfaces.** There are two interfaces, one with the data-link layer, and the other with the transport layer.
2. **Tables.** All devices maintain a series of tables that control their communications, provide routing information, allow for end-to-end acknowledgements, and protect the privacy of these communications. These tables include:

    a) *The session table* that defines communications security;
    b) *The transport table* which supports end-to-end acknowledgement to assure successful data packet delivery;
    c) *The route table* that includes routing information such as graph ID and source, and destination addresses;
    d) When source routing is used, the *source-route table* contains a list of up to eight device addresses that should be used along the path;
    e) *The timetable* indicates the timeslot that the NM has allocated to establish communication between the source and destination devices;
    f) *The service table* manages graphs that are used to route data packets from their source to their destination.

3. **Network Management.** This module manages communication between devices on the WirelessHART network and the NM.
4. **Security Module**. This module buffers the network layer’s end-to-end encrypted communication between source and destination devices.

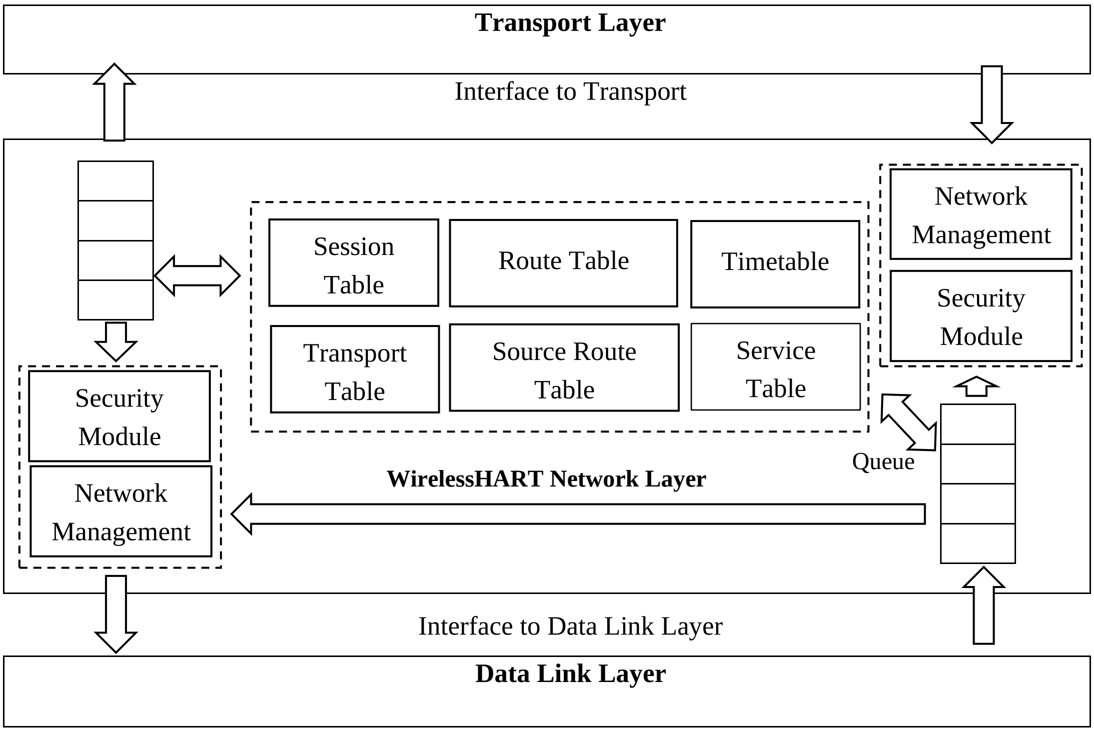


Figure 2.10 Overall design of the Network layer [29].

2.6.3.2 Types of Routing in Network Layer

In the WirelessHART network, three methods are used to route data packets in the network layer, each of which suits a different purpose. Their objective is to reliably deliver data packets on time. WirelessHART provides *Graph-Routing*, *Source-Routing*, and *Proxy Routing*. All of these must be supported by all network devices.

2.6.3.2.1 Graph-Routing

A graph is a collection of paths that connect network devices to send data packets from a source device to a destination device. Graph-routing is used for process data such as sending sensor readings, reporting alarms, and sending commands to actuators. This type of routing provides redundant communication paths between a source and a destination device in case of a path failure. This technique increases data packet delivery in an industrial environment. Because it forms the primary routing type in this thesis, further details are presented in Section 6.6.

2.6.3.2.2 Source Routing

Source-routing refers to a static directed path from a source device to a destination device. Since each data packet in source-routing carries addresses of network devices along the whole path in the header, as seen in Figure 2.11, intermediate devices do not require prior knowledge of the source path. As a result, when a data packet is routed, each network device uses the address of the next network device in the list to determine the following device that will serve as the next hop. This process continues until the data packet reaches the destination device. Path redundancy is not supported by source-routing. Subsequently, the packet is lost if any of the intermediate links fail. Therefore, source-routing is significantly less reliable than graph-routing. The aim of this type of route is to perform network diagnostics it is used for testing paths or troubleshooting network paths. It is not utilised for data processing.

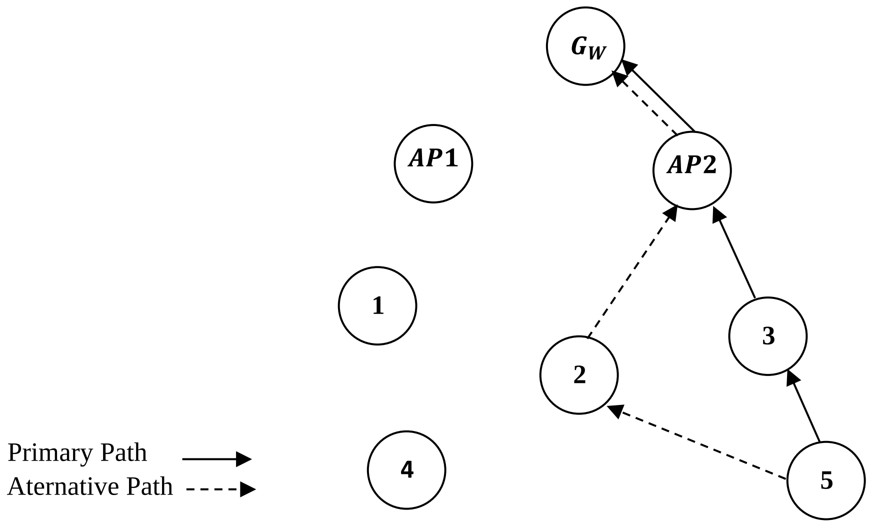


Figure 2.11 Source Routing.

Two potential source paths for sending a data packet from 5 to G_w are shown in Figure 2.11. In this situation, node 5 will generate a data packet whose Network layer Protocol Data Unit (NPDU) header [37] (see Figure 2.13) contains one of the paths listed in Table 2.4, which are configured by the NM in the routing table of node 5.

Table 2.4 Potential source paths in routing table.

Source Path	Path in routing table
Primary Path	5 → 3 → AP2 → G_w
Alternative Path	5 → 2 → AP2 → G_w

2.6.3.2.3 Proxy Routing

This type of special routing is only used when a device is joining the network. Another device, already in the network, mediates the communications between NM and a joining node.

First, a new sensor node will search for an advertisement message, which is an invitation provided by an existing node to new devices wishing to join the wireless network. The sensor node sending an advertisement is called the *proxy* and acts as the new device's parent during this procedure. The joining process then begins, which involves many message exchanges with the NM via the proxy device.

The process of a new device entering a WirelessHART network is shown in Figure 2.12 [38]. After receiving advertisement messages, the device may begin requesting to join. If more than one sensor node accepts join requests, the new device must choose the best candidate based on particular criteria. This is done on the basis of the highest signal strength and the selected node is now regarded as the new device's proxy.

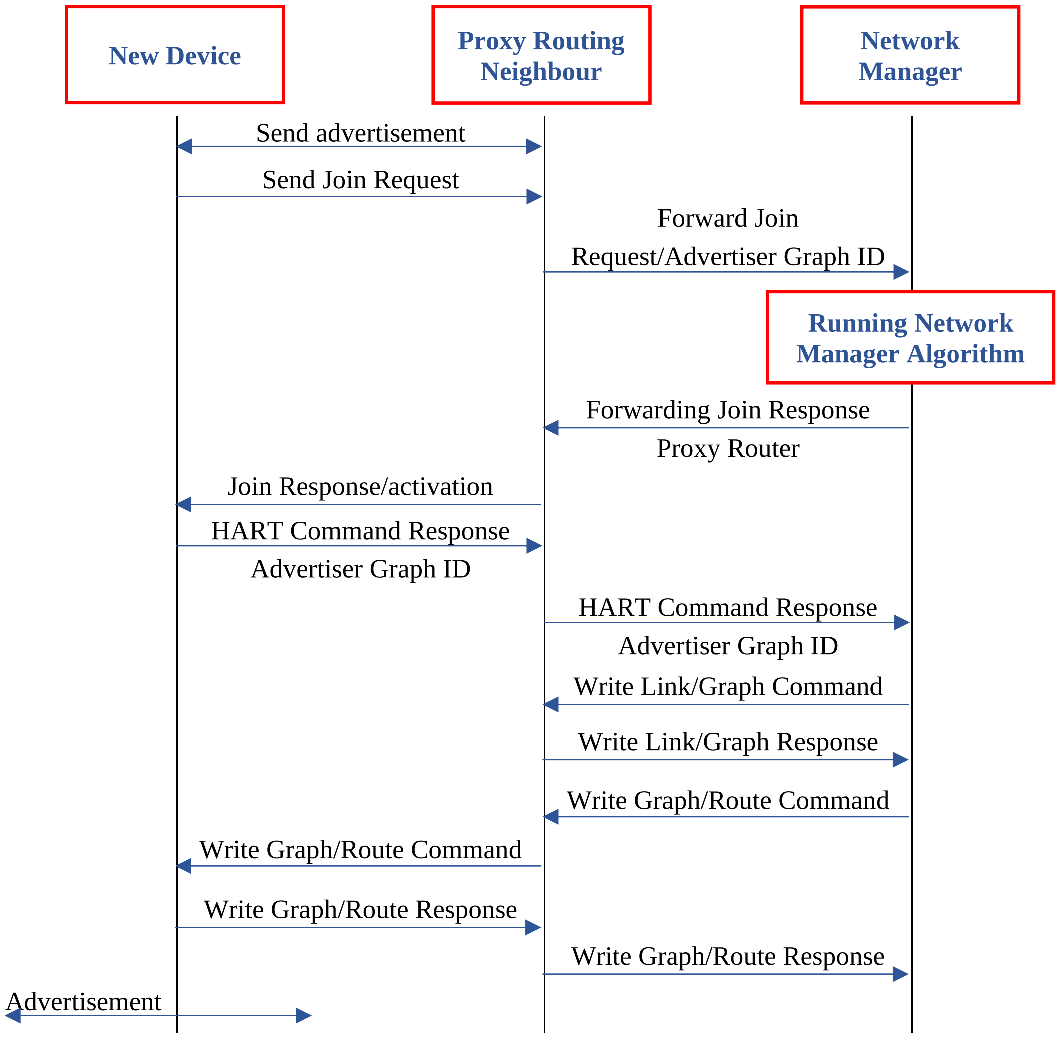


Figure 2.12 Joining process [38].

A join link will be supplied with the request. It must be forwarded by the proxy using its own routing path. As a result, the new device must utilise the proxy's graph ID, and the proxy must have a graph route to the NM. When the NM receives a join request, it allocates network resources (such as links and routes) according to the management algorithm. After all relevant network resources are set and reserved along the path, the NM sends a join activation/response instruction to the new device. So, the routing information is disseminated prior to the transfer of data packets.

2.6.3.3 NPDU Header

Figure 2.13 shows the structure of the NPDU header of the network layer.

Network Layer

NPDU header

Control Byte (1 byte)	TTL Time To Live (1 byte)	ASN Snippet (2 byte)	Graph ID (2 byte)	Source /Destination Address (2/8 byte)	Source Address (2/8 byte)	Expanded Routing information (optional) (2/4/6/8/10 byte)

Figure 2.13 WirelessHART NPDU structure.

Table 2.5 shows the fields required to route the data packet to its final destination.

Table 2.5 NPDU header fields.

Field	**Definition**
Control Byte	Set the address bits based on source and destination address.
Time-To-Live (TTL)	TTL determines how many hops a data packet can travel before being discarded.
ASN Snippet	Absolute Slot Number Snippet provides real-time network performance metrics and diagnostics.
Graph ID	The graph ID is used to route the packet to its final destination.
Source Destination Address	Defines the source and destination addresses.
Expanded Routing information	It assigns extra fields if other types of routing are used, such as source or proxy routing.

2.6.4 Transport Layer

The transport layer ensures data packets are successfully communicated across multiple hops to their final destinations. The WirelessHART network enables acknowledged and unacknowledged services to occur. An unacknowledged service enables devices to send data packets without guaranteeing successful data packet transmission. This service is utilised in procedures repeated on a regular basis, such as publishing process data. In contrast, the primary benefit provided by the acknowledged service is that it allows devices to send data packets and confirm their delivery. This method is best suited for management commands where it is necessary to keep track of the arrival of the data packet.

2.6.5 Application Layer

In the OSI model, the application layer is designed as the closest layer to the end user, and it is a command-based layer. It is used to send data packets from field devices to the NM, and to send commands from the NM to field devices. Each command is distinguished by a unique command number that defines the contents of its corresponding message. Commands from the gateway or field devices are the basis for

HART communication, and the command number, embedded in the communication, determines the message content. Each HART command performs only one of the following functions [39]:

- READ reads data from a field device to return the requested data,
- WRITE writes data to the field device to determine or save the field device's configuration,
- COMMAND causes the device to perform an action, such as an operation of the network or the configuration of a field device.

WirelessHART commands involve a set of between 768 to 1023 commands that support NM and gateway functions. The implemented commands are classified into several categories, including managing source and graph route commands, managing link and superframes commands, bandwidth management commands, as well as network health reporting [38].

2.7 Graph-Routing Mechanisms: Types, Construction, and Implementation

Graph-routing is one of the primary methods for data packet routing in WirelessHART networks. The term refers to a routing structure that forms a directed end-to-end connection between network devices in which all wireless sensor nodes on the way to the destination are pre-configured with the necessary graph information to specify the neighbours to which data packets may be forwarded to reach their final destination. All types of uplink graphs that use the gateway as a root can thus be used in WirelessHART.

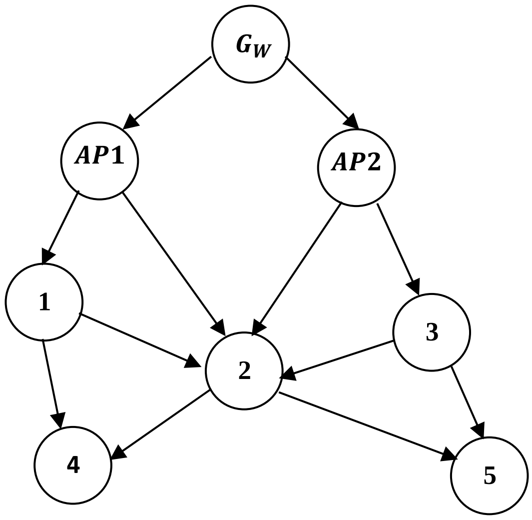
2.7.1 Types of Graph-Routing in WirelessHART network

There are three types of graph-routing in a WirelessHART network which address different communication requirements [2]:

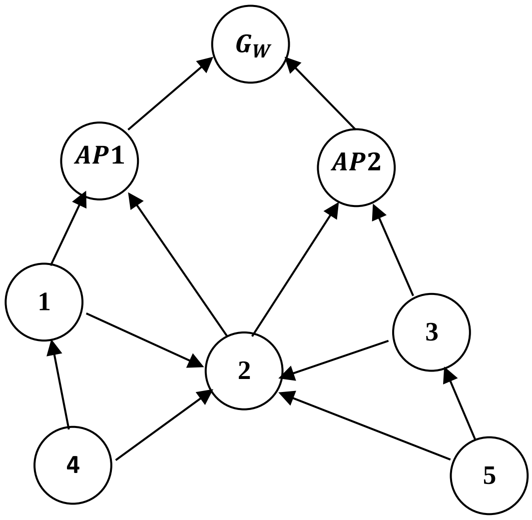
- **Uplink graph** (denoted by G_U): a graph that connects all sensor nodes to the gateway. It is used for forwarding data packets from all sensor nodes to the gateway.
- **Downlink graph** (denoted by G_n): defined per node, this graph allows control messages to be forwarded from the gateway to each sensor node.
- **Broadcast graph** (denoted by G_B): a graph that connects the gateway to all sensor nodes. This graph can be utilised to distribute common data or control information across the entire network.

To explain graph-routing in the WirelessHART network, $G(D, E)$ is used to represent the WirelessHART network topology, where the set of vertices $D = G_w \cup D_N \cup APs$,

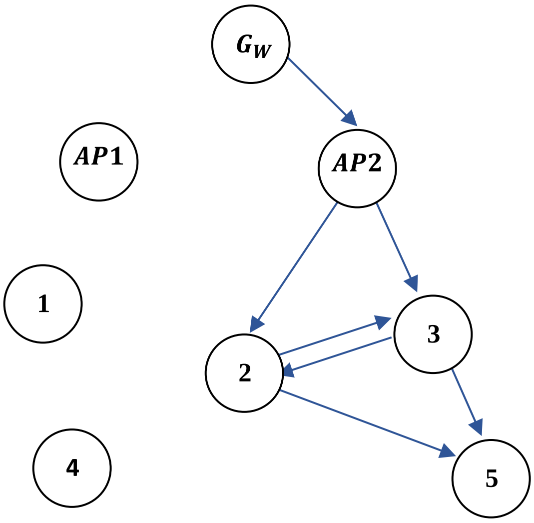
G_w refers to a specific gateway, D_N refers to the set of sensor nodes, APs refers to the access points, and E refers to the set of edges. Additionally, $G_B(D_B, E_B)$ and $G_U(D_U, E_U)$ are used to represent the broadcast graph and uplink graph respectively. The downlink graph for node $n \in D_N$ is denoted by $G_n(D_n, E_n)$. Figure 2.14 illustrates an example, the graph-routing of G_B, G_U and G_n for a WirelessHART mesh network, which consists of the G_w, two access points '$AP1$' and '$AP2$', and four nodes denoted from 1 to 5.



a. G_B



b. G_U



c. $G_n, n = 5$

Figure 2.14 Graph-routing types $\boldsymbol{G_B}$, $\boldsymbol{G_U}$ and $\boldsymbol{G_n}$.

2.7.2 Graph-Routing Construction Process

The NM is responsible for pre-configuring graphs and communication paths on all sensor nodes. Based on regular reports sent by the sensor nodes [31], the NM incorporates information about the network, such as the neighbours for each sensor node in the network and their signal levels and then uses this information to build a complete graph of the network in that moment. Each such graph has an ID called a *graph ID* that is introduced by the NM to each sensor node prior to its use and routing. After configuring the graph for the entire network and downloading the graph ID for each sensor node in the network, the basic graph-routing algorithm within the WirelessHART network uses the first available link technique to send a data packet between two sensor nodes when the source node writes a specific graph ID for the final destination in the NPDU header [37] (see Figure 2.13).

2.7.2.1 Graph ID

The graph ID is used to route each data packet to its final destination, as the graph ID offers a list of sensor nodes, any of which may be used to forward the packet towards its final destination, as shown in Table 2.6. When the graph ID value is less than 0x0100, this indicates that it is a valid graph; when it is equal to 0xFFFF, however, the graph is invalid. Once the overall graph table is generated, this is then transferred through a series of commands from the NM to the sensor nodes across the network [33].

Table 2.6 Graph table [35].

Field	Definition
Graph ID	Unique graph (ID).
Ref Neighbour	List of references to neighbours that are the next hop toward the destination

2.7.2.2 Neighbour Discovery

The neighbour discovery process is a vital step for each sensor node within the network area, as this enables the identification of potential communication links with other sensor nodes within communication range. Each sensor node maintains a list of discovered sensor nodes in its table of neighbours, as depicted in Table 2.7. This information is then periodically provided in health reports, based on the application of commands 780 and 787 [38]. The table of neighbours is thus centralised when the NM uses the information from health reports to adjust the overall network graph. This requires sensor nodes to continuously listen for communications from their existing neighbours and from new neighbours. The discovery process thus generates discovery links that are shared by all sensor nodes in the network: the sensor nodes listen for these links and periodically transmit on any valid options, facilitating the discovery of new neighbours.

The neighbour table entries collocate a variety of properties and statistics related to each neighbour, as shown in Table 2.7. These include both basic neighbour identity information and historical performance statistics, which contributes significantly to the sensor node's ability to establish effective communication with its neighbours. The process also tracks how well a sensor node communicates with its neighbours by noting the average communication strength. After the sensor node actively communicates with a neighbour, it sends periodic signals to check if that connection is still active. In any case of communication failure, a timer will begin, and the sensor node will keep trying the connection until that neighbour or its links are removed from the table. Whenever a data packet arrives, the table is updated: thus, if the table is full and a new neighbour is found, the oldest entry is removed to make space for the new one. The details in the neighbour table thus help manage and maintain effective communication within the sensor network.

Table 2.7 Neighbour table [35].

Field	**Definition**
Node ID	Unique identifier (ID) of the neigbour sensor node.
Time Source Flag	Flag indicating if device should take time synchronization from this neighbour.
Status	Status information regarding this neighbour (e.g., Path failure).
Last Time Communicated	Time when last communicated with this neighbour.
Time Path Failure Timer	Resets to path fail interval after each successful communication. The PATH_FAILURE is invoked whenever Path Failure Timer reaches zero.
Avg RSL	Average received signal level (in dBm) for packets received from neighbour.
Packets Transmitted	Number of data packets transmitted to the neighbour.
Missed Ack Packets	Number of packets for which an expected ACK was not received.
Packets Received	Number of good data packets received from the neighbour.

2.7.2.3 Building Graphs in WirelessHART Networks

A graph route provides the routing information required to guide the delivery of each data packet from source sensor nodes to its final destination. A graph is therefore defined as any directed list of paths that connect two sensor nodes within the network. The overall routing information is assembled by the NM using both the table of neighbours and diagnostic information reported by the sensor nodes. Once the routing information for each of the sensor nodes is known, the graph of the network can be activated.

No single sensor node knows the entire route; instead, the data packet is forwarded along the path to the corresponding graph ID in steps until it reaches its destination. The sensor nodes receive the data packet and then forward it along the prescribed set of paths belonging to the graph to its destination. In a properly configured network, all sensor nodes will have at least two neighbours in the graph through which they may send data packets, which ensures redundancy and enhances reliability. A routing table is illustrated in Table 2.8. Using graph-routing, a sensor node routing a data packet performs a lookup in the graph table by using the graph ID before sending the data packet to any of its listed neighbours.

Table 2.8 Route table [32].

Field	**Definition**
Route ID	Unique route (ID).
Ref Destination ID	A reference to the destination of this Route.
Ref Graph ID	A reference to the graph used to get packets to the destination.

2.7.3 Graph-Routing Implementation Example

An example of uplink graph-routing in mesh topology is illustrated in Figure 2.15. The red and blue arrows illustrate graph-routing using graph IDs for configured neighbours: in this case, sensor node 4 communicates with gateway (G_w) using graph ID 1. Sensor node 4 may thus forward a packet to either sensor node 1 or sensor node 5 in order to send it on that graph. From those sensor nodes, the packet may then take several alternative routes; however, it is guaranteed to eventually arrive at G_w if graph ID 1 is followed. In a similar manner, sensor node 4 can send packets on graph ID 2 in order to communicate with sensor node 2.

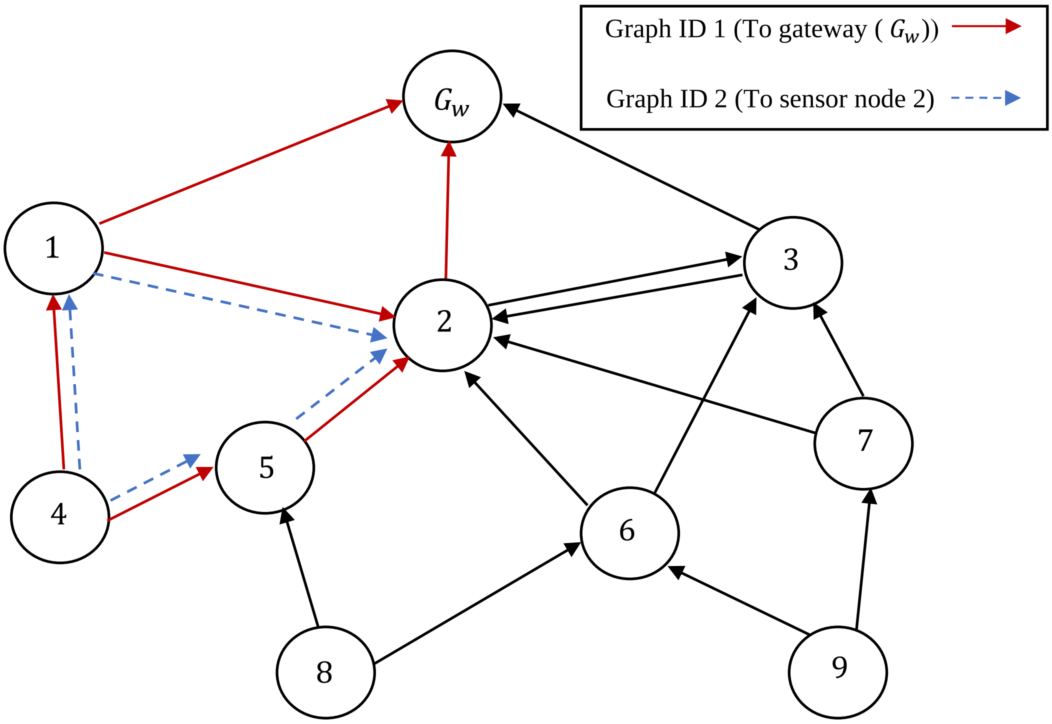


Figure 2.15 Graph-Routing, Concept Graph IDs.

2.7.4 Routing Strategy

The routing algorithm is not specified in the WirelessHART network but provides details on the routing strategy as follows [28]:

1. A direct route to the gateway should be taken if one exists,
2. When constructing graph-routing, a minimum number of hops to be considered is 2,
3. When constructing source routing, a maximum number of hops to be considered is 8,
4. The primary path is the first available path for graph and source routing,
5. For proxy routing, the signal strength is used to determine the primary path.

2.7.5 Routing Requirements

Routing is a key part of the NM's tasks, so the NM must develop the network's overall routing requirements. The NM needs information about the network, information about communication requirements, and information about the capabilities of the network devices themselves, to adjust routes in the network [32]. The following are the NM's routing requirements.

- Maintaining an internal representation of the whole network, which it utilises to build graph and source routes.
- Collecting network statistics and neighbour table information from each device in the network through periodic health reports, which are used to adjust connections and signal levels, which are then used to generate routes.
- Making decisions about making new connections and choosing between existing ones based on the communication information.
- Building route tables for graph-routing.
- Building source route lists for source routing.
- Verifying sure that no circular loops exist in any path.

2.8 Routing in WirelessHART Network

This section analyses state-of-the-art work that has been conducted on routing algorithm in IWSNs, with a focus on those associated with the WirelessHART network. According to the WirelessHART network, there may be path redundancy in the routes, making the IWSNs more reliable [2]. Most of the work explored in this analysis builds graphs or routes with redundant paths. In Section 2.8.1, the routing algorithms proposed for WirelessHART network are presented and their characteristics are highlighted. Section 2.8.2 covers the observations made for the routing algorithms from analysing the criteria used in the construction of paths and the types of paths created, plus the techniques, metrics, topology, and implementation used.

2.8.1 Analysis of Routing Algorithms

Several routing algorithms have been proposed in the existing literature to better adhere to stringent IWSN standards. Using the mesh topology of WirelessHART, [15] introduces a graph-routing technique called *Exhenced Least-Hop First Routing* (ELHFR). As an uplink graph-routing algorithm based on Breadth-First Search (BFS) and employing the Received Signal Level (RSL) information, ELHFR takes advantage of the fact that WirelessHART's NM has sufficient resources to generate routing paths for all network nodes. First, it treats the gateway as the starting point of a connected graph that describes the network topology. The BFS tree is used by all nodes to find the shortest path. By using the AvrRSL (Average Received Signal Level) as a sorting criterion for selecting neighbours in the lower levels, the ELHFR produces a sub-graph that includes all of the shortest paths from a sensor node to the gateway. After the topological graph is partitioned into many sub-graphs, the shortest paths from each sensor node to the gateway can be determined. Since sensor nodes in the ELHFR can only communicate with neighbours on a lower level, route redundancy cannot be ensured. For the ELHFR, the least-hop metric is the sole relevant one for connecting nodes. As a result, it is unable to maximise the lifetime of the network because it disregards the communication load.

Sequential Reliable Downlink Routing (SRDR) and SRDR-OPT are proposed in [16] to achieve high data packet delivery and real-time communication within industrial wireless environments through using greedy algorithms. The SRDR graph is built iteratively and, during each iteration, selects a sensor node in the topology and adds it to the resultant graph, along with connections to its neighbours. The metric to select sensor nodes and connections is based on the typical number of hops from the gateway. Lowering the number of hops between sensor nodes and the gateway, lowers latency and the utilisation of communication resources.

A minimum-hop load-balancing routing algorithm is proposed for WirelessHART [17] to achieve redundancy and minimis end-to-end communication delay to fulfil real-time demands. The offered routing algorithm consists of two phases. During the first phase, minimum-hop graphs are formed with maximum possible path redundancy, and during the second phase, device load balancing is provided to deliver a long network lifetime. The experimental results demonstrated that load-balancing techniques significantly improved the network lifetime.

The *Joint Routing Algorithm for Maximising Network Lifetime* (JRMNL) was proposed by [40] to prolong the lifetime of IWSNs. The authors considered the residual energy and transmission capacity of neighbouring nodes and the communication load of the path as an exponent-weighted cost function to select paths. It was shown that the JRMNL improves a network's lifetime much more than ELHFR [15].

The Reliable and Efficient Routing ('Re-add') algorithm is suggested for WirelessHART in [18] for data packet delivery and network lifetime improvement. In the Re-add algorithm, at least two neighbour nodes are maintained for every node to support the hop-level retransmission delivery ratio increase. The link selection process that decreases the potential retransmission number and balances the network's residual energy considers the energy model and link quality. The Re-add routing algorithm outperformed WirelessHART's routing algorithms [16], [41]: it enhanced the network lifetime and improved the data packet delivery.

According to [41], energy usage in WirelessHART can be effectively balanced by creating a pre-emptive *Energy-Balanced Graph-Routing* (EBGR) algorithm for the network node. This suggested algorithm initially applies a BFS set of rules to separate the network into different levels. Subsequently, a graph-routing algorithm redistributes the energy usage to nodes that have fewer routing activities, by decreasing the links to the nodes that have more routing activities. This aids in improving the network's lifetime as the created graphs are re-structured. Compared against other approaches, such as the ELHFR proposed by [15], EBGR enhanced energy usage and improved the network's lifetime.

Graph Route Lifetime Maximization (GRLM) is proposed for WirelessHART in [19] to satisfy industrial demands for long-term stable communication. Firstly, the authors formulated the problem of maximising the network lifetime and proved it was an NP-hard issue. Secondly, they proposed an optimal algorithm based on a linear relaxation algorithm, integer programming, and a greedy heuristic algorithm to increase the WirelessHART network lifetime. After conducting experiments on a physical testbed, GRLM increased the WirelessHART network lifetime by 60%.

A *Conflict-Aware Real-time routing (CAR)* algorithm is offered in [42] to reduce conflicts among transmissions in sensor nodes which could significantly contribute to communication delays in WirelessHART. By introducing conflict delays to routing decisions, the CAR approach can service more real-time flows while satisfying the required deadlines. After conducting experiments on an IWSN testbed, CAR appeared to cause a three-fold enhancement in the IWSN's real-time capacity.

Shi et al. [43] emphasise that a significant limitation of existing IWSNs in IIoT applications is their restricted scalability because of their centralised approach to scheduling and routing. They resolve this challenge by suggesting a *Distributed Graph routing and autonomous Scheduling* (DiGS) approach, which enables a network system to handle its graph paths and communication schedule.

Similar to [41], an *Energy-Balancing Routing algorithm based on Energy Consumption* (EBREC) is proposed by Han et al. [20] to ensure the lifetime of a network is extended. The BFS algorithm obtains a layered network structure and creates a routing path based on the energy usage of each network layer. It also applies multipath routing so that the shortest route is used for network communications, while the channel's redundancy is also assured. Their energy-balancing routing protocol was shown to result in efficient and balanced energy usage in WirelessHART networks, while deterring potential interruptions to a network's lifetime due to untimely energy exhaustion within a single sensor node.

Using the fan-shaped network structure, [21] propose the use of a hierarchical clustering routing protocol, referred to as *Adaptive Freeshape Clustering* (AFC), aimed at enhancing energy use and the routing approach in WirelessHART networks. This proposed solution includes the utilisation of a network area called the Reign of Interest (RoI), which is classed into different clusters shaped as fans, and these clusters extensively use a competition-based process to choose a Cluster Head (CH). Considering that Cell Nodes (CNs) and Cluster Members (CMs) are close to each other in this formation, and the CN manages the converge-casting of data, network nodes would not need direct communication with CHs. Thus, AFC aids in the reduction of energy usage for data converge-casting.

In the literature, different reinforcement learning models have been used for data delivery, energy consumption, and latency optimisation. One of these models is Q-Routing where the network nodes learn which of their neighbours delivers the best routes for a destination node. Besides, the nodes cannot select the routes. The *Q-Learning Reliable Routing with Multiple Agents* (QLRR-MA) approach is presented in [44], which built routing graphs in a centralised way using the Q-Routing model. This approach demonstrates that in a significant portion of cases, average network latency is reduced.

Similarly, [45] offers the *Q-learning Graph-Routing Lifetime Enhanced* (QGRLE) algorithm to improve lifetime, latency, and data packet delivery performance in IWSN metrics. This proposed algorithm periodically reconstructs the routing graph while data packet delivery metrics, lifetime, and latency, experience dynamic optimisation. Power resources are fully accounted for by considering the nodes' residual energy. After conducting simulations, the QGRLE algorithm was shown to be effective in that it improved lifetime and latency performance.

The *MultiPath Routing* (MPR) algorithm proposed in [46] provides industrial WMNs with low-cost planning, high data packet delivery, and low-level latency. This algorithm builds three main paths, each consisting of multiple nodes with different hop numbers. The multipath routing algorithm prioritises data transmission over the shortest path, but alternative paths are always ready in case of transmission errors. This multipath routing algorithm has been simulated on three existing graph-routing algorithms, including [44].

The MPR algorithm demonstrated a significant reduction in average network latency, the enhancement of expected network lifetime, and the improved ratio of data packet delivery.

2.8.2 Comparative Routing Algorithms

The following items are used by [2] with some items added, for this thesis, to categorise the routing algorithms proposed for IWSNs: algorithm objectives, architecture used, types of routing graphs (broadcast, uplink, and downlink), methods used to propose the algorithm, criteria for defining paths, provide path redundancy, and ways to present, implement, and measure the performance of these algorithms. The following list of items will be considered.

- **Objective.** This is the author's aim for the algorithm. This is necessary because routing algorithms are created to perform a variety of problem-solving tasks or to improve network characteristics.
- **Topology.** WirelessHART adopts a cluster, mesh or star topology, and can be used for large and scalable industrial control systems.
- **Graph-routing.** Three different graphs are described in WirelessHART: uplink, downlink, and broadcast. Each of these graphs possesses distinct characteristics, and researchers often chose specific types of graphs during the construction of their proposed graph-routing algorithms. This item aims to identify the types of graphs that a given proposal addresses.
- **Techniques.** A variety of approaches are used to problem-solve and enhance important network characteristics. This item demonstrates the approach used for IWSN algorithm development.
- **Criteria of paths.** Providing routing algorithms using pre-defined criteria enables the node to attempt effectively and correctly to establish its first connection. This process also allows scheduling algorithms to assign multiple 'links' in primary paths to allow nodes to make appropriate successive connection attempts, should the initial one fail.
- **Redundancy Path (RP).** This is a key feature designed to withstand the industrial environment that ensures an increased data packet delivery in IWSN communication. It focuses on the fail-safe that data packets will establish new pathways with neighbours if the primary path fails.
- **Presentation and implementation.** This item provides clarity regarding the authors' approach to communicating their proposal. This includes their means of presenting the proposal and the validation methodology used, such as simulations or testbeds.
- **Performance metrics evaluated.** This includes measurements involving algorithms performance, like packet loss, data packet delivery, and energy consumption.

Table 2.9 compares the algorithms concerning their constructed routes and graphs, route construction objectives, route definition criteria, techniques used, and the definition of a primary path for each node. It is observed that most algorithm work in the table do not implement all the graphs suggested in the WirelessHART network. This is because if one graph is generated, then other graphs can be generated following the same method. By focusing on graph-routing in this comparison, all these works provide path redundancy to improve data packet delivery. Furthermore, most of the work used mesh topology, one used cluster topology [21], another triangle topology [46], and three worked on hierarchical topology [40], [41],and [20]. Most algorithms aim to prolong the network lifetime, since the use of battery-powered wireless sensor nodes is predominant in IWSNs. Most of these previous studies employ the BFS algorithm to divide sensor nodes into layers, which determine the next hop or path taken. To select the best path, it uses the lowest number of hops along the path from the source sensor node to the gateway as the main criterion, because this metric reduces latency and the use of communication resources. Furthermore, two works in the table present how machine learning can be used to enhance graph-routing algorithms, with a focus on the latency and reliability of IWSN. Finally, due to the limitations of traditional techniques, most these previous algorithms disregard the achievement of balance between IWSN requirements.

Table 2.9 Objectives and methodology comparison of the routing algorithms of IWSNs.

Algorithm	Year	Objective	Type of Topology	Type of Routing	Technique	Criteria of paths	RP
ELHFR [15]	2009	Data packet delivery and stability.	Mesh topology	Graph-routing (Uplink)	BFS, RSL	Select next hop with Highest RSL	Yes
SRDR & SRDR-OPT [16]	2011	Data packet delivery and real-time communication.	Mesh topology	Graph-routing (Broadcast; Uplink; Downlink)	Greedy algorithm	Lowest No. of hop	Yes
Minimum hop load balancing graph-routing algorithm [17]	2013	Prolong the network lifetime	Mesh topology	Graph-routing (Uplink; Downlink)	BFS	Minimum-hop end-to-end communication	Yes
JRMNL [40]	2013	Prolong the network lifetime	Hierarchical topology	Graph-routing (Uplink; Downlink)	An exponent-weighted cost function	Minimum link cost function based on communication load, residual energy, and link transmission power	Yes
Re-add [18]	2014	Prolong the network lifetime	Mesh topology	Graph-routing (Uplink)	Link Selection based on energy model & link quality	Optimal link according to energy efficiency and quality of links and No. of hops	Yes
EBGR [41]	2015	Prolong the network lifetime	Hierarchical topology	Graph-routing (Uplink)	BFS	Lower traffic load	Yes
GRLM [19]	2016	Prolong the network lifetime	Testbed topology	Graph-routing (Uplink; Downlink)	Integer Programming; Linear Programming; Greedy heuristics.	Highest flow Priority	Yes
CAR [42]	2018	Real-time communication	Testbed topology	Graph-routing (Uplink)	By incorporating conflict delays into the routing decisions	Small conflict delay	Yes

Algorithm	Year	Objective	Type of Topology	Type of Routing	Technique	Criteria of paths	RP
DiGS [43]	2018	Scalability and data packet delivery	Mesh and testbed topology.	Graph-routing (Uplink)	Extending RPL, the routing protocol for low-power IPv6 networks; Scheduling approach	Smallest accumulated expected transmission count (ETX)	Yes
EBREC [20]	2019	Prolong the network lifetime	Hierarchical topology	Graph-routing (Uplink)	BFS	Highest energy	Yes
AFC [21]	2019	Save and balance of energy consumption	Cluster topology	Graph-routing (Broadcast)	Clustering	Remaining Energy and quality of link connectivity	No
QLRR-MA [44]	2020	Latency, energy consumption, and data delivery	Mesh topology	Graph-routing (Uplink)	Machine Learning. Reinforcement Learning (Q-Learning)	Each node has learning agent, rewards given to agent when its average data latency decreases.	Yes
QGRLE [45]	2021	Latency, prolong the network lifetime and data packet delivery	Mesh topology	Graph-routing (Uplink)	Machine Learning. Q-learning	Lowest costs based on average No. of hops; RSSI of connection; residual energy. Q-learning to alter the weights and reconstruct graph	Yes
MPR [46]	2022	Latency, prolong the network lifetime and data packet delivery	Triangle topology	Multipath	Planning and deployment algorithm	Shortest hops	Yes

Table 2.10 presents characteristics about the presentation, implementation, and performance matrices of the routing algorithms of IWSNs. The algorithms are all implemented in simulated environments, which are OMNET++, MATLAB, Cooja, NS-2 and NS-3. Only two proposals present real experiments in the table. Most work presents pseudo code, flowcharts, or both to explain their algorithms. The performance metrics evaluated are for the lifetime of the network, Packet Delivery Ratio (PDR), transmission delay, packet loss ratio, and degree of imbalance energy.

Table 2.10 Implementation of different algorithms in the state of the art.

Algorithm	Presentation	Implementation	Performance metrics
ELHFR [15]	No pseudo code is presented	Simulation in OMNET++	**Reliability:** Packet lost ratio; Throughput; **Latency:** End to End delay
SRDR & SRDR-OPT [16]	Presents pseudo code for the algorithm	Own simulator	**Reliability:** percentage of reachable nodes; Rate of success routes construction; recovery overhead to regain connectivity; **Real-time:** Average latency
Minimum hop load balancing graph routing algorithm [17]	Presents flowchart for the algorithm	Own simulator	**Lifetime:** Energy consumption for each sensor node
JRMNL [40]	No pseudo code is presented	MATLAB simulation	**Lifetime:** Network lifetime; Average transmission power per route
Re-add [18]	Presents pseudo code for the algorithm	No simulation platform details were given	**Lifetime:** No. of successful transmissions; No. of packet received; Average residual energy; Average latency; Packet loss ratio
EBGR [41]	Presents flowchart for the algorithm	MATLAB simulation	**Lifetime:** Energy consumption; Network lifetime
GRLM [19]	Presents pseudo code for the algorithm	Real experiment	**Lifetime:** Network lifetime; Delivery Ratios of Flows
CAR [42]	Presents pseudo code for the algorithm	Experiments on a WSAN testbed; Own simulator	**Real-time:** Acceptance ratio; End-to-end delay; Execution time
DiGS [43]	Presents pseudo code for the algorithm	Contiki based on Cooja simulation; physical testbed	**Reliability:** End-to-end reliability; end-to-end latency; Energy consumption per received packet
EBREC [20]	Presents pseudo code for the algorithm	MATLAB simulation	**Lifetime:** Network lifetime; Remaining energy
AFC [21]	No pseudo code is presented	Contiki 3.0 based on Cooja simulation	**Save Energy:** Network lifetime; Degree of energy imbalance

Algorithm	Presentation	Implementation	Performance metrics
QLRR-MA [44]	Presents pseudo code for the algorithm	NS-2 simulation	**Latency:** Average network latency; **Energy:** Expected network lifetime; **Reliability:** Packet delivery ratio; Percentage of reliable nodes
QGRLE [45]	Presents pseudo code and flowchart for the algorithm	NS-3 simulation	**Latency:** Accumulative delay; average delay; **Reliability:** packet reception ratio; **Lifetime:** expected lifetime
MPR [46]	Presents pseudo code for the algorithm	NS-2 simulation	**Reliability:** Packed delivery rate; **Lifetime:** energy consumption-based network lifetime; **Latency:** average network latency

2.9 Summary

While Chapter 1 offered an introduction to IWSNs and the motivation for conducting this thesis, Chapter 2 has completed the presentation of the WirelessHART description as widely standard for IWSNs. This helps clarify the characteristics, architecture, and open system interconnection model of the WirelessHART network.

Moreover, this chapter has presented a literature review of the most important graph-routing algorithms proposed until 2022 in WirelessHART network, and has discussed the techniques and criteria used to build graph routes to achieve desired objectives. From this summary of the existing research, it can be observed that the work to date lacks the application of clustering and optimisation techniques to build a reliable graph-routing algorithm that balances energy consumption.

This thesis will proceed into Chapter 3 by first addressing the effect of changing the topology of the WirelessHART network in the basic graph-routing algorithm by using clustering techniques. Then it will use optimisation techniques to select graph-routing paths based on the requirements of the IWSNs. Therefore, these two elements are described in detail in the next chapter.

Chapter 3

Clustering, Optimisation Techniques, and Preliminaries

3.1 Overview

This chapter provides the theoretical descriptions of previous research concerning clustering and optimisation techniques, which are used in this thesis to build reliable graph-routing algorithms in the WirelessHART network. Section 3.2 presents the first technique, used to design the novel topology of the WirelessHART network, the concept of clustering in wireless networks and the methods that are used to establish clustering. Section 3.3 discusses the second technique used to construct graph paths for graph-routing in the WirelessHART network. This section covers the scope, general classifications, and definitions of optimisation techniques, as well as a comprehensive literature comparison of the literature on how optimisation techniques are applied to enhance wireless networks. Because of its importance in the thesis, the Covariance Matrix Adaptation-Evolution Strategy (CMA-ES) algorithm is thoroughly detailed in Section 3.4. The final section of this chapter contains some of the preliminary material used throughout the thesis.

3.2 Clustering Techniques

A clustering technique is a common strategy to design a network's topology, which organises wireless sensor nodes into groups known as *clusters*, as shown in Figure 3.1. The literature review has demonstrated that it is an energy-saving strategy. Therefore, clustering a significant role in prolonging the lifetime of wireless networks and increasing data packet delivery [47], [48]. Because the sensor nodes in each cluster elect a leader known as the Cluster Head (CH), denoted by the red circle in Figure 3.1, the remaining sensor nodes are referred to as *Cluster Members* (CMs), indicated by blue circles. Since CMs transmit their data packets to their respective CHs, over shorter distances communication within the cluster is called *intra-cluster communication*. Each cluster's CH is responsible for sending and receiving data packets to and from the gateway (G_w), either as a single hop if the G_w is within its communication range, or as a multi-hop with other CHs if the G_w is outside its communication range – this is referred to as *inter-cluster communication*. Consequently, three primary steps form the clustering procedure: cluster formation, CH selection, and data communication.

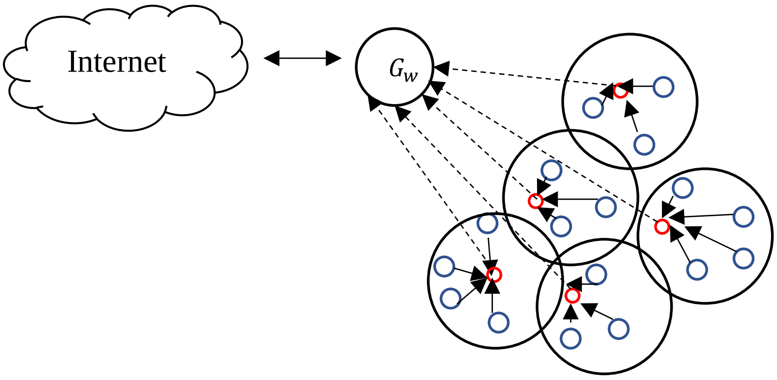


Figure 3.1 The structure of clustering in wireless networks.

3.2.1 Objectives of Clustering

Clustering objectives help meet the needs of different applications in wireless networks. The following are the primary objectives of using clustering in wireless networks:

- **Scalability.** This is an objective that describes how well a network can add more wireless sensor nodes, which increases the communication overhead of the network, without affecting the network's performance. This challenge can be addressed through wireless network clustering techniques, such as applying an energy-efficient distributed clustering algorithm [49] for all levels of clusters in the network area, or using the hierarchical geographic method [50] to keep overhead per data packet constant.
- **Energy consumption.** As the most important objective in wireless networks, clustering techniques aim to reduce energy consumption and thus prolong the network's lifetime. The most energy-consuming task in wireless networks involves transferring data packets from sensor nodes to the gateway [8], because the sensor nodes in wireless networks are small and battery powered. In addition, to their basic functions (for example, sensing and computation) nodes act as routers when delivering data packets to the gateway. Because of their limited communication range, not all sensor nodes can communicate directly with the

gateway, so they forward data packets to other nodes within their communication range. To reduce energy consumption hierarchical or layered clustering techniques have been proposed, dividing the network into different layers [47], [51], [52]. CH selection is also important, due to CHs consuming more energy than CMs, with methods, like fuzzy logic [53], [54] AI [55], and the heuristics method [56] all explored.

- **Fault tolerance.** Faulty nodes must be dealt with to improve data packet delivery and stability. Depleted batteries at some sensor nodes, collisions, radio interference, or environmental factors can all result in faulty nodes. Fault-tolerant clustering techniques try to avoid connectivity or coverage impairments and the loss of data packets using a variety of detection strategies [57]–[59]. This problem is challenging, as network applications may experience data loss due to excessive detection delay.
- **Load balancing.** It is important to ensure uniform load distribution between the sensor nodes in the network area as this helps avoid unbalanced energy consumption, network congestion, data loss, and inefficiency in supporting real-time and data-intensive applications. Clustering techniques can be used to balance the network load by varying the number of clusters or CHs [60].
- **Data aggregation/ fusion.** Since a large number of sensor nodes sense the same data in the physical environment, there is a greater chance of data redundancy [47]. In clustering, the CH aggregates all data received from its CMs and forwards the aggregated data to the gateway. This can significantly reduce the number of transmissions which improves throughput and decreases energy consumption in the wireless network.
- **Security.** Clustering techniques can be used to improve the security of a wireless network through the introduction of different techniques to resolve attacks and detect malicious nodes. For example, detecting malicious nodes and preventing these being CHs [61], or establishing periodic authentication between CHs and CMs to establish secure channels [62].
- **Stable network topology**. The CH maintains information concerning its CMs, such as node ID, location, and energy level. When a CM dies or moves to another cluster, these changes are immediately registered and communicated to the NM. Re-clustering can then maintain the network topology effectively.

3.2.2 Methods for Establishing Clusters

In a wireless network, clusters can be created using two approaches [24], [63]:

- Grouping sensor nodes and selecting one of them as a CH. The grouping may be based on various criteria, such as the number of sensor nodes, the size of the cluster, the number of clusters, or physical proximity.

- Identifying CHs and inviting other sensor nodes to join a neighbouring CH. This technique is based on parameters such as the member nodes' proximity to the CH and/or the CH's proximity to the gateway.

3.2.3 The Fundamental Methods of CHs Election

The following are some of general approaches that can be taken when choosing CHs [24], [63]:

- **Energy-based.** Sensor nodes with high energy and resources are determined as CHs [63]. The problem with this approach is that most wireless networks are limited in resources and homogeneous (i.e., consisting of sensor nodes with the same capabilities such as battery power or sensing range). Therefore, this technique may be ineffective in many situations. Furthermore, even if a sensor node with high energy can be identified and selected as a CH, such as in a heterogeneous network, being a CH for an extended period can quickly deplete the node's power and result in its death.
- **Randomness.** This is a solution that attempts to spread CH responsibility between sensor nodes; for example, a Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm is the most often used clustering technique [47]. Even though this approach works well in homogenous networks, any change or imbalance in wireless networks can cause serious runtime problems in some CHs, such as constant energy consumption or an unbalanced use of resources.
- **Selection of a CH based on the network and node configurations.** Here CHs are selected based on a variety of criteria, including the resources available, the number of neighbours, the location, etc.
- **Centralised CH selection.** Here parameters used for selecting CHs are collected in a central node (usually the gateway) and compared, evaluated, and processed for selecting CHs. Due to the comparison of all sensor nodes, centralised approaches will generate universal results but large or highly complex networks may have a high computational overhead.

3.3 Optimisation Techniques

Optimisation techniques are a set of mathematical operations that are written as algorithms and used to find the best possible solution to complicated optimisation problems. These algorithms use the objective function f, which, depending on the optimisation problem, can be maximised or minimised. In other words, difficult or complex optimisation problems usually have several potential solutions, so optimisation algorithms evaluate objective functions to define which candidate solution is the best one. These problems may have a single-objective function or a multiple-objective function [64].

The objective function of any optimisation algorithm can be expressed in general as

$$Maximise/Minimise \ \ f_1(x), f_2(x), \dots \dots \dots, f_N(x),$$
$$optimise \ subject \ to \ \ x = [x_1, x_2, \dots \dots, x_n] \qquad (3.1)$$

Where $f_1, f_2, \dots \dots \dots, f_N$ are the objectives sought by the optimisation algorithm, whether to maximise or minimise. When $N = 1$, it is referred to as a single-objective optimisation, while a multiple-objective optimisation occurs when $N \geq 2$. Here, x is a search space. A search space can be represented as a vector of values corresponding to different search points. An n-dimensional search point is $[x_1, x_2, \dots \dots, x_n]$, where the i^{th} search point is denoted by x_i. Any value of x_i from among all points in the search space x that minimises the objective function is called a *solution* or *minimiser*. In Figure 3.2 an example of an optimisation problem, observe that the minimum x^* is the best solution in the local search space, but lower points may exist if a global search is conducted. Therefore, every optimisation algorithm needs to address the exploration and exploitation of a search space. *Exploration* (global search) is the process of looking at entirely new regions of a search space, whilst *exploitation* (local search) is the process of looking in regions of a search space in order to find good solutions as quickly as possible.

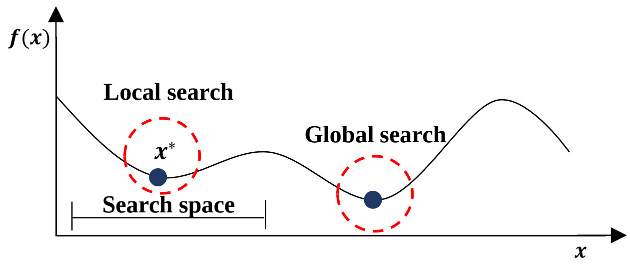


Figure 3.2 A one-dimensional optimisation problem.

However, a *single-objective function* means finding the best solution for a specific objective, where all the points converge on a single point, and as a result that point is the best solution. In contrast, the value of *multiple-objective functions* depends on two or more conflicting objectives, the points converge at two or more points, and the best solution through a trade-off among them should be selected based on the optimisation technique used. As an example, from a wireless network perspective, if a wireless network just needs to minimise energy consumption, this is a single objective, so there will be one solution. But if the network needs maximum data packet delivery, minimum delay, and minimum energy consumption, then in it has multiple objectives to achieve and will need a trade-off between them based on the wireless network's requirements to select the best possible solution. Optimisation techniques have been used to solve problems in many different fields, such as finance, engineering design, system and database design, and wireless networks [25]. Therefore, no single method is available for solving all optimisation problems efficiently. However, several methods have been developed to solve these optimisation problems, which may be categorised into two basic techniques: deterministic optimisation and stochastic optimisation [64], as illustrated in Figure 3.3.

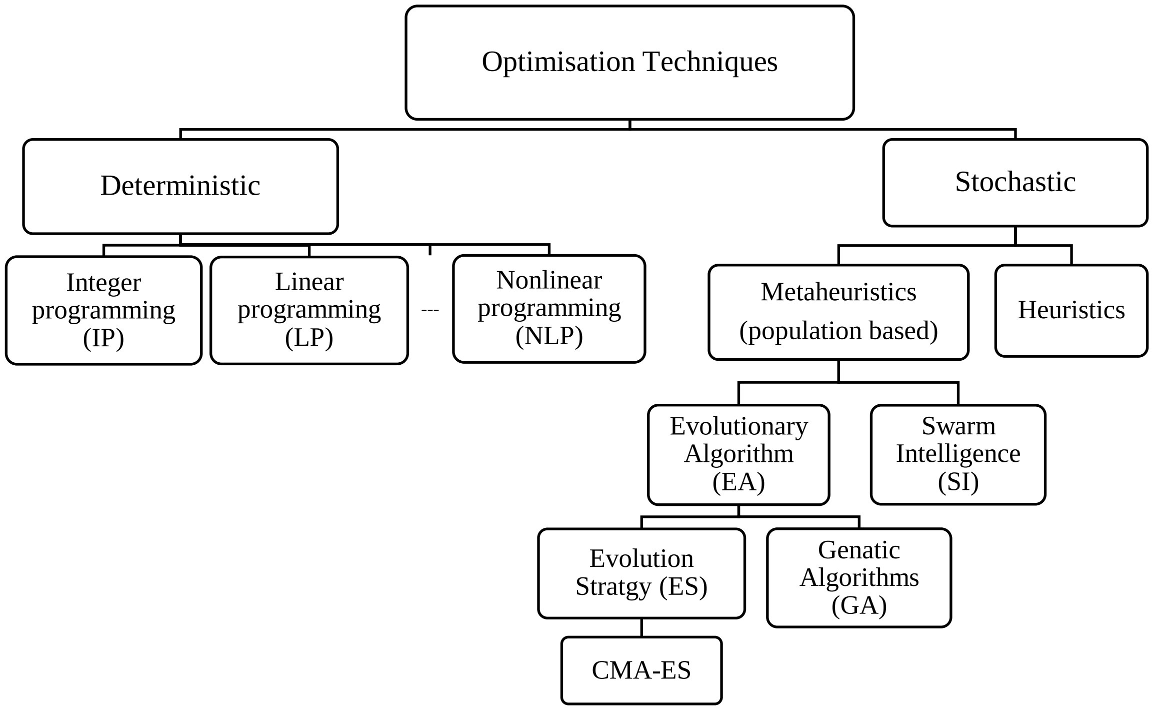


Figure 3.3 The overall classification of optimisation techniques.

Deterministic optimisation techniques, such as Integer Programming (IP), Linear Programming (LP), or Non-Linear Programming (NLP), are used to find the best solution to a specific problem and provide a theoretical guarantee of its solution. However, deterministic algorithms can have problems resolving extremely complex and difficult optimisation functions, and consequently can take a long time to solve. This is due to large searching spaces and intricate problem structures [65].

Stochastic optimisation techniques aim to reach proper solutions to multiple problems using random search processes. Stochastic optimisation techniques are more flexible and efficient than deterministic approaches because the execution times required to find the best solutions can be controlled. Stochastic processes are divided into heuristics and metaheuristics [64]. Heuristics are strategies employed to solve a particular problem without guaranteeing a global search (exploration), whereas metaheuristics are generic strategies adapted to solve multiple problems [66]. These techniques have been widely used to solve optimisation problems in WSNs that exhibit high computational complexity [25].

Glover [67] derived the term 'metaheuristic' from two Greek terms, 'meta' meaning 'high level' and 'heuristic' meaning 'solution or technique'. A metaheuristic, then, is a high-level technique that intelligently uses heuristics to effectively find the best potential solutions for problems that have no deterministic solution. Some metaheuristic algorithms prefer local search (exploitation), while others prefer global search (exploration). In other words, it is necessary to explore the search region at the beginning of each metaheuristic algorithm to identify better solutions. Thus, the algorithm should have a high exploration ability. But some metaheuristic algorithms have the exploitation ability to enhance the solution and get closer to the best solution as they approach the end of the computation process [68]. When formulating the objective function f, several parameters or metrics that are associated with the problem statement are considered. Population-based metaheuristic algorithms can be further classified into two major classes, namely: Swarm Intelligence (SI) and Evolutionary Algorithm (EA).

SI is a form of intelligence that is defined as emulating the behaviour of certain organisms that allows them to identify a source of food or track prey. Some examples of the most widely used swarm-based algorithms in the WSNs are: Particle Swarm Optimisation (PSO) [69], Artificial Bee Colony (ABC) [70], Ant Colony Optimisation (ACO) [71], Grey Wolf Optimiser (GWO) [72], Whale Optimisation Algorithm (WOA) [73], Cuckoo Search (CS) algorithm [74], and Firefly Algorithm (FA) [75].

EA is a class of population-based optimisations inspired by natural selection. Natural selection advances the theory that individuals with traits that are beneficial to their survival can survive through the generations, with each generation passing down their survival characteristics to the next. Evolution happens gradually through the process of selection, and the population continually grows better adapted to the environment in which it lives. EA uses biological-inspired mechanisms, such as selection reproduction, recombination, and mutation, as search operators. Candidate solutions to the problem play the roles of individuals in a population, and the objective function determines the quality of eventual solutions. Although different variants of EA exist, notably Genetic Algorithms (GA) and Evolution Strategies (ES), they all share the same fundamental structure shown in Figure 3.4 to find the best solution [76].

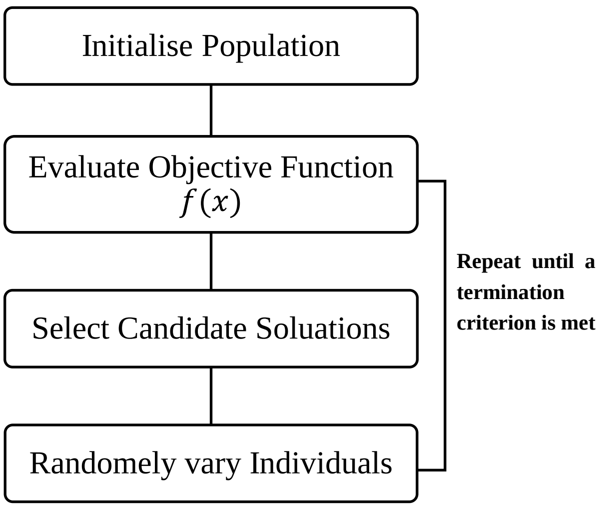


Figure 3.4 Basic Steps of EA.

The EA is applied in a loop, and an iteration of the loop is called a *generation*. In each generation, variation often generates new individuals (candidate solutions), typically in a stochastic manner. The objective function $f(x)$ for an individual then returns a numeric value, generally referred to as *fitness*, signifying the quality of the solution presented by that individual. Individuals with a high fitness level are those who offer the best solutions to the problem at hand. The selection process targets highly fit individuals for survival as parents, based on their fitness or objective function value. Figure 3.4 depicts the iterative process that occurs until either an acceptable solution is found, or a predetermined number of generations have elapsed.

GAs use a chromosome-like data structure to iteratively improve candidate solutions to a problem by applying recombination operators that are designed to keep important information. The first step in implementing GA is to generate a population of 'chromosomes'. These chromosomes are then evaluated according to their objective functions, using operators borrowed from natural genetics [77].

ES was proposed in the mid-1960s by Rechenberg [78] and was further developed by Schwefel [79] to optimise candidate solutions composed of real-value parameters. Selection, mutation, and recombination are applied to ES as search operators, as shown in Figure 3.5.

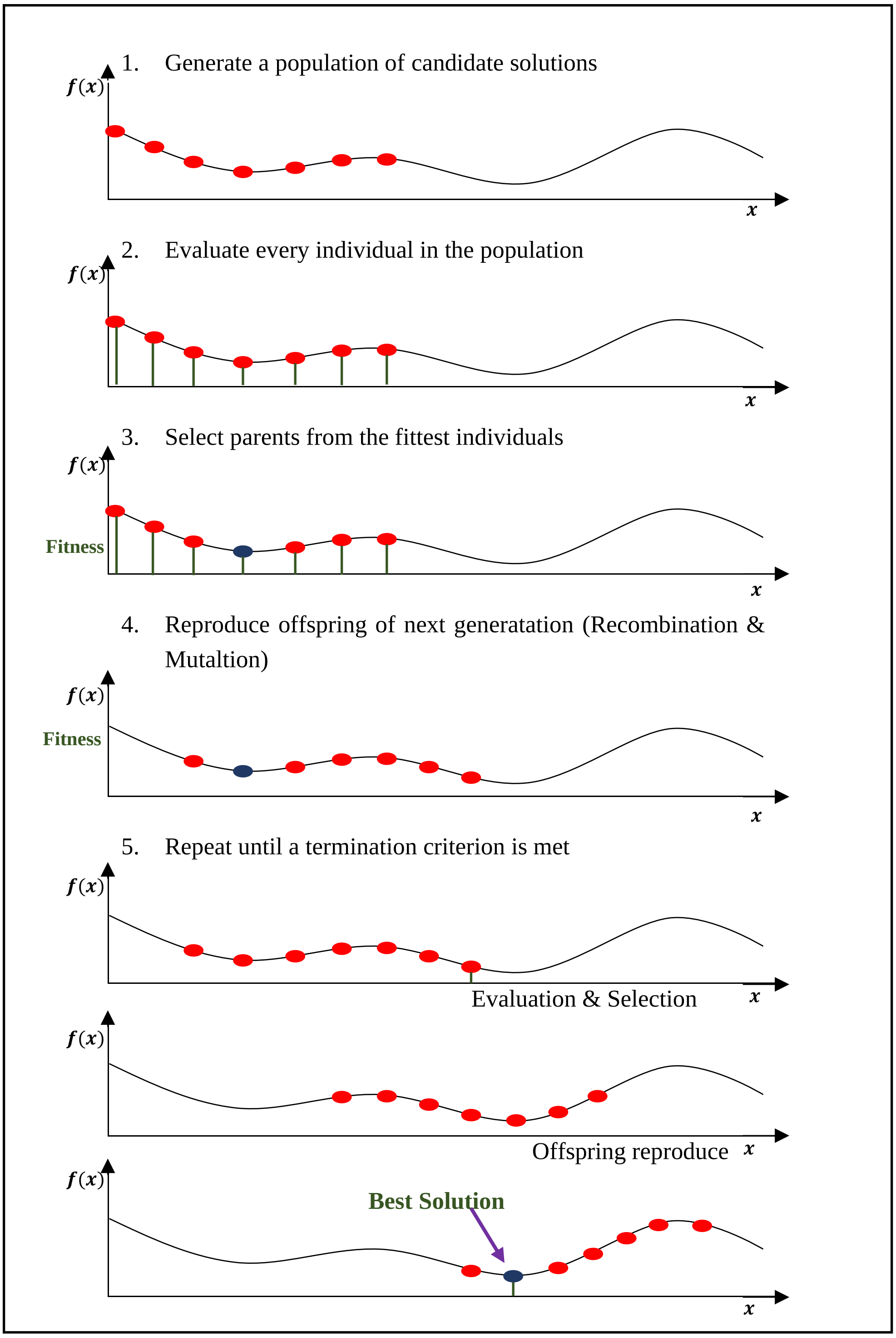


Figure 3.5 Basic Steps of ES.

ES algorithms monitor and update the mutation operator, which is adding (normally distributed) random values to each component of an individual dynamically at each generation. At this point their objective function is evaluated. Once the offspring is generated by the process of mutation, the selection of candidates to parent the next generation is based on the fitness ranking of individuals in the current offspring. This procedure is iterated until the objective is fully optimised.

One of the most recent and powerful versions of the ES algorithm, which was used in this thesis, is the *Covariance Matrix Adaptation-Evolution Strategy* (CMA-ES) [80], proposed by Nikolaus Hansen and Andreas Ostermeier in 2001. It is noteworthy that the original name of the algorithm was *Completely Derandomized Self-Adaptation in Evolution Strategies* and that it is almost the same algorithm. Furthermore, it has been adopted as a standard tool for continuous optimisation in many research laboratories [26] and industrial environments worldwide. CMA-ES has also seen widespread application in a variety of domains, including WSNs [81], deep neural networks [82], security of networks [83], and inverse reinforcement learning [84].

3.3.1 Application of Metaheuristic Algorithms in the Routing of Wireless Networks

This section analyses state-of-the-art research that has been conducted on metaheuristic algorithms to improve routing algorithms in WSNs. These research works have been published in journals, with a focus on the period 2015–2022, as shown in Table 3.1. The following items are used by [25], with some items added to present the metaheuristic algorithms that have been used to enhance wireless networks, the goals of using it, the tasks that metaheuristic algorithms undertake to improve the proposed routing algorithm, and the simulation platform.

Table 3.1 Optimisation enhancement techniques used in routing for wireless sensor networks.

Algorithm	Research Ref.	Objective	Task of Optimisation Routing Techniques	Implementation Tools
GA	[85]	Prolong the network lifetime	Selecting the paths and CHs	MATLAB R2012b and C programming language
	[86]	Prolong the network lifetime	Selecting the paths	MATLAB R2010a simulation
	[87]	Energy efficiency and prolong the network lifetime	Selecting the paths	MATLAB R2014a and C programming language + OMNeT++ simulation
	[88]	Ensure the QoS requirements	Selecting the paths	MATLAB is used along with NS-2 simulation
Multiobjective GA + CS	[89]	Balance of energy consumption between sensor nodes	Selecting the paths and CHs	MATLAB R2020 simulation

Algorithm	**Research Ref.**	**Objective**	**Task of Optimisation techniques in routing**	**Implementation Tools**
GA and GWO	[90]	Energy efficiency and prolong the network lifetime	Selecting the paths	MATLAB simulation
FA	[91]	Prolong lifetime and throughput of the network	Selecting the paths	NS-2 simulation
ACO	[92]	Prolong the network lifetime	Selecting the optimal link cost	No simulation platform details were given
	[93]	Energy balance and prolong the network lifetime	Selecting the paths	NS-2 simulation
	[94]	Prolong the network lifetime	Selecting the paths	No simulation platform details were given
	[95]	Energy balance and prolong the network lifetime	Selecting the paths	No simulation platform details were given
	[96]	Prolong the network lifetime	Selecting the paths	NS-2 simulation
	[97]	Prolong the network lifetime	Selecting the paths	MATLAB R2013a simulation
	[98]	Energy balance and prolong the network lifetime	Selecting the paths	C++
	[99]	Ensure the QoS requirements, security guarantees and prolong the network lifetime	Selecting the paths	MATLAB simulation
	[100]	Reduce the energy consumption of the network	Selecting the paths	VC++ programming language
	[101]	Prolong the network lifetime	Route discovery	NS-2 simulation
	[102]	Save energy consumption	Selecting the paths	MATLAB simulation
	[103]	Reduce the energy consumption	Selecting the paths	Java programming language
	[104]	Prolong the network lifetime	Selecting the paths	No simulation platform details were given
	[105]	Prolong the network lifetime	Selecting the paths	No simulation platform details were given
	[106]	Energy efficient and prolong the network lifetime	Neighbour node discovery	NS-2 simulation

	[107]	Reduce the energy consumption	Selecting the paths	MATLAB simulation
	[108]	Energy efficient and network security	Selecting the paths	NS-2 simulation
FA and ACO	[109]	Prolong the network lifetime	Selecting the paths and CHs	MATLAB R2016b simulation
ACO and GSO	[110]	Energy efficient	Selecting the paths	NS-2 simulation
BOA and ACO	[111]	Prolong the network lifetime	Selecting the paths and CHs	MATLAB R2018a simulation
PSO	[112]	Data packet delivery, network coverage and energy consumption	Selecting the CHs and paths	OMNeT++ platform
	[113]	Prolong the network lifetime	Distribute the traffic load over the CHs	MATLAB R2012b and C programming language
	[114]	Prolong the network lifetime	Selecting parent sensor nodes and the paths	MATLAB simulation
	[115]	Energy efficient and energy balance	Selecting the paths	MATLAB simulation
	[116]	Prolong the network lifetime	Selecting the optimal rendezvous points	MATLAB simulation
	[117]	Ensure the QoS requirements	Selecting the paths	NS-2 simulation
ABC	[118]	Prolong the network lifetime	Selecting the paths and CHs	OMNeT++ platform
	[119]	Prolong the network lifetime	Selecting the paths and CHs	Nature-inspired tool for sensor simulation (NITSS)
	[120]	Reduce the energy consumption and prolong the network lifetime	Selecting the paths and CHs	MATLAB simulation
WOA	[121]	Energy efficient and prolong the network lifetime	Selecting the paths	NS-2 simulation
GWO, TSA and FGSA	[122]	Prolong the network lifetime	Selecting the paths and CHs	NS-2 simulation

The table 3.1 shows the following:

- Most of the research used optimisation algorithms in wireless networks to prolong the lifetime of the network through reduced energy consumption, energy efficiency, or a balance of energy consumption between sensor nodes in the network area. Three of the research studies focused on QoS requirements, and one of them also included a security guarantee in its objective.

- Most of the proposed routing algorithms use optimisation to find the best routing paths, depending on specific criteria defined by objective functions. Some researchers use optimisation to find the best CHs to enhance the network's performance. One [106] used optimisation to discover the neighbouring nodes of each sensor node, and another [116] to select the optimal rendezvous points based on the number of data packets received from other sensor nodes. Furthermore, optimisation is used to assign a smaller number of sensor nodes for each cluster to reduce traffic load over the CHs, as present in [113].
- Some work incorporates two or more optimisation algorithms that are usually used for determining optimal paths and CHs, based on criteria specified in its objective functions.
- MATLAB, followed by NS-2, were the predominant simulation platforms for the majority of the previous research. OMNeT++, C/C++, and Java were used in some of the work.

3.3.2 Do We Need Optimisation Techniques in WirelessHART Networks?

Optimisation is required to build a well-functioning design, as per the requirements of the network. Wireless network optimisation is typically required to achieve a desired objective, such as reducing energy consumption or prolonging the network's lifetime. As mentioned in the previous sub-section, metaheuristic algorithms are widely used to improve the performance of routing algorithms in WSNs. This is because of their ability to design routing algorithms effectively and easily under special requirements and environmental conditions, enabling the best designs and strategies and thereby improving the overall performance of wireless networks.

In the WirelessHART network, some of the major challenges in selecting the best graph paths of the graph-routing algorithm to achieve the required objectives are balancing energy consumption between wireless sensor nodes, reducing End-to-End Transmission (E2ET) time, and increasing data packet delivery. With more than one goal present, all these goals need to converge to find the best possible solution. In this context, optimisation or high-level procedures are required. Optimisation techniques to find the best solution in a centralised manner may, therefore, be useful for IWSN and future IIoT protocols.

3.4 CMA-ES

CMA-ES is a derivative-free, efficient stochastic method for black-box optimisation [80]. It relies on a multivariate normal (Gaussian) distribution to sample a population (candidates for solutions), iteratively, instead of using the whole population [123]. This is undertaken first to explore the search space and then to evaluate it using the objective function f. Therefore, the need for a finite-dimensional search space is explicit when using CMA-ES to find potential solutions. A multivariate normal distribution is an n-dimensional normal distribution, which can be denoted as $N(m, C)$. Here, $m \in R^n$ is the mean vector and C is the symmetric positive covariance matrix, which determines the centre and shape of the distribution. This normal distribution is the most convenient way to generate candidate solutions because it is isotropic and rotationally invariant. It is also the most stable distributor in R^n. These factors make the normal distribution an especially appealing candidate for a randomised search.

3.4.1 Black Box Optimisation

A black box search scenario is used to minimise or maximise an objective, fitness, or cost function. A black box search algorithm is outlined in Figure 3.6, where the algorithm continuously repeats the process of sampling-evaluation-update.

$$f: x \in R^n \to R, \quad x \to f(x) \text{ is objective function} \tag{3.2}$$

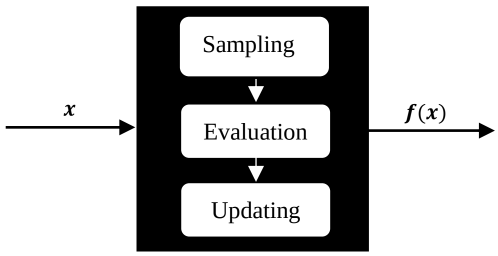


Figure 3.6 Black box optimisation.

The objective is to find one or more search points (candidate solutions), $x \in R^n$ with a function value, $f(x)$, that is as small as possible. Black box optimisation refers to the situation in which the function values of evaluated search points are the only accessible information on f.

3.4.2 Steps of the Original CMA-ES Algorithm

The original CMA-ES algorithm has the following steps [80], [123]–[125].

1. **Initialisation Parameters**

The first step in CMA-ES is to initialise the parameters, as shown in [80]. Table 3.2 defines what each parameter used in the algorithm is and what its initial value is, whether initialised randomly or based on prior knowledge. Each of these parameters can be calculated as a step in the algorithm for the next generations, as discussed in the following five sections.

Table 3.2 Initialisation parameters of CMA-ES [80].

Parameters	Definition	Initial Value
$\boldsymbol{\lambda}$	Population size, number of offspring or sample size or candidate solutions that will be generated and evaluated in each iteration.	$\boldsymbol{\lambda} = (4 + round(3 * log(n)) * 10$; Where $\lambda \geq 2$
$\boldsymbol{g}$	The number of generations g iteration number.	Typically, $\boldsymbol{g} \geq 10$, $\boldsymbol{g} \in N_0$ where N_0 natural numbers including zero
$\boldsymbol{m}^{\boldsymbol{g}}$	Mean vector at generation g represents the center of the multivariate normal distribution used to generate candidate solutions.	$\boldsymbol{m} \in \boldsymbol{R}^{\boldsymbol{n}}$, where $\boldsymbol{R}$ positive real numbers
$\boldsymbol{C}$	The covariance matrix C is symmetric and positive definite, and determines the shape and spread of the distribution.	$\boldsymbol{C} = I$, is the identity matrix, $\boldsymbol{C} \in \boldsymbol{R}^{\boldsymbol{n \times n}}$
$\boldsymbol{p}_{\boldsymbol{c}}^{\boldsymbol{g}}$	The length of evaluation path for C at generation g.	$\boldsymbol{p}_{\boldsymbol{c}}^{\boldsymbol{0}} = 0$, where $\boldsymbol{p} \in \boldsymbol{R}^{\boldsymbol{n}}$
$\boldsymbol{p}_{\boldsymbol{\sigma}}^{\boldsymbol{g}}$	The length of evaluation path for σ at generation g.	$\boldsymbol{p}_{\boldsymbol{\sigma}}^{\boldsymbol{0}} = 0$, where $\boldsymbol{p} \in \boldsymbol{R}^{\boldsymbol{n}}$
$\boldsymbol{E}[\lVert\boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I})\rVert]$	The expected length of a $(0, I)$ N-normally distributed random vector.	$\boldsymbol{E}[\lVert\boldsymbol{N}(\boldsymbol{0}, \boldsymbol{I})\rVert] = \sqrt{n} \times (1 - 1/(4 \times n) + 1/(21 \times n^2)$.
$\boldsymbol{VarMin}$	Lower bound of decision variables.	$\boldsymbol{VarMin} = 0$
$\boldsymbol{VarMax}$	Upper bound of decision variables.	$\boldsymbol{VarMax} = 100$
$\boldsymbol{\sigma}$	Step size controls the step length in search space.	$\boldsymbol{\sigma}^{\boldsymbol{0}} = 0.3 * (VarMax - VarMin)$;

2. Generating Candidate Solutions

In each iteration of CMA-ES, λ candidate solutions x_i are generated from a multivariate normal distribution [126] with mean vector m and covariance matrix C. Candidate solutions x_i can be generated using a standard method in equation (3.3) [124].

$$x_i^{g+1} \sim m^g + \sigma^g N_i(0, C^g) \qquad for\ i = 1,2, \dots \dots \dots, \lambda \tag{3.3}$$

Where x_i^{g+1} represents the i^{th} search point (candidate solution, individual, offspring or object parameters/variables) generated at generation $g + 1$. m^g and σ^g are mean vector of the search distribution and 'overall' standard deviation (step size) at generation g, respectively. $N_i(0, C^g)$ is a multivariate normal distribution with a zero mean and a covariance matrix C^g of the search distribution at generation g.

3. Evaluation Objective Function

The objective function $f(x)$ of the candidate solutions x_i are then evaluated. The objective function should be a scalar-valued function that assigns a fitness value to each candidate solution, where higher values indicate better solutions. The fitness values are used to guide the search process and determine which candidate solutions should be selected for the next iteration.

4. Calculation of Weights

The purpose of weighting is to give more weight to the better candidate solutions and less weight to the weaker candidate solutions. This helps guide the search process toward the best solution. Therefore, a weight w_i is assigned to each candidate solution x_i based on its fitness value f_i as follows [123].

$$w_i \ = \ exp(-f_i \ / \ 2) \ / \ sum(exp(-f_j \ / \ 2)) \tag{3.4}$$

Where f_j is the fitness value of candidate solution x_j.

5. Selection And Recombination: Update Mean Vector

In this step, calculating the mean vector $m^{(g+1)}$ for each generation is updated using the weighted average of the candidate solutions to reflect the current best candidate solutions. By updating the mean vector $m^{(g+1)}$ in each iteration, the CMA-ES algorithm can adapt to changes in the distribution of these candidate solutions and effectively converge towards the global minimum/maximum of the objective function. The update of the mean vector $m^{(g+1)}$ is performed using the following equation [80].

$$m^{(g+1)} = \sum_{i}^{\mu} w_i \, x_{i:\lambda}^{(g+1)} \tag{3.5}$$

The number of effective solutions, μ_{eff}, is derived from

$$\mu_{eff} = 1 \Big/ \sum_{m=1}^{\mu} w^2 \tag{3.6}$$

6. Update of the Covariance Matrix

The purpose of this phase is to compute the update of the covariance matrix C for generation $g + 1$ candidate solutions, where the covariance matrix of the next generation $g + 1$ is dependent on the learning curve derived from the covariance matrix g.

In the following, the updated covariance matrix, $C^{(g+1)}$ at generation $g + 1$, of the CMA-ES is described by two steps:

First, the evolution path (called the *cumulation*) $p_c^{(g+1)}$, which is the sequence of steps the strategy takes over several generations it is calculated by [123]

$$p_c^{(g+1)} = (1 - c_c) \times p_c^g + \sqrt{c_c(2 - c_c)\mu_{eff}} \tag{3.7}$$

Where c_c denotes the learning rate for the cumulation regarding the rank-one update of c,

$$c_c = (4 + \mu_{eff})/(4 + g + 2 \times (\mu_{eff}/g)) \quad (3.8)$$

Second, $p_c^{(g+1)}$ is used to generate the $C^{(g+1)}$ according to the equation (3.9) [123]:

$$C^{(g+1)} = \left(1 - c_1 - c_\mu\right) \times C^g + c_1 \times \left(p_c^{(g+1)} + \left(c_c \times (2 - c_c)\right) \times C^g\right) \quad (3.9)$$

Where

- c_1 denotes the learning rate for the rank-one update of c,

$$c_1 = 2/(n + 1.3)^2 + \mu_{eff} \quad (3.10)$$

- c_μ denotes the learning rate for the rank- μ update of c,

$$c_\mu = \min\,(1 - c_1, \alpha_\mu \times \left((\mu_{eff} - 2 + 1/\mu_{eff})/((n + 2)^2 + \alpha_\mu \times \mu_{eff}/2)\right) \quad (3.11)$$

7. Update of Step-Size Control

The Step-Size σ^g, or Cumulative Step length Adaption (CSA) indicates the overall scale of the distribution. The CMA-ES algorithm exploits evolution path p_σ to control σ^g which can be applied, as below [124].

First, the evolution path p_σ^{g+1} is computed based on the evolution path p_σ^g by

$$p_\sigma^{g+1} = (1 - c_\sigma) \times p_\sigma^g + \sqrt{c_\sigma(2 - c_\sigma)\mu_{eff}} \times m^{(g+1)}/c^g \quad (3.12)$$

Where c_σ is a learning rate for the cumulation of σ,

$$c_\sigma = \mu_{eff} + 2/n + \mu_{eff} + 5 \quad (3.13)$$

Second, by using the equation (3.12), the length of the evolution path defines the step size σ for generation $g + 1$.

$$\sigma^{g+1} = \sigma^g \times exp(\frac{c_\sigma}{d_\sigma}(\frac{\left\|p_\sigma^{g+1}\right\|}{E[\|N(0,I)\|]} - 1)^{0.3} \quad (3.14)$$

Where d_σ is the damping parameter for the step-size update,

$$d_\sigma = 1 + c_\sigma + 2 \times max\left(\sqrt{\mu_{eff} - 1/n + 1} - 1{,}0\right) \quad (3.15)$$

The flowchart of CMA-ES [80] which is used in this thesis as shown in Figure 3.7:

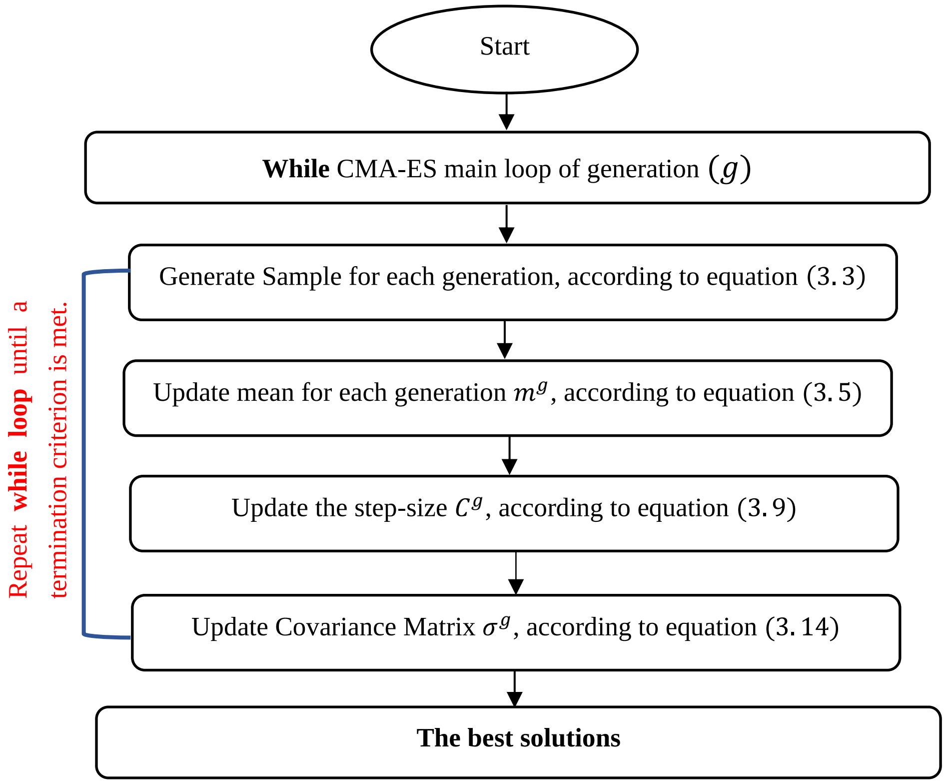


Figure 3.7 Flowchart of CMA-ES algorithm.

Steps 2–5 are repeated until stopping criteria are met. The three ways to specify the stopping criteria of the CMA-ES algorithm are as follows:

1. The algorithm is run for a specified number of generations, which is set in the first step of the CMA-ES algorithm as the $MaxIt$;
2. When the CMA-ES algorithm is not making progress towards the solution, it should be terminated;
3. Stopping when the CMA-ES algorithm reaches a predetermined objective function value.

Once the stopping criteria are met, the best candidate solution, as determined by its fitness value, is returned as the result.

3.4.3 Why CMA-ES?

The following are some of the most important features about CMA-ES that made it the best choice in this research:

- The CMA-ES algorithm does not require gradient analytic computing, which has difficulty finding the best solution due to the lack of differentiability [26]. As a result, CMA-ES can be used to solve multimodal or noisy problems.

- The CMA-ES algorithm was tested on some general objective functions [127] and, when comparing the results to those of other optimisation algorithms, it was found to be reliable and highly competitive. This was particularly true in terms of accuracy and speed because of its ability to find the best solution in very few generations, thereby reducing time delay (Appendix A).
- For its application, the CMA-ES does not require extensive parameter adjustment. Indeed, the selection of internal strategy parameters is entirely automated.

3.5 Preliminaries

In this section, the groundwork for the rest of the thesis by explaining the required concepts of graph-routing, providing a justification of the methods used for the validation, and explaining the simulation environment, assumptions, energy model used, system parameters, and performance metrics.

3.5.1 Concepts of Graph-Routing

A set of vertices V and a set of edges E form a structure known as a graph, where $G = (V; E)$. Figure 3.8 shows that in a wireless network, vertices are network devices, also called sensor nodes, and edges are the connections between them. Furthermore, an edge exists between two sensor nodes if they are within their communication range.

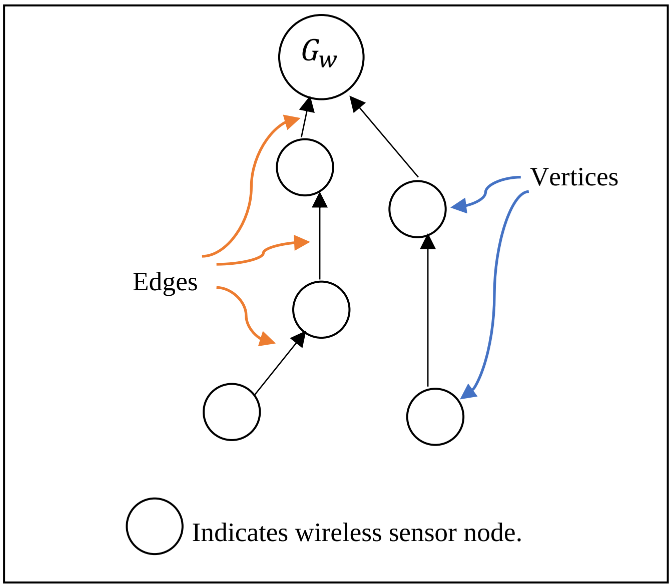


Figure 3.8 Examples of vertices and edges.

This section presents concepts of graph-routing that are considered in this thesis.

a) **Orientation.** The direction of a given edge is shown by an arrow, which indicates the direction in which data packets are moving.

b) **Source node.** This is a sensor node that has data packets to send to the final destination.

c) **Final destination.** The last network device to receive the data packet in this work is the gateway.

d) **Neighbours.** Two sensor nodes are neighbours if there is an edge in the graph connecting them. In this thesis, each sensor node has at least two neighbours if the gateway is out of its communication range.

e) **Route/Path.** This is a sequence of edges that send a data packet in the same direction to its final destination. Except for the first and last edges, each edge shares a vertex with the edge before it and the edge after it.

f) **Cycle.** This is a chain where certain sensor nodes are interconnected so that they create loops. When there are cycles in a network, a data packet can get stuck in an infinite loop and never reach the gateway.

g) **Hop.** This is the movement of a data packet from one sensor node to a neighbour along the path to the gateway.

h) **Multi-hop.** The routing of data packets from the source node to their final destination may, in some cases, take several hops; this is a *multi-hop path*.

3.5.2 Validation Method Justification

Simulation is an effective way to forecast performance when there is no real network available for performance measurements. It provides an insight into the evaluation of the efficiency of an algorithms' real-world performance without the, potentially very demanding, time and cost of a physical test-bed implementation [2]. In addition, it enables analysts to easily test performance under a wide variety of network conditions.

3.5.2.1 Simulation Environment

Simulations were conducted for this thesis using MATLAB on a Windows 10 workstation running on an Intel (R) core™ i7 processor with 16 GB RAM. In this work, the simulation focused on implementing wireless sensor nodes, links, access points, and network manager within the MATLAB environment. This was done to study how the graph-routing algorithm might be improved.

MATLAB is an acronym for Matrix Laboratory. MATLAB is a software programme developed by MathWorks [128] that is widely used for numerical computation, deep learning, engineering applications, and data analytics. It is regarded as one of the top Artificial Intelligence (AI) programming languages. The primary reasons for the use of MATLAB in this work are as follows:

1. MATLAB's programming capacity, its most significant feature, which is easy to use and learn and can support user-developed functions.
2. MATLAB offers an interactive environment with an extensive library of built-in mathematical functions that are accurate and reliable, making it ideal for developing wireless networks.
3. It is easy to integrate MATLAB code with other programming languages, including C/C++, Python, Fortran, Excel, and Java.
4. MATLAB can run on a variety of operating systems, such as Windows, Linux, and macOS.
5. The primary reason for using MATLAB is that it has a built-in Optimisation Toolbox™ that contains mathematical operations and data analysis of optimisation algorithms. These are used to find the parameters of objective functions that minimise or maximise objectives while satisfying constraints. The toolbox includes solvers for Linear Programming (LP), Quadratic Programming (QP), Mixed-Integer Linear Programming (MILP), Non-Linear Programming (NLP), Second-Order Cone Programming (SOCP), constrained linear least squares, and non-linear least squares. It was observed that most literature reviews that used optimisation techniques in WSN used MATLAB, because all optimisation algorithms are built into MATLAB.

3.5.3 Assumption

The network area in this thesis, which is about the uplink graph-routing algorithm, consists of a number of wireless sensor nodes, the gateway, and two APs, which are dispersed in a square network field. Figure 3.9 illustrates an example of a network area of the mesh topology that consists of one red rhombus shape, which represents the gateway (G_w) positioned in the centre of the network area; two blue squares are the APs, located 10 m to the right and 10 m to the left of the G_w, along with 50 wireless sensor nodes placed at random.

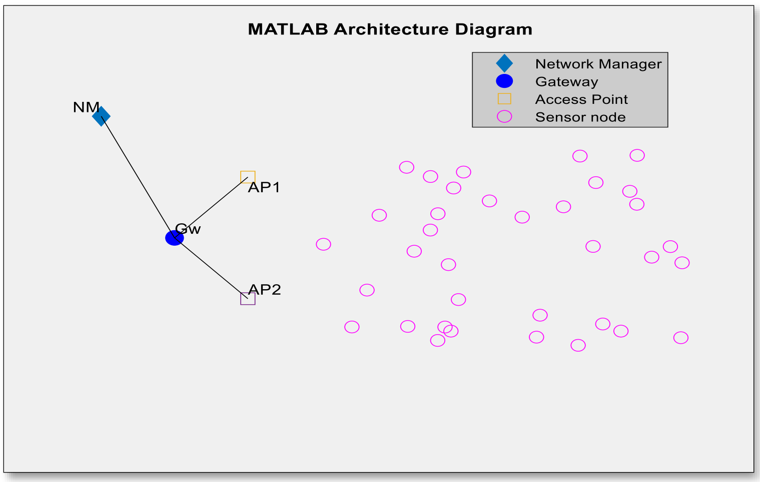


Figure 3.9 A diagram of MATLAB architecture.

This thesis assumes the following network properties, as per the general requirements of typically configuring the WirelessHART network [28]:

1. Within a 100-by-100-metre region, one G_w is in the center, and two APs are placed 10 metres to the right and left of the G_w.
2. The connections between the G_w and the various APs were considered to be reliable and wired.
3. The NM has information about the location and battery status of each sensor node and the G_w.
4. In a two-dimensional space, battery-powered sensor nodes are deployed and cannot be recharged after deployment.
5. All sensor nodes can send data packets to the G_w.
6. All sensor nodes are homogeneous and have the same capabilities in terms of their initial power level.
7. After deployment, sensor nodes are stationary.
8. A unique ID is assigned to each sensor node.
9. Each sensor node acts as a router, allowing it to receive and forward data packets from and to other sensor nodes or a gateway.

3.5.4 Energy Consumption Model

Data transmission and reception use a lot of power in IWSNs. To evaluate and compare the power consumption of the wireless sensor nodes more accurately, the simulator had to be adapted to use an energy model. The focus of our energy model is how much energy a sensor node uses when sending and receiving a data packet. It is based on the energy model in [20], which is used in most of the literature to determine energy losses during transmission and reception in IWSNs, as illustrated diagrammatically in Figure, 3.10.

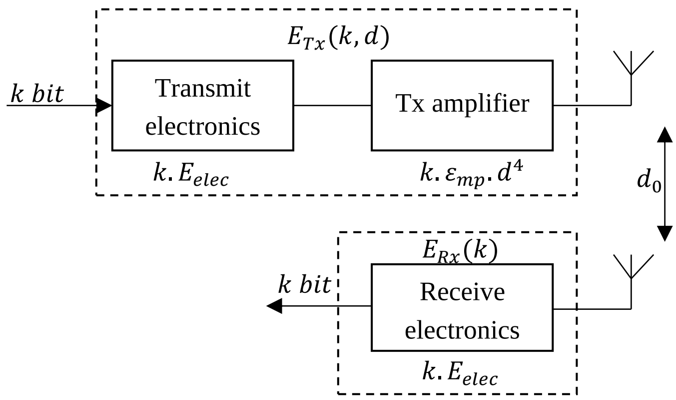


Figure 3.10 Radio energy model.

During transmission, the sensor node consumes energy to operate radio electronics and the power amplifier. During reception, the sensor node consumes energy to operate these radio electronics. Moreover, the free space (d_0 power loss) model is utilised if the distance is less than a predetermined threshold between transmitter and receiver; otherwise, the fading multipath model (d^4 power loss) is used. The energy consumed in transmitting a k-bit packet over a distance d is,

$$E_{Tx}(k,d) = \begin{cases} kE_{elec} + k_{\epsilon_{fs}} d^2, & d < d_0 \\ kE_{elec} + k_{\epsilon_{mp}} d^4, & d \geq d_0 \end{cases} \quad (3.16)$$

and in receiving the message,

$$E_{Rx}(k) = kE_{elec} \quad (3.17)$$

Where E_{elec} is the consumed energy per bit to run the transceiver circuit, the energy loss factors in the hardware emission power amplification process are ϵ_{fs} and ϵ_{mp}, and the distance threshold is d_0. The energy consumption has a square relationship with the distance when the data transmission distance is less than to d_0; if the data transfer distance is higher than or equal to d_0, the energy consumption is four times that distance. This is why multi-hop communication is more effective in IWSNs.

3.5.5 System Parameters

This section describes the parameters used to configure the network simulation for this research. In Section 3.5.5.1, the parameters of the network area that depend on the WirelessHART network in general are described, including node and area parameters, physical layer parameters, network layer parameters, and energy model values. The parameters used in the CMA-ES algorithm are covered in Section 3.5.5.2.

3.5.5.1 Network Parameters

To simulate networks approximatively, wireless sensor nodes are considered to operate in a square simulation area with a fixed communication range of 35 m for all sensor nodes; the communication range distance is based on the WirelessHART network's 100 × 100 m^2 network area [34]. 50 or 100 sensor nodes were used to verify the algorithms' performance under varying node densities. The number of sensor nodes selected for this research simulation has been determined by two arguments. The first claims that IWSN applications primarily concerned with reducing latency, conventionally have no more than 50 sensor nodes [129]. The second uses evidence from industrial environments to affirm that a WirelessHART network can comprise 100 sensor nodes [130].

Each simulation begins with the initialisation of the NM, gateway, and APs. The NM then configures the relevant network (routes and links), based on its knowledge of each node in the network, including its location and its battery status. This data is derived from the health reports sent by the sensor nodes every 15 minutes. When a new node joins the network, it receives network configurations from the NM after each update. Parameter settings for the simulation are summarised in Table 3.3.

Table 3.3 System parameters.

Item	**Definition**
	Node and area parameters
Simulation area	100 × 100 m^2
Number of nodes	50 and 100
Nodes positions	Random
Gateway (G_w)	One G_w
G_w position	Central
Access Points (APs)	Two Aps
	Physical Layer
Physical layer	IEEE 802.15.4
Propagation Model	O-QPSK
Communication range	35 m
Transmission power	0 dBm
Node initial energy	0.5 J
Maximum Packet size	133 Bytes
Radio frequency	2.4 GHz
Medium Access Control (MAC)	TDMA with 10 ms time slot
	Network Layer
Routing algorithm	Uplink graph-routing algorithm
	Energy Model [20]
$\boldsymbol{E_{elec}}$	50 nJ/bit
$\boldsymbol{E_{da}}$	5 nJ/bit/singal
$\boldsymbol{\epsilon_{fs}}$	0.01 nJ/bit/m^2
$\boldsymbol{\epsilon_{mp}}$	0.000013 nJ/bit/m^4
$\boldsymbol{d_0}$	35 m

3.5.5.2 CMA-ES parameters

The CMA-ES algorithm does not require tuning of the parameters except for population size λ, where strategy parameters are considered part of the algorithm design. This is a feature of CMA-ES. The aim is to have a well-performing algorithm as observed in [80]. Therefore, λ is set to $4 + \lfloor 3 \log(n) \rfloor$ as suggested in [80] where n is the number of the variables that are in the optimise routing table in this model (See Chapter 5, Section 5.4.1). The parameter σ specifies the direction of the algorithm as $0.3 \times (VarMax - VarMin)$, where $VarMax$, $VarMin$ are upper and lower bound to the optimise routing table decision, respectively. Table 3.4 shows the CMA-ES parameters.

Table 3.4 CMA-ES Parameters.

Parameters	Value
Population size (λ)	$4 + \lfloor 3 \log(n) \rfloor$
Number of the variables (n)	Optimise Routing table
Specifies the direction (σ)	$0.3 \times (VarMax - VarMin)$
$VarMax$	Upper bound to the routing table decision
$VarMin$	Lower bound to the routing table decision

3.5.6 Performance Metrics

The performance of the graph-routing algorithm can be measured by against the three requirements of IWSN applications: high data packet delivery, balanced energy consumption, and low End-to-End Transmission (E2ET) time. To evaluate these requirements, the following metrics have been defined.

Communication reliability.

- **Packet Delivery Ratio (PDR)** is the ratio of data packets successfully delivered to the gateway to the total number of data packets sent by source nodes.

$$PDR = \frac{P_{Received}}{\sum_{i=1}^{n} P_{Generated_i}} \times 100 \tag{3.18}$$

In the equation, $P_{Received}$ represents the total number of data packets received by the gateway, whereas $P_{Generated_i}$ is the total number of data packets generated by the source nodes, while i and n represents the number of sensor nodes.

Energy consumption.

- **Total Consumed Energy (TCE)** is the total energy consumed by a sensor node during the communication process. It shows the rate at which the energy source is drained over time.

$$TCE = \sum_{i=1}^{n} TCE + CE_i \tag{3.19}$$

Where n is the number of sensor nodes, CE_i is the current energy of the sensor node i after the end of the simulation time.

- **Energy Imbalance Factor (EIF)** is the mean standard deviation of the residual energy of all wireless sensor nodes and is used to demonstrate how efficient the graph-routing algorithm is in terms of energy balance.

$$EIF = \frac{1}{n} \sqrt{\sum_{i=1}^{n} (RE_i - RE_{avg})^2} \tag{3.20}$$

Where n is the number of sensor nodes, RE_i is the residual energy of the i^{th} sensor node, and RE_{avg} denotes the average residual energy of all sensor nodes.

Transmission time.

- **End-to-End Transmission (E2ET) time** represents the time a data packet takes to transit from the source node to the gateway (See Chapter 5, Section 5.4.2.3).

3.5.7 Error Estimation

Error estimation is used to calculate the accuracy of the simulation results. It is the process of determining the margin of error, and, thus, the uncertainty associated with the simulation. In this research, multiple simulation runs were performed to test the disparate random placements of sensor nodes, where each run presented a different node topology. The result metrics were collected after the completion of these simulation runs for each topology. The mean of these results was then calculated. However, these simulation results could be affected by various sources of uncertainty, including model parameters, random variables, and measurement errors.

It is, therefore, important to calculate the error estimation. This requires a statistical method that considers variability in the simulation results. The Standard Error of the Mean (SEM) is a widely accepted method. It measures the degree of variation in the samples' mean to assess the uncertainty of the results for the different simulation runs. Here is the general formula for calculating the SEM.

$$SEM = \frac{s}{\sqrt{n}} \tag{3.21}$$

Where s is the standard deviation of the sample, which in this study is represented by the results of a specific metric, and n is the sample size, which comprises the number of iterations of each simulation run. The SEM measures the accuracy of the mean of the ratios. This is used to determine the *error interval*: the range of values within which the true mean is likely to fall.

Once the SEM has been calculated, it can be used to compute the error interval using the following formula:

$$Error\ interval\ = \pm\ (SEM\ \times\ t - value) \tag{3.22}$$

Where $t - value$ refers to the statistical value used to calculate the error interval of the sample mean.

Therefore, based on the mean error intervals in the results, an error estimation of ± 3% has been included, offering insight into their accuracy. For example, if the PDR results were to be estimated at 95% accuracy with an error interval of ± 3%, this would indicate the degree of uncertainty associated with the results. The true value, therefore, could be between 92% and 98% with a certain level of confidence.

3.6 Summary

The first part of this chapter discussed the solution methods used in this thesis; namely, clustering and optimisation techniques. It explained the concept of clustering techniques, their objectives, and the methods used to form the clusters and select CHs in general. Optimisation techniques were presented in detail along with their overall classification, followed by a survey of the state-of-the-art research work that applies these techniques to wireless networks. Optimisation can be used to improve the performance of graph routing by constructing paths in a centralised manner, building in redundancy to increase data packet delivery and, in clustered networks, to select the best CHs.

The second part of the chapter outlined various general elements that will be used throughout the thesis, including the concepts of graph-routing, validation methods, explanation of the simulation environment, energy model, plus a list of assumptions concerning the network area, system parameters, and performance metrics.

Chapter 4

Mitigating the Hotspot Problem in IWSNs Using an Unequal Clustering Algorithm

4.1 Overview

The use of graph-routing in the WirelessHART network offers the benefit of increased data packet delivery due to path redundancy and multi-hop network paths. Nonetheless, this use creates a hotspot challenge resulting from unbalanced energy consumption because it typically relies on a simple sensor mesh communication system. This chapter studies the effect of unequal clustering topologies on the performance of the graph-routing algorithm in IWSNs, and compares unequal clustering topologies to mesh topology in terms of data packet delivery, energy consumption, and balance in the consumption of energy between sensor nodes in the network area. Section 4.2 describes the hotspot problem in relation to mesh topology. Section 4.3 discusses the technique of unequal clustering and its desired objective. Section 4.4 discusses related works that use unequal clustering in wireless networks to alleviate the hotspot issue. The system model that this work uses is detailed in Section 4.5. Then the analysis of the simulation results is presented in Section 4.6. The final section of this chapter contains a summary.

4.2 Problem Description

Typically, the WirelessHART network transfers data packets from battery-powered sensor nodes to a central system, the NM, through the gateway. Furthermore, the WirelessHART network uses a mesh topology with multi-hop communications, allowing all sensor nodes to act as routers for data packets, from other nodes, until these data packets reach the gateway. In other words, any sensor node in the network area can transmit its data packet to the closest neighbouring sensor node, rather than directly communicating with the gateway if the gateway is out of its communication range.

In general, intermediary sensor nodes within the network will have several neighbouring sensor nodes to send data packets to, so if a sensor node is unable to communicate with one neighbour it can transmit the data packet through an alternative neighbour. However, the routing and forwarding of data packets consume the battery life of the wireless sensor node. Therefore, the sensor nodes closer to the gateway are overburdened with high traffic loads compared to those further away, as data packets from the entire region are forwarded through the former to reach the gateway, as shown in Figure 4.1. Overloaded nodes will expire much faster than other sensor nodes, a phenomenon known as the ‘hotspot problem’, potentially resulting in network partitioning due to this imbalance in energy consumption. For example, certain sensor nodes can quickly consume energy resulting in the WirelessHART network having an undesirably short lifetime, whereas other sensor nodes are able to slowly consume energy and have a long lifetime. The network lifetime depends on the shortest-lived sensor node because any node containing a dead battery becomes unreachable. As a result, due to this issue, sensor nodes that are further away from the gateway may be unable to reliably transmit data packets to the gateway.

Therefore, executing an energy-saving procedure within the sensor nodes is insufficient in a centralised WirelessHART network, as they may still suffer from unbalanced energy consumption. Consequently, the balance of energy consumption distribution among the wireless sensor nodes has become a key factor that must be considered when executing WirelessHART network energy-saving procedures.

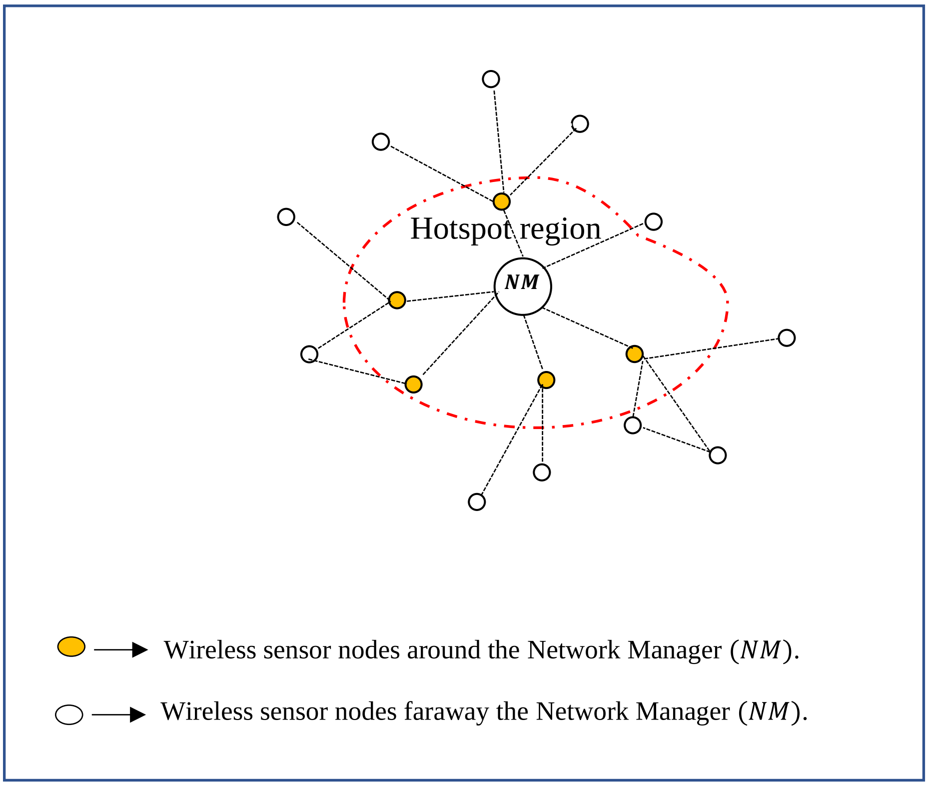


Figure 4.1 Hotspot problem.

To mitigate the hotspot problem when dealing with limited energy resources in IWSNs, this thesis proposes using the clustering technique, and making the clusters different sizes based on their distance to the gateway, which is known as ‘unequal clustering’. The aim is to balance energy consumption. Following this reasoning, the present study proposes using the basic graph-routing algorithm of the WirelessHART network based on unequal clustering.

4.3 Unequal Clustering Methods

Unequal clustering has been proposed in various research [131]–[138] to address the hotspot problem, because it can help balance the load between sensor nodes in the network area [131]. Unequal clustering places the smallest clusters near to gateway and increasing their size with distance from gateway. Thus, the distance of the CHs from the gateway is directly proportional to the size of the cluster, as shown in Figure 4.2. Smaller clusters near the gateway (G_w) have fewer CMs and more limited traffic within the cluster. Thus, the smaller the cluster size, the more limited energy intra-cluster traffic consumes, and the greater the focus on traffic between clusters. Likewise, the clusters with a larger size and distance from G_w specify a greater number of CMs: the less energy the inter-cluster traffic consumes, the more it focuses on intra-cluster traffic. With unequal clustering, all sensor nodes in the network area may consume similar amounts of energy, so a sensor node near G_w may spend the same amount of energy as a sensor node that is distant from G_w [133]. Therefore, unequal clustering eliminates the hotspot problem through efficient load balancing between the sensor nodes.

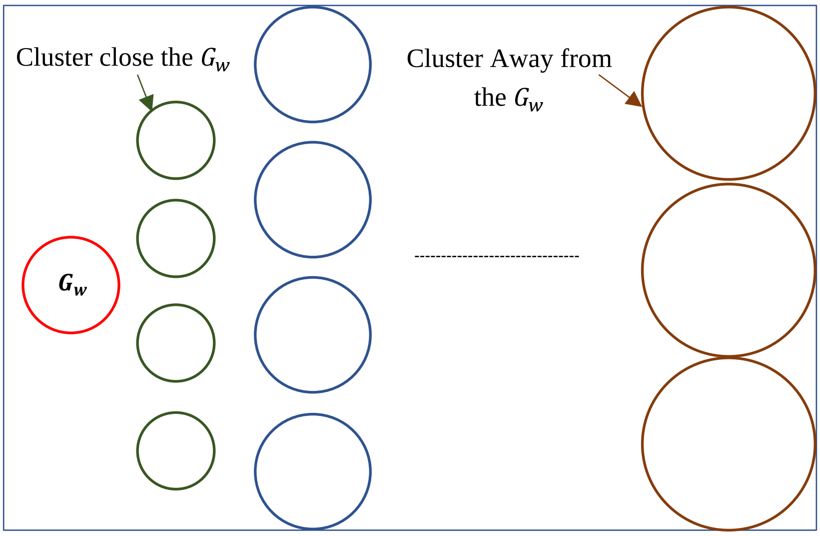


Figure 4.2 Example of unequal clustering technique.

Unequal clustering techniques can be classified into three categories: probabilistic, pre-set, and deterministic [139].

Probabilistic. In probabilistic approaches to CH selection, a probability is initially assigned to each sensor node which is utilised to determine CHs randomly or via a hybrid method. Simplicity and low energy consumption makes this approach popular.

Pre-set. Pre-set approaches are not dynamic, the location of clusters or CHs are assigned prior to deployment in the physical world.

Deterministic. The deterministic approach is more effective and reliable than probabilistic approaches because the chosen CH is based on specific parameters such as residual energy, and the gateway's distance.

4.4 Unequal Clustering Applied to the Routing of Wireless Networks

This section highlights some of the basic clustering algorithms and examines state-of-the-art related work that has used unequal clustering techniques for balancing energy consumption in wireless networks.

The first cluster-based routing algorithm for WSN was the LEACH algorithm [47]. Clusters are formed after each sensor node generates a random number between 0 and 1 at the start of each round. A node is elected as a CH if this number is smaller than the threshold value. Subsequently, CMs join their nearest CH based on their Received Signal

Strength (RSS). The key shortcomings of the algorithm are that the energy level of the sensor nodes is not considered in the CH selection process, and communication between the CH and the BS occurs though a single hop. The problem with the LEACH algorithm's single hop is addressed by the Energy-Efficient Hierarchical Clustering (EEHC) algorithm [140], which divides the network area into layers to make it more manageable. Then, data packets are sent from sensor nodes in the lower layer to the BS in the top layer via multi-hop communication. However, this approach is still unaware of the energy level needed to select CHs in each cluster. Nonetheless, a significant number of algorithms have been developed to date [141] in an attempt to improve the performance of the LEACH algorithm.

The Hybrid Energy-Efficient Distributed Clustering Appdoroach (HEED) is a multi-hop clustering algorithm [142]. In the cluster formation phase, the CHs are periodically selected based on the residual energy of the sensor node, and on the proximity of the sensor node to its neighbours. However, as with most clustering techniques that do not take the cluster size into account during cluster formation, the network still suffers from the hotspot problem, which significantly reduces its lifetime [139].

Energy-Efficient Unequal Clustering (EEUC) [131] is proposed as a probabilistic clustering technique that uses unequal clustering. Each node in a cluster generates a random integer between 0 and 1 to participate in a final competition to select CHs, and the final CHs are selected based on their amount of residual energy. After that, it divides the network into clusters of unequal size based on their distance from the BS in order to save the energy of the sensor nodes near the BS, and uses multi-hop communication to send data packets to the BS.

Similarly, the Probability-Driven Unequal Clustering ('PRODUCE') algorithm [132] divides the network area into hierarchies of unequal clustering. In this algorithm, the transmission distance is limited to the radio energy model's transmission distance threshold. Consequently, it performs better in small networks but it has a scalability problem.

The Energy-Aware Unequal Clustering Fuzzy (EAUCF) algorithm described by [133] uses fuzzy logic to produce a competition radius that reduces the number of cluster nodes closest to the gateway, thus also reducing the number of transmitted data packets. CHs are elected at random, and the radius of each CH competition is determined by the residual energy rate and gateway distance.

The Unequal Clustering Algorithm for Wireless Sensor Networks ('UHEED') [134] algorithm uses the competition radius formula of the EEUC algorithm [131] to create unequal clustering. This makes the performance of the HEED algorithm [142] better in terms of balancing energy consumption.

The Density controlled Divide-and-Rule (DDR) algorithm [135] presents the static clustering of sensor nodes and optimal CH selection. These are based on the energy in each round in order to solve the unbalanced energy utilisation that causes energy holes in the network. The disadvantage of this technique is that it divides the network into a predetermined number of clusters without taking network size into account.

Recently [136] proposed an unequal clustering-based routing algorithm that could address the hotspot problem by using an unequal fixed grid-based cluster along with a mobile sink for data collection from the CHs. Then this data is delivered to the BS in the WSN to reduce the multi-hops between the CHs and BS. The combination of these two techniques enhances uniform energy consumption throughout the network. Moreover, [137] presents a routing algorithm according to the unequal clustering techniques for WSNs. This proposed approach adopts appropriate strategies for low-power routing by introducing an unequal-sized cluster structure, allowing flexibility in multi-hop communication, and mitigating the problem of hotspots. Furthermore, by implementing unequal size partitioning, in the number of sensor nodes per cluster, the proposed algorithm ensures that the load on the CHs is balanced in the network. In [138], researchers proposed an unequal cluster formation mechanism known as the Energy-efficient Multi-hop Routing with Unequal Clustering (EMUC) algorithm; this is based on a probabilistic model of the EEUC topology and the residual energy of nodes in chosen CHs. To further improve energy efficiency and extend the network's lifetime, a minimum transmission energy algorithm is used to determine the shortest path between the CH and its CMs.

This section has reviewed several algorithms designed to mitigate the hotspot problem and prolong network lifetime in wireless networks. Different approaches have been proposed by these unequal clustering techniques, including, but not limited to, the selection of CHs, double CHs, competition CHs, and the use of mobile sinks. However, graph-routing of the WirelessHART network has not been applied to the unequal clustering algorithms proposed, nor has there been a focus on developing static unequal clustering that would make the network more stable.

The originality of the current research lies in an attempt to improve data packet delivery and energy consumption by using unequal clustering algorithms on the WirelessHART network; while also improving the DDR algorithm to accommodate any network size. Finally, in this research, a graph-routing algorithm is applied between the CHs and the gateway to reduce multi-hop communication.

4.5 Network Model

This section details the system model that this work uses. Consider a WirelessHART network with an area of $(M * M)$ square with N sensor nodes deployed randomly across it. The gateway's storage, computational, and battery power are unrestricted. However, all the sensor nodes have limited battery power, and work as routers in addition to their normal functions. The location of all wireless sensor nodes is known by the gateway (G_w), and may be found in the routing table that calculates the distance between each wireless sensor node and the G_w. Three models can be considered for building the basic graph-routing algorithm of the WirelessHART network in this research: (1) mesh topology, which is the basic topology of the WirelessHART network; (2) pre-set unequal clustering, which is developed in this research; and (3) probabilistic unequal clustering, which is the EEUC topology [131]. Table 4.1 lists the various notations used in this chapter.

Table 4.1 Notations of network models.

Notation	Meaning
General Parameters	
$M * M$	Simulation area
N	Nunmber of sensor nodes
d_0	Communication range
WDDR topology Parameters	
S	Number of concentric layers
r	The distance between the boundary of the first concentric layer and G_w
EEUC topology Parameters [131]	
h_i	Tentative CH_i
b	A weighted factor with a value between [0,1]
d_{max}	Maximum distance from the CHs to G_w
d_{min}	Minimum distance from the CHs to G_w
$d(h_i - G_w)$	The distance between h_i and G_w
R^0_{Comp}	The maximum value of the pre-defined competition radius

4.5.1 Graph-Routing in Mesh Topology

Figure 4.3 illustrates the basic topology of a WirelessHART network with its field devices deployed in a mesh topology. In this example network, there is one G_w and 13 sensor nodes (labelled $a - n$). All communication takes place by routing data packets from the sensor node to G_w via intermediate sensor nodes.

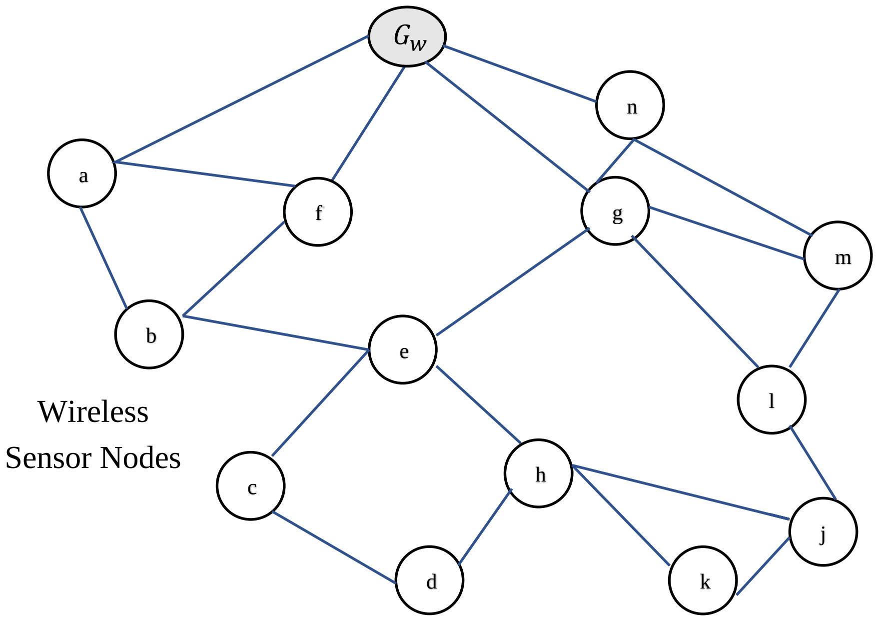


Figure 4.3 Mesh topology of the WirelessHART network.

All sensor nodes in the network must be able to route data packets on behalf of other sensor nodes. The routing of data packets from their source node to G_w requires several hops. Within this network, nodes a, f, g, and n are one hop from the G_w. However, since mesh connectivity is involved, redundant paths are included to improve data packet delivery. In Figure 4.3, sensor node a can communicate with the G_w both directly and via sensor node f if the direct route becomes blocked. Sensor nodes c, d, j and k are several hops away; hence, all intermediate devices (for example, sensor nodes g and e) must be capable of receiving and forwarding data packets. The fundamental problem with this topology is that the sensor nodes closest to G_w are overburdened with higher traffic loads relative to those further away, because data packets from the entire region are routed through the former to reach G_w. Overloaded nodes will die considerably faster than the rest of the sensor nodes, risking network partitioning.

By supporting mesh communication technology, IWSNs can be installed in a wide range of topologies such as star, cluster, or tree. However, WirelessHART network compatible sensor nodes can also be deployed in unequal cluster layouts to address the hotspot problem and achieve balanced energy consumption. This is discussed in greater detail in the next sub-section.

4.5.2 Graph-Routing in Unequal Clustering Topology

This sub-section describes the two models of unequal clustering techniques that have been applied in this work. The first model is the WirelessHART DDR (WDDR) topology, a pre-set unequal clustering technique that has been developed in this research. The second is the EEUC topology [131], a probabilistic unequal clustering technique.

The primary and redundant paths of the graph-routing algorithm between the CHs and the gateway are subsequently generated while the CMs in the cluster communicate directly with its CH.

4.5.2.1 First model: Pre-set Unequal Clustering

Pre-set unequal clusters can maintain the stability of the network area because the number of clusters is fixed, so there is no need to change the clusters after the clustering process. In this model, the pre-set approach to unequal clustering applies a generalised density-controlled divide-and-rule (DDR) algorithm [135], with its proposed enhancement in this research, known as the WDDR topology, to be adaptable to any network size. The WDDR topology divides the network area into several concentric square layers based on the size of the network. Except for the first layer around the G_w, each layer has four clusters, as illustrated in the pseudocode of Algorithm 4.1.

Algorithm 4.1 Pseudocode of WDDR topology.

Input:
  $Coordinates\ of\ network\ area\ and\ d_0$
Output:
  Clusters $of\ WDDR$
Start
   $Determine\ the\ coordinate\ of\ G_w\ as\ Centre\ Point$
   $G_w = [\mathrm{Xcoor_G}, \mathrm{Ycoor_G}]$
   $S = round(M/d_0)$ ▷ Number of concentric layers
   $Calculate\ r_1 = (M/2)/S$ ▷ Find r_1 for first concentric layer
   For $i = 1{:}\,S$ ▷ Divide network area to layers
      $r_{1+i} = r_1 * i$ ▷ Find r_s for S^{th} concentric layer
      $TopRight\ (i,:) = [(XG_w + r_{1+i}), (YG_w + r_{1+i})];$
      $BottomRight\ (i,:) = [(XG_w + r_{1+i}), (YG_w - r_{1+i})];$
      $TopLeft\ (i,:) = [(XG_w - r_{1+i}), (YG_w + r_{1+i})];$
      $BottomLeft\ (i,:) = [(XG_w - r_{1+i}), (YG_w - r_{1+i})];$ ▷ Calculate coordinate of each square layer [135]
   End
   For $j = 1{:}\,S - 1$ ▷ Creating clusters for each layer
   $A = [(XtopRight_j, YtopRight_j)\ (XtopRight_j, YtopRight_j + r_1)];$
   $B = [(XtopLeft_j, YtopLeft_j)\ (XtopLeft_j - r_1, YtopLeft_j)];$
   $D = [(XbottomRight_j, YbottomRight_j)(XbottomRight_j + r_1, YbottomRight_j)];$
   $E = [(XbottomLeft_j, YbottomLeft_j)(XbottomLeft_j, YbottomLeft_j - r_1)];$
    End
   Deploy SNs;
   For $C = 1{:}\,Clusters$ ▷ Selecting CH of all clusters
      For $n = 1{:}\,Node(Senor\ nodes\ in\ cluster_c(count))$
         $Node(n).Curretenergy = MaxResidualEnergy_N$
         $Node(n).CH = Node(n)$
      End
   End
End

In relation to the pseudo code for building the pre-set clustering, there are four basic steps involved. The first step is to determine the coordinates of the G_w (XG_w, YG_w) of the network area as the Centre Point (CP), as shown in lines 6 and 7 in Algorithm 4.1. The second step is to find the distance from CP to the boundary of the first concentric layer by calculating the value r_1, as observed in lines 8–11. This will be calculated and derived by calculating value S, which defines the number of concentric layers around the gateway based on the size of the network area, and d_0, as given in equations (4.1) and (4.2):

$$S = round\left({}^{M}/_{d_0}\right) \tag{4.1}$$

$$r_1 = \frac{\left({}^{M}/_{2}\right)}{S} \tag{4.2}$$

Hence, to find r_{1+i} is the distance between the G_w and boundary for any concentric layer will be calculated as:

$$r_{1+i} = r_1 * i^{th} \tag{4.3}$$

Where i^{th} is the ordinal number of a given concentric layer.

In the third step, as shown in lines 12–15, the coordinates for the corner boundaries of each square layer based on its own value r and the coordinates of the G_w (XG_w, YG_w) can be calculated, as given in equations below [135].

$$TopRight_i = [(XG_w + r_{1+i}), (YG_w + r_{1+i})] \tag{4.4}$$

$$TopLeft_i = [(XG_w - r_{1+i}), (YG_w + r_{1+i})] \tag{4.5}$$

$$BottomRight_i = [(XG_w + r_{1+i}), (YG_w - r_{1+i})] \tag{4.6}$$

$$BottomLeft_i = [(XG_w - r_{1+i}), (YG_w - r_{1+i})] \tag{4.7}$$

Where i indicates the number of the current square layer while $TopRight_i$ and $TopLeft_i$ are the (x, y) coordinates of the top right and top left corners of square layer i, respectively. $BottomRight_i$ and $BottomLeft_i$ are the (x, y) coordinates the of bottom right and bottom left corners of square layer i, respectively.

For example, if the network size of wireless sensor nodes is $400 * 400\ m^2$, and the $d_0 = 100$, then based on equations (4.1) and (4.2), the number of concentric layers will be $S = 4$, as shown in Figure 4.4.

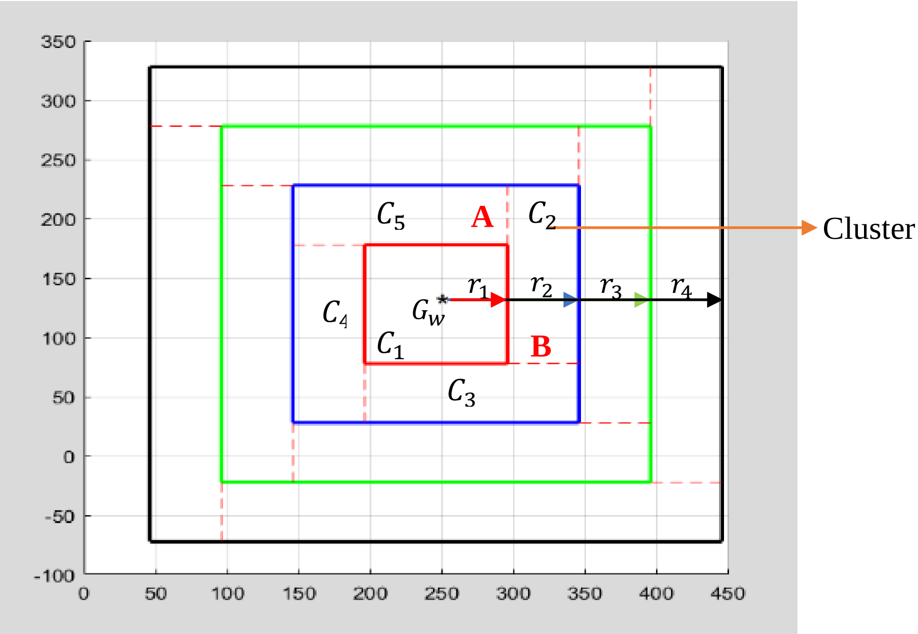


Figure 4.4 Concentric layers and clusters of the WDDR topology.

This is followed in step four by dividing the area between every two concentric layers into four rectangular areas as segments, as shown in lines 17–21. Each segment will be considered a cluster, denoted as C (see Figure 4.4). To from these clusters, A, B, D, E lines are obtained between the corner coordinates for each layer and the corresponding point in the next layer, as demonstrated in the equations below.

$$A = [(TopRight_j)\ (XtopRight_j, YtopRight_j + r_j)] \quad (4.8)$$

$$B = [(TopLeft_j)\ (XtopLeft_j - r_{j+1}, YtopLeft_j)] \quad (4.9)$$

$$D = [(BottomRight_j)\ (XbottomRight_j + r_j, YbottomRight_j)] \quad (4.10)$$

$$E = [(BottomLeft_j)\ (XbottomLeft_j, YbottomLeft_j - r_j)] \quad (4.11)$$

Where $(XtopRight_j, YtopRight_j + r_j)$ are the corresponding point coordinates for layer j's $TopRight$ corner, $(XtopLeft_j - r_{j+1}, YtopLeft_j)$ are the corresponding point coordinates for layer j's $TopLeft$ corner, $(XbottomRight_j + r_j, YbottomRight_j)$ are the corresponding point coordinates for layer j's $BottomRight$ corner, and $(XbottomLeft_j, YbottomLeft_j - r_j)$ are the corresponding point coordinates for layer j's $BottomLeft$ corner.

For instance, as depicted in Figure 4.4, cluster C_2 of layer j is formed by connecting the coordinates of $TopRight_j$ and $BottomRight_j$ with their corresponding points using equations (4.8) and (4.10). Also, in the same way, in the C_3, C_4, and C_5 clusters of the same layer. Where C_3 was formed by connecting the coordinates of $BottomRight_j$ and $BottomLet_j$ with their corresponding points using equations (4.10) and (4.11), C_4 was formed by connecting the coordinates of $BottomLeft_j$ and $TopLeft_j$ with their corresponding points using equations (4.11) and (4.9) and C_5 was formed by connecting the coordinates of $TopLeft_j$ and $TopRight_j$ with their corresponding points using equations (4.9) and (4.8).

4.5.2.1.1 Cluster Head Selection for WDDR Topology

In the WDDR topology, each cluster has its own CH, and the number of CHs remains constant during network operation. Sensor nodes will be connected to the nearest CH to transmit their data packets, regardless of whether the CH is in the same cluster or another cluster. This will reduce the communication distance. The proposed technique is to perform the selection of the CHs based on the maximum residual energy rate between sensor nodes in the same cluster.

4.5.2.1.2 Enhancements of WDDR Topology Over DDR Topology

Therefore, the WDDR topology is enhanced in four ways compared to the DDR topology:

- The DDR topology uses a static area network of size $120 * 120\, m^2$, but the WDDR amends this so that it is flexible with any square network size using equation (4.1).
- The DDR topology uses a fixed value of r to calculate coordinates for each concentric layer. WDDR can calculate and derive this value based on network size using equation (4.2).
- The DDR topology uses static equations to create clusters [135], while WDDR can create clusters depending on the number of layers using the value of r for each layer.
- In DDR, CHs are selected from all clusters except the first concentric layer (around gateway). To improve data packet delivery and reduce delay, WDDR selects CH from all clusters including the first concentric layer. In addition, a sensor node can select from the nearest CHs if its own CH is further away.

4.5.2.2 Second model: Probabilistic Unequal clustering

The probabilistic approach of unequal clustering applies the Energy-Efficient Unequal Clustering (EEUC) topology [131] to the topology of the WirelessHART network. There are two stages to this.

First stage: tentative CHs are randomly selected to compete for final CHs with probability T which is a predefined threshold:

a) Randomly, each sensor node generates a value from 0 to 1.

b) If the random value for the node is smaller than T, the node will be selected as a CH candidate; otherwise, other nodes go into sleeping mode until the CH selection stage ends.

c) Suppose CHs in the WirelessHART network can be expressed as: CHs= $\{h_1, h_2, h_3, \dots \dots, h_i\}$.

Second stage: each tentative h_i needs to determine its own competition radius, which is a function of its distance to the G_w. To control cluster size, it should be directly proportional to the distance to G_w. The following is the formula for competition range, R_{comp} [131]:

$$h_i.R_{comp} = \left(1 - b\ \frac{d_{max} - d(h_i - G_w)}{d_{max} - d_{min}}\right) R^0_{Comp} \qquad (4.12)$$

As shown in Figure 4.5, each tentative CH maintains a set of its adjacent tentative CHs for final CH selection based on the highest residual energy. Suppose h_i becomes a tentative CH, it has competition R_{comp}. If h_i is in h_j's competition diameter or h_j is in h_i's competition diameter, the tentative CH h_j is neighbouring. The aim is to prevent the addition of a second CH, h_j, within h_i's competition diameter, if h_i becomes a CH at the end of the competition [131].

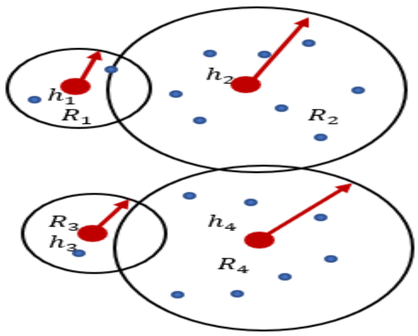


Figure 4.5 The competition among tentative CHs of EEUC.

4.5.3 Impact of Unequal Clustering on Graph-Routing in WirelessHART Networks

This work builds on the graph-routing algorithm of the WirelessHART network, based on mesh and unequal clustering topologies. In a mesh topology, the graph is built between all the wireless sensor nodes in the network area, relying on the graph-routing algorithm in WirelessHART, referred to in detail in Chapter 2, that depends on selecting the first path available to transmit a data packet from the source sensor node to the receiver node to reach the final destination.

Contrary to mesh topology, the unequal clustering method within the WirelessHART network significantly influences its graph density. This method involves the assignment of Cluster Heads (CHs) responsible for transmitting data packets to the gateway.

Consequently, it builds graph paths between CHs in the network area, resulting in a reduction in the number of hops between sensor nodes and the gateway.

Within this method, the data packet is sent directly from the CM to its CH in the same cluster, while all other sensor nodes in the same cluster remain asleep to reduce the energy dissipation of all CMs. Sequentially, the CH sends the data packet to the upper CH; this process is repeated until the data packet is received by the gateway from the closest CH. This approach significantly enhances energy efficiency while notably impacting the delivery of data packets.

The foundation of inter-cluster communication is based on building a graph-routing algorithm specifically between CHs, as outlined below.

4.5.3.1 Graph-Routing Mechanism Between CHs

A graph route is a directed list of paths that connect sensor nodes in the network, thereby providing redundant routes to the final destination. Each CH may have multiple graph routes going through it, even to the same neighbours. Individual CHs only know the next hop along the graph route from themselves to a list of neighbours. In other words, to the individual sensor node, the graph route in the graph table has an ID and a list of neighbours. Data packets are transmitted over the graph route's ID-associated pathway until they reach their destination. Essentially, intermediate CHs look up the graph route ID to find the neighbour CHs. In the graph-routing algorithm of this study the graph routes between CHs are generated as follows.

1. For each CH there are two paths (ensuring redundancy and enhancing data packet delivery): the primary and redundant paths.
2. Depending on gateway proximity, the source CH retains at least two upper neighbour CHs in the neighbouring table for the next hop.
3. There is a primary path for the neighbouring CH which has the highest residual energy and the closest distance, and a redundant path for another CH. The network's redundancy paths can prevent data packet loss due to node failure. If neighbouring CHs have the same remaining energy, the source CH selects the closer CH as the primary path. Similarly, if two neighbouring CHs have the same distance from the source CH, the source CH selects the one that has higher remaining energy as the primary path.
4. If the next hop is not a gateway, the data packet proceeds to Step 2; otherwise, the CH chooses the gateway as the primary path for transmitting the data packet.

4.6 Simulation Experiments

In this study, simulations were conducted in MATLAB R2021a, and a graph-routing algorithm was formulated for the WirelessHART network. It was assumed that the area of the system model is 100 x 100 m^2. The graph-routing algorithm was run for three types of network topologies: mesh, which builds graphs between all sensor nodes; WDDR, which constructs graphs between CHs; and EEUC, which also builds graphs between CHs. Comparisons were made between diverse numbers of sensor nodes to assess the impact of node density on energy consumption metrics, average Energy Imbalance Factor (EIF), total consumed energy, and Packet Delivery Ratio (PDR). Initially, the network area was composed of 50 wireless sensor nodes positioned randomly, then of 100 to facilitate the study of scalability and traffic load increase on the network [129],[130]. Once deployed, the sensor nodes were stationary and each one was given a unique ID. The maximum packet size that should be sent over a WirelessHART network was 133 bytes. Each wireless sensor node could communicate across 35 metres [34], and homogeneity was assumed, meaning that their sizes and energies were identical. For every round, the energy model calculated the energy consumed by each sensor node during data transmission. In each round, a timestamp calculated the number of packets sent, received, and dropped, as well as the energy consumed.

4.7 Evaluation Results and Analysis

4.7.1 Data Packet Delivery Evaluation

The percentage of all data packets received by the gateway that refer to PDR was regarded as an indicator for the data packet delivery of the network. This is one of the most significant requirements of IWSNs. Therefore, the ratio of the number of data packets in each topology was counted at the end of the simulation run.

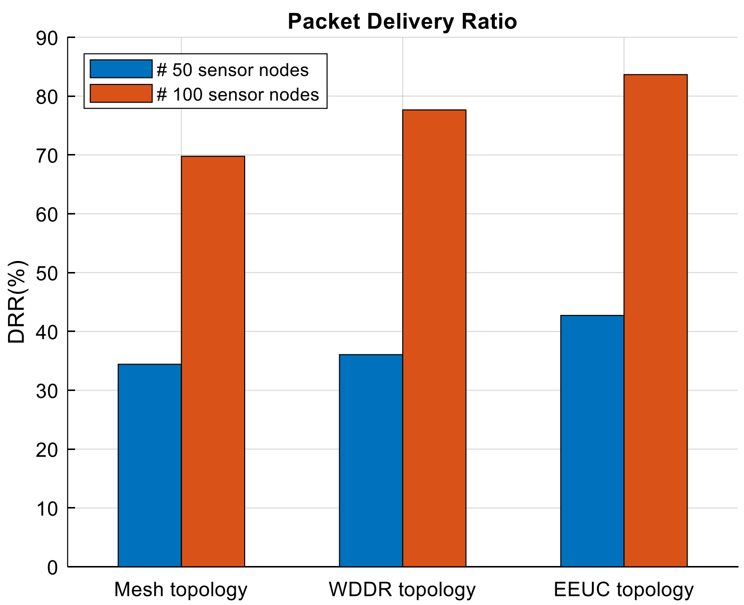


Figure 4.6 PDR results of mesh, WDDR and EEUC.

In Figure 4.6, the results of the PDR with 50 sensor nodes distributed randomly in the network area show that the PDR of graph-routing with the EEUC topology is approximately 42%, that of the WDDR topology is approximately 36%, and that of mesh topology is 34%. However, the results demonstrate that the PDR of the graph-routing algorithm with mesh topology performed slightly lower than the other algorithms. Firstly, this result is reasonable due to the mesh topology used, where each source node in the network area, which has a data packet ready to transmit to the gateway, will transmit the data packet along the path using multi-hop and select its nearest neighbour node available to reach the gateway. This results in more retransmissions because the number of multi-hops increases, especially with sensor nodes located far from the gateway which, in turn, leads to increased drop probability.

The second reason is that when there are fewer sensor nodes in a network area, there is less redundancy in the paths by which data packets can reach the gateway if the primary path fails. However, in mesh topology, as well, it is also observed that PDR increased with an increase in the number of sensor nodes, as shown in Figure 4.6. This increase means that the number of neighbouring nodes, for each sensor node that can communicate with them, has increased. Thus, increasing the redundancy probability of paths in case the primary path fails. However, due to the characteristics of mesh topology, the graph-routing algorithm with mesh topology still has a lower PDR than other algorithms.

Furthermore, the results indicate that graph-routing with EEUC and WDDR topologies enhances the PDR of the WirelessHART network due to two features in the topologies of EEUC and WDDR. Retransmissions are decreased as a result of unequal clustering, as the graph-routing algorithm supports redundant paths and multi-hops between CHs only if the CH is outside the communication range of the gateway, and any CH can communicate with the gateway directly if the gateway is within its communication range. These elements help to ensure the successful receipt of data packets at the gateway.

The WDDR topology improved the PDR results slightly compared to the mesh topology. In this algorithm, there were observed some of sensor nodes in the network area with the same initial energy value after the simulation run ended. This means these sensor nodes were incapable of communicating with the gateway directly, or with other sensor nodes in the network area, because they were out of their communication range. Therefore, the connectivity between sensor nodes is not ensured. This is called an *isolated node*. This phenomenon appeared with the WDDR topology, particularly for sensor nodes in the clusters far from the gateway, due to the algorithm's static cluster approach, which ensures that clusters remain the same once formed. As a result, the WDDR topology may not perform properly because sensor nodes were not considered when the clusters were formed, thus reducing the data packet delivery of the network. In contrast, the EEUC topology improved the PDR results by approximately 8%, compared to the mesh topology, by performing new unequal clustering at each round and resulting in improved data packet delivery.

However, as the number of sensor nodes increases, the WDDR topology works better in dense networks because it reduces the number of isolated nodes in static clusters. Although these nodes are still there, there are not as many as in the less dense network, which therefore makes the network more reliable.

4.7.2 Energy Consumption Evaluation

The energy of sensor nodes is one of the critical constraints of IWSNs. The energy consumption rate of a sensor node often depends on the routing algorithm used. Low-power consumption means that the entire network lifetime will be extended and the stability of the network will be enhanced. To illustrate this, the total energy consumption of the three algorithms is compared in Figure 4.7.

Mesh topology consumes more energy than the EEUC topology, which consumes the least energy. The Total Consumed Energy (TCE) in the WDDR topology is almost the same as that in the EEUC topology; however, it is slightly more than the EEUC topology. From Figure 4.7, it is evident that the graph-routing of the WirelessHART network with EEUC topology is also superior to other algorithms in terms of conserving energy, regardless of the number of sensor nodes in the network area.

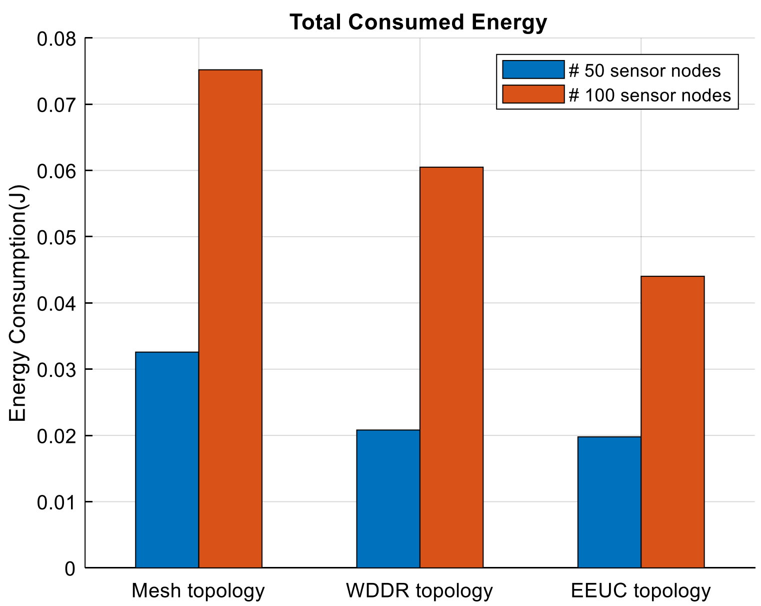


Figure 4.7 Energy consumption results of mesh, WDDR and EEUC.

The major reason for this is the reliance on CHs to construct graphs and transmit packets collected from the sensor nodes of their clusters to the gateway via a graph-routing algorithm, thereby making significant energy savings. However, the WDDR topology demonstrates slightly higher energy consumption compared to the EEUC topology, which is attributable to its static cluster approach that can sometimes select CHs that are distant from their CMs. Thus, wasting some energy on sensor nodes in the same cluster.

In all algorithms, energy consumption increases with the number of sensor nodes in the network, as shown in Figure 4.7. However, an increase in the number of sensor nodes in the mesh topology results in the consumption of substantially more energy than in the other algorithms. This is because increasing the number of sensor nodes results in additional multi-hop network paths and redundant paths, which raise energy consumption across the whole network.

Moreover, it is interesting to observe that energy consumption in the WDDR topology with fewer sensor nodes is almost the same as that in the EECU algorithm, although with an increase in the number of sensor nodes the WDDR topology's energy consumption clearly rose (see Figure 4.7) because the network's connectivity level increased, as shown in the PDR results of Figure 4.6.

4.7.3 Energy Imbalance Factor Evaluation

Figure 4.8 shows the average Energy Imbalance Factor (EIF) results, which measures the average standard deviation of residual energy between sensor nodes in the network area. The maximum EIF of mesh topology was recorded at approximately 61% with 100-node density and 50% with 50-node density compared to the WDDR and EEUC topologies. However, the average EIF in all algorithms increased with the number of sensor nodes.

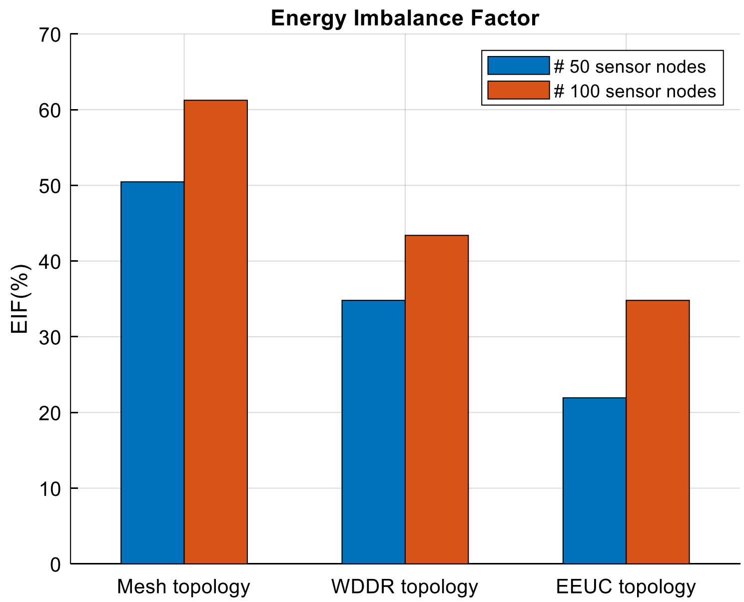


Figure 4.8 Average EIF results of mesh, WDDR and EEUC.

The increase in EIF obtained in the mesh topology is due to the fact that increasing the number of hops results in longer paths; particularly with sensor nodes distant from the gateway, which, as shown in Figure 4.6, reduced the network connectivity level when using the mesh topology compared to other topologies. Therefore, the energy consumption level and EIF are particularly high using graph-routing without WDDR or EEUC. However, a significant reduction in energy consumption imbalance was observed when using graph-routing with either WDDR or EEUC topologies. Compared with mesh topology, the imbalance of energy consumption of the WDDR and EEUC topologies with 50 sensor nodes decreased by approximately 15% and 28%, and with 100 sensor nodes, by approximately 17% and 26%. Because they depend on building the graph-routing between the CHs, which reduces the overhead on the sensor nodes around the gateway, while the CMs communicate directly in each cluster with its CHs which reduces the number of hops by using short-distance communications.

It is also noteworthy that EIF of the WDDR topology significantly increases when the density of the network is low compared to EEUC for a variety of reasons. Firstly, clusters are formed using preassigned information and so are static, with network or node conditions not considered. Secondly, the presence of isolated nodes in clusters far from the gateway has a significant impact on the reach of data packets to the gateway. Thirdly, because there is no distance control between CMs and CHs in the same cluster, the CHs that have the highest residual energy in the cluster are selected regardless of their location. So, the CHs may be faraway or isolated from some CMs, causing an increasing imbalance of energy consumption.

Furthermore, as the number of sensor nodes in the EEUC topology increased, a rise in the EIF was noted. This is due to the increased number of random tentative CHs used by the algorithm to select the final CHs in each round, thus increasing the network's overhead. This suggests that the WDDR topology may be more effective than the EEUC topology in terms of balancing energy consumption in higher density networks.

4.8 Summary

The principal objective of adopting an unequal clustering topology for graph-routing in the WirelessHART network is to study its affect on the performance of the network. The sensor field is divided into clusters of unequal size, which helps to save the nodes' energy nearest to the gateway. This mitigates the impact of the hotspot problem which causes partitioning of the network. This chapter has examined how two types of unequal clustering topologies (WDDR and EEUC) affect the performance of graph-routing in a WirelessHART network, particularly with regard to energy consumption, average EIF, and Packet Delivery Ratio (PDR).

It was observed that graph-routing with the EEUC topology delivered a better performance than WDDR for PDR and EIF. However, in the 50-node-density network, the energy consumption for the graph-routing of the WDDR and EEUC topologies converged to some extent. For all three metrics in this study, EEUC and WDDR scale better than mesh topology as the number of sensor nodes increases; thus, graph-routing with a mesh topology is less suitable for scalability.

The drawbacks of the WDDR topology that led to reduced data packet delivery and increased EIF are the isolated nodes observed after running the simulation in some clusters far from the gateway. Furthermore, the distance between the CH and its CMs was not taken into account, resulting in increased energy consumption. To address these shortcomings, optimisation techniques were implemented to select the CHs and create new clusters for any isolated nodes if they appeared. This is discussed in Chapter 6 of the thesis.

The next chapter also focuses on the ability of the graph-routing algorithm to enhance the graph path selection in mesh topology based on IWSN requirements using optimisation techniques.

Chapter 5

Enhancing the Graph Paths of the Graph-Routing Algorithm Using CMA-ES

5.1 Overview

Accomplishing effective communication in IWSNs requires a reduction in End-to-End Transmission (E2ET) time, a balance in energy consumption, and an improvement in data packet delivery. In order to satisfy these requirements, effective communication depends significantly on the graph-routing algorithm. However, as this algorithm involves the application of a first-path-available approach combined with path redundancy to transmit data packets from the sensor nodes to the gateway it disregards the IWSN requirements. Consequently, it has a negative impact on the performance and lifetime of the network.

This chapter discusses the efficacy of applying optimisation techniques, specifically the CMA-ES algorithm, to facilitate the creation of graph paths focusing on the requirements of the IWSN for the graph-routing algorithm. Section 5.2 problem description and Section 5.3 describes the disadvantages of the graph-routing algorithm when applied to WirelessHART networks. Section 5.4 provides a detailed description of the proposed graph paths for graph-routing based on CMA-ES selection. Section 5.5 describes the setup of the simulation and presents performance evaluation. Section 5.6 compares the performances of proposed graph-routing paths and existing graph-routing algorithms based on IWSN requirements. The final section of the chapter contains a summary.

5.2 Problem Description

The graph-routing algorithm of the WirelessHART network employs a first-path approach, to transmit data packets from a source node to the gateway. However, this approach can create some challenges that affect the requirements of IWSNs. Firstly, industrial environments often generate high levels of noise, which may lead to a decline in the performance of the routing algorithm [18]. However, the data packet delivery of wireless communication can be improved by using redundant routes, which can be applied by the graph-routing algorithm at the network layer [2]. Retransmission is an effective method for increasing data packet delivery, but it also increases E2ET. Secondly, industrial automation imposes stringent end-to-end delay requirements on data communication. Such delays are increased further by conflicts between transmissions where two paths share a sensor node (sender or receiver) [42]. WirelessHART networks do not permit multiple transmissions to take place simultaneously on the same channel; hence, a channel can only support one transmission at a time across the network. A conflict delay occurs when a data packet is delayed because it conflicts with another data packet that is scheduled in the current time-slot [42]. Lastly, the workload of sensor nodes around a gateway must also be considered since, due to centralisation in WirelessHART network, nodes closer to the gateway are often overburdened with high traffic loads compared to those further away. This is because packets from the entire region are forwarded through the former to reach the gateway, leading to an imbalance in energy consumption that reduces the life-time of the network.

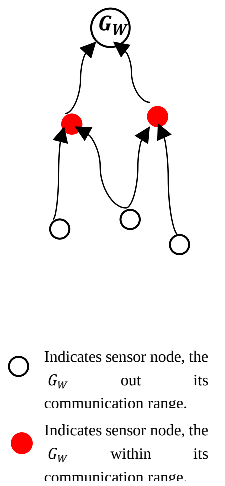
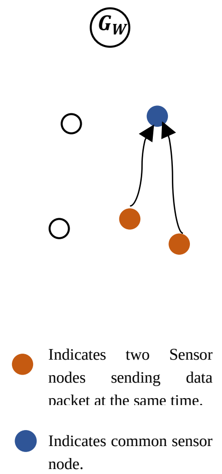
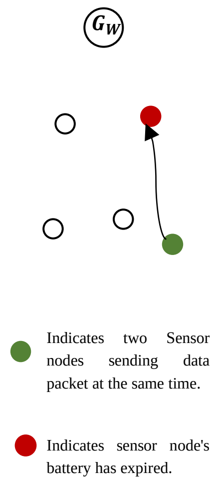
Therefore, when creating and selecting the graph paths of the graph-routing algorithm for the WirelessHART network, all these challenges must be addressed by striking a balance between the essential requirements of real-life IWSNs, which are balancing the energy consumption of sensor nodes, increasing data packet delivery, and reducing E2ET. However, these requirements can be relatively difficult to achieve due to interference and noise in industrial environments, which cause constant redundancy, high latency as a result of redundancy, and unbalanced energy consumption. To address these adequately, optimisation or high-level procedure algorithms are required. The use of optimisation techniques for creating and selecting best paths in a centralised manner may thus be useful for WirelessHART networks and future IIoT protocols.

5.3 Optimisation Problems Formulation for Graph-Routing in IWSNs

In order to use optimisation techniques to design the graph-routing algorithm of WirelessHART networks, it is necessary to identify the problem formulation using the following items: the problems faced by routing in WirelessHART networks, any constraints in the network, and the objectives (minimise or maximise) that are sought, as shown in Table 5.1. The following list of items will be discussed.

- The problem, identifies the situation or case that the existing graph-routing cannot handle and that requires optimisation.
- The objective, represents the values that should be minimised or maximised;
- Search points, which are the values that can be manipulated to minimise or maximise the objective functions.
- Constraints, which are undesirable system limitations that must be taken into account while selecting search points.

Table 5.1 The problem formulation for the graph-routing algorithm.

Items	**Graph-routing in WirelessHART networks**		
Problem	The hotspot problem 	Conflict delay 	Lost data packets 
Objectives	Minimise energy consumption	Minimise delay time	Maximise data packet delivery of the network
Search points	Wireless sensor node	Wireless sensor node	Wireless sensor node
Constraints	Limited energy	Limited resources	Limited energy and resources

The following are the aims of graph-routing designs for WirelessHART networks.

- A graph-routing algorithm should mitigate the hotspot problem of WirelessHART networks. For this problem, the objective is to minimise the energy consumption of sensor nodes around the gateway (G_W) to achieve balanced energy consumption between sensor nodes in the network via selecting sensor nodes with higher residual energy along the path of routing, which is the solution. The constraints of this option are that all wireless sensor nodes are battery powered and small.
- A graph-routing algorithm should reduce conflict delay in WirelessHART networks. For this problem, the objective is to minimise E2ET time between a

pair of sensor nodes by selecting appropriate sensor nodes along the path of routing that avoid conflict with another node, which has already sent its data packet, at common nodes, which is the solution. The constraints of this option are the limited number of wireless sensor nodes, especially around the gateway.

- A graph-routing algorithm should increase data packet delivery in WirelessHART networks. For this problem, the objective is to maximise the reach of a data packet from the source sensor node to the gateway by using path redundancy. If the sender node fails to receive acknowledgment from the receiver node after sending a data packet, it tries again by selecting another sensor node in its communication range. The constraints of this option involve the limited number of neighbour nodes in the gateway direction of the source sensor nodes.

After formulating a graph-routing problem with objectives (the values to be minimised or maximised) and constraints, it becomes easy to identify the best solutions using one of the optimisation techniques or by combining two or more of them.

5.4 Description of the System Model of Graph Paths Based on CMA-ES

This work focuses on how sensor nodes in IWSN monitoring systems create the uplink graph paths for the graph-routing algorithm to be used when transmitting sensor data to the gateway. Mesh topologies were selected because WirelessHART networks are commonly of this type, with static sensor nodes powered by batteries. The network is assumed to operate with this topology during simulations. Nodes are also assumed to inform the NM about poor connections with neighbours so that the NM can remove these connections from the network topology.

Firstly, this research proposes three graph-routing algorithms for creating and selecting graph-routing paths with a single objective, depending on the minimised Euclidean distance between sensor nodes (called PODis), the maximised residual energy (called POEng), and the minimised E2ET for each data packet between the transmitter and receiver (called POE2E). Using an optimised routing table, which keeps a list of the neighbours of each sensor node within its effective communication range in the gateway direction, the receiver node for each hop along the best paths of all objective functions is carefully chosen. As a result, this helps reduce overhead in the network and the energy consumed to maintain live sensor nodes throughout the network.

Secondly, after computing these objective functions, it is necessary to select the best solution by means of the proposed graph-routing algorithm termed ‘best path graph-routing with CMA-ES’ (BPGR-ES), which uses multiple objectives for creating and selecting the graph-routing paths.

5.4.1 Optimise Routing Table

After the deployment of sensor nodes in the network area, each sensor node submits its neighbour table in the Data-Link Layer (DLL) to the NM. The routing formation prior to the data transfer at the network layer can use this neighbour table and connected graph paths to construct a routing table for each sensor node. As wireless conditions change frequently in industrial environments, the NM must also frequently reconfigure and re-disseminate the routing graphs, which leads to increased energy consumption. To address this, as shown in the pseudo code in Algorithm 5.1, an optimised routing table saves the storage space required, and reduces the large overhead for the neighbour table by allowing each sensor node to select its neighbouring sensor nodes closest to the gateway. The neighbour table for each sensor node retains all its neighbours in each direction in the network area within an effective communication range.

Algorithm 5.1 Optimise Routing Table.

1: **Input:**
2: $Source\ Node$
3: **Output:**
4: $Optimise\ routing\ table\ baesd\ on\ Neighbouring\ Table$
5: **Start**
6: $CurrentHop = Source\ Node$
7: $CurrentHop\ read\ neigbouring\ Table$
8: $\boldsymbol{For}$
9: $\boldsymbol{If}\ Node(CurrentHup).neighbouringTable == G_w$
10: Calculate $Distance$ between $(Node(CurrentHop), G_w)$;
11: Add $Node\ G_w\ to$ routing table
12: $\boldsymbol{For}$
13: $Node(CurrentHop).neighbouringTable$
14: Calculate $Distance_{Hup}(Node(CurrentHop), Node\ (DNode))$
15: $\boldsymbol{If}\ Distance\_Hup < Distant$
16: Add $Node\ (CurrentHop)\ to$ routing table
17: **End**
18: **End**
19: **End**
20: **End**

To build the optimised routing table in this model for each sensor node in the network (as shown in lines 6 and 7 of Algorithm 5.1), the current node reads the neighbour table at the DLL, then identifies the Euclidean distance between the current sensor node and the gateway, as shown in lines 9 and 10. Euclidean distance is the distance between a sensor node and the gateway or between two sensor nodes in the network area with coordinates; for example, if sensor node i has coordinates (x_i, y_i) and the gateway has coordinates (x_g, y_g), the following equation can be used to calculate Euclidean distance [143]:

$$Euclidean\ distance = \sqrt{\left(x_i - x_g\right)^2 + (y_i - y_g)^2} \qquad (5.1)$$

The current sensor node verifies if the gateway is available in the neighbour table in lines 8–11. This indicates that the current sensor node can communicate directly with the gateway, as each sensor node has a communication range. From lines 12–16, if the current sensor node is outside the communication range of the gateway, it selects its neighbour nodes from the optimised routing table where the distance between the neighbour node and gateway is less than the distance between the current sensor node and gateway.

5.4.2 Propose Graph-Routing Paths with a Single-Objective

The system model selects the three single-objective graph paths for graph-routing in WirelessHART network based on CMA-ES, as presented in Figure 5.1, which portrays a flowchart of the proposed three single-objective graph-routing paths. The main operations are generating samples, evaluating the graph-routing paths based on three objective functions, adapting the covariance matrix, the path evolution, and the global step size, and outputting single-objective paths (PODis, POEng, and POE2E).

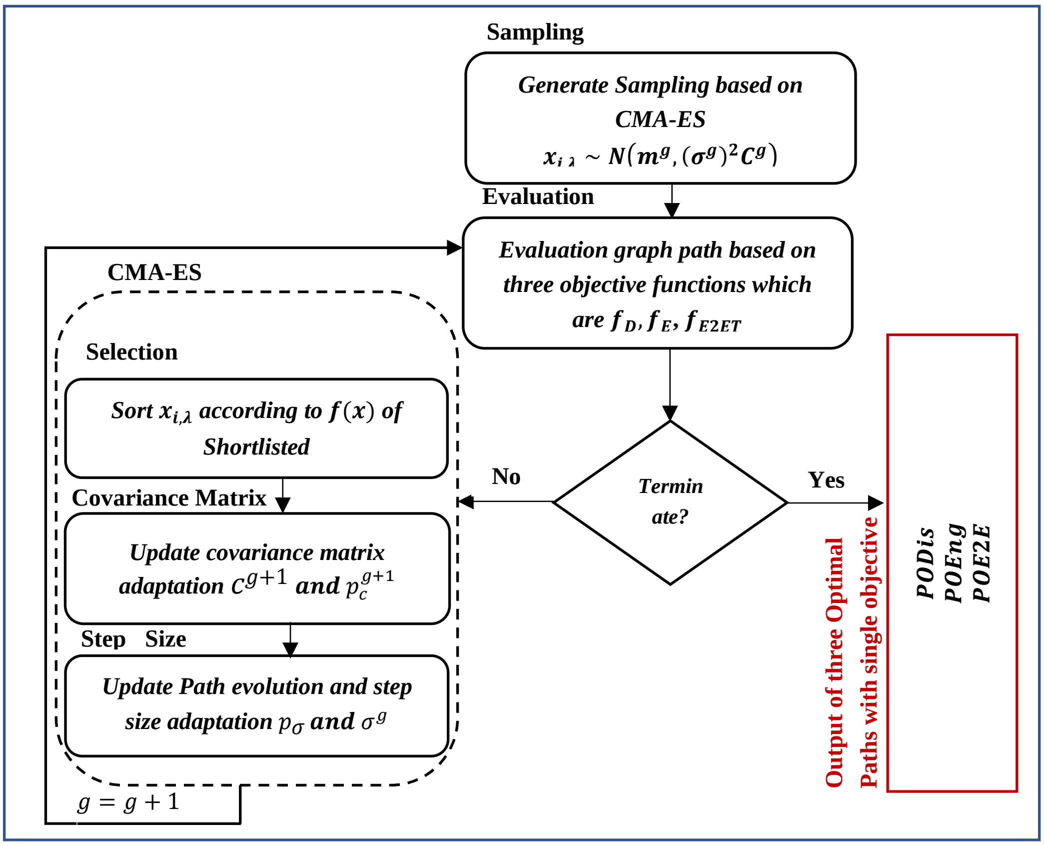


Figure 5.1 Flowchart of the proposed three single-objective graph paths.

Objective functions are used in our model to select the graph-routing path for each source node in the WirelessHART network. To achieve the first objective function which is called f_D, the Euclidean distance between the sensor nodes and their neighbours heading in the gateway direction is calculated. Subsequently, this is utilised to select the next hop to minimise transmission distance [144]. The second objective function is the residual energy for each node which is used to avoid dead nodes and low energy nodes. This is known as f_E. The third objective function is f_{E2ET} where the E2ET time is then considered in the selection of the next hop as a way to reduce conflict delay. The pseudo code of the three single-objective functions of CMA-ES, considered to select the graph path of the graph-routing algorithm, is shown in Algorithm 5.2 and discussed in greater detail below.

Algorithm 5.2 Objective Functions of Select the Graph-Routing Paths based on of CMA-ES.

```
Input:
   Optimised Routing Table of Source Sensor Node
Output:
   Graph paths based on CMAES algorithm
Start
For (RoutingTable)                                   ▷ The first objective function f_D
     Calculate Distance between(FinalDestination, Node(RouingTable))
      If Distance < MinD
        SelectedNodeD = Node(RouingTable);
              Find MinD = Distance;
     End
End
For (RoutingTable)                                   ▷ The second objective function f_E
     Find Energy of Node(RouingTable))
        If Node(RouingTable).CurrentEnrgy ≥ MaxE
          SelectedNodeE = Node(RouingTable);
              Find MaxE = Node(RouingTable).CurrentEnrgy;
       End
End
For (RoutingTable)                                   ▷ The third objective function f_E2ET
  Delta = FindDeltaFromPropagationModel();
  Calculate Distance between(FinalDestination, Node(RouingTable))
   Calculate E2ET = (c1 * nodeprocessor) + (c2 * d * delta);
        If E2ET ≤ minDelay
          SelectedNodeDelay = Node(RouingTable);
              Find MinDelay = E2ET;
       End
End
```

5.4.2.1 Minimise Communication Distance, $(\boldsymbol{f_D})$

Minimum communication distance is defined as the shortest distance between the currently sending node (source node) and its neighbours in the gateway direction, and is achieved by minimising D at the source node with the lowest communication cost. Thus,

$$f_D = Minimise\ (DPathFromnode_i, j) \tag{5.2}$$

Where $DPathFromnode_i, j$ is the Euclidean distance in the optimised routing table between the source node i and its neighbour node j toward the gateway and where f_D searches for the neighbouring node with the shortest distance for the source node i.

5.4.2.2 Maximise Residual Energy, $(\boldsymbol{f_E})$

Maximum residual energy is defined as the residual energy in the sensor nodes after they perform sensing, communication operations, and computation. Sensor nodes with higher residual energy are more likely to be selected as the next hop in the graph path, and each sensor node periodically uploads its residual energy value to NM. Thus,

$$RE_i = CE_i - ConE_i \tag{5.3}$$

Where RE_i is the residual energy of sensor node i, CE_i denotes the current energy of sensor node i and $ConE_i$ denotes the consumed energy of sensor node i.

$$f_E = Maximise\ (RE_i\ ) \tag{5.4}$$

Where $Maximum\ (RE_i\ )$ of $\boldsymbol{f_E}$, is the maximum residual energy of sensor node i. To ensure the quality of communication and increase data packet delivery in the IWSNs, each currently sending sensor node looks at an optimised routing table to locate the sensor node that has the maximum residual energy in the required path.

5.4.2.3 Minimise End-to-End Transmission Time, $(\boldsymbol{f_{E2ET}})$

The E2ET time measure proposed in this research refers to the time estimated for a given pair of nodes in the WirelessHART network to exchange a data packet. WirelessHART is a TDMA-based network protocol. Therefore, each communication is time-synchronised and this provides a timescale for nodes in the network. A fixed-length timeslot shared by all network devices is the basic time unit of communication activity. The timeslot provides a time base for scheduling the transmission of process data. In WirelessHART, a timeslot has a duration of 10 ms, which is sufficient to send or receive one packet per channel and its accompanying acknowledgement, including the guard-band times required for network-wide synchronisation.

In wireless networks, several mechanisms for time synchronisation are applied [145]–[147]. Time synchronisation algorithms use specific time message exchange mechanisms to reduce transmission delay. In this research, in order to obtain a definitive means of identifying the time estimated for data packet exchange between any two sensor nodes in the WirelessHART network, a Two-way Time Message Exchange (TTME) estimation model was applied between each pair of nodes in the sensor network, which uses uplink and downlink transmissions to figure out and fix the time difference between sensor nodes, as in [148]. This allowed the development of an equation that simulates the actual transmission time for each packet between the transmitter and the receiver based on the propagation model in a WirelessHART network [34], as shown in Figure 5.2.

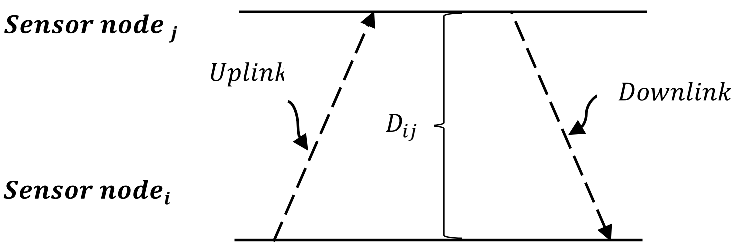


Figure 5.2 Two-way time message exchange between node i and node j.

The uplink-downlink time was modelled using equation (5.5); after it was calculated for each sensor node in the network area, sensor nodes send their time information to NM periodically:

$$T_{i,j} = c1\tau_i + c2D_{i,j}\partial_l \quad (5.5)$$

Where $c1 \in [0,1]$ and $c2 \in [0,1]$ are the node's processing delay time and channel delay time coefficients, respectively, τ_i is the processing time required by node i to process a data packet, $D_{i,j}$ is the distance between the transmitter node i and the receiver node j, and $\partial_l \in [0,1]$ is the delay time required to transfer a data packet from the transmitter node i and the receiver node j through channel l.

$$f_{E2ET} = Minimise\left(T_{i,j}\right) \quad (5.6)$$

Where the $Minimum(T_{i,j})$ of f_{E2ET}, is the minimum time estimated from source node i to receiver node j, which in the optimised routing table is a neighbour node of the source node i.

At the end of the CMA-ES, three graph-routing paths exist; specifically, PODis, POEng and POE2E. Finally, to select the graph BPGR-ES path and achieve a balance between the three objectives above and energy consumption between sensor nodes in the network, adaptation of the CMA-ES was employed to select the graph path based on three objectives.

5.4.3 Adaptation of CMA-ES for Multiple-Objectives Graph-Routing Path Selection

To meet the graph path selection criterion in an actual WirelessHART network, data packets are forwarded via the graph paths, as shown in Algorithm 5.3. In relation to the pseudo-code for the selection of BPGR-ES paths, two situations (equality or inequality) apply to the graph-routing paths of the BPGR-ES algorithm based on PODis, POEng, and POE2E.

Algorithm 5.3 Selection Graph-Path of Graph-Routing (BPGR-ES).

1: **Input:**
2: $PODis$; $POEng$; $POE2E$ ▷ Three single-objective graph paths
3: **Output:**
4: Graph-routing path *(BPGR-ES)*
5: ***Start***
6: $\boldsymbol{If}$ $isequal$ $((PODis, POEng)$ && $(POEng, POE2ET))$ ▷ In case equality
7: $GrapPath = POE2E;$
8: **End**
9: $\boldsymbol{If}$ $isequal(POEng, PODis)$
10: $GraphPath = PODis;$
11: $\boldsymbol{Else\ if}$ $isequal(PODis, POE2E)$
12: $GraphPath = POE2E;$
13: $\boldsymbol{Else\ if}$ $isequal(POEng, POE2E)$
14: $GrapPath = POEng;$
15: **End**
16: $\boldsymbol{If}$ $isnoequal$ $((PODis, POEng)$ && $(POEng, POE2ET)$ ▷ In case inequality
17: $Check\ number\ of\ the\ hops\ for\ all\ the\ paths$
18: $Select\ GraphPath < number\ of\ the\ hops$
19: $\boldsymbol{If}$ $all\ paths\ OR\ two\ paths\ have\ the\ same\ number\ of\ the\ hops$
20: $Check\ energy\ of\ the\ last\ node\ before\ the\ G_w\ in\ these\ paths$
21: $Select\ GraphPath > energy\ last\ node\ before\ the\ G_w$
22: **End**
23: **End**

In the case of equality, as shown in lines 6–8 of Algorithm 5.3, two cases arise: if all single-objectives of the objective functions for transmission are achieved in one path, this will be the graph-routing path of the BPGR-ES algorithm. Hence, this must be selected as the final path. In the second case in lines 9–12, if two potential single-objective graph paths follow the same path, priority will be given to any one of them as the graph-routing path of the BPGR-ES algorithm where two objectives have been achieved. In a situation where there is inequality between the single-objective graph paths, as observed in lines 16–21, the graph-routing of the BPGR-ES path with the least number of hops will be selected to reduce energy consumption and increase data packet delivery. However, as several single-objective graph paths may have the same number of hops, a further check on the residual energy of the last sensor node around the gateway is added to achieve balanced energy consumption. Subsequently, priority is given to the graph-routing path of the BPGR-ES algorithm, which has the highest residual energy at the last sensor node before the gateway.

5.5 Simulation Experiments

To study how the PODis, POEng, POE2E, and BPGR-ES perform under different numbers of sensor nodes, extensive simulations were conducted to evaluate their performance in two scenarios comparable with the baseline uplink algorithms; specifically, the Exhenced Least-Hop First Routing (ELHFR) algorithm [15] and the Energy-Balancing Routing algorithm based on Energy Consumption (EBREC) [20].

In Chapter 5 and Chapter 6, the duration of each simulation run was 4 hours, or approximately 10,000 rounds, with the initial 30 seconds ignored as the network's start-up period. As each run of the simulation presented a different node topology, with respect to the spatial distribution of sensor nodes, the performance metrics generated were for different values. Several simulations were conducted to verify whether algorithms produced similar performance levels, and therefore 15 random topologies were run for each algorithm in order to obtain a statistical mean for the results, which then produced the graphs shown throughout Chapters 5 and 6. Thus, the TCE, average EIF, PDR, and E2ET results for several repetitions of the simulation for each algorithm were obtained.

5.6 Evaluation of Results and Analysis

5.6.1 Data Packet Delivery Evaluation

This research focuses on a critical factor to evaluate the data packet delivery of the network: the PDR. Figure 5.3 show the PDR results for different node densities in a 100 ×100 m^2 network area. It can be seen that an increase in the delivery of data packets to the gateway resulted in a corresponding decrease in the packet loss ratio. This was due to all algorithms exploiting path redundancy and thereby boosting the data packet delivery of the network. More specific details of this ostensibly simple variation in results are explained below.

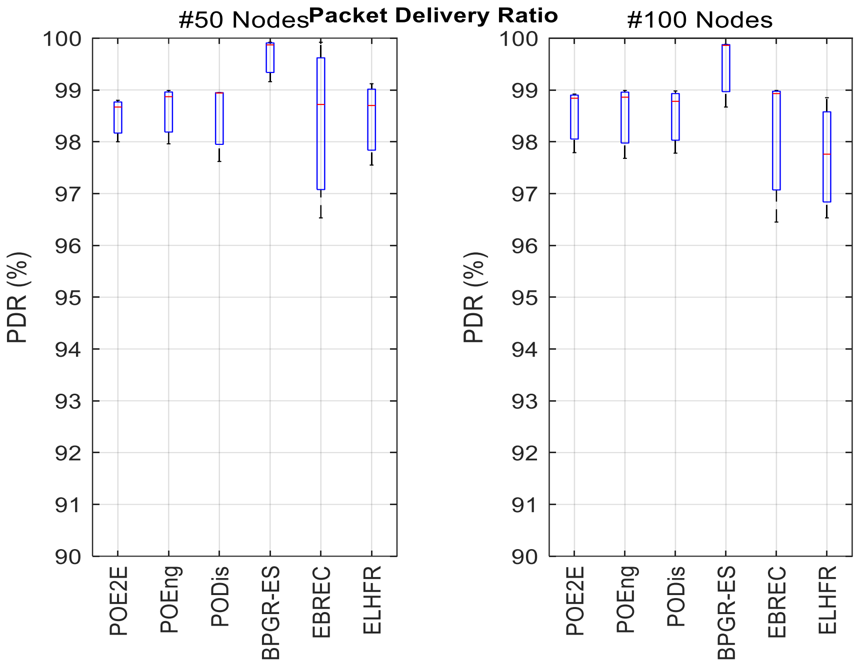


Figure 5.3 PDR results of proposed graph-routing paths with mesh topology.

Figure 5.3 illustrates the effect on the PDR of varying the number of sensor nodes: an increase to 100 sensor nodes caused a negligible decrease in the PDR across most algorithms compared to a 50-node density. This is logical, as the number of sensor nodes increases the traffic load in the network intensifies, causing congestion in some areas and data packet loss throughout the network. In consequence, the PDR is negatively affected.

However, as Figures 5.3 demonstrate, the proposed BPGR-ES approach attained the highest PDR results compared to other options, even when the number of sensor nodes was increased. This is explained by the mechanisms of the BPGR-ES algorithm which are used: it creates an optimised routing table, calculated according to Euclidean distance, which allows communication with all neighbouring nodes located in the direction of the gateway, regardless of whether they are at the same level or lower levels. This results in an increase in the availability of neighbouring sensor nodes outside the communication range of the gateway, and, therefore, produces an increase in the PDR. This was also observed in all single-objective graph paths, which also used the optimised routing table. Furthermore, by using the optimised routing table to select the neighbour for the next hop, the BPGR-ES not only ensures delivery of the data packet within its deadline, but also achieves the most reliable possible paths route by retransmitting data packets when the next hop is unable to receive them. Consequently, the number of lost data packets due to path redundancy is reduced, improving network throughput. Taking these important techniques into account explains the observed reduction in the packet-miss ratio. However, although the BPGR-ES presented similar PDR results for different network densities, a negligible decline in the PDR result was observed in the 100-node network compared to the 50-node density.

In general, the POE2E, POEng, and PODis graph paths all exhibited effective PDR results, achieving approximately 98% (see Figure 5.3). This is not unexpected: the single-objective graph paths use a similar approach to that of the BPGR-ES, including path redundancy and an optimised routing table, but with each focusing on just one requirement. As shown in Figure 5.3, this results in a slightly reduced PDR compared to the BPGR-ES approach.

The PDR results for both sensor node densities were lower for the ELHFR algorithm than for other algorithms, but they were still good. Since the ELHFR algorithm only permits sensor nodes to establish connections with neighbours located at lower levels in the BFS tree, fewer neighbours are typically available in the lower levels; therefore, it does not guarantee path redundancy for every sensor node [15].

5.6.2 Energy Consumption Evaluation

The performance of the proposed approaches was evaluated with respect to energy consumption in terms of both TCE and average EIF of the energy balance. This is important, as the WirelessHART network is centralised, making balancing energy consumption between sensor nodes a key target [149].

Figure 5.4 illustrates the energy consumption results of the 100 ×100 m^2 network area for densities of 50 sensor nodes and 100 sensor nodes. It is evident that, in contrast to the other algorithms, the BPGR-ES algorithm reduced the total energy consumption of the 50-node network. However, in the denser networks, a comparable energy consumption total was achieved by both the proposed BPGR-ES approach and the EBREC. This is due to the EBREC algorithm taking into consideration the remaining energy of a sensor node when communicating with the nodes in the network. Nevertheless, the BPGR-ES approach fares better concerning TCE because, as shown in Figure 5.3, the network connection is better.

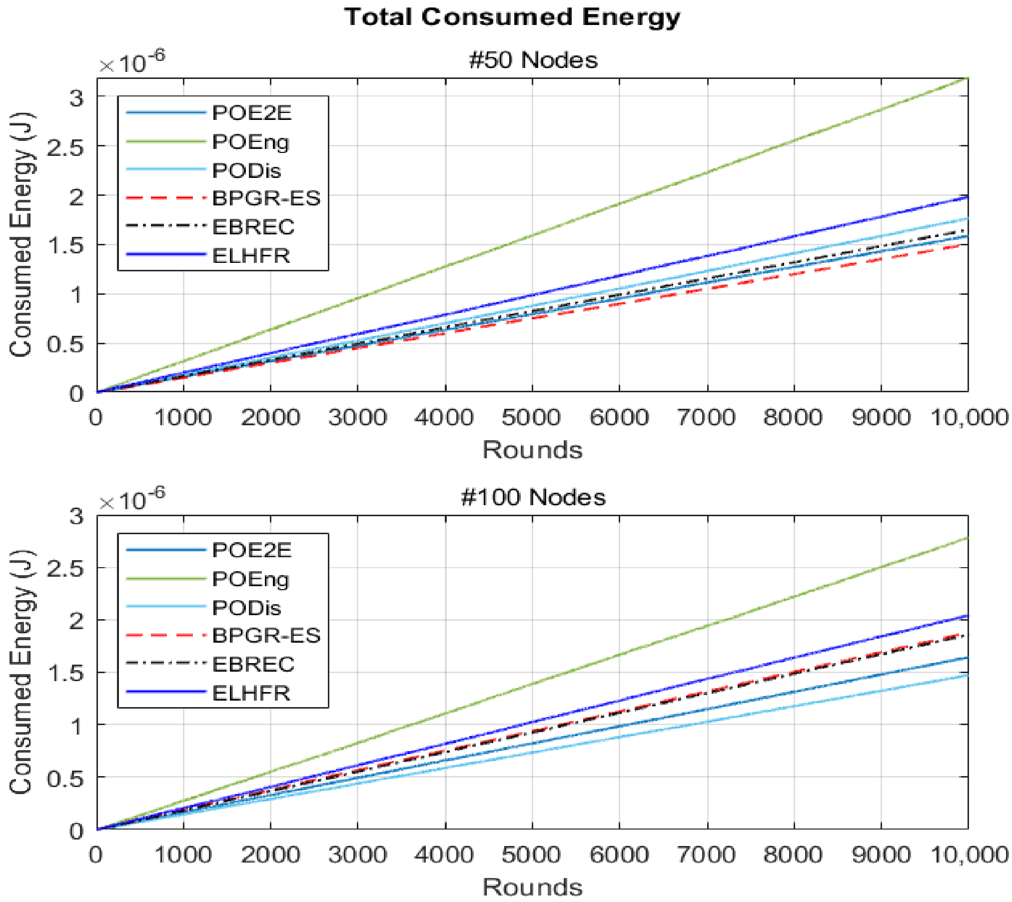


Figure 5.4 TCE results of proposed graph-routing paths with mesh topology.

It is of note that in the PODis, a clear reduction in total energy consumption is experienced by increasing the density of the sensor nodes. An increased number of sensor nodes enables the shortest path to be selected in the most expeditious manner, thereby significantly reducing energy consumption. This stems from reliance upon the optimised routing table, which determines the neighbours for each sensor node by calculating hops with the least Euclidean distance to the gateway. However, this significantly increases the energy consumption imbalance of the network sensor nodes (see Figure 5.5). Thus, it can be concluded that the PODis fails to balance energy consumption. This is a logical outcome as the energy consumption of the sensor nodes located nearest to the gateway will increase due to their constantly being selected. According to Euclidean distance, these nodes will always provide the shortest path from the source nodes sited further away from the gateway.

In POEng, the TCE sharply increased in the different network densities compared to other algorithms. This is illustrated in Figure 5.4. Because it selects neighbouring sensor nodes by seeking those with the highest residual energy from an optimised routing table without taking the distance from the gateway into account. This multiplies the number of hops that have been noticed while the simulation is running: increasing energy consumption compared to other algorithms where proximity to the gateway is not considered to reduce the number of hops. Typically, the number of hops in POEng is higher than other graph paths. Consequently, as shown in Figure 5.5, the high energy consumption of the POEng has a clear impact on the energy consumption imbalance among the network nodes.

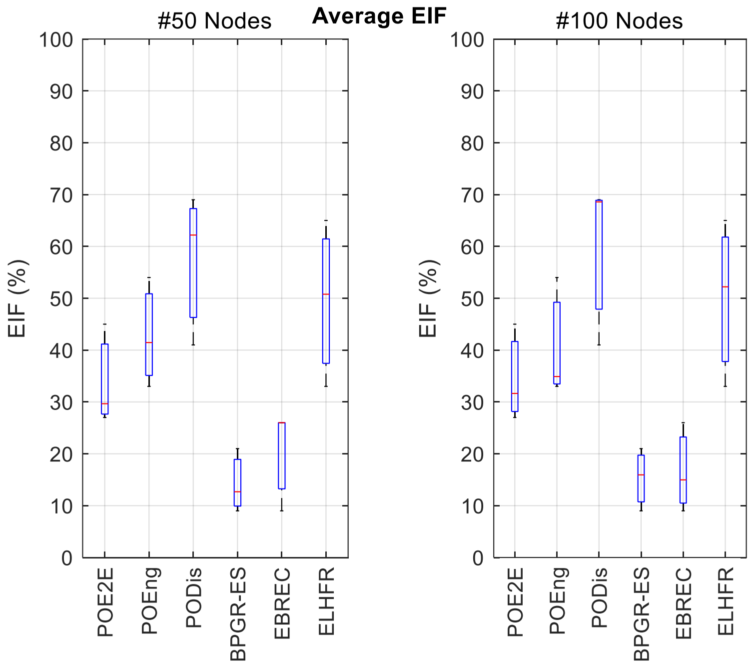


Figure 5.5 EIF results of proposed graph-routing paths with mesh topology.

The average EIF of the proposed BPGR-ES approach was also significant, being the smallest among those tested. This is elucidated in Figure 5.5, which presents the average EIF results for a network area of 100 ×100 m^2 with a density of 50 and 100 sensor nodes. The high use of the sensor nodes around the gateway compared to other nodes within the network resulted in a reduction in the average residual energy, leading to a corresponding increase in the average EIF. The selection of graph paths by the BPGR-ES algorithm, however, takes into account all IWSN requirements. Therefore, the energy of all sensor nodes in the network is closer to the average energy than in any of the other approaches. In addition, the BPGR-ES approach only selects the sensor nodes with the highest remaining energy adjacent to the gateway when all graph paths have the same number of hops. Therefore, the proposed BPGR-ES algorithm achieves a better balance in terms of energy consumption between sensor nodes in the network area than other algorithms. In contrast, although all sensor nodes in the EBREC algorithm route their data packets via neighbouring sensor nodes that have a greater residual energy, this is insufficient to ensure a balance in energy consumption between all sensor nodes in the network. In particular, if several sensor nodes forward their data packets to the same node, the selected sensor node will take on a critical role, potentially leading to imbalances in energy consumption in the network.

In addition, the ELHFR algorithm significantly increases average EIF due to its consideration of least-hop as the only selection metric. This does not consider the increased energy consumption of the nodes around the gateway, as sensor nodes closer to the gateway become overburdened with high traffic loads compared to those further away. Therefore, these overloaded nodes will expire much faster than the other sensor nodes due to such imbalances in energy consumption.

5.6.3 End-to-End Transmission Time Evaluation

A further experiment examined the proposed approaches in terms of End-to-End Transmission (E2ET) time. Monitoring systems often have delay needs of fewer than 100 ms, whereas factory automation has even stricter delay requirements ranging from 2 to 25 ms [23].

Figures 5.6 and 5.7 show the various E2ET results for the network topologies for a 10,000-round run of each algorithm's E2ET results for the 100 ×100 m^2 network area of 50 and 100 sensor nodes, respectively. The simulation results clearly show that the E2ET results of all algorithms increased with the 100-node density compared to the 50-node density, where the increased network traffic prompted a rise in the nodes' multi-hop behaviour and path redundancy. Consequently, an increase in the E2ET occurred due to several data packet retransmissions from the sensor node to the gateway, which caused queuing and other delays.

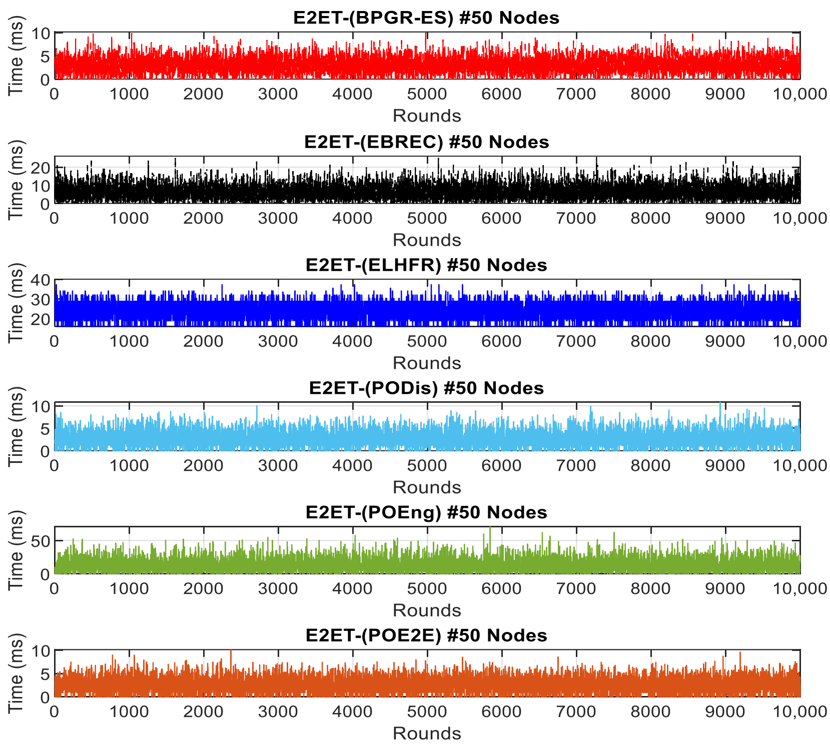


Figure 5.6 E2ET time results for 50-node of proposed graph-routing paths with mesh topology.

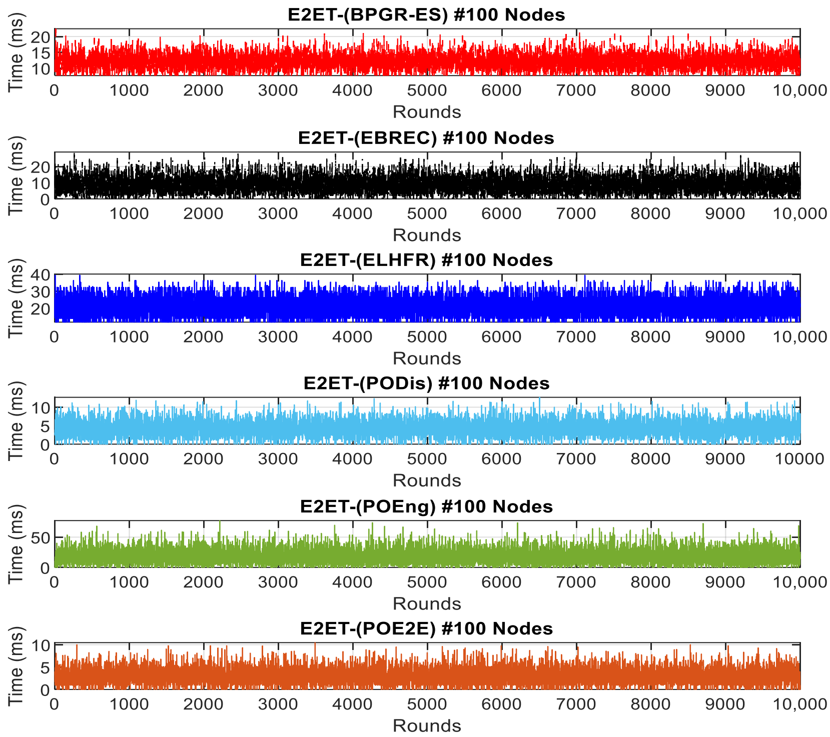


Figure 5.7 E2ET time results for 100-node of proposed graph-routing paths with mesh topology.

It is nevertheless evident that the single-objective POE2E of the graph-routing algorithm gave the lowest E2ET results compared with all other algorithms at both the 50-node and 100-node network densities, where the highest transmission time of POE2E in a dense network reached 11 ms, as shown in Figure 5.9, which presents the Cumulative Distribution Function (CDF) of E2ET time results that allows for easier and clearer comparison between different graph-routing algorithms. This is justifiable for the following reasons: the POE2E calculates the estimated time of a pair of sensor nodes to select the least transmission time between a source node and receiver node to form the next hop in the best path, and improves packet delivery by preventing the use of unreliable paths. All delays due to the retransmission of lost packets are, therefore, reduced. Moreover, the POE2E decreases congestion at the nodes by selecting the neighbours that can best deliver the data packet within its deadline. This is achieved by using an optimised routing table to choose the next hop from available neighbours, hence facilitating faster delivery of data packets and reducing the delay. Even if transmission of the network data packets is broken, the intermediate nodes will not spend any time searching for the next hop for the retransmission of data packets, which accelerates packet forwarding procedures in terms of reaching the gateway.

This was also observed in the E2ET results of the PODis, where transmission times reached 11 ms at 50-node density and 12 ms at 100- node density, as shown in Figures 5.8 and 5.8, respectively. Some delays occurred in the E2ET results of the PODis approach compared with POE2ET because the PODis approach selects graph paths as the shortest paths, increasing network traffic, especially at sensor nodes around the gateway, and prompting delays. In addition, the BPGR-ES approach selects graph paths as in the POE2ET approach in one case if this is the best path among all of the single-objective graph paths but prioritises the balance of energy consumption in the other cases. Thus, as shown in Figure 5.8, compared to the POE2ET approach, the transmission time increased up to 21 ms for the BPGR-ES.

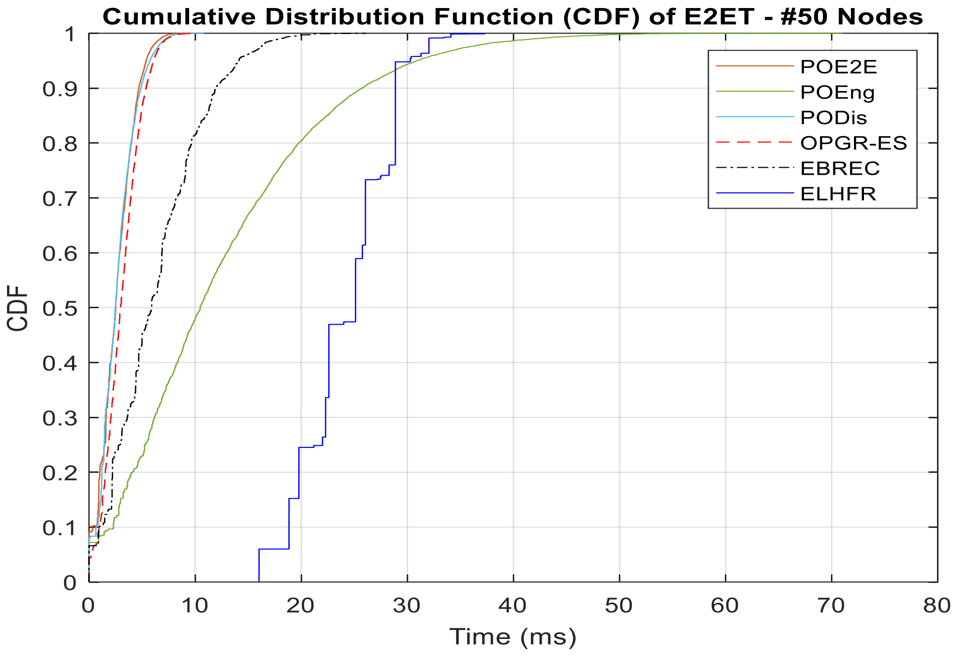


Figure 5.8 CDF of E2ET time results for 50-node.

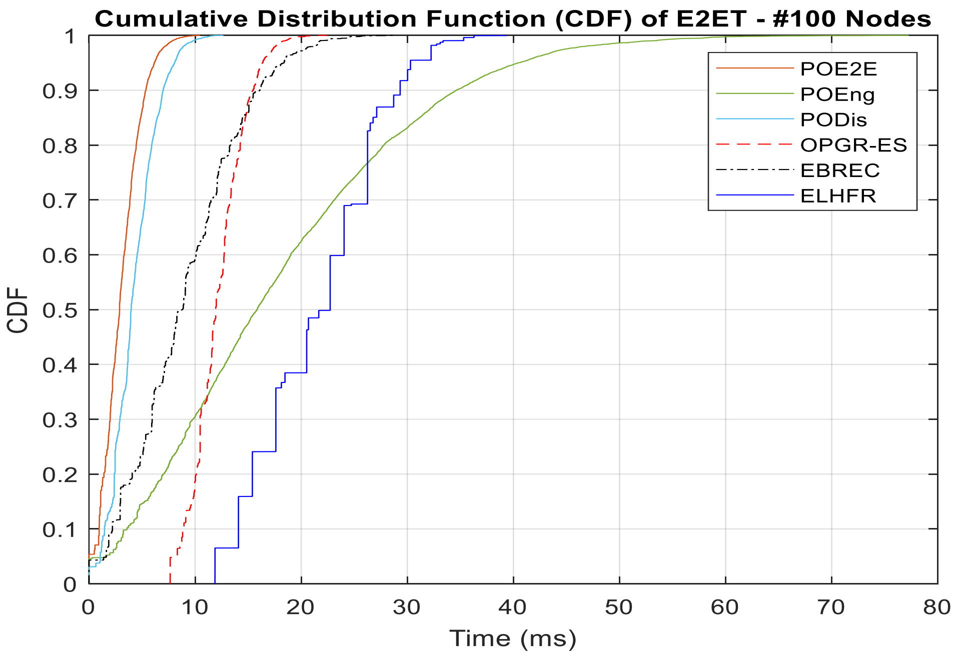


Figure 5.9 CDF of E2ET time results for 100-node.

Figure 5.8 illustrates a significantly sharp increase in the E2ET results of POEng, with the transmission time reaching 72 ms. The main reason for this is the increased number of hops, which caused data packets to arrive at the gateway with a noticeable delay. In the ELHFR and EBREC algorithms, however, because of the BFS tree, data packets could not avoid heavily congested regions. Therefore, it may have taken a long time for them to look up options in the neighbouring table to locate the next hop node in the path. Consequently, this increased the transmission time due to the retransmission of multiple data packets striving to reach the gateway.

5.7 Performance Comparison

Table 5.1 presents the performance comparison based on the results in Section 5.6 comparing the proposed POE2ET, POEng, PODis, and BPGR-ES approaches with the state-of-the-art graph-routing algorithms. The performance comparison considered the following four aspects.

- Criteria of paths: the primary path and formula specified by the graph-routing algorithm for each sensor node (i.e., the path by which a sensor node will attempt to send a data packet for the first time) and what the criterion is for this selection.
- Data Packet Delivery: the ratio of delivery of data packets to the gateway, measured by taking the average of the PDR results for each algorithm across all of the network's densities.
- Balance of energy consumption: the ratio of energy consumption balance between all sensor nodes in the network area is determined by averaging the EIF results for each algorithm at different network densities.
- Transmission time: the lower and higher transmission times in the E2ET results for each algorithm through various densities of the network determine.

Table 5.2 Comparison of proposed graph paths with graph-routing algorithms.

Algorithm	Criteria of paths	Packet Delivery Ratio	Balance Energy Ratio	Transmission Time
POE2ET	Lower transmission time of CMA-ES	98%	69%	Between 4 to 11 ms
POEng	Highest residual energy of CMA-ES	98%	61%	Between 12 to 72 ms
PODis	Shortest distance of CMA-ES	98%	34%	Between 4 to 12 ms
BPGR-ES	Multiple Objectives of CMA-ES	99%	85%	Between 4 to 21 ms
EBREC [20]	Highest residual energy of BFS	98%	79%	Between 8 to 27 ms
ELHFR [15]	Highest received signal level of BFS	98%	48%	Between 11 to 39 ms

An analysis of the performance of the four proposed graph paths of the graph-routing algorithm in this research and existing graph-routing algorithms, based on IWSN's requirements results in the following observations.

- The POE2ET graph path ensures data packet delivery and reduces transmission time in the network. In addition, it guarantees, to some degree, the balance of energy consumption between the sensor nodes in the network area compared to other single-objective paths. This is achieved by avoiding the selection of a sensor node on its path that simultaneously conflicts with another sensor node.
- The POEng graph path ensures data packet delivery in the network. It does not, however, attempt to reduce the number of hops when selecting sensor nodes along its path, resulting in a negative impact on energy consumption (see Figure 5.4), transmission time, and energy balance.
- The PODis graph path, like the POE2ET ensures data packet delivery and reduces transmission time in the network. Even though it reduces network energy consumption (see Figure 5.4), it does not achieve a balance of energy consumption between the sensor nodes in the network area. This is because node selection is based on the shortest graph paths of the graph-routing algorithm, causing sensor nodes near the gateway to be overloaded.
- The BPGR-ES graph path of the graph-routing algorithm selects graphs based on multi-objectives. This ensures data packet delivery, effective transmission time, and energy balance in the network. Although the BPGR-ES graph path has a longer transmission time than the POE2ET and PODis graph paths, it is still within the range of industrial automation transmission times.
- The EBREC [20] algorithm uses the BFS algorithm and selects graph paths based on residual energy. It ensures data packet delivery and, to some extent, energy balance and transmission time. However, as sensor node selection is dependent on the higher energy value of the sensor node on the next hop along the graph path, the EBREC algorithm is more effective at balancing energy consumption between sensors on all single-objective graph paths.
- The ELHFR [15] algorithm uses the BFS algorithm and selects graph paths based on received signal level. This ensures the data packet delivery of the network. It also uses the BFS algorithm to divide the network area into layers. This may help reduce energy consumption (see Figure 5.4) and transmission time compared to the POEng graph path, and balance energy consumption between sensor nodes in the network area compared to the PODis graph path.

5.8 Summary

This chapter adopts a CMA-ES to establish the graph paths of a graph-routing algorithm for the WirelessHART network that also provide path redundancy. Firstly, this research proposed three graph paths, each based on a single-objective function for CMA-ES, according to the different performance requirements of IWSNs considered in this research: (1) a graph path based on the minimum distance between sensor nodes in the direction of the gateway (PODis); (2) a graph path based On maximum residual Energy (POEng); (3) a graph path based on the minimum End-to-End transmission time (POE2E). Secondly, this research proposes a graph path of graph-routing evolution strategy algorithm (BPGR-ES) that selects the best hops on the basis of multiple objectives to achieve balanced energy consumption as well as a balance among IWSN requirements.

The results revealed a reduction in E2ET across all network densities for the POE2ET graph paths of the graph-routing algorithm. Additionally, the PDR values were good for all proposed approaches, which increased the data packet delivery of the network. Despite the fact that TCE for PODis graph paths of the graph-routing algorithm outperformed BPGR-ES in small networks, and that TCE for BPGR-ES and EBREC algorithms was somewhat similar in dense networks, the BPGR-ES algorithm achieved an 85% better energy balance among all sensor nodes in the network in terms of energy balance. It is also noteworthy that all single-objective graph paths of the graph-routing algorithm did not achieve balanced energy consumption over a mesh topology.

Therefore, the next chapter strives to implement single-objective graph paths of a graph-routing algorithm with unequal clustering topology to evaluate its performance, especially in terms of energy balance.

Chapter 6

Evaluation of Single-Objective Graph Paths Using Pre-set Unequal Clustering

6.1 Overview

As seen in Chapter 5, the application of multi-hop network paths and path redundancy via graph-routing can significantly enhance the data packet delivery of IWSNs. However, the centralised management of the WirelessHART network promotes an imbalance in the energy consumption of the battery-powered wireless sensor nodes. This creates a hotspot problem in the graph-routing of the network. Chapter 5 also discussed the performance of single-objective graph paths within the graph-routing algorithm of the WirelessHART network in mesh topology. Simulations were conducted using the CMA-ES algorithm to optimise the graph-routing algorithm through creating and selecting the best graph paths. The present research revealed that this approach is ineffectual in terms of achieving balanced energy consumption.

Chapter 6, therefore, addresses this problem by exploring the effect of combining the single-objective graph paths of the graph-routing algorithm with a pre-set unequal clustering topology. Section 6.2 describes the problems engendered by the application of a single-objective graph path in a mesh topology. Section 6.3 provides a literary review. Section 6.4 provides a detailed description of the system model proposed for the single-objective graph paths of the graph-routing algorithm. Sections 6.5 and 6.6 explain the setup of the simulation and present its performance evaluation. Section 6.7 compares the performance of single-objective graph-routing paths in mesh topology with the IWDDR topology. The final section of this chapter provides a summary.

6.2 Problem Description

This chapter focuses on the resolution of two key problems: to optimise the balance of energy and node isolation.

1: The single-objective graph paths of a graph-routing algorithm in mesh topology focus on just one of the requirements of the IWSN. They do not achieve a balanced energy consumption, and may cause hotspot problems in the network area. This includes: a graph path based on the shortest distance between sensor nodes in the direction of the gateway (PODis); a graph path based on maximum residual energy (POEng); and a graph path based on minimum end-to-end transmission time (POE2E).

To optimise the balance of energy consumption in the graph paths mentioned above, this research evaluates the performance of the three single-objective paths when using a pre-set unequal clustering topology. This topology is provided by the Improve WirelessHART Density Controlled Divide-and-Rule (IWDDR) topology.

2: The IWDDR topology is based on the WDDR topology which was developed in Chapter 4 of this thesis. Observation of the WDDR topology revealed an interesting phenomenon: due to its static approach, it isolates nodes – sensor nodes that are incapable of communicating with other sensor nodes in the same cluster. This phenomenon appears in the clusters located at a distance from the gateway, as these tend to be larger than those in closer proximity. In this chapter, the IWDDR topology detects isolated nodes in each static cluster of the WDDR topology and mitigates them. Lost data packets and latency are overcome by creating new clusters for the isolated nodes. The objective function of CMA-ES is then used to select the best Cluster Head (CH). Selection is achieved by considering node centrality between the nodes in the same cluster and the distance between other CHs or the gateway.

6.3 Avoiding Isolated Nodes During Cluster Formation

The isolation of sensor nodes in the network area can be caused by clustering algorithms that are inadequately designed or that randomly select CHs. A node isolated from its cluster presents two possibilities: if the gateway is in communication range, it can communicate directly with the gateway, but, if it is out of communication range, it cannot communicate with any other sensor nodes, it is completely isolated. In the first case, the node will consume an excessive amount of energy, instigating an imbalance in energy consumption between the sensor nodes in the network area. In the second case, the data packet will be lost. Considerable previous research effort has been devoted to finding a method for mitigating isolated nodes during cluster formation. This section focuses on the most current and advanced of these research solutions.

The authors in [150] proposed Regional Energy Aware Clustering with Isolated Nodes (REAC-IN), which improves the CH selection process and solves the problem of node isolation. The REAC-IN cluster is founded on the LEACH protocol [47]. It determines whether the isolated node should send its data directly to the gateway or via the CH. The selection of CHs is determined by three factors: the distance between the isolated node and the gateway, the residual energy in each sensor node, and the average amount of energy in all sensor nodes within each cluster.

Authors in [59] proposed the Distributed Fault-tolerant Clustering and Routing (DFCR) algorithm for run-time recovery of nodes that do not have a CH within range. This solves the problem of node isolation and makes the algorithm fault-tolerant as well as energy-efficient. On the other hand, the Weibull distribution fault model [151] can be used to attempt failure recovery by encouraging isolated nodes in the network to relay their data packets to a neighbouring node that belongs to a cluster.

The authors in [152] proposed an uneven clustering routing algorithm based on optimal clustering. The algorithm is self-adaptive, enabling it to deal with isolated nodes by directing them to send their data packets to the nearest cluster, and thus via the CH to the gateway. Compared to the LEACH algorithm, this technique proves more effective in reducing energy consumption and prolonging the network lifetime [47].

An alternative solution to the problem of node isolation, is the use of an energy-efficient clustering technique with isolated nodes (EEC-IN) [153]. In EEC-IN, a membership handshake method is proposed. During cluster formation, an isolated node is invited to shake hands and become a cluster member with its nearest CH, calculated according to the available residual energy. This technique increases the stability and lifetime of the network.

A Threshold-sensitive Energy-Efficient sensor Network with Isolated Nodes (TEEN-IN) was proposed in [154]. This is comparable to REAC-IN [150], except the formation of the research cluster is based on Threshold-sensitive Energy-Efficient sensor Network protocols (TEEN) [155]. According to the TEEN-IN protocol, the CH selection is determined by the ratio between the energy of the sensor nodes and the average energy in a cluster. In comparison with the TEEN protocol, which randomly selects the CH to accommodate the isolated node, TEEN-IN reduces energy consumption by an average of 10%.

The Joint Routing and resource allocation with Isolated Nodes technique (JR-IN) was suggested by [156] to deal with isolated nodes in unequal clustering. In this technique, the network area is divided into layers, and each layer is further subdivided into circular clusters that become smaller as they move towards the sink nodes. In addition, an isolated node located in any layer will act as a CH for that layer if the original CH fails to transmit aggregated data to the sink node. JR-IN achieves effective results in terms of network lifetime, throughput, and delay.

A recent solution, proposed for WSNs in the IoT, implements an efficient Second-Fold Clustering (SFC) algorithm [157] which divides the network into grids. The selection of the CH is determined by calculating the residual energy and the degree of connectivity between sensor nodes in the same cluster. Any isolated nodes within communication range are then clustered, and their CH is chosen based on their zonal degree of connectivity. This technique uniformly distributes sensor nodes and reduces energy consumption. However, it is not suitable for heterogeneous networks.

The above literature review has shown that, in general, most solutions to isolated nodes involve connecting them to the closest CH or creating new clusters for them; however, each algorithm uses a specific technique to achieve this end. The result is a reduction in energy consumption and an extended network lifetime.

6.4 Network Model

This section provides a detailed description of the proposed system model. The deficiencies of using single-objective graph paths for graph-routing in the mesh topology of the WirelessHART network were presented in Chapter 5: unbalanced energy consumption leading to hotspots, low coverage, and short network lifetime. To address these challenges, the proposed method principally evaluates single-objective graph paths using pre-set unequal clustering. The flowchart in Figure 4.1 illustrates this concept in more detail.

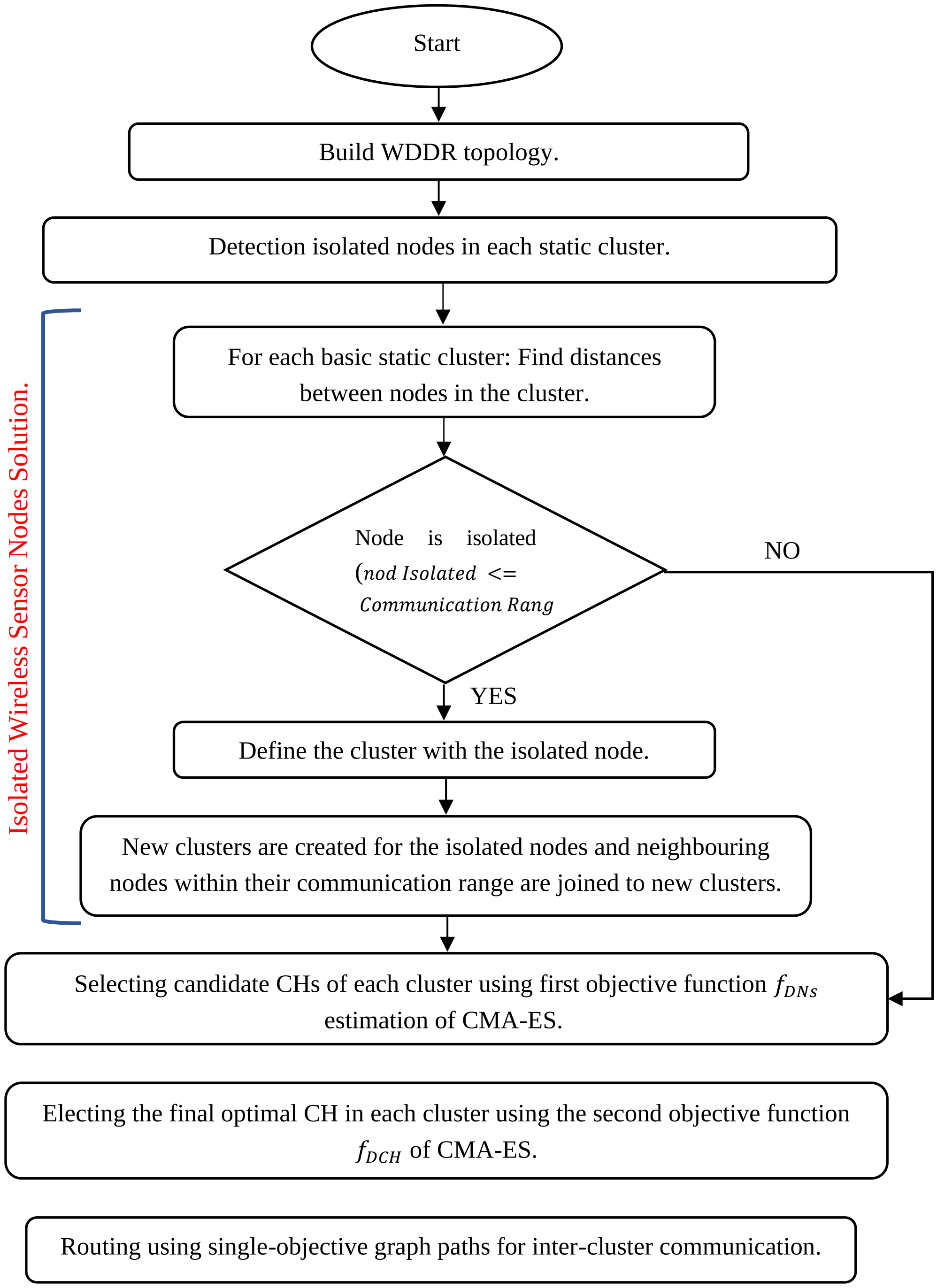


Figure 6.1 Flowchart of the proposed methodology.

6.4.1 Clustering Formation Stage

Balanced energy consumption in this research work is achieved by using the pre-set unequal clustering process, as seen in the WDDR topology (see Chapter 4). Then, the isolated sensor nodes are initially detected in each static cluster in the WDDR topology, based on the communication range of sensor nodes in the same cluster. Using Algorithm 1, a new cluster is made based on the location of these nodes. Candidates for efficient CHs are selected based on their location in relation to other CHs and the centrality of the node in the cluster. Therefore, this pre-set unequal clustering algorithm is called the Improved WDDR (IWDDR) topology.

6.4.2 Detection of Isolated Sensor Nodes in Each Static Cluster

As illustrated in the pseudo code in Algorithm 6.1, the proposed technique makes it possible to find isolated nodes, from the static approach of the pre-set unequal clustering of the WDDR topology, and to create new clusters for them.

Algorithm 6.1 Detection isolated sensor nodes in the clusters.

```
1: Input:
2:   Main Static Clusters of WDDR topology
3: Output:
4:   New Clusters of isolated sensor nodes of IWDDR topology
5: Start
6:   For 1:C9                                        ▷ Number of basic static clusters
7:      Find distances between nodes in each static clusters
8:      Check (nodIsolated <= CommunicationRange);
9:        If nodIsolated
10:          Define the cluster which has an Isolated node.
11:          Calculate means of this cluster.
12:             If mod(nodIsolated) == 0
13:                Means result is even for vertical cluster.
14:                Looking for the y positions to separationY.
15:                Add sensor nodes in new clusters.
16:           Else
17:                Means result is odd for horizontal cluster.
18:               Looking for the x positions to separationX.
19:              Add sensor nodes in new clusters.
20:      End
21:  End
22:       Select CHs using CMAES
23:     Objective Functions of CMAES
24: End
```

Algorithm 6.1 carries out this function, as demonstrated in lines 6-8 regarding the identification of isolated nodes in every static cluster. First, it defines the Euclidean distance between sensor nodes in the same cluster, and then it checks whether there is any sensor node outside the communication range of other nodes in the same cluster. An isolated node is determined in the case where a sensor node is incapable of communicating with another sensor node. The goal is that when selecting any node as a CH it can communicate with any other sensor node in its cluster.

The mean location of an isolated node is calculated, as seen in lines 9-19 which, in turn, serves as the basis for the generation of the new cluster. This effectively necessitates the capacity of all sensor nodes within the same cluster to communicate with one another. The IWDDR topology divides static clusters into vertical and horizontal clusters as shown in Figure 6.2a, where C1, C2, ..., and C9 are the names of the clusters. Hence, sensor nodes close to the isolated nodes and within its communication range in the new cluster are incorporated following the generation of the new cluster, which, in turn, is based upon the location mean of the isolated node's coordinates. This is exemplified by C9 dividing into two clusters to generate a new cluster for node 29 as it is incapable of communicating with sensor nodes 43, 63, 10, 24, and 68. Then incorporation sensor nodes 30, 5, and 52 with a new cluster because they can communicate with an isolated node. Thereby making it an isolated node, as portrayed in Figure 6.2b. Finally, the CMA-ES algorithm is used to select the best CHs from each cluster. The following section expands on this process.

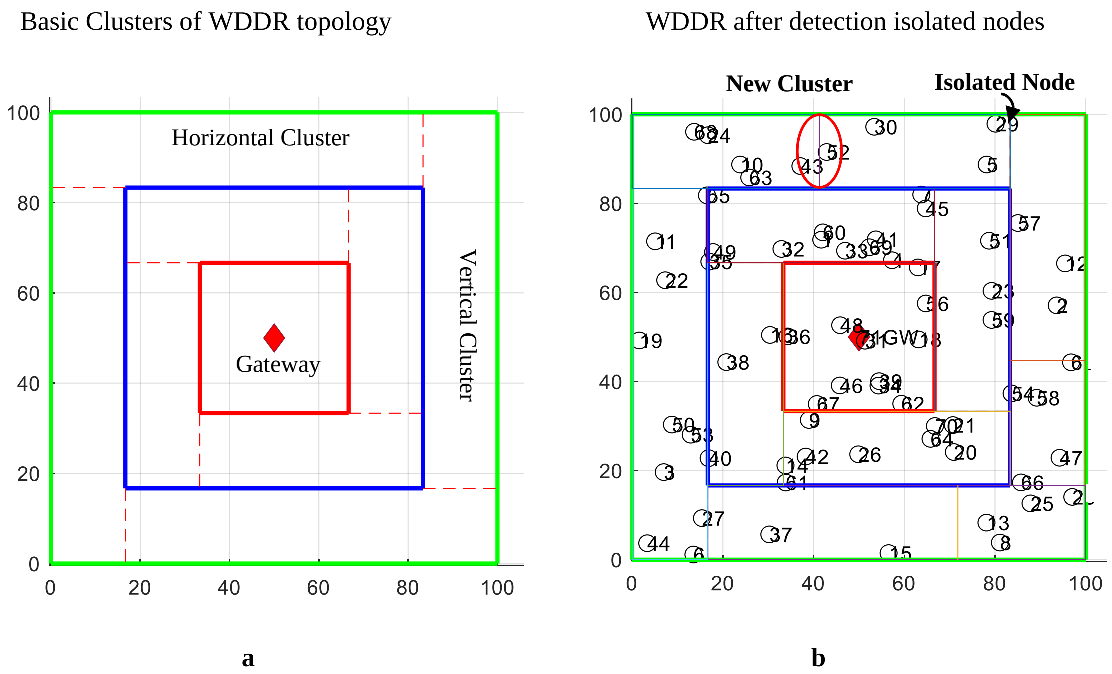


Figure 6.2 WDDR topology before and after detection isolated node.

6.4.3 Optimisation of Cluster Heads Selection

An appropriate selection of CHs needs to be considered in this type of clustering because of the static approach of the IWDDR topology. This affects communication with the nodes in their own cluster and other CHs or the gateway in WirelessHART networks. It is noted that energy consumption is diminished when the transmission distance is shorter [111].

Therefore, the final CHs are selected using the CMA-ES algorithm based on two steps. First, tentative CHs are selected for each cluster using node centrality. Second, the best CH is chosen based on its location in relation to the gateway or other CHs. To diminish the overhead of the network, rather than altering the CHs in each round, CHs are re-selected via the same stages when they consume half their initial energy, which represents the energy threshold in this research.

Initially, several tentative CHs are selected in each cluster to compete for the role of actual CH. This uses the first objective function f_{DNs} of CMA-ES, which selects tentative CHs which are sensor nodes central to their own neighbour nodes in the same cluster. Other nodes keep sleeping until the CH selection phase ends.

Suppose that node i becomes a tentative CH, called s_i. s_i has a second objective function f_{DCH}, which is a function of its distance to the gateway and other CHs that will be discussed later. The goal is that if s_i becomes a CH at the end of the competition, this will be the best CH based on the CMA-ES algorithm where the best CHs are situated most proximally to their neighbours, the gateway, or other CHs. The reason for this is to reduce energy waste, minimise time delays and improve data packet delivery.

1. **Node centrality:** The extent to which a CH is situated centrally in relation to neighbouring nodes in the same cluster is representative of the node's centrality. The length of the transmission path largely determines the energy dissipation of the node. When the chosen node possesses less transmission distance towards the gateway, the energy consumption of the node is smaller [111]. The distance from CH to the normal sensors is illustrated in the pseudo code in Algorithm 6.2 below.

Algorithm 6.2 Pseudocode of node centrality.

```
1: Input:
2:   Sensor nodes in each cluster
3: Output:
4:   Tentitave CHs in each cluster
5: Start
6:  For  j = 1: Clusters in network area
7:     SelectNode
8:     MinNodeDis = Communication Range
9:      For  i = 1: Nodes in cluster (s_i)
10:        Set  NodesDistsSum to 0
11:         If SelectNode ≈ s_i
12:           If NodesDistsSum < MinNodeDis
13:              MinNodeDis = NodesDistsSum
14:      End
15:  End
16:  Find f_DNs = Minmum (MinNodeDis) as Tentitave CHs
17:  End
```

This pseudo code enables tentative CHs to be selected from a group of sensor nodes in the same cluster. The selection is based on the distance between the nodes: those with a minimum distance to all the other sensor nodes, excluding themselves, are selected as tentative CHs. Lines 7–8 demonstrate initialising each sensor node in the cluster as *selectNode* for all input sensor nodes in the same cluster, and *MinNodeDist* sets the maximum communication range between two sensor nodes. For each *selectNode* the distance between the input node and all other sensor nodes in the same cluster (excluding itself) is computed. This is demonstrated in lines 9–10. If the distance is smaller than the current minimum distance for the *selectNode*, then the minimum distance for *MinNodeDist* needs to be updated, as shown in lines 12-13.

Once these steps have been completed for all the sensor nodes in each cluster, the objective function, f_{DNs}, identifies the sensor nodes with the minimum distance from other sensor nodes in the same cluster and selects them as tentative CHs.

$$f_{DNs} = Minmum\ (MinNodeDist) \qquad (6.1)$$

Additionally, the best CHs should support inter-cluster communication, whether with the gateway or other CHs, to deliver data packets to the gateway; not all tentative CHs necessarily communicate with the gateway or other CHs. This research needed to control the distance between CHs capable of directly communicating with the gateway, or other CHs through the selection of best CH which reside at the minimum communication distance. Therefore, the second objective function f_{DCH} of CMA-ES algorithm serves as the basis for the control of distance and the selection of best CHs.

2. **Minimum distance between the CHs and the gateway:** First, each tentative CH checks if it can connect directly with the gateway; if not, it will check the distance between itself and other CHs closer to the gateway. The distance of the transmission path determines the energy consumption of the node. For instance, the CH requires more energy for data transmission when the gateway is situated far from it. Thus, the sudden drop in a CH's energy may occur due to higher energy consumption. Hence, the node with a lesser distance from the gateway is preferred during data transmission [111]. Equation (6.2) demonstrates the objective function of distance between the gateway and the CH:

$$f_{DCH} = Minmum\ (Tentative\ CH_i, G_w) \qquad (6.2)$$

Where, the term $dis(Tentitive\ CH_i,\ G_w)$ represents the distance between gateway (G_w) and tentative CH_i in each cluster.

6.4.4 Communication Stage

Following the establishment of unequal clustering in the network area, two phases forward data between the gateway and sensor nodes. In the first phase, intra-cluster paths are enacted by direct communication as a single hop, where each member sensor node in the cluster connects to its CH to transmit its respective data packets within the first phase. In the second phase, graph-routing builds single-objective graph paths using the CMA-ES between CHs for transmitting data packets to the gateway. This is accomplished via multi-hop communication between them: these are inter-cluster paths.

To save the energy of sensor nodes around the gateway and achieve balanced energy consumption, a CH in any cluster can communicate directly with the gateway if it is within its communication range, without the need to forward data packets through CHs around the gateway.

6.5 Simulation Experiments

Sections 6.5 and 6.6 explain the setup of the simulation and present its performance evaluation. Simulations were conducted using MATLAB. Fifty and 100 sensor nodes were employed in order to verify the performance of the POE2E, POEng, and PODis, under varying node densities and random deployment. Since each run of the simulation presented a different deployment, with respect to the spatial distribution of the sensor nodes, their performance metrics generated different values. Several simulations were therefore conducted to verify whether the POE2E, POEng, and PODis, produced similar performance levels over 15 random deployments, and to obtain statistical means for the results. In this research, the performance of single-objective graph paths of the graph-routing algorithms was measured by the PDR, TCE, average EIF, and E2ET, respectively.

6.6 Evaluation of Results and Analysis

6.6.1 Data Packet Delivery Evaluation

As before, an essential criteria is used to evaluate the data packet delivery of the network: the PDR. This is illustrated in Figure 6.3. The rate at which data packets are delivered to their destination at the gateway reflects the efficacy of network communication between sensor nodes in the network area and the gateway.

Figure 6.3 demonstrates that the single-objective graph paths of the graph-routing algorithm, POE2E, POEng, and PODis, achieved high PDR results with a pre-set unequal clustering topology as with a mesh topology. This indicates that more data packets reached the gateway.

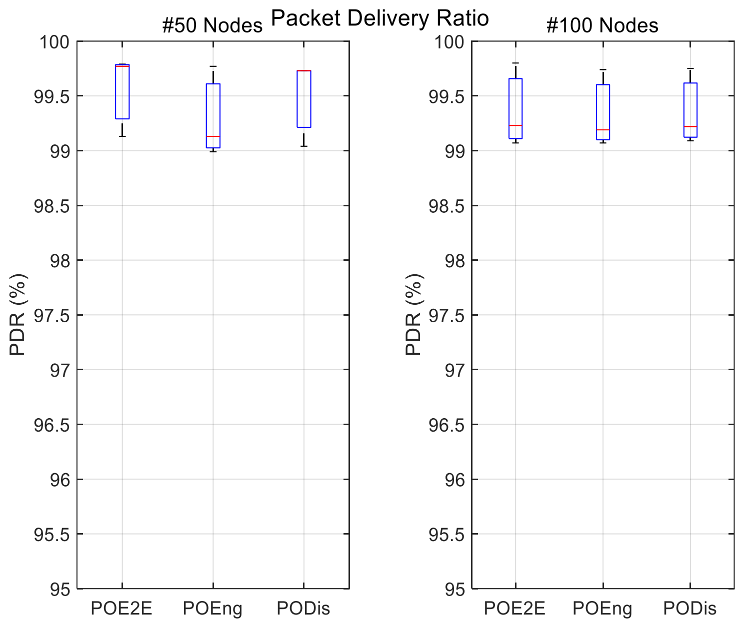


Figure 6.3 PDR results of single-objective paths with pre-set unequal clustering.

The proposed method facilitated a decrease in the ratio of missing data packets and an improvement in the data packet delivery of the entire network. This was expected because the method mitigates node isolation, ensuring all nodes can communicate with CHs to send their data packets to the gateway. It also applies path redundancy to provide support in the event of a failure during the transmission process. Furthermore, the method ensures the most effective CH in each cluster is selected. To guarantee that all sensor nodes in the same cluster can communicate with the CH, the node in the centre of the cluster is selected. In addition, the minimum distance from the CH to the gateway is calculated to ensure data packets sent by the CH can reach their destination. These factors explain the observed improvement in the data packet delivery of the network's communication.

6.6.2 Energy Consumption Evaluation

The performance of the proposed POE2E, PODis, and POEng with the IWDDR topology were evaluated with respect to energy consumption in terms of both the TCE and the average EIF of the energy balance.

The results revealed that the selection of the single-objective graph paths by the graph-routing algorithm using IWDDR topology optimises the energy consumption of the whole network (See Figure 6.4). This is in comparison to the total energy consumption results exhibited by the mesh topology, where more energy was consumed as a result of the increased number of hops along the path (See Chapter 5, Section 5.6.2, Figure 5.4).

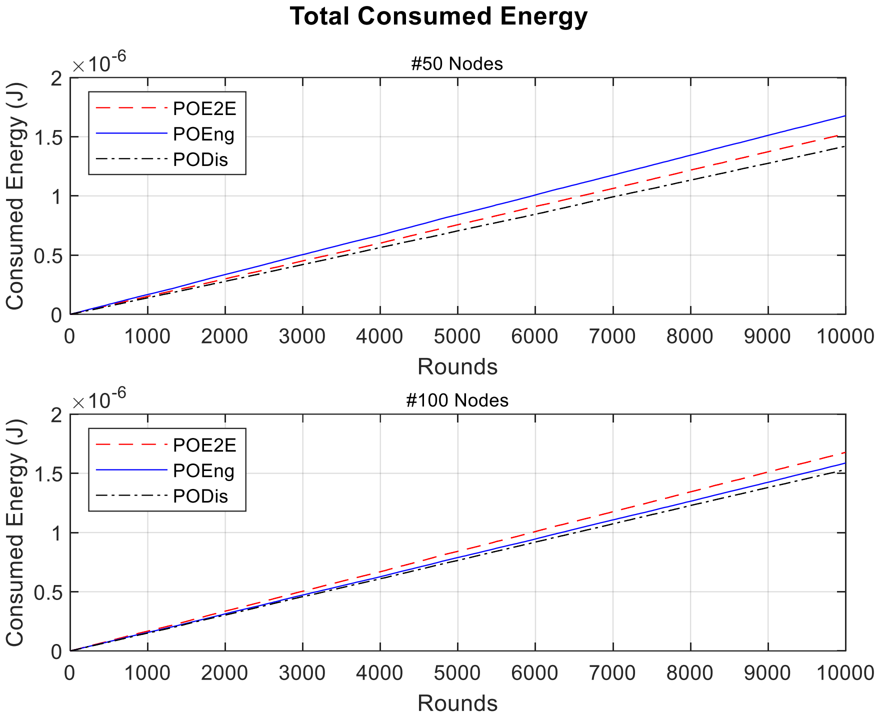


Figure 6.4 TCE results of single-objective paths with pre-set unequal clustering.

The proposed method promotes energy conservation by reducing the distance the data packet travels. This is achieved by selecting the CH closest to the gateway and applying node centrality. The method not only reduces total energy consumption, but also prevents imbalanced consumption, which can generate a hotspot problem, as illustrated in Figure 6.5.

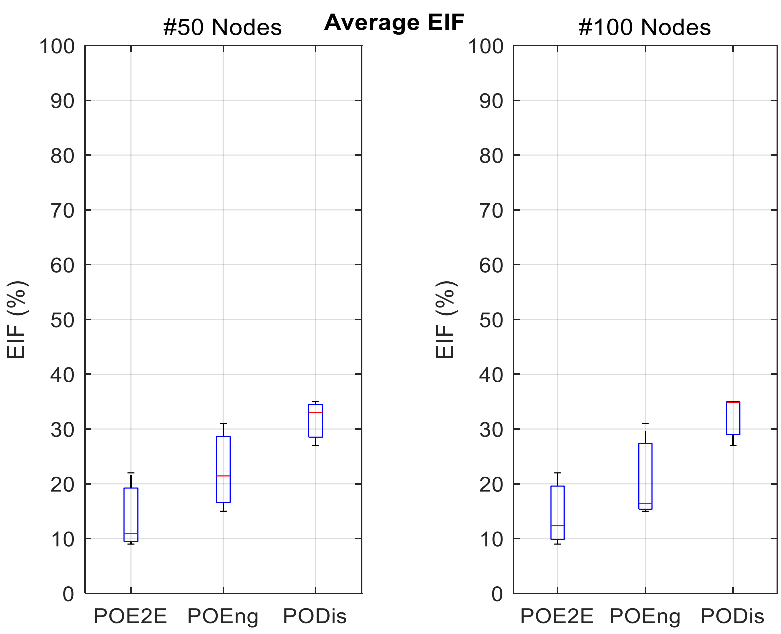


Figure 6.5 EIF results of single-objective paths with pre-set unequal clustering.

When compared with mesh topology, the single-objective graph paths of the graph-routing algorithm with the IWDDR achieved an effective balance of energy consumption. The standard deviation ratios of the residual energy for all the sensor nodes in the POE2E, POEng, and PODis graph paths in the network had a lower EIF than that achieved in the mesh topology (See Chapter 5, Section 5.6.2, Figure 5.5). This showed a decrease of 18%, 20%, and 29%, respectively, in a 50-node density network and by 19%, 18%, and 33%, respectively, in a 100-node density network. It is evident, therefore, that pre-set unequal clustering optimises the single-objective paths of the graph-routing algorithm to achieve a balance in energy consumption between the sensor nodes in the network area. This fulfils the objective of this research.

The proposed method utilises three techniques to attain the required balance in energy consumption. First, the pre-set unequal clustering topology facilitates the single-objective graph paths of the graph-routing algorithm to conserve the energy of the sensor nodes located near the gateway. This is achieved by reducing overheads on these sensor nodes. The second technique enables CHs which are not in the cluster adjacent to the gateway to communicate with the gateway, if it is in their communication range. The resulting efficient load balancing supports balanced energy consumption. Third, isolated nodes in the static clusters are addressed by the creation of new clusters. This promotes energy balance in the network area, as isolated nodes can have two effects on unbalanced energy consumption in the network. If the gateway is within their communication range, the nodes send their data packets directly to the gateway, requiring them to consume their energy faster than other sensor nodes that already belong in clusters. Alternatively, if the gateway is outside their communication range, the isolated nodes cannot communicate with any CHs. Their inability to send their data packets reduces the data packet delivery of network communication. In addition, the nodes keep their energy on while running the network.

6.6.3 End-to-End Transmission Time Evaluation

An additional experiment was conducted to assess the performance of the proposed approach in terms of the E2ET time; this is the time taken for a data packet to be sent from the source sensor node to the gateway.

Figures 6.6 and 6.7 show the variation in the E2ET time of the three single-objective graph paths of the graph-routing algorithm for a 10,000-round run of each algorithm under 50 and 100 sensor nodes, respectively. Compared to mesh topology (See Chapter 5, Section 5.6.3, Figures 5.8 and 5.9), the CDF of E2ET time for the POE2E and the PODis in the pre-set unequal clustering topology were between 3 and 13 ms in each round. In general, this is slightly higher than the E2ET time obtained from a mesh topology, where the E2ET results were between 4 and 12 ms for the POE2E and PODis. This is due to the lower number of sensor nodes in the cluster surrounding the gateway, causing congestion in the queue of data packets.

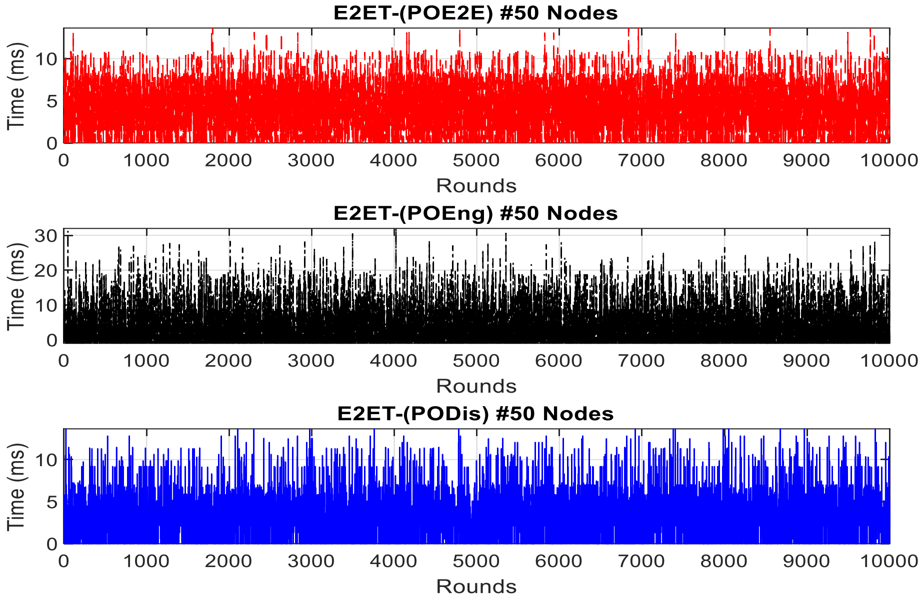


Figure 6.6 E2ET results for 50-node of single-objective paths with pre-set unequal clustering.

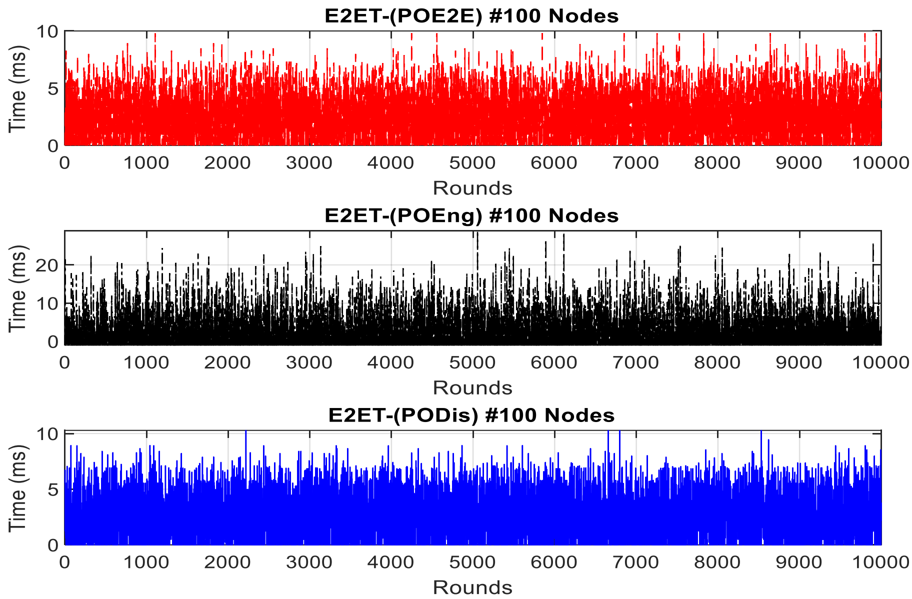


Figure 6.7 E2ET results for 100-node of single-objective paths with pre-set unequal clustering.

It is also worth noting that the POEng gave lower E2ET results in the pre-set unequal clustering than the mesh topology. This was a result of the reduced number of hops required in this topology. However, the POEng still exhibited higher transmission times than other single-objective graph paths from the graph-routing algorithm. This is logical due to the selection process applied in the proposed method. The source node focuses on communicating with the gateway via the CH with the shortest distance to the member nodes in its own cluster or with other more appropriate CHs. This reduces the number of hops as only a single hop is required when communicating intra-cluster. In addition, the CHs can communicate directly with the gateway if it is in their communication range. Finally, each CH can communicate with other CHs in the direction of the gateway.

6.7 Performance Comparison

This section is divided into two sub-sections. Section 6.7.1 compares the results of single-objective graph paths with mesh and pre-set unequal clustering to determine if changing the topology has an impact on the results. Section 6.7.2 compares the results of single-objective graph paths with pre-set unequal clustering and the results of a multiple-objective graph path that is BGR-ES with mesh topology. The aim of this comparison is to identify the best graph paths of the graph-routing algorithm according to the IWSN's requirements in WirelessHART networks.

6.7.1 Performance Comparison of Single-Objective Graph Paths with Different Topologies

This section compares the effect of the mesh topology and the IWDDR topology on the performance of the three single-objective graph paths, POE2E, POEng, and PODis, of the graph-routing algorithm. Their effectiveness was measured in terms of data packet delivery, balance of energy consumption, and transmission time. The results are illustrated in Table 6.1.

Table 6.1 Performance comparison of single-objective graph paths with different topologies.

Graph Paths	**POE2E**		**POEng**		**PODis**	
Topology	**Mesh**	**IWDDR**	**Mesh**	**IWDDR**	**Mesh**	**IWDDR**
Packet Delivery Ratio	98%	99%	98%	99%	98%	99%
Balance energy Ratio	69%	88%	61%	81%	34%	66%
Transmission Time	Between 4 to 11 ms	Between 4 to 13 ms	Between 12 to 72 ms	Between 3 to 31 ms	Between 4 to 12 ms	Between 3 to 13 ms

The performance analysis revealed the following key findings:

- The POE2E graph path using unequal clusters enhanced the balancing of energy consumption between sensors in the network area by 19% compared with mesh topology. This graph path was also judged to be the most effective in terms of energy balance when compared to other single-objective graph paths. It slightly outperformed POEng in both topologies, and outperformed PODis with mesh topology and the IWDDR topology by approximately 34% and 22%. Furthermore, the POE2E graph path using the IWDDR topology achieved convergent results for mesh topology in terms of data packet delivery and E2ET time. It therefore ensured communication data packet delivery while reducing delay in both topologies.

- It is interesting to note that the POEng graph path using the IWDDR topology improved the energy balance of the network by approximately 19% compared to the mesh topology. Moreover, the E2ET time was significantly reduced. This was a result of the decrease in hops engendered by focusing on graph paths between CHs. In comparison with the PODis, the POEng with mesh topology and the IWDDR topology achieved a superior balance of energy consumption, showing an increase of approximately 27% and 15%, respectively. Because POEng is dependent on the maximum residual energy of the sensor node in the next hop along the graph path, while PODis chooses the next node to which the data packet is sent – selected according to its distance along the path, with the minimum distance being the deciding factor. Consequently, the same sensor nodes may be selected multiple times, particularly those located closest to the gateway. It is for this reason that the PODis is the graph-routing algorithm’s single-objective graph path with the least balanced energy consumption.
- The PODis graph path with the IWDDR topology, comparable to the mesh topology, ensures data packet delivery and reduces transmission time in the network. Even though it was the least effective in terms of balancing energy consumption between sensor nodes in the network area, it improved energy balance by approximately 31% compared to mesh topology. This is because, with mesh topology, any sensor node around the gateway could be selected as the next hop in the PODis graph path, resulting in all sensor nodes around the gateway contending with increased energy consumption compared to other sensor nodes in the network. However, with the IWDDR topology, it is just the CHs around the gateway that are overburdened, while other SMs within the clusters save their energy, resulting in an increased energy balance in this topology.

6.7.2 Performance Comparison of Single-Objective Graph Paths with IWDDR to the BPGR-ES with Mesh Topology

After the three single-objective graph paths (POE2E, POEng, and PODis) proved effective with the IWDDR topology in terms of balanced energy consumption, as evidenced in subsection 6.7.1, the present section aims to compare their performances with the IWDDR topology to the performance of the multiple-objective graph path BPGR-ES, which utilises a mesh topology. As illustrated in Table 6.2, the comparison focused on the achievement of the IWSN requirements of data packet delivery, balance of energy consumption, and transmission time.

Table 6.2 Performance comparison of single-objective graph paths with IWDDR to the BPGR-ES with mesh topology.

Graph Paths	**BPGR-ES**	**POE2E**	**POEng**	**PODis**
Topology	**Mesh**	**IWDDR**	**IWDDR**	**IWDDR**
Packet Delivery Ratio	99%	99%	99%	99%
Balance energy Ratio	85%	88%	81%	66%
Transmission Time	Between 4 to 21 ms	Between 4 to 13 ms	Between 3 to 31 ms	Between 3 to 13 ms

It is apparent from Table 6.2 that the pre-set unequal clustering topology has effectively enhanced the performance of the single-objective graph paths POE2E and POEng. An explanation for this is that the results of the single graph paths in this type of topology, and the multiple-objective graph path of BPGR-ES in mesh topology, converged in terms of data packet delivery and energy consumption balance between the sensor nodes in the network area. However, in terms of transmission time, the combination of POE2E with a pre-set unequal clustering topology outperformed the BPGR-ES, although the performance of the BPGR-ES remained within the required transmission time range for industrial automation [23]. The single-objective graph path POEng still experienced high E2ET time compared with the single-objective graph path POE2E and the multiple-objective graph path BPGR-ES. This is logical due to the next-hop selection process of these graph-routing paths. The next sensor node in a POEng graph path is selected based on its amount of residual energy, and in a PODis graph path, it is selected based on the shortest distance. This creates a queue of sensor nodes with these characteristics, causing congestion and increasing transmission time.

The purpose of this thesis is to identify the best WirelessHART graph-routing strategy to achieve a balance between the IWSN requirements that this thesis focused on. The data in Table 6.2 enables the graph paths of the graph-routing algorithm to be ranked according to their performance. The highest-performance graph path, successfully balancing all IWSN requirements, was realised by POE2E combined with pre-set unequal clustering. This was closely followed by the BPGR-ES graph path. This attained similar results to POE2E save for a slightly longer transmission time, although this still remained within the strict transmission time requirements of IWSN applications. These results indicate, therefore, that in IWSN applications that are time-bound and critically sensitive to delay, such as factory automation, and automotive and aerospace applications [23], the POE2E graph path combined with pre-set unequal clustering is superior to the BPGR-ES multi-objective graph path. The third graph path to successfully balance the IWSN requirements was POEng combined with pre-set unequal clustering. This achieved a better energy consumption balance than PODis, which is considered the least likely graph path to achieve a balance between IWSN requirements.

6.8 Summary

This chapter examined how a pre-set unequal clustering topology, Improved WirelessHART Density Controlled Divide-and-Rule (IWDDR), affects the performance of single-objective paths of graph-routing (POE2E, POEng, and PODis). Particularly with regard to the balance of energy consumption. In addition, the WDDR topology was improved by reducing the isolated node problem in the WDDR topology, and then using CMA-ES to select the best CH for each cluster.

Using the above optimisation method, it was discovered that single-objective graph paths for graph-routing with the IWDDR topology outperformed the mesh topology of these paths in PDR, and therefore significantly improving the network's performance and data transmission efficiency. Even though E2ET performed better in the POE2E and PODis mesh topologies, it was best in the POEng topology with pre-set unequal clustering than in a mesh topology. Furthermore, the TCE decreased, which was achieved by reducing the length of data packet transmission and applying pre-set unequal clustering in order to enhance the balance of energy consumption within the cluster and between clusters.

Chapter 7 draws together the overall conclusions of the thesis and suggests future research directions.

Chapter 7

Conclusion and Future Research Works

7.1 Overview

This concluding chapter will summarise the key research findings in relation to the research aims and provide recommendations for future work. Section 7.2 presents the conclusion of this thesis' contributions, and in Section 7.3, the thesis statement is revisited. Section 7.4 presents several directions for future work derived from the limitations and possible extensions of this research.

7.2 Research Contributions

This thesis has focused on enhancing the performance of existing IWSN standards in terms of balancing energy consumption, increasing data packet delivery, and reducing End-to-End Transmission (E2ET) time. The research has centred on the premise that this can be achieved through improving the network topology and addressing the challenges of establishing effective graph paths in the graph-routing algorithm, which is the main routing method in existing standards. This research has applied the characteristics of WirelessHART, the global standard for IWSNs, to setup a simulation environment for the network. Furthermore, two techniques have been employed to enhance the performance of WirelessHART networks under centralised management.

The main contributions of this thesis are summarised in the following points:

- **First,** a class of clustering techniques known as unequal clustering has been implemented to examine the effect on the performance of basic graph-routing algorithm when changing the WirelessHART network topology (Section 4.5). The aim of this technique was to address the hotspot problem, and thus enhance energy consumption balance between the sensor nodes in the network area. The pre-set, unequal clustering that has been proposed as a solution is known as

WirelessHART Density-Controlled Divide-and-Rule (WDDR). The construction of static clustering by the WDDR topology is determined by the location of the gateway and the size of the network area. Once the wireless sensor nodes are deployed, a Cluster Head (CH) is selected within each cluster. This selection is determined by the maximum remaining energy of the sensor nodes in that cluster. In the WDDR topology, the basic graph-routing algorithm of WirelessHART aims to reduce the number of hops on the communication paths being built between the CHs. This increases the data packet delivery of the network and reduces the overheads on the sensor nodes around the gateway, thus balancing energy consumption.

Three types of network topology have been tested by the graph-routing algorithm of the WirelessHART network in the simulation environment, and their performance has been compared and evaluated (Section 4.7). The topologies comprised mesh, which builds graphs between all sensor nodes; WDDR, which constructs graphs between CHs; and EEUC [131], which also builds graphs between CHs. Despite an isolated nodes phenomenon being observed in the WDDR topology, the findings indicate that unequal clustering techniques deliver superior results for graph-routing compared to the mesh topology, with regard to total energy consumption, balancing energy consumption between the sensor nodes in the network area, and the data packet delivery of the network. This promotes an increase in the lifetime of the WirelessHART network.

- **Second,** an optimisation technique using a metaheuristic algorithm known as the Covariance-Matrix Adaptation Evolution Strategy (CMA-ES) has been implemented. This enables the graph-routing algorithm to construct graph paths based on the IWSN's requirements. Using this technique, four contrasting approaches were undertaken to build the graph paths of the graph-routing algorithm. Three of these approaches are based on single-objective functions, and one is based on multiple objectives (Section 5.3). The single-objective graph-routing approaches have the following acronyms: POE2E, POEng, and PODis. The multiple objective approach is known as BPGR-ES.

  These approaches were tested in the WirelessHART network with mesh topology. In this topology, each sensor node has multiple neighbour nodes in the direction of the gateway. This enables path redundancy to be exploited, leading in turn to an increase in data packet delivery. **However, each proposed graph-routing approach has a different objective:**
  - POE2E aims to build a graph path with the lowest transmission time, thereby reducing total energy consumption and E2ET time.
  - POEng aims to build a graph path with the highest residual energy for each sensor node along the path, thereby increasing data packet delivery.
  - PODis aims to build a graph path with the shortest possible distance, reducing total energy consumption and E2ET time.

- BPGR-ES aims to build a graph path that achieves a balance between the three graph paths above and a balance of energy consumption between sensor nodes in the network area. This balance energy consumption, reduce total energy consumption and E2ET time, and increase data packet delivery.

To evaluate and compare the performance of the graph-routing algorithms, as seen in Section 5.6, the simulation environment implemented graph paths for POE2E, POEng, PODis, and BPGR-ES of the graph-routing algorithm. It also implemented ELHFR [15] and EBREC [20], two relevant, state-of-the-art graph-routing algorithms. Considering the given scenarios and simulations performed, all proposed graph paths presented high data packet delivery in the uplink graphs due to ensuring path redundancy. Compared to the two state-of-the-art graph-routing algorithms, BPGR-ES achieved a balance in energy consumption between the sensor nodes in the network area. In contrast, the single-objective graph paths and the ELHFR produced a decrease in the ratio of balanced energy consumption. The graph paths POE2E, POEng, and PODis, therefore, exhibited an imbalance in energy consumption when operated in the WirelessHART network with the mesh topology. However, the BPGR-ES, POE2E, and PODis succeeded in reducing the E2ET time more than the other graph-routing algorithms. The POEng, on the other hand, increased the E2ET time significantly due to the increased number of hops along this graph path.

- **Third,** the three single-objective graph path approaches were applied using a pre-set unequal clustering topology to examine how effectively energy consumption could be balanced in this topology. These clusters were created using a pre-set, unequal clustering algorithm known as IWDDR (Section 6.4). This is based on the WDDR topology but mitigates the issue of isolated nodes, thus ensuring all sensor nodes in the network area can communicate with the gateway or other sensor nodes. In addition, the CMA-ES algorithm is exploited to select the best CH in each static cluster. The graph paths POE2E, POEng, and PODis were implemented using IWDDR topology in the simulation environment with different network densities to evaluate their performance (Section 6.6). In comparison to the mesh topology, pre-set unequal clustering enabled the POE2E, POEng, and PODis to achieve a balance in energy consumption between the sensor nodes in the network area, and a slightly increased data packet delivery. Furthermore, POE2E and PODis maintained comparable E2ET times to the mesh topology, while a clear decrease in E2ET time was achieved by the graph path of the POEng.

7.3 Thesis Statement Revisited

In this section, the thesis statement is repeated from Section 1.3, while the remainder of this section provides recommendations to enhance the performance of the graph-routing algorithm of the WirelessHART network. The thesis statement is restated as follows:

This thesis asserts that by changing the topology of the WirelessHART network based on the cluster technique, using optimisation techniques to construct more efficient graph-routing paths, and exploiting path redundancy, the performance of graph-routing can be significantly improved in terms of data packet delivery, balanced energy consumption, and transmission time. Furthermore, it can achieve a balance between these requirements.

Following an analysis of the proposed paths applied by graph-routing in the WirelessHART network, using either a mesh or pre-set unequal clustering topology, this thesis makes the following recommendations for designing an effective IWSN in real industrial environments: The most effective graph path of the graph-routing algorithm, in terms of both performance and design, is the BPGR-ES multiple-objective graph path with the basic mesh topology of the WirelessHART network. This recommendation is based on two conclusions. First, its ability to achieve an effective balance between the IWSN's requirements in terms of data packet delivery, energy consumption balance, and E2ET time. Second, it utilises minimal computation processes at both the sensor nodes and the network area, resulting in improved resource utilisation and reduced energy consumption.

However, in oil and gas smart factories, WirelessHART networks may be used to monitor and control various processes, including pipeline and wellhead monitoring. For example, in pipeline monitoring, wireless sensor nodes may be used to measure the pressure, temperature, or flow rate of fluid inside the pipeline. If there is a leak, sensor nodes can quickly detect it and transmit data to the NM, which can then take appropriate action to prevent further damage [5], [158]. Therefore, in such applications, even a small delay in the transmission of data could have serious consequences, such as increased environmental damage, loss of product, and increased repair costs. Therefore, a POE2E graph path with pre-set unequal clustering may outperform the BPGR-ES graph path, particularly in such circumstances, due to its slightly lower transmission time. It is worth noting, however, that the POE2E graph path may involve higher computation requirements at the sensor nodes, owing to the additional tasks of electing CHs and changing the network area topology of the WirelessHART network. This increased computation may also result in a faster depletion of energy resources. Therefore, designers of the IWSN should carefully consider these trade-offs and choose the most appropriate graph path and topology for their specific application.

Additionally, this thesis recommends avoiding using the shortest distance graph path PODis of the graph-routing algorithm in centralised WirelessHART networks. Despite its effectiveness in reducing the total energy consumption of the network and ensuring data packet delivery, this path fails to achieve energy consumption balance in both mesh and unequal clustering topologies. The primary drawback lies in PODis's focus on the sensor nodes around the gateway, resulting in higher energy consumption for these nodes compared to other sensor nodes in the network area. This imbalance of energy consumption in the centralised network between the sensor nodes in the network area is a significant concern due to the hotspot problem that affects the overall performance efficiency of the network.

In conclusion, the research suggests that the use of clustering and optimisation techniques can improve graph-routing algorithm performance of WirelessHART networks, especially in centralised management networks, the IIoT, and in Industry 4.0 protocols. These techniques promote more effective resource utilisation, improved network efficiency and scalability, and more robust and flexible routing solutions. However, the effective implementation of these techniques requires a thorough evaluation and simulation process to be undertaken for each standard and application. This is because the requirements and constraints of each standard and application may differ, and the use of clustering and optimisation techniques may not always result in improved performance. In addition, it is important to consider such factors as the computational complexity, the cost of implementation, and the scalability of these techniques when evaluating their feasibility for a particular standard or application.

7.4 Recommendations for Future Work

7.4.1 POE2E, POEng, PODis, and BPGR-ES Evaluation

The research in this thesis focused on the characteristics of WirelessHART and compared graph-routing performance in two topologies: mesh and cluster. Further research should be undertaken to evaluate the performance of the four different paths of the graph-routing algorithm with other IWSN standards, such as ISA 100.11a. This evaluation may involve other topologies, such as star, tree, ring or cluster, as well as various scenarios. Furthermore, it is feasible to conduct experiments in real-world industrial applications, thus providing valuable insights into the graph-routing algorithm's performance.

7.4.2 IWSN Standards

An important area for future research, and one that would be extremely valuable to researchers in a IWSN field, would be the creation of an IWSN standards simulator that implements the full OSI reference model. This could be used as a reference tool to evaluate and validate proposed algorithms in the domain of industrial monitoring and control. Although [38] implemented the WirelessHART network in a NS-2 simulator, it is currently experiencing technical difficulties due to compatibility issues with recent Linux and NS-2 releases. The implementation of a full OSI model simulator for IWSN standards would give researchers an alternative to expensive testbeds, thereby enabling the performance of IWSN algorithms to be comprehensively evaluated.

7.4.3 Future Directions

With the goal of enhancing the efficacy of the routing algorithm in IWSNs, this sub-section discusses a number of challenges and unresolved issues that may provide three directions for future research.

- **Real-time and critical data**: The handling of real-time and critical data in IWSNs presents a significant challenge that could be addressed by further research [159]. The aim would be to create routing algorithms that can provide the delivery of real-time and critical data with minimal latency and maximum data packet delivery. A variety of techniques could be explored to achieve this, including prioritising real-time and critical data traffic, implementing fault-tolerant routing strategies [160], and enhancing network scalability to accommodate high traffic demands. Efficient handling of real-time and critical data is crucial for critical industrial automation applications, such as oil refineries, where delays or inaccuracies in data collection can impact process control, product quality, and safety [161]. Further research in this area could contribute to reliable and efficient operation of IWSNs in challenging environments.
- **Security issues**: Security is a major concern in IWSNs as sensor nodes may be vulnerable to a range of security threats [3]. For example, eavesdropping is where an attacker intercepts and listens to the communication between sensor nodes [162]. Tampering is where an attacker modifies or corrupts the data packets being transmitted [163]. Whereas Denial of Service (DoS) is where the normal functioning of the network is disrupted by being overwhelmed with traffic [164]. In a Sybil attack, the attacker creates multiple fake identities to gain control over the network [165]. A Man-In-The-Middle (MITM) attack is where the communication between two sensor nodes is intercepted and manipulated [166]. Attacks such as these pose a significant threat to the functionality and security of IWSNs. Future research could, therefore, focus on developing secure graph-routing algorithms that can guarantee the confidentiality and integrity of transmitted data packets in the network. This could be achieved by integrating cryptographic techniques such as encryption and digital signatures, and by implementing secure routing protocols that can detect and defend against distinct types of attacks, such as eavesdropping, tampering, and malicious nodes.

- **Quality of Service (QoS) requirements:** IWSNs often have strict requirements in terms of QoS, including delay, data packet delivery, and energy efficiency. For example, in a smart grid system, IWSNs are used for monitoring and control of various components, such as power generation sources, distribution substations, transformers, and power consumption points [11]. In the context of a smart grid system, IWSNs need to meet strict QoS requirements in terms of low delay, high data packet delivery, and energy efficiency to ensure reliable and efficient operation of the grid. Failure to meet these QoS requirements can lead to operational inefficiencies, incorrect decisions, and potential risks to the stability and data packet delivery of the power grid. Meeting these requirements can be a challenging task, particularly when trade-offs between different objectives need to be considered. Future work could focus on developing graph-routing algorithms in IWSNs that can address these trade-offs while simultaneously affording multiple QoS objectives. One approach would be to explore alternative optimisation techniques, such as GA [77], PSO [69], or ACO [71], to balance conflicting objectives and provide a solution that meets the desired QoS requirements. These optimisation techniques would need to maintain overall performance by exploring a large solution space and by finding the most effective solution contingent on a set of predefined criteria. An alternative approach could be to develop optimisation algorithms that can adjust routing paths according to changing network conditions and QoS requirements. This would enable IWSNs to provide flexible and adaptable QoS provisioning, while simultaneously maintaining performance and stability.

Appendix A

A.1 Test Optimisation Techniques

To improve the performance of the graph-routing algorithm used in IWSNs, this research sought to identify and implement the most effective optimisation technique. This was achieved by testing the performance of 18 optimisation algorithms against eight common objective functions and datasets. The optimisation objective was to improve accuracy and minimise transmission time. The test, therefore, focused on two criteria: the results of the objective functions (cost functions) and the number of iterations required to reach the final results. In this test, all cost functions with a fixed final result of zero were selected; this enabled the number of iterations required by the optimisation algorithm to be ascertained. The tests conducted in the MATLAB simulator depended on the two datasets outlined below:

- Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. From http://www.sfu.ca/~ssurjano.
- Mostapha Kalami Heris, (URL: https://yarpiz.com), Yarpiz, 2015.

In addition, as they are widely used for testing optimisation algorithms, the following functions were selected [127]:

1. Sphere Function:

$$f(x) = \sum_{i=1}^{d} x_i^2,$$

$$\textit{The function is usually evaluated on } x_i \in [-5.12, 5.12]$$

2. Ackley Function:

$$f(x) = -a \exp\left(-b\sqrt{1\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1),$$

$$\textit{The function is usually evaluated on } x_i \in [-32.768, 32.768]$$

3. Griewank Function:

$$f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-600{,}600]$$

4. Rastrigin Function:

$$f(x) = 10d + \sum_{i=1}^{d} [x_i^2 - 10\cos\ (2\pi x_i]\ ,$$

$$The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-5.12{,}5.12]$$

5. Rotated Hyper-Elliposoid Function:

$$f(x) = \sum_{i=1}^{d} \sum_{j=1}^{i} x_j^2\ , The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-65.536{,}65.536]$$

6. Sum of Different Powers Function:

$$f(x) = \sum_{i=1}^{d} |x_i|^2\ , \qquad The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-1{,}1]$$

7. Sum of Different Powers Function:

$$f(x) = \sum_{i=1}^{d} i x_i^2\ , \qquad The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-10{,}10]$$

8. Zakharov Function:

$$f(x) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + \ (\sum_{i=1}^{d} 0.5ix_i)^4,$$

$$The\ function\ is\ usually\ evaluated\ on\ \ x_i \in [-5{,}10]$$

Where all the objective functions above have a global minimum result of zero, $f(x) = 0$. The decision variables that are lower bound and upper bound in each optimisation algorithm are denoted by x_i. To ensure fair testing, all optimisation algorithms were given the same maximum permitted number of iterations, which was set to 1000. This was considered sufficient to permit a fair evaluation of the efficiency of the algorithm.

The optimisation algorithms that were tested were metaheuristic. These were [167]: Particle Swarm Optimisation (PSO), Binary and Real-Coded Genetic (BRCG), algorithm, Artificial Bee Colony (ABC), Firefly Algorithm (FA), Bees Algorithm (BeA), Invasive Weed Optimisation (IWO), Teaching-Learning-Based Optimisation (TLBO), Shuffled Frog Leaping Algorithm (SFLA), Harmony Search (HS), Differential Evolution

(DE), Shuffled Complex Evolution (SCE-UA), Bees algorithm (Probabilistic) (BeP), Imperialist Competitive Algorithm (ICA), Biogeography-Based Optimisation (BBO), Covariance Matrix Adaptation Evolution Strategy (CMA-ES), Cultural Algorithm (CA), MoathFlame Algorithm (MFA), and real-coded Simulated Annealing (SA).

A.2 Test Simulation Results for the Selected Optimisation Algorithms

Table 7.1 presents the simulation results for the 18 optimisation algorithms tested against the eight selected objective functions. If the result of the Cost Function (CF) equals 0, this indicates that the optimisation algorithm attained the global minimum result. The number of iterations required to reach the final result can, therefore, be used to define the speed of the algorithm.

As shown in Table 7.1, the PSO, BRCG, TLBO, DE, and CMA-ES algorithms, produced the most successful results. Out of all these optimisation algorithms, the CMA-ES proved more accurate: it achieved the best results with all CFs and reached the global minimum result in less than 1000 iterations. The PSO, BRCG, TLBO, and DE algorithms, on the other hand, failed to reach the final result with the Ackley function. Furthermore, the CMA-ES algorithm required fewer iterations to find the best solution than the other optimisation algorithms. For example, the CMA-ES with the Griewank function needed 51 iterations to reach the final solution, whereas the PSO, BRCG, TLBO, and DE algorithms needed 92, 145, 66, and 74 iterations, respectively. Similarly, with the Sum Squares function, the CMA-ES reached the final solution in only 466 iterations compared to the PSO, BRCG, TLBO, and DE algorithms which needed 543, 659, 519, and 585 iterations, respectively. The CMA-ES algorithm was, therefore, adopted as the best optimisation method for this research. A key reason for its selection was its ability to consistently achieve effective results with different CFs.

Table A.1 Comparison of test simulation results for optimisation algorithms

Optimisation Algorithms		Sphere	Ackley	Griewank	Rastrigin	Rotated Hyper-Ellipsoid	Sum of Different Powers	Sum Squares	Zakharov
PSO	**CF**	**0**	**8.88E-16**	**0**	**0**	**0**	**0**	**0**	**0**
	Iteration	**541**	**1000**	**92**	**74**	**543**	**547**	**543**	**533**
BRCG	**CF**	**0**	**8.8818E-16**	**0**	**0**	**0**	**0**	**0**	**0**
	Iteration	**643**	**1000**	**145**	**32**	**662**	**640**	**659**	**647**
ABC	**CF**	8.19E-27	9.67E-13	6.10E-08	0	6.25E-24	9.5782E-30	8.1627E-22	9.8857E-25
	Iteration	1000	1000	1000	148	1000	1000	1000	1000
FA	**CF**	9.0921E-26	0.000009979	0	0	8.3512E-24	8.6565E-19	9.8854E-22	8.0439E-16
	Iteration	1000	1000	382	185	1000	1000	1000	1000
BeA	**CF**	8.32E-22	9.67E-10	0	0	9.82E-14	9.94E-19	7.33E-18	8.91E-52
	Iteration	1000	1000	532	466	1000	1000	1000	1000
IWO	**CF**	5.90E-10	9.77E-06	5.2184	7.04E-11	9.51E-14	8.5991E-08	8.36E-07	9.51E-14
	Iteration	1000	1000	1000	1000	1000	1000	1000	1000
TLBO	**CF**	**0**	**8.88E-16**	**0**	**0**	**0**	**0**	**0**	**0**
	Iteration	**517**	**1000**	**66**	**42**	**518**	**514**	**519**	**515**
SFLA	**CF**	9.185E-55	0	0	0	9.55E-17	9.6159E-33	8.5164E-1	63
	Iteration	1000	213	168	110	1000	1000	1000	1000

Optimisation Algorithms		Sphere	Ackley	Griewank	Rastrigin	Rotated Hyper-Ellipsoid	Sum of Different Powers	Sum Squares	Zakharov
HS	**CF**	8.06E-13	7.34E-05	1.51E-08	8.24E-06	9.92E-09	9.05E-09	6.11E-06	9.37E-08
	Iteration	1000	1000	1000	1000	1000	1000	1000	1000
DE	**CF**	**0**	**8.88E-16**	**0**	**0**	**0**	**0**	**0**	**0**
	Iteration	**578**	**1000**	**74**	**35**	**583**	**572**	**585**	**579**
SCE-UA	**CF**	8.45E-67	8.88E-16	8.88E-16	0	8.58E-14	9.90E-63	9.65E-55	9.36E-63
	Iteration	1000	1000	1000	319	1000	1000	1000	1000
BeP	**CF**	8.23E-07	8.88E-16	7.89E-05	0	9.38E-10	8.51E-26	9.98E-34	9.16E-39
	Iteration	1000	1000	1000	168	1000	1000	1000	1000
ICA	**CF**	9.91E-11	8.88E-16	0	0	8.56E-46	8.09E-09	9.57E-94	8.79E-66
	Iteration	1000	1000	25	19	1000	1000	1000	1000
BBO	**CF**	0	8.88E-16	0	0	0	9.13E-05	9.29E-05	9.94E-68
	Iteration	396	1000	23	19	405	1000	1000	1000
CMA-ES	**CF**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	Iteration	**470**	**44**	**51**	**21**	**481**	**471**	**466**	**478**
CA	**CF**	9.58E-41	8.88E-16	0	0	9.88E-117	9.63E-100	9.96E-140	9.82E-94
	Iteration	1000	1000	556	83	1000	1000	1000	1000

Optimisation Algorithms		Sphere	Ackley	Griewank	Rastrigin	Rotated Hyper-Ellipsoid	Sum of Different Powers	Sum Squares	Zakharov
MFA	**CF**	1.3731E-35	5.5278E-36	1.5436E-34	1.1589E-34	2.9001E-35	1.0513E-36	2.1105E-36	2.0506E-34
	Iteration	1000	1000	1000	1000	1000	1000	1000	1000
SA	**CF**	9.63E-18	9.95E-07	0	0	9.90E-25	9.04E-25	9.47E-28	6.50E-28
	Iteration	1000	1000	608	271	1000	1000	1000	1000

References

[1] Xu, L. D.; He, W.; Li, S. Internet of Things in Industries: A Survey. *IEEE Transactions on Industrial Informatics* **2014**, *10* (4), 2233–2243.

[2] Nobre, M.; Silva, I.; Guedes, L. Routing and Scheduling Algorithms for WirelessHART Networks: A Survey. *Sensors* **2015**, *15* (5), 9703–9740.

[3] Raza, M.; Aslam, N.; Le-Minh, H.; Hussain, S.; Cao, Y.; Khan, N. M. A Critical Analysis of Research Potential, Challenges, and Future Directives in Industrial Wireless Sensor Networks. *IEEE Communications Surveys and Tutorials* **2018**, *20* (1), 39–95.

[4] Salam, H. A.; Khan, B. M. IWSN-Standards, Challenges and Future. *IEEE Potentials* **2016**, *35* (2), 9–16.

[5] Devan, P. A. M.; Hussin, F. A.; Ibrahim, R.; Bingi, K.; Khanday, F. A. A Survey on the Application of WirelessHART for Industrial Process Monitoring and Control. *Sensors* **2021**, *21* (15), 4951.

[6] Benomar, Z.; Campobello, G.; Segreto, A.; Battaglia, F.; Longo, F.; Merlino, G.; Puliafito, A. A Fog-Based Architecture for Latency-Sensitive Monitoring Applications in Industrial Internet of Things. *IEEE Internet of Things Journal* **2021**, *10* (3), 1908–1918.

[7] Mishra, A.; Agrawal, D. P. Evaluation of Suitability of Current Industrial Standards in Designing Control Applications for Internet of Things Healthcare Sensor Networks. *Journal of Sensor and Actuator Networks* **2019**, *8* (4), 54.

[8] Costa, C. M.; Baltus, P. Design Methodology for Industrial Internet-of-Things Wireless Systems. *IEEE Sensors Journal* **2021**, *21* (4), 5529–5542.

[9] Cabrini, F. H.; Valiante Filho, F.; Rito, P.; Barros Filho, A.; Sargento, S.; Venâncio Neto, A.; Kofuji, S. T. Enabling the Industrial Internet of Things to Cloud Continuum in a Real City Environment. *Sensors* **2021**, *21* (22), 7707.

[10] Lin, C.; Han, G.; Qi, X.; Du, J.; Xu, T.; Martínez-García, M. Energy-Optimal Data Collection for Unmanned Aerial Vehicle-Aided Industrial Wireless Sensor Network-Based Agricultural Monitoring System: A Clustering Compressed Sampling Approach. *IEEE Transactions on Industrial Informatics* **2021**, *17* (6), 4411–4420.

[11] Faheem, M.; Butt, R. A.; Raza, B.; Ashraf, M. W.; Begum, S.; Ngadi, Md. A.; Gungor, V. C. Bio-Inspired Routing Protocol for WSN-Based Smart Grid

Applications in the Context of Industry 4.0. *Transactions on Emerging Telecommunications Technologies* **2019**, *30* (8), e3503.

[12] *Industrial Wireless Sensor Network Market Share Report, 2025*; GVR-2-68038-325-6.

[13] Chen, D.; Nixon, M.; Mok, A. Why WirelessHART. In *WirelessHART^{TM}*; Springer US, 2010; pp 195–199.

[14] Winter, J. M.; Kunzel, G.; Muller, I.; Pereira, C. E.; Netto, J. C. Study of Routing Mechanisms in a WirelessHART Network. In *Proceedings of the IEEE International Conference on Industrial Technology*; 2013; pp 1540–1545.

[15] Zhao, J.; Liang, Z.; Zhao, Y. ELHFR: A Graph Routing in Industrial Wireless Mesh Network. In *2009 IEEE International Conference on Information and Automation, ICIA 2009*; Zhuhai/Macau, China, 2009; pp 106–110.

[16] Han, S.; Zhu, X.; Mok, A. K.; Chen, D.; Nixon, M. Reliable and Real-Time Communication in Industrial Wireless Mesh Networks. In *17th IEEE Real-Time and Embedded Technology and Applications Symposium*; Chicago, IL, USA, 2011.

[17] Memon, A.; Hong, S. Minimum-Hop Load-Balancing Graph Routing Algorithmfor Wireless HART. *International Journal of Information and Electronics Engineering* **2013**, *3* (2).

[18] Zhang, Q.; Li, F.; Ju, L.; Jia, Z.; Zhang, Z. Reliable and Energy Efficient Routing Algorithm for WirelessHART. In *International Conference on Algorithms and Architectures for Parallel Processing*; Springer, Cham, 2014; Vol. 8630 LNCS, pp 192–203.

[19] Wu, C.; Gunatilaka, D.; Saifullah, A.; Sha, M.; Tiwari, P. B.; Lu, C.; Chen, Y. Maximizing Network Lifetime of WirelessHART Networks under Graph Routing. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*; 2016; pp 176–186.

[20] Han, X.; Ma, X.; Chen, D. Energy-Balancing Routing Algorithm for WirelessHART. In *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*; Institute of Electrical and Electronics Engineers Inc., 2019; Vol. 2019-May.

[21] Xiao, G.; Shi, J.; Sun, N.; Chen, Y.; Zhang, Y. Adaptive Freeshape Clustering for Balanced Energy Saving in the WirelessHART Networks. *Complexity* **2019**, *2019*, e2836981.

[22] Parwekar, P.; Rodda, S.; Kalla, N. A Study of the Optimization Techniques for Wireless Sensor Networks (WSNs). In *Proceedings of Fourth International Conference INDIA*; Springer: Singapore, 2018; Vol. 672, pp 909–915.

[23] Oyewobi, S. S.; Hancke, G. P. A Survey of Cognitive Radio Handoff Schemes, Challenges and Issues for Industrial Wireless Sensor Networks (CR-IWSN). *Journal of Network and Computer Applications* **2017**, *97*, 140–156.

[24] Fanian, F.; Kuchaki Rafsanjani, M. Cluster-Based Routing Protocols in Wireless Sensor Networks: A Survey Based on Methodology. *Journal of Network and Computer Applications* **2019**, *142*, 111–142.

[25] Al Aghbari, Z.; Khedr, A. M.; Osamy, W.; Arif, I.; Agrawal, D. P. Routing in Wireless Sensor Networks Using Optimization Techniques: A Survey. *Wireless Personal Communications* **2020**, *111* (4), 2407–2434.

[26] Zhao, Y.; Li, W.; Liu, A. Improved Grey Wolf Optimization Based on the Two-Stage Search of Hybrid CMA-ES. *Soft Computing* **2020**, *24* (2), 1097–1115.

[27] Neves Valadão, Y. das; Künzel, G.; Müller, I.; Pereira, C. E. Industrial Wireless Automation: Overview and Evolution of WIA-PA. *3rd IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control CESCIT* **2018**, *51* (10), 175–180.

[28] HCF. HCF_SPEC-290: WirelessHART Device Specification https://library.fieldcommgroup.org/20290/TS20290/1.1 (accessed 2023 -02 -23).

[29] Chen, D.; Nixon, M.; Mok, A. Overview. In *WirelessHART^{TM}*; Springer US, 2010; pp 3–14.

[30] K.C, K. Wireless Mesh Network: A Survey. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*; 2016; pp 1966–1970.

[31] Chen, D.; Nixon, M.; Mok, A. Discourses on the Mesh Network. In *WirelessHART^{TM}*; Springer US, 2010; pp 107–131. https://doi.org/10.1007/978-1-4419-6047-4_9.

[32] HCF. HCF_SPEC-85 FCG TS20085: Network Management Specification https://library.fieldcommgroup.org/20085/TS20085 (accessed 2022 -03 -08).

[33] HCF. HCF_SPEC-155: Wireless Command Specification https://library.fieldcommgroup.org/20155/TS20155/2.0 (accessed 2023 -02 -23).

[34] HCF. HCF_SPEC-065: 2.4GHz DSSS O-QPSK Physical Layer specification https://library.fieldcommgroup.org/20065/TS20065/1.1 (accessed 2021 -05 -19).

[35] HCF. HCF_SPEC-075, FCG TSG TS20075: TDMA Data Link Layer Specification https://library.fieldcommgroup.org/20075/TS20075/1.2 (accessed 2023 -02 -23).

[36] Chen, D.; Nixon, M.; Mok, A. Data Link Layer. In *WirelessHART^{TM}*; Springer US, 2010; pp 19–27.

[37] Chen, D.; Nixon, M.; Mok, A. Network Layer and Transport Layer. In *WirelessHART^{TM}*; Springer US, 2010; pp 29–38.

[38] Zand, P.; Mathews, E.; Havinga, P.; Stojanovski, S.; Sisinni, E.; Ferrari, P. Implementation of WirelessHART in the NS-2 Simulator and Validation of Its Correctness. *Sensors* **2014**, *14* (5), 8633–8668.

[39] HCF. HCF_SPEC-099: Command Summary Specification https://library.fieldcommgroup.org/20099/TS20099/10.1/#page=26 (accessed 2023 -02 -23).

[40] Zhang, S.; Yan, A.; Ma, T. Energy-Balanced Routing for Maximizing Network Lifetime in WirelessHART. *International Journal of Distributed Sensor Networks* **2013**, *2013*.

[41] Hong, S. H.; Ding, Y. M.; Li, X. H.; Luo, Z.; Kim, J. B. An Energy-Balancing Graph-Routing Algorithm for WirelessHART Networks. In *2015 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*; 2016; pp 239–245.

[42] Wu, C.; Gunatilaka, D.; Sha, M.; Lu, C. Real-Time Wireless Routing for Industrial Internet of Things. In *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation*; Orlando, FL, USA, 2018.

[43] Shi, J.; Sha, M.; Yang, Z. DiGS: Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*; 2018; pp 354–364.

[44] Künzel, G.; Cainelli, G.; Müller, I.; Pereira, C. E.; Indrusiak, L. S. A Reliable and Low-Latency Graph-Routing Approach for IWSN Using Q-Routing. In *2020 IEEE X Brazilian Symposium on Computing Systems Engineering (SBESC)*; 2020; pp 1–8.

[45] Yuxin, W.; Gaofeng, N.; Hui, T. A Graph Routing Algorithm Enhancing Wireless Sensor Networks Lifetime. In *2021 International Conference on Space-Air-Ground Computing (SAGC)*; IEEE: Huizhou, China, 2021.

[46] Abdulrab, H.; Hussin, A.; Abd Aziz, A.; Awang, A.; Ismail, I.; Devan, M. Reliable Fault Tolerant-Based Multipath Routing Model for Industrial Wireless Control Systems. *Applied Sciences* **2022**, *12* (2).

[47] Heinzelman, W. R.; Chandrakasan, A.; Balakrishnan, H. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proceedings of the Hawaii International Conference on System Sciences*; IEEE, 2000; p 223.

[48] Singh, S. K.; Singh, M. P.; Singh, D. K. Energy Efficient Homogenous Clustering Algorithm for Wireless Sensor Networks. *International Journal of Wireless & Mobile Networks* **2010**, *2* (3), 49–61.

[49] Neamatollahi, P.; Naghibzadeh, M.; Abrishami, S.; Yaghmaee, M.-H. Distributed Clustering-Task Scheduling for Wireless Sensor Networks Using Dynamic Hyper Round Policy. *IEEE Transactions on Mobile Computing* **2018**, *17* (2), 334–347.

[50] Koutsonikolas, D.; Das, S. M.; Hu, Y. C.; Stojmenovic, I. Hierarchical Geographic Multicast Routing for Wireless Sensor Networks. *Wireless Networks* **2010**, *16* (2), 449–466.

[51] Choudhary, A.; Kumar, S.; Gupta, S.; Gong, M.; Mahanti, A. FEHCA: A Fault-Tolerant Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. *Energies* **2021**, *14* (13), 3935.

[52] Wang, M.; Zeng, J. Hierarchical Clustering Nodes Collaborative Scheduling in Wireless Sensor Network. *IEEE Sensors Journal* **2022**, *22* (2), 1786–1798.

[53] Jung, K.; Lee, J.-Y.; Jeong, H.-Y. Improving Adaptive Cluster Head Selection of Teen Protocol Using Fuzzy Logic for WMSN. *Multimedia Tools and Applications* **2017**, *76* (17), 18175–18190.

[54] Fanian, F.; Kuchaki, M. Memetic Fuzzy Clustering Protocol for Wireless Sensor Networks: Shuffled Frog Leaping Algorithm. *Applied Soft Computing* **2018**, *71*, 568–590.

[55] Sharma, S.; Sethi, D.; Bhattacharya, P. Artificial Neural Network Based Cluster Head Selection in Wireless Sensor Network. *International Journal of Computer Applications* **2015**, *119*, 34–41.

[56] Vijayalakshmi, K.; Anandan, P. A Multi Objective Tabu Particle Swarm Optimization for Effective Cluster Head Selection in WSN. *Cluster Comput* **2019**, *22* (5), 12275–12282.

[57] Lin, J.-W.; Chelliah, P. R.; Hsu, M.-C.; Hou, J.-X. Efficient Fault-Tolerant Routing in IoT Wireless Sensor Networks Based on Bipartite-Flow Graph Modeling. *IEEE Access* **2019**, *7*, 14022–14034.

[58] Deshpande, V. V.; Bhagat Patil, A. R. Energy Efficient Clustering in Wireless Sensor Network Using Cluster of Cluster Heads. In *2013 Tenth International Conference on Wireless and Optical Communications Networks (WOCN)*; 2013; pp 1–5.

[59] Azharuddin, M.; Kuila, P.; Jana, P. K. Energy Efficient Fault Tolerant Clustering and Routing Algorithms for Wireless Sensor Networks. *Computers & Electrical Engineering* **2015**, *41*, 177–190.

[60] Gupta, G.; Younis, M. Load-Balanced Clustering of Wireless Sensor Networks. In *IEEE International Conference on Communications, 2003. ICC '03.*; 2003; Vol. 3, pp 1848–1852 vol.3.

[61] Amine, D.; Nassreddine, B.; Bouabdellah, K. Energy Efficient and Safe Weighted Clustering Algorithm for Mobile Wireless Sensor Networks. *Procedia Computer Science* **2014**, *34*, 63–70.

[62] Bala Krishna, M.; Doja, M. N. Deterministic K-Means Secure Coverage Clustering with Periodic Authentication for Wireless Sensor Networks. *International Journal of Communication Systems* **2017**, *30* (4), e3024.

[63] Shahraki, A.; Taherkordi, A.; Haugen, Ø.; Eliassen, F. Clustering Objectives in Wireless Sensor Networks: A Survey and Research Direction Analysis. *Computer Networks* **2020**, *180*, 107376.

[64] Yang, X.-S. *Engineering Optimization: An Introduction with Metaheuristic Applications*; John Wiley & Sons, 2010.

[65] Liberti, L.; Kucherenko, S. Comparison of Deterministic and Stochastic Approaches to Global Optimization. *International Transactions in Operational Research* **2005**, *12* (3), 263–285.

[66] Houssein, E. H.; Saad, M. R.; Hussain, K.; Shaban, H.; Hassaballah, M. A Review of Metaheuristic Optimization Algorithms in Wireless Sensor Networks. In *Metaheuristics in Machine Learning: Theory and Applications*; Studies in Computational Intelligence; Springer International Publishing: Cham, 2021; pp 193–217.

[67] Glover, F. Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers & Operations Research* **1986**, *13* (5), 533–549.

[68] Ebrahimi Mood, S.; Javidi, M. M. Energy-Efficient Clustering Method for Wireless Sensor Networks Using Modified Gravitational Search Algorithm. *Evolving Systems* **2020**, *11* (4), 575–587.

[69] Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*; 1995; Vol. 4, pp 1942–1948 vol.4.

[70] Karaboga, D.; Basturk, B. On the Performance of Artificial Bee Colony (ABC) Algorithm. *Applied Soft Computing* **2008**, *8* (1), 687–697.

[71] Mirjalili, S. Ant Colony Optimisation. In *Evolutionary Algorithms and Neural Networks: Theory and Applications*; Mirjalili, S., Ed.; Studies in Computational Intelligence; Springer International Publishing: Cham, 2019; pp 33–42.

[72] Mirjalili, S.; Mirjalili, S. M.; Lewis, A. Grey Wolf Optimizer. *Advances in Engineering Software* **2014**, *69*, 46–61.

[73] Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Advances in Engineering Software* **2016**, *95*, 51–67.

[74] Yang, X.-S.; Deb, S. Cuckoo Search: Recent Advances and Applications. *Neural Computing & Applications* **2014**, *24* (1), 169–174.

[75] Yang, X.-S.; Slowik, A. Firefly Algorithm. In *Swarm Intelligence Algorithms*; CRC Press, 2020.

[76] Greenwood, G. W. Finding Solutions to NP Problems: Philosophical Differences between Quantum and Evolutionary Search Algorithms. In *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*; 2001; Vol. 2, pp 815–822 vol. 2.

[77] Reeves, C.; Rowe, J. E. *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*; Springer Science & Business Media, 2002.

[78] Rechenberg, I. Evolutions strategien. In *Simulations methoden in der Medizin und Biologie*; Springer: Berlin, Heidelberg, 1978; pp 83–114.

[79] Schwefel, H. *Evolution and Optimum Seeking: The Sixth Generation*; 1993.

[80] Hansen, N.; Ostermeier, A. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary computation* **2001**, 159–195.

[81] Akbarzadeh, V.; Ko, A. H.-R.; Gagné, C.; Parizeau, M. Topography-Aware Sensor Deployment Optimization with CMA-ES. In *Parallel Problem Solving from Nature, PPSN XI*; Springer: Berlin, Heidelberg, 2010; pp 141–150.

[82] Loshchilov, I.; Hutter, F. CMA-ES for Hyperparameter Optimization of Deep Neural Networks. arXiv April 25, 2016.

[83] Hu, G.-Y.; Zhou, Z.-J.; Zhang, B.-C.; Yin, X.-J.; Gao, Z.; Zhou, Z.-G. A Method for Predicting the Network Security Situation Based on Hidden BRB Model and Revised CMA-ES Algorithm. *Applied Soft Computing* **2016**, *48*, 404–418.

[84] Rückert, E.; Neumann, G.; Toussaint, M.; Maass, W. Learned Graphical Models for Probabilistic Planning Provide a New Class of Movement Primitives. *Frontiers in Computational Neuroscience* **2013**, *6*.

[85] Gupta, S. K.; Jana, P. K. Energy Efficient Clustering and Routing Algorithms for Wireless Sensor Networks: GA Based Approach. *Wireless Personal Communications* **2015**, *83* (3), 2403–2423.

[86] Yao, G.; Dong, Z.; Wen, W.; Ren, Q. A Routing Optimization Strategy for Wireless Sensor Networks Based on Improved Genetic Algorithm. *Journal of Applied Science and Engineering* **2016**, *19* (2), 221–228.

[87] Wang, T.; Zhang, G.; Yang, X.; Vajdi, A. Genetic Algorithm for Energy-Efficient Clustering and Routing in Wireless Sensor Networks. *Journal of Systems and Software* **2018**, *146*, 196–214.

[88] Baroudi, U.; Bin-Yahya, M.; Alshammari, M.; Yaqoub, U. Ticket-Based QoS Routing Optimization Using Genetic Algorithm for WSN Applications in Smart Grid. *Journal of Ambient Intelligence and Humanized Computing* **2019**, *10* (4), 1325–1338.

[89] Zhao, X.; Zhong, W.; Navaei, Y. D. A Novel Energy-Aware Routing in Wireless Sensor Network Using Clustering Based on Combination of Multiobjective Genetic and Cuckoo Search Algorithm. *Wireless Communications and Mobile Computing* **2022**, *2022*, e6939868.

[90] Patra, B. K.; Mishra, S.; Patra, S. K. Genetic Algorithm-Based Energy-Efficient Clustering with Adaptive Grey Wolf Optimization-Based Multipath Routing in Wireless Sensor Network to Increase Network Life Time. In *Intelligent Systems*; Springer Nature: Singapore, 2022; pp 499–512.

[91] Manshahia, M. A Firefly Based Energy Efficient Routing in Wireless Sensor Networks. *IEEE African Journal of Computing and ICT* **2015**, *8*.

[92] Mohajerani, A.; Gharavian, D. An Ant Colony Optimization Based Routing Algorithm for Extending Network Lifetime in Wireless Sensor Networks. *Wireless Networks* **2016**, *22* (8), 2637–2647.

[93] Tong, M.; Chen, Y.; Chen, F.; Wu, X.; Shou, G. An Energy-Efficient Multipath Routing Algorithm Based on Ant Colony Optimization for Wireless Sensor Networks. *International Journal of Distributed Sensor Networks* **2015**, *11* (6), 642189.

[94] Sun, Y.; Dong, W.; Chen, Y. An Improved Routing Algorithm Based on Ant Colony Optimization in Wireless Sensor Networks. *IEEE Communications Letters* **2017**, *21* (6), 1317–1320.

[95] Enxing, Z.; Ranran, L. Routing Technology in Wireless Sensor Network Based on Ant Colony Optimization Algorithm. *Wireless Personal Communications* **2017**, *95* (3), 1911–1925.

[96] Zhao, Z.; Hou, M.; Zhang, N.; Gao, M. Multipath Routing Algorithm Based on Ant Colony Optimization and Energy Awareness. *Wireless Personal Communications* **2017**, *94* (4), 2937–2948.

[97] Arjunan, S.; Sujatha, P. Lifetime Maximization of Wireless Sensor Network Using Fuzzy Based Unequal Clustering and ACO Based Routing Hybrid Protocol. *Applied Intelligence* **2018**, *48* (8), 2229–2246.

[98] Jiang, A.; Zheng, L. An Effective Hybrid Routing Algorithm in WSN: Ant Colony Optimization in Combination with Hop Count Minimization. *Sensors* **2018**, *18* (4), 1020.

[99] Rathee, M.; Kumar, S.; Gandomi, A. H.; Dilip, K.; Balusamy, B.; Patan, R. Ant Colony Optimization Based Quality of Service Aware Energy Balancing Secure Routing Algorithm for Wireless Sensor Networks. *IEEE Transactions on Engineering Management* **2021**, *68* (1), 170–182.

[100] Zou, Z.; Qian, Y. Wireless Sensor Network Routing Method Based on Improved Ant Colony Algorithm. *Journal of Ambient Intelligence and Humanized Computing* **2019**, *10* (3), 991–998.

[101] Li, X.; Keegan, B.; Mtenzi, F.; Weise, T.; Tan, M. Energy-Efficient Load Balancing Ant Based Routing Algorithm for Wireless Sensor Networks. *IEEE Access* **2019**, *7*, 113182–113196.

[102] Li, F.; Liu, M.; Xu, G. A Quantum Ant Colony Multi-Objective Routing Algorithm in WSN and Its Application in a Manufacturing Environment. *Sensors* **2019**, *19* (15), 3334.

[103] Jain, A.; Pathak, A. Ant Colony Optimization and Excess Energy Calculations Based Fast Converging Energy Efficient Routing Algorithm for WSNs. *Wireless Personal Communications* **2019**, *109* (4), 2305–2328.

[104] Arora, V. K.; Sharma, V.; Sachdeva, M. A Multiple Pheromone Ant Colony Optimization Scheme for Energy-Efficient Wireless Sensor Networks. *Soft Computing* **2020**, *24* (1), 543–553.

[105] Anandh, S. J.; Baburaj, E. Energy Efficient Routing Technique for Wireless Sensor Networks Using Ant-Colony Optimization. *Wireless Personal Communications* **2020**, *114* (4), 3419–3433.

[106] Nayyar, A.; Singh, R. IEEMARP- a Novel Energy Efficient Multipath Routing Protocol Based on Ant Colony Optimization (ACO) for Dynamic Sensor Networks. *Multimedia Tools and Applications* **2020**, *79* (47), 35221–35252.

[107] Xiao, J.; Li, C.; Zhou, J. Minimization of Energy Consumption for Routing in High-Density Wireless Sensor Networks Based on Adaptive Elite Ant Colony Optimization. *Journal of Sensors* **2021**, *2021*, e5590951.

[108] Wang, X. Low-Energy Secure Routing Protocol for WSNs Based on Multiobjective Ant Colony Optimization Algorithm. *Journal of Sensors* **2021**, *2021*, e7633054.

[109] Wang, Z.; Ding, H.; Li, B.; Bao, L.; Yang, Z.; Liu, Q. Energy Efficient Cluster Based Routing Protocol for WSN Using Firefly Algorithm and Ant Colony Optimization. *Wireless Personal Communications* **2022**, *125* (3), 2167–2200.

[110] Reddy, D. L.; C., P.; Suresh, H. N. Merged Glowworm Swarm with Ant Colony Optimization for Energy Efficient Clustering and Routing in Wireless Sensor Network. *Pervasive and Mobile Computing* **2021**, *71*, 101338.

[111] Maheshwari, P.; Sharma, A. K.; Verma, K. Energy Efficient Cluster Based Routing Protocol for WSN Using Butterfly Optimization Algorithm and Ant Colony Optimization. *Ad Hoc Networks* **2021**, *110*, 102317.

[112] Elhabyan, R. S. Y.; Yagoub, M. C. E. Two-Tier Particle Swarm Optimization Protocol for Clustering and Routing in Wireless Sensor Network. *Journal of Network and Computer Applications* **2015**, *52*, 116–128.

[113] Azharuddin, M.; Jana, P. K. Particle Swarm Optimization for Maximizing Lifetime of Wireless Sensor Networks. *Computers & Electrical Engineering* **2016**, *51*, 26–42.

[114] Sohan, R.; Mittal, N.; Singh, U.; Sohi, B. S. An Optimal Tree-Based Routing Protocol Using Particle Swarm Optimization. In *Nature Inspired Computing*; Springer: Singapore, 2018; pp 117–124.

[115] Wang, J.; Gao, Y.; Liu, W.; Sangaiah, A. K.; Kim, H.-J. An Improved Routing Schema with Special Clustering Using PSO Algorithm for Heterogeneous Wireless Sensor Network. *Sensors* **2019**, *19* (3), 671.

[116] Tabibi, S.; Ghaffari, A. Energy-Efficient Routing Mechanism for Mobile Sink in Wireless Sensor Networks Using Particle Swarm Optimization Algorithm. *Wireless Personal Communications* **2019**, *104* (1), 199–216.

[117] Ghawy, M. Z.; Amran, G. A.; AlSalman, H.; Ghaleb, E.; Khan, J.; AL-Bakhrani, A. A.; Alziadi, A. M.; Ali, A.; Ullah, S. S. An Effective Wireless Sensor Network Routing Protocol Based on Particle Swarm Optimization Algorithm. *Wireless Communications and Mobile Computing* **2022**, *2022*, e8455065.

[118] Ari, A. A. A.; Yenke, B. O.; Labraoui, N.; Damakoa, I.; Gueroui, A. A Power Efficient Cluster-Based Routing Algorithm for Wireless Sensor Networks: Honeybees Swarm Intelligence Based Approach. *Journal of Network and Computer Applications* **2016**, *69*, 77–97.

[119] Mann, P. S.; Singh, S. Artificial Bee Colony Metaheuristic for Energy-Efficient Clustering and Routing in Wireless Sensor Networks. *Soft Computing* **2017**, *21* (22), 6699–6712.

[120] Wang, Z.; Ding, H.; Li, B.; Bao, L.; Yang, Z. An Energy Efficient Routing Protocol Based on Improved Artificial Bee Colony Algorithm for Wireless Sensor Networks. *IEEE Access* **2020**, *8*, 133577–133596.

[121] Sahu, S.; Silakari, S. A Whale Optimization-Based Energy-Efficient Clustered Routing for Wireless Sensor Networks. In *Soft Computing: Theories and Applications*; Springer Nature: Singapore, 2022; pp 333–344.

[122] Chouhan, N.; Jain, S. C. Tunicate Swarm Grey Wolf Optimization for Multi-Path Routing Protocol in IoT Assisted WSN Networks. *Journal of Ambient Intelligence and Humanized Computing* **2020**.

[123] Auger, A.; Hansen, N. Tutorial CMA-ES: Evolution Strategies and Covariance Matrix Adaptation; 2012. https://doi.org/ACM 978-1-4503-1178.

[124] Hansen, N. The CMA Evolution Strategy: A Comparing Review. In *Towards a New Evolutionary Computation: Advances in the Estimation of Distribution Algorithms*; Springer: Berlin, Heidelberg, 2006; pp 75–102. https://doi.org/10.1007/3-540-32494-1_4.

[125] Hansen, N. The CMA Evolution Strategy: A Tutorial. arXiv April 4, 2016. https://doi.org/10.48550/arXiv.1604.00772.

[126] Tong, Y. L. *The Multivariate Normal Distribution*; Springer Science & Business Media, 2012.

[127] Surjanovic, S.; Bingham, D. Virtual Library of Simulation Experiments: Test Functions and Datasets https://www.sfu.ca/~ssurjano/index.html (accessed 2023 -02 -24).

[128] MathWorks - Makers of MATLAB and Simulink https://uk.mathworks.com/ (accessed 2023 -02 -24).

[129] Åkerberg, J.; Gidlund, M.; Björkman, M. Future Research Challenges in Wireless Sensor and Actuator Networks Targeting Industrial Automation. In *2011 9th IEEE International Conference on Industrial Informatics*; 2011; pp 410–415.

[130] Saifullah, A.; Xu, Y.; Lu, C.; Chen, Y. Real-Time Scheduling for WirelessHART Networks. In *2010 31st IEEE Real-Time Systems Symposium*; IEEE: San Diego, CA, USA, 2010; pp 150–159.

[131] Li, C.; Ye, M.; Chen, G.; Wu, J. An Energy-Efficient Unequal Clustering Mechanism for Wireless Sensor Networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*; 2005; p 8 pp. – 604.

[132] Kim, J.-H.; Sajjad Hussain, C.; Yang, W.-C.; Kim, D.-S.; Park, M.-S. PRODUCE: A Probability-Driven Unequal Clustering Mechanism for Wireless Sensor Networks. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)*; 2008; pp 928–933.

[133] Soro, S.; Heinzelman, W. B. Prolonging the Lifetime of Wireless Sensor Networks via Unequal Clustering. In *Proceedings - 19th IEEE International Parallel and Distributed Processing Symposium, IPDPS 2005*; IEEE Computer Society, 2005; Vol. 2005, pp 8–15.

[134] Ever, E.; Luchmun, R.; Mostarda, L.; Navarra, A.; Shah, P. UHEED - an Unequal Clustering Algorithm for Wireless Sensor Networks; Rome, Italy, 2012.

[135] Ahmad, A.; Latif, K.; Javaid, N.; Khan, Z. A.; Qasim, U. Density Controlled Divide-and-Rule Scheme for Energy Efficient Routing in Wireless Sensor Networks. In *2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*; 2013; pp 1–4.

[136] Singh, S. K.; Kumar, P.; Singh, J. P. An Energy Efficient Protocol to Mitigate Hot Spot Problem Using Unequal Clustering in WSN. *Wireless Personal Communications* **2018**, *101* (2), 799–827.

[137] Elkamel, R.; Messouadi, A.; Cherif, A. Extending the Lifetime of Wireless Sensor Networks through Mitigating the Hot Spot Problem. *Journal of Parallel and Distributed Computing* **2019**, *133*, 159–169.

[138] El Assari, Y. Energy-Efficient Multi-Hop Routing with Unequal Clustering Approach for Wireless Sensor Networks. Rochester, NY 2020.

[139] Biswas, T.; Kumar, S.; Singh, T.; Gupta, K.; Saxena, D. A Comparative Analysis of Unequal Clustering-Based Routing Protocol in WSNs. In *Advances*

in Intelligent Systems and Computing; Springer Verlag, 2019; Vol. 900, pp 53–62.

[140] Bandyopadhyay, S.; Coyle, E. J. An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No.03CH37428)*; 2003; Vol. 3, pp 1713–1723 vol.3.

[141] Daanoune, I.; Abdennaceur, B.; Ballouk, A. A Comprehensive Survey on LEACH-Based Clustering Routing Protocols in Wireless Sensor Networks. *Ad Hoc Networks* **2021**, *114*, 102409.

[142] Younis, O.; Fahmy, S. HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks. *IEEE Transactions on Mobile Computing* **2004**, *3* (4), 366–379.

[143] Danielsson, P.-E. Euclidean Distance Mapping. *Computer Graphics and Image Processing* **1980**, *14* (3), 227–248. https://doi.org/10.1016/0146-664X(80)90054-4.

[144] Ullah, U.; Shahid, A. R.; Irfan, M.; Qadir, J.; Nawaz, M.; Qureshi, R. A Stable and Reliable Short-Path Routing Scheme for Efficient Acoustic Wireless Sensor Networks (AWSNs). *IEEE Access* **2020**, *8*, 1458–1474.

[145] Koo, J.; Panta, R. K.; Bagchi, S.; Montestruque, L. A Tale of Two Synchronizing Clocks. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*; Association for Computing Machinery: New York, NY, USA, 2009; pp 239–252.

[146] Cho, H.; Kim, J.; Baek, Y. Enhanced Precision Time Synchronization for Wireless Sensor Networks. *Sensors* **2011**, *11* (8), 7625–7643.

[147] Gong, F.; Sichitiu, M. L. CESP: A Low-Power High-Accuracy Time Synchronization Protocol. *IEEE Transactions on Vehicular Technology* **2016**, *65* (4), 2387–2396.

[148] Shi, F.; Tuo, X.; Yang, S.; Li, H.; Shi, R. Multiple Two-Way Time Message Exchange (Ttme) Time Synchronization for Bridge Monitoring Wireless Sensor Networks. *Sensors* **2017**.

[149] Sha, M.; Gunatilaka, D.; Wu, C.; Lu, C. Empirical Study and Enhancements of Industrial Wireless Sensor-Actuator Network Protocols. *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)* **2017**, *4* (3), 261–266.

[150] Leu, J.-S.; Chiang, T.-H.; Yu, M.-C.; Su, K.-W. Energy Efficient Clustering Scheme for Prolonging the Lifetime of Wireless Sensor Network With Isolated Nodes. *IEEE Communications Letters* **2015**, *19* (2), 259–262.

[151] Lee, J.-J.; Krishnamachari, B.; Kuo, J. Aging Analysis in Large-Scale Wireless Sensor Networks. *Ad Hoc Networks* **2008**, *6* (7), 1117–1133.

[152] Li, J.; Huo, J. Uneven Clustering Routing Algorithm Based on Optimal Clustering for Wireless Sensor Networks. *Journal of Communications* **2016**.

[153] Mhatre, M.; Kumar, A.; Jha, C. K. Energy-Efficient WSN Using Membership Handshaking Clustering Technique for Isolated Nodes. In *Pervasive Computing: A Networking Perspective and Future Directions*; Springer: Singapore, 2019; pp 145–152.

[154] Bria, R.; Wahab, A.; Alaydrus, M. Energy Efficiency Analysis of TEEN Routing Protocol with Isolated Nodes. In *2019 Fourth International Conference on Informatics and Computing (ICIC)*; 2019; pp 1–5.

[155] Manjeshwar, A.; Agrawal, D. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *ipdps*; 2001; Vol. 1.

[156] Bhagyalakshmi, L.; Suman, S. K.; Sujeethadevi, T. Joint Routing and Resource Allocation for Cluster Based Isolated Nodes in Cognitive Radio Wireless Sensor Networks. *Wireless Personal Communications* **2020**, *114* (4), 3477–3488.

[157] Vimal, V.; Singh, U.; Kumar, A.; Gupta, K.; Rashid, M.; Saket, K.; Padmanaban, S. Clustering Isolated Nodes to Enhance Network's Life Time of WSNs for IoT Applications. *IEEE Systems Journal* **2021**, *15* (4), 5654–5663.

[158] Nobre, M.; Silva, I.; Guedes, A. Performance Evaluation of WirelessHART Networks Using a New Network Simulator 3 Module. *Computers & Electrical Engineering* **2015**, *41*, 325–341.

[159] Chen, D.; Nixon, M.; Mok, A. *WirelessHART^{TM}: Real-Time Mesh Network for Industrial Automation*; Springer US, 2010.

[160] Muhammed, T.; Mehmood, R.; Albeshri, A.; Alzahrani, A. HCDSR: A Hierarchical Clustered Fault Tolerant Routing Technique for IoT-Based Smart Societies. In *Smart Infrastructure and Applications: Foundations for Smarter Cities and Societies*; Springer International Publishing: Cham, 2020; pp 609–628.

[161] Tramarin, F.; Mok, A. K.; Han, S. Real-Time and Reliable Industrial Control Over Wireless LANs: Algorithms, Protocols, and Future Directions. *Proceedings of the IEEE* **2019**, *107* (6), 1027–1052.

[162] Xu, X.; Hu, H.; Liu, Y.; Tan, J.; Zhang, H.; Song, H. Moving Target Defense of Routing Randomization with Deep Reinforcement Learning against Eavesdropping Attack. *Digital Communications and Networks* **2022**, *8* (3), 373–387.

[163] Obaidat, M. S.; Woungang, I.; Dhurandher, S. K.; Koo, V. Preventing Packet Dropping and Message Tampering Attacks on AODV-Based Mobile Ad Hoc Networks. In *2012 International Conference on Computer, Information and Telecommunication Systems (CITS)*; 2012; pp 1–5.

[164] Misra, S.; Venkata, P.; Isaac, K.; Sasikumar, N.; Fredun, S. An Adaptive Learning Routing Protocol for the Prevention of Distributed Denial of Service Attacks in Wireless Mesh Networks. *Computers & Mathematics with Applications* **2010**, *60* (2), 294–306.

[165] Patel, S. T.; Mistry, N. H. A Review: Sybil Attack Detection Techniques in WSN. In *2017 4th International Conference on Electronics and Communication Systems (ICECS)*; 2017; pp 184–188.

[166] Bhushan, B.; Sahoo, G.; Rai, A. K. Man-in-the-Middle Attack in Wireless and Computer Networking — A Review. In *2017 3rd International Conference on Advances in Computing,Communication & Automation (ICACCA) (Fall)*; 2017; pp 1–6.

[167] Heris, M. Yarpiz https://yarpiz.com/about (accessed 2023 -02 -24).