Clustering and Routing Protocols for Wireless Sensor Networks: Design and Performance Evaluation

by

Riham Elhabyan

Thesis submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In this thesis, we propose a suite of Evolutionary Algorithms (EA)-based protocols to solve the problems of clustering and routing in Wireless Sensor Networks (WSNs). At the beginning, the problem of the Cluster Heads (CHs) selection in WSNs is formulated as a single-objective optimization problem. A centralized weighted-sum multi-objective optimization protocol is proposed to find the optimal set of CHs. The proposed protocol finds a predetermined number of CHs in such way that they form one-hop clusters. The goal of the proposed protocol is to enhance the network's energy efficiency, data delivery reliability and the protocol's scalability. The formulated problem has been solved using three evolutionary approaches: Genetic Algorithms (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) and we assessed each of their performance. Then, a PSO-based hierarchical clustering protocol that forms two-hop clusters is proposed to investigate the effect of the number of CHs on network's energy efficiency. This protocol enhances the WSN's energy efficiency by setting an upper bound on the number of CHs and trying to minimize the number of CHs compared to that upper bound. It also maximizes the protocol's scalability by using two-hop communication between the sensor nodes and their respective CHs. Then, a centralized weighted-sum PSO-based protocol is proposed for finding the optimal inter-cluster routing tree that connects the CHs to the Base Station (BS). This protocol is appropriate when the CHs are predetermined in advance. The proposed protocol uses a particle encoding scheme and defines an objective function to find the optimal routing tree. The objective function is used to build the trade-off between the energy-efficiency and data delivery reliability of the constructed tree. Finally, a centralized multi-objective Pareto-optimization approach is adapted to find the optimal network configuration that includes both the optimal set of CHs and the optimal routing tree. A new individual encoding scheme that represents a joint solution for both the clustering and routing problems in WSNs is proposed. The proposed protocol uses a variable number of CHs, and its objective is to assign each network node to its respective CH and each CH to its respective next hop. The joint problem of clustering and routing in WSNs is formulated as a multi-objective minimization problem with a variable number of CHs, aiming at determining an energy efficient, reliable (in terms of data delivery) and scalable clustering and routing scheme. The formulated problem has been solved using two state-of-the-art Multi-Objective Evolutionary Algorithms (MOEA), and their performance has been compared.

The proposed protocols were developed under realistic network settings. No assumptions were made about the nodes' location awareness or transmission range capabilities. The proposed protocols were tested using a realistic energy consumption model that is based on the characteristics of the Chipcon CC2420 radio transceiver data sheet. Extensive simulations on 50 homogeneous and heterogeneous WSN models were evaluated and compared against well-known cluster-based sensor network protocols.

Acknowledgements

It is difficult to put into words my gratitude to my Ph.D. supervisor Dr. Mustapha C.E. Yagoub. His enthusiasm and motivation have helped to make my Ph.D. experience much more interesting and productive. He has provided me with encouragement, inspiration, priceless advice and friendship, resources and financial support. Professor Yagoub, I am in debt with you.

I would like to thank Dr. Abdulmotaleb El Saddik and Dr. Tony White for their valuable comments and inputs during my research.

I am deeply grateful to my parents who raised me with a love of science and supported me at all times. To my children Ahmed, and Maya, I can not turn back time; however, I will try to make up for the times that I have missed spending with you. Most of all, I wish to thank my loving, supportive, encouraging, and patient husband, Mohamed, whose faithful support during my studies is so appreciated.

Above all, I give thanks to God for giving me strength and inspiration to follow my dream.

I dedicate this thesis to my parents for giving me life, and to Mohamed, Ahmed and Maya for sharing it with me.

Table of Contents

List of Tables viii

List of Figures xi

1 Introduction 1

1.1 Thesis Motivation . . . 3
1.2 Research Contributions . . . 4
1.3 List of Publications . . . 7
1.4 Thesis Outline . . . 8

2 Wireless Sensor Networks and Evolutionary Algorithms 9

2.1 Wireless Sensor Networks . . . 9
2.1.1 Applications of Wireless Sensor Networks . . . 11
2.1.2 Limitations, Requirements and Challenges of Wireless Sensor Networks 14
2.1.3 Link Quality Metrics in Wireless Sensor Networks . . . 15
2.2 Evolutionary Algorithms . . . 17
2.2.1 Genetic Algorithms . . . 17
2.2.2 Differential Evolution . . . 20
2.2.3 Particle Swarm Optimization . . . 22

3 Literature Review 24

3.1 Heuristic Approaches . . . 24
3.1.1 Low Energy Adaptive Clustering Hierarchy (LEACH) . . . 24
3.1.2 Hybrid Energy-Efficient Distributed Clustering (HEED) . . . 26
3.1.3 Energy-Efficient Clustering Scheme (EECS) . . . 27
3.1.4 Energy Efficient Heterogeneous Clustered (EEHC) Scheme . . . 29

3.1.5 Enhanced Heterogeneous LEACH Protocol for Lifetime Enhancement (EHE-LEACH) . . . 31
3.1.6 Single-hop and Multi-hop Energy-Efficient Clustering Protocols (S-EECP) and (M-EECP) . . . 33
3.2 Meta-heuristic Approaches . . . 34
3.2.1 LEACH-Centralized (LEACH-C) . . . 34
3.2.2 Energy-aware Clustering for WSNs using PSO Algorithm (PSO-C) 35
3.2.3 Energy Balanced Unequal Clustering Protocol (EBUC) . . . 37
3.2.4 A Novel Genetic Algorithm in LEACH-C Routing Protocol for Sensor Networks (GA-C) . . . 39
3.2.5 A Evolutionary Approach for Load Balanced Clustering Problem for WSN (GA-LBC) . . . 40
3.2.6 Energy-aware Evolutionary Routing Protocol for Dynamic Clustering of WSNs (EAERP) . . . 41
3.3 Supplementary Remarks . . . 42
3.4 The System Model . . . 43
3.4.1 The WSN Model . . . 45
3.4.2 The Energy Consumption Model . . . 45
3.4.3 Overview of the Proposed Protocols . . . 46
3.4.4 Simulation Settings . . . 47

4 Weighted-sum based Optimization Protocols for Clustering in WSN 49
4.1 Introduction and Motivation . . . 49
4.2 Weighted-sum Approach for Multi-objective Optimization . . . 51
4.3 One-hop Clustering Protocol . . . 52
4.3.1 Individual Initialization . . . 53
4.3.2 Individual Evaluations . . . 54
4.3.3 The Steady-state Phase . . . 57
4.3.4 Experimental Results . . . 57
4.4 Hierarchical Clustering Protocol . . . 64
4.4.1 Particle Initialization . . . 66
4.4.2 Particle Evaluations . . . 69
4.4.3 The Steady-state Phase . . . 71
4.4.4 Experimental Results . . . 72
4.5 Conclusion . . . 76

5 PSO-based Routing Protocol for Clustered WSN **78**
5.1 Introduction and Motivation . . . 78
5.2 Particle Initialization . . . 79
5.2.1 Particle Encoding Process . . . 80
5.2.2 Particle Decoding Process . . . 80
5.3 Particle Evaluation . . . 84
5.3.1 Energy Efficiency . . . 84
5.3.2 Data Delivery Reliability . . . 85
5.4 Experimental Results . . . 85
5.5 Conclusion . . . 91

6 Pareto-based Optimization Protocol for Clustering and Routing in WSN **92**
6.1 Introduction and Motivation . . . 92
6.2 Pareto-based Multi-objective Optimization . . . 94
6.2.1 Basic Concepts . . . 94
6.2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II) . . . 96
6.2.3 Speed-constrained Multi-objective PSO (SMPSO) . . . 97
6.2.4 Performance assessment of different MOEAs . . . 99
6.3 Joint Clustering and Routing Approach for WSN . . . 103
6.3.1 Individual Initialization . . . 103
6.3.2 Routing Tree Construction . . . 107
6.3.3 Individual Evaluations . . . 110
6.3.4 Determining the Best Compromise Individual . . . 112
6.4 Calculation of the Objective Functions . . . 113
6.4.1 Energy Efficiency . . . 113
6.4.2 Scalability . . . 114
6.4.3 Data Delivery Reliability . . . 115
6.5 Experimental Results . . . 116
6.5.1 Performance Evaluation of NSGA-II and SMPSO . . . 116
6.5.2 Performance Evaluation of the Proposed Protocol . . . 120
6.6 Conclusion . . . 132

7 Conclusions and Future Research Directions **135**

7.1 Conclusions 135

7.2 Future Research Directions 137

References **139**

List of Tables

3.1 Comparison of clustering protocols with respect to clustering attributes . . 44

3.2 Summery of the WSN simulation settings for the proposed protocols 48

4.1 The evolutionary algorithms parameters settings for the proposed one-hop clustering protocol 58

4.2 Mean and Standard Deviation for the fitness values for the proposed one-hop clustering protocol 59

4.3 Mean PDR and standard deviation in $WSN\#1$ for PSO-OC 63

4.4 Mean PDR and standard deviation in $WSN\#2$ for PSO-OC 63

4.5 Mean for average consumed energy per node and standard deviation in $WSN\#1$ for PSO-OC 65

4.6 Mean for average consumed energy per node and standard deviation in $WSN\#2$ for PSO-OC 65

5.1 PSO algorithm settings for TPSO-CR 87

5.2 Mean for average consumed energy per node and standard deviation in $WSN\#1$, for TPSO-CR 90

5.3 Mean for average consumed energy per node and standard deviation in $WSN\#2$, for TPSO-CR 90

6.1 Individual decoding process to assign a CH to each node 106

6.2 Parameters setting of NSGA-II and SMPSO 117

6.3 Mean and standard deviation for the HV Indicator 117

6.4 Mean and standard deviation for the Epsilon Indicator 118

6.5 The Average number of non-dominated solutions per run 120

6.6 Average objective functions values for NSGA-II and SMPSO 121

6.7 Minimum objective functions values for NSGA-II and SMPSO 121

6.8 Simulation settings for SMPSO-CR 123

6.9 Average number of cluster heads per round for $WSN\#1$, for SMPSO-CR . 124
6.10 Average number of cluster heads per round for $WSN\#2$, for SMPSO-CR . 124
6.11 Mean and standard deviation for the average consumed energy per node and standard deviation in $WSN\#1$, for SMPSO-CR 127
6.12 Mean and standard deviation for the average consumed energy per node in $WSN\#2$, for SMPSO-CR 127

List of Abbreviations

BS	Base station
CH	Cluster head
DE	Differential evolution
EA	Evolutionary Algorithm
EAERP	Energy-aware evolutionary routing protocol for dynamic clustering of WSNs
EBUC	Energy balanced unequal clustering protocol
EECS	Energy-efficient clustering scheme
EEHC	Energy efficient heterogeneous clustered scheme
EHE-LEACH	Enhanced heterogeneous LEACH protocol
GA	Genetic algorithm
GA-C	Genetic algorithm-based clustering protocol
GA-LBC	Evolutionary approach for load balanced clustering problem
GPS	Global positioning system
HEED	Hybrid energy-efficient distributed clustering
HV	Hypervolume indicator
LEACH	Low energy adaptive clustering hierarchy
LEACH-C	LEACH centralized
LQI	Link quality indicator
M-EECP	Multi-hop energy-efficient clustering protocol
MOEA	Multi-objective evolutionary algorithms
NP	Non-deterministic Polynomial

NSGA-II	Non-dominated sorting genetic algorithm II
OCP	Optimal Coverage Problem
PDR	Packet delivery rate
PSO	Particle swarm optimization
PSO-C	Energy-aware clustering for WSNs using PSO algorithm
PSO-HC	PSO-based approach for hierarchical clustering
PSO-OC	PSO-based one-hop clustering protocol
RSSI	Received signal strength indicator
S-EECP	Single-hop energy-efficient clustering protocol
SA	Simulated annealing
SMPSO	SMPSO-based approach for joint clustering and routing in WSN
SMPSO	Speed-constrained multi-objective particle swarm optimization
TDMA	Time division multiple access
TPC	Transmission Power Control
TPSO-CR	Two-tier particle swarm optimization for clustering and routing protocol
WSA	weighted-sum approach
WSN	Wireless sensor network

List of Figures

2.1 Generalized View of a clustered WSN [25] 11

4.1 Schedule of set-up and steady-state phases in a given round, in the proposed one-hop clustering approach 57
4.2 Comparison of GA, DE and PSO for different network sizes [100 - 500] . . 60
4.3 Average number of unclustered nodes per round for $WSN\#1$ for PSO-OC 61
4.4 Average number of unclustered nodes per round for $WSN\#2$ for PSO-OC 62
4.5 Example of two different particles and their respective CH candidates, upper bound = 5, red nodes are CHs 67
4.6 Cluster Formation Process for PSO-HC 68
4.7 TDMA Scheduling in PSO-HC 69
4.8 Schedule of set-up and steady-state phases in a given round, in PSO-HC . 71
4.9 Average number of unclustered nodes per round for $WSN\#1$ for PSO-HC 72
4.10 Average number of unclustered nodes per round for $WSN\#2$ for PSO-HC 73
4.11 PDR for $WSN\#1$ for PSO-HC 74
4.12 PDR for $WSN\#2$ for PSO-HC 75
4.13 Average energy consumed per node in $WSN\#1$ for PSO-HC 75
4.14 Average energy consumed per node in $WSN\#2$ for PSO-HC 76

5.1 A wireless sensor network with 20 sensor nodes and 2 CHs (n_1 and n_8) . . 81
5.2 Example of priority-based encoding and decoding process for an arbitrary particle P_i 83
5.3 Throughput for $WSN\#1$, for TPSO-CR 88
5.4 Throughput for $WSN\#2$, for TPSO-CR 89

6.1 HV enclosed by the non-dominated solutions A,B, and C [130]. 102
6.2 Workflow of the Proposed Approach 104

6.3 A wireless sensor network with 20 sensor nodes 105
6.4 Final assignment of the sensor nodes to their respective next hop 106
6.5 The generated clusters that correspond to the final assignments 107
6.6 Final assignment of the sensor nodes to their respective next hop 108
6.7 Final assignment of the sensor nodes to their respective next hop 109
6.8 Boxplots of the HV obtained by NSGA-II and SMPSO in the evaluated problem, for different network sizes [100 - 500] 118
6.9 Boxplots of the Epsilon obtained by NSGA-II and SMPSO in the evaluated problem, for different network sizes [100 - 500] 119
6.10 Average number of unclustered nodes per round for $WSN\#1$, for SMPSO-CR 125
6.11 Average number of unclustered nodes per round for $WSN\#2$, for SMPSO-CR 126
6.12 Throughput for $WSN\#1$, for SMPSO-CR 128
6.13 Throughput for $WSN\#2$, for SMPSO-CR 129
6.14 Average number of CHs nodes per round for $WSN\#1$, for SMPSO-CR . . 130
6.15 Average number of CHs per round for for $WSN\#2$, for SMPSO-CR 130
6.16 Average consumed energy per node for $WSN\#1$, for SMPSO-CR 131
6.17 Average consumed energy per node for $WSN\#2$, for SMPSO-CR 132
6.18 Throughput for $WSN\#1$, for SMPSO-CR 133
6.19 Throughput for $WSN\#2$, for SMPSO-CR 133

List of Algorithms

1 Pseudo-code of the proposed TPSO-CR protocol 86

2 The main steps of the NSGA-II Algorithm 97

3 Non-dominated Sorting 98

4 Crowding distance calculation 98

5 Crowding Selection Operator 98

6 The main steps of the SMPSO Algorithm 100

7 The Dijkstra algorithm to find the SPT (and its cost) that connects the CHs to the BS 111

Chapter 1

Introduction

Wireless sensor networks (WSN) have attracted significant attention from the research community and the industry in the last few years. The main reason for the recent research efforts and rapid development of WSNs is their potential application in a wide range of contexts including military operations, environment monitoring, surveillance systems, health care, environmental monitoring and public safety [1, 2].

A common practice in these applications is the deployment of a large number of sensors, often randomly deployed over the entirety of the geographical region of interest. These sensor nodes sense the environment, communicate with neighboring nodes, and in many cases perform basic computations on the data being collected [3, 4].

In order to realize the existing and potential applications for WSNs, sophisticated and extremely efficient routing protocols are needed. However, the inherent properties of individual sensor nodes pose additional challenges to the communication protocols in terms of energy consumption and throughput. These properties include limited power supply and short transmission range. Due to these unique inherent characteristics, it is a challenging task to select or propose a new routing protocol for a specific WSN application [5].

Using clustering techniques in WSNs can help solve some of those concerns. In clus-

tering, the network nodes are organized into smaller clusters and a *Cluster Head* (CH) for each cluster is elected. Sensor nodes in each cluster transmit their data to their respective CH and the CH aggregates data and forwards them to a central *Base Station* (BS) [6]. The fact that only the CH is transmitting information out of the cluster helps avoiding collisions between the sensors inside of the cluster because they do not have to share the communication channel with the nodes in other clusters [7].

The objective of clustering is to search among a group of sensor nodes to find a set of nodes that can act as cluster heads. For a given network topology, it is difficult to find the optimal set of CH nodes. Finding the optimal set of CHs has been proven to be a Non-deterministic Polynomial (NP)-hard optimization problem [8, 9, 10, 11, 12, 13].

Once the WSN has been divided into clusters, the communication between the nodes can be either intra-cluster or inter-cluster. Intra-cluster communication comprises the data exchanges between the member nodes and their respective CH. Inter-cluster communication includes transmission of the data between the CHs and the BS.

The process by which data are forwarded efficiently between the CHs and the BS, the inter-cluster communication, is an important aspect and essential feature of WSNs. A simple method to accomplish this task is for each CH to exchange data directly with the BS, a single hop-based approach. Another method is to allow intermediate nodes to participate in forwarding data packets between the CH and the BS which is a multihop-based approach [3]. However, in a WSN, individual nodes have limited communication range and form an ad hoc network over a shared wireless medium. Furthermore, the BS is usually located far away from the sensing area and is often not directly reachable to all nodes due to limited communication range and signal propagation problems. A more realistic approach is to use the multihop inter-cluster communication model. For a more reliable data communication, data packets need to be routed using a multihop communication model [14].

The basic function of a routing protocol is to select a route, from the set of available

routes, which is most efficient based on some specific criteria. Once the optimal set of CHs is elected in the clustering phase, the next step is to find the optimal routing tree from the CHs to the BS while minimizing the total cost of that tree. Routing is, at its most basic level, an optimization problem [15]. Moreover, finding an optimal routing tree is known to be NP-hard problem [16, 17].

Therefore, polynomial-time algorithms are infeasible to use for selecting the optimal set of cluster heads and finding the optimal routing tree, due to their high computational complexity in real-time communications systems. Solutions to NP-hard problems involve searches through vast spaces of possible solutions. Evolutionary computation approaches have been applied successfully to solve a variety of these problems [18, 19].

1.1 Thesis Motivation

Clustering sensor nodes is an efficient topology control method to maximize the network's energy efficiency, data delivery reliability and scalability. In order to have an efficient clustering and routing protocol, three main issues should be considered:

- Low energy consumption during clustering and routing to enhance the network energy efficiency.
- Good link quality between the sensor nodes to enhance the data delivery reliability.
- Good distribution of the cluster heads to minimize the number of unclustered nodes and therefore maximize the number of nodes which participate in the communication. That in turn will enhance the protocol's scalability.

Energy efficiency, data delivery reliability and scalability are key requirements in WSNs [20, 21]. The primary objective of this thesis is to address these requirements via designing a suite of clustering and routing protocols for WSNs.

Many clustering and routing protocols have been proposed for WSNs. However, the performance of these protocols is limited by challenges related to determining an accurate radio model for the sensor nodes in the network. A common limitation is the use of an idealized energy model [22, 23]. A discrete radio model should be used for more accurate and realistic calculation of the power consumption and to determine which links between sensor nodes are available for transmission [22, 23, 24].

In this thesis, we aim at lowering the barrier to designing and developing more realistic clustering and routing protocols. To achieve that goal, we designed the protocols under no assumptions about the nodes' location awareness or transmission range capabilities. Moreover, we tested the protocols under a more realistic energy consumption model, a discrete radio model that is based on the characteristics of the Chipcon CC2420 radio transceiver's data sheet, for realistic computation of the energy consumption.

1.2 Research Contributions

The purpose of this thesis is twofold. First, to design, implement and test protocols to solve the problems of clustering and routing in WSNs. Since both the cluster head selection and finding the optimal routing tree have been proven to be NP-hard problems, evolutionary computation approaches were used to solve these problems. The proposed protocols take into consideration the following properties: the network's energy efficiency, the network's data delivery reliability, and the protocol's scalability. Second, to test and investigate the performance of the proposed protocols against well-known clustering protocols, under a realistic energy consumption model.

In this thesis, the problem of CHs selection in WSN is formulated as a single-objective optimization problem. A centralized weighted-sum multi-objective optimization protocol is proposed to find the optimal set of CHs. The proposed protocol finds a predetermined number of CHs in such way that they form one-hop clusters. The goal of the proposed

protocol is to enhance the network's energy efficiency, data delivery reliability and the protocol's scalability. The formulated problem has been solved using three evolutionary approaches: Genetic Algorithms (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO) to assess their performance.

Then, in order to study the effect of minimizing the number of CHs on the network's energy efficiency, a PSO-based hierarchical clustering protocol that forms two-hop clusters is proposed. The proposed protocol's objective is to enhance the network's energy efficiency by setting an upper bound on the number of CHs and minimizing the number of CHs compared to that upper bound. Furthermore, it improves the protocol's scalability by using two-hop communication between the sensor nodes and their respective CHs.

Then, a centralized weighted-sum PSO-based protocol is proposed for finding the optimal inter-cluster routing tree. This protocol is appropriate when the CHs are predetermined in advance. The proposed protocol uses a particle encoding scheme and defines an objective function to find the optimal routing tree. The objective function is used to build the trade-off between the energy-efficiency and data delivery reliability of the constructed tree.

Finally, a centralized multi-objective Pareto-optimization approach is adapted to find the optimal network configuration that includes both the optimal set of CHs and the optimal routing tree. A new individual encoding scheme that represents a joint solution for both the clustering and routing problems in WSNs is proposed. The proposed protocol uses a variable number of CHs, and its objective is to assign each network node to its respective CH and each CH to its respective next hop. The joint problem of clustering and routing in WSNs is formulated as a multi-objective minimization problem with a variable number of CHs, aiming at determining an energy efficient, reliable (in terms of data delivery) and scalable clustering and routing scheme. The formulated problem has been solved using two state-of-the-art Multi-Objective Evolutionary Algorithms (MOEA), and their performance has been compared.

Furthermore, we developed the protocols under realistic network settings. No assumptions were made about the nodes' location awareness or transmission range capabilities. The protocols were also tested using a realistic energy consumption model that is based on the characteristics of the Chipcon CC2420 radio transceiver's data sheet. Extensive simulations on 50 homogeneous and heterogeneous WSN models were evaluated and compared against some of the well-known clustering protocols.

The main contributions of this thesis include the design and development of:

1. A One-hop clustering protocol that adapts a centralized weighted-sum multi-objective optimization approach to find a predetermined number of CHs and form one-hop clusters. This protocol defines an objective function that takes into consideration the network's energy efficiency, data delivery reliability and the protocol's scalability.

2. A PSO-based hierarchical clustering protocol that forms two-hop clusters. The main objective of this protocol is to enhance the network's energy efficiency by lowering the number of CHs. Moreover, it enhances the protocol's scalability by using two-hop communication between the sensor nodes and their respective CHs.

3. A PSO-based routing protocol that finds the optimal inter-cluster routing tree. A new particle encoding scheme is proposed to find a complete routing tree solution, and a multi-objective fitness function is defined to evaluate the constructed tree based on its energy efficiency and data delivery reliability.

4. A Pareto-based optimization protocol to find a joint solution for both the clustering and routing problems. To achieve that goal, a particle mapping scheme is proposed and the objective functions are defined to consider the following properties: the network's energy efficiency, data delivery reliability and the protocol's scalability.

1.3 List of Publications

The following publications by the author are relevant to the work presented in this thesis.

Journal Papers

[1] R. S. Elhabyan and M. C. Yagoub, "Two-tier particle swarm optimization protocol for clustering and routing in wireless sensor network," *Journal of Network and Computer Applications, Elsevier*, vol. 52, June 2015. Pages 116-128.

[2] R. S. Elhabyan and M. C. Yagoub, "Realistic approach to clustering and routing in wireless sensor networks using particle swarm optimization and Dijkstra's algorithm," *Special Issue of the IEEE Information Reuse and Integration.* Accepted.

[3] R. S. Elhabyan and M. C. Yagoub, "Pareto-based optimization protocol for clustering and routing in wsn," *Applied Soft Computing, Elsevier.* Submitted.

Conference Papers

[1] R. Elhabyan and M. Yagoub, "Evolutionary algorithms for cluster heads election in wireless sensor networks: Performance comparison," in *Science and Information Conference 2015*, 2015. Pages 1070-1076. Accepted.

[2] R. Elhabyan and M. Yagoub, "PSO-HC: Particle swarm optimization protocol for hierarchical clustering in wireless sensor networks," in *International Conference on Col-*

laborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014, Oct 2014. Pages 417-424.

[3] R. Elhabyan and M. Yagoub, "Energy efficient clustering protocol for WSN using PSO," in *Global Information Infrastructure and Networking Symposium (GIIS), 2014*, Sept 2014. Pages 1-3.

[4] R. Elhabyan and M. Yagoub, "Particle swarm optimization protocol for clustering in wireless sensor networks: A realistic approach," in *IEEE 15th International Conference on Information Reuse and Integration (IRI), 2014*, Aug 2014. Pages 345-350.

[5] R. Elhabyan and M. Yagoub, "Weighted tree-based routing and clustering protocol for WSN," in *26th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2013*, May 2013. Pages 1-6.

1.4 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 gives the background knowledge about WSNs. Chapter 3 presents a review of the related work about clustering protocols based on heuristic and meta-heuristic approaches. In Chapter 4, we give a detailed explanation of the system model that was used to develop and test the proposed protocols. Chapter 5 gives a detailed description of the two proposed clustering protocols, the one-hop clustering protocol and the hierarchical clustering protocol. Chapter 6 gives a detailed description of the PSO-based routing protocol that is used to find the optimal routing tree. Chapter 7 gives a detailed description the Pareto-based Optimization Protocol that is used to solve the joint problems of clustering and routing in WSNs. Finally, Chapter 8 concludes this research work and highlights a few future directions.

Chapter 2

Wireless Sensor Networks and Evolutionary Algorithms

2.1 Wireless Sensor Networks

Wireless Sensor Networks (WSNs) have emerged as a powerful technological platform with tremendous and novel applications. It has become an important technology for realizing many applications including both simple phenomena monitoring applications and heavy-duty data streaming applications including military operations, environment monitoring, and surveillance systems.

A WSN usually consists of tens to thousands of sensor nodes that communicate through wireless channels for information sharing and cooperative processing [2]. Usually, the nodes are statically deployed over vast areas. However, they can also be mobile and capable of interacting with the environment.

Though the sensor nodes can work autonomously, they work in a collaborative way to sense the physical parameters of an environment. WSN nodes can sense the environment, communicate with neighboring nodes, and in many cases perform basic computations on

the data being collected [3, 4]. These features make WSN an excellent choice for many applications [2].

Routing is a key process to be considered in WSN. Due to the limited transmission range of each node, it may be necessary for one sensor node to use other sensor nodes in forwarding a packet to its destination, usually the Base Station (BS). The task of finding and maintaining routes in WSNs is non-trivial due to the energy restrictions and transmission range restrictions. To minimize energy consumption, routing protocols proposed in the literature for WSNs employ some well-known routing strategies such as clustering.

Clustering protocols in WSN aim at grouping the sensor nodes into clusters and electing a cluster head (CH) for each cluster. In order to realize an energy efficient WSN, the CHs can aggregate the data sent from the cluster members and send them directly to the BS. A clustering protocol is mainly a two layer protocol. The first layer is used to select the optimal set of CHs while the second layer is responsible for transmitting the data to the BS. Figure 2.1 shows the generalized view of WSNs, which consists of a BS, CHs and sensor nodes (devices) deployed in a geographical region.

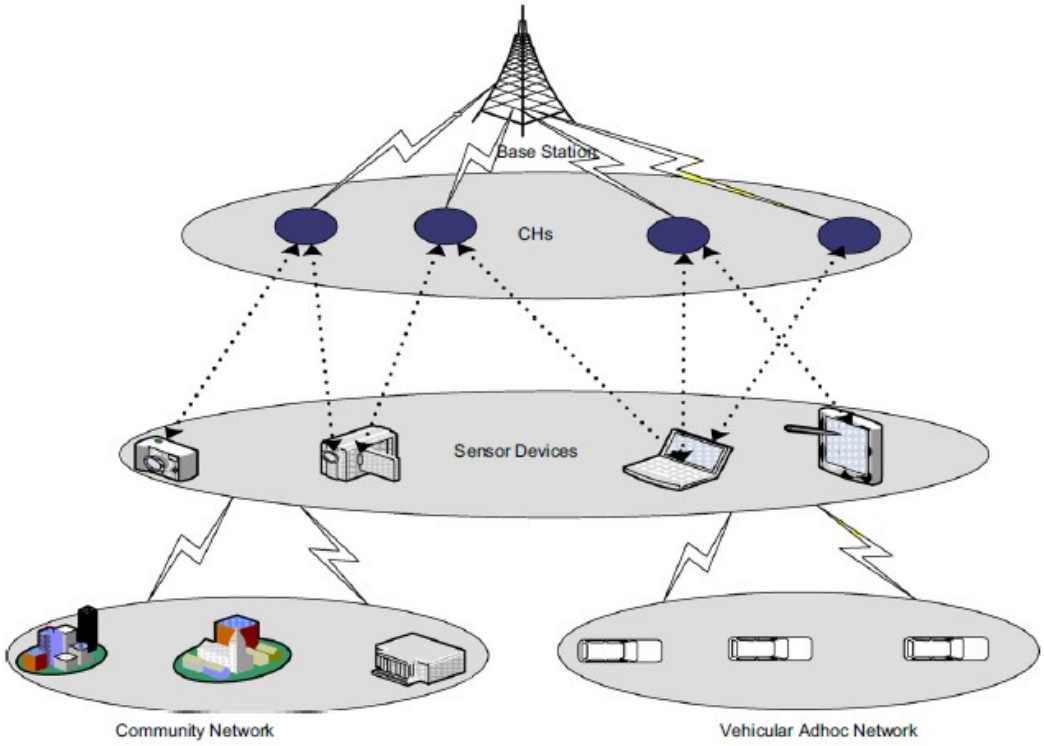


Figure 2.1: Generalized View of a clustered WSN [25]

The clustering protocol in WSN should not only facilitate the data transmission, but also consider the sensor nodes' constraints. It should also meet the WSN requirements including the energy efficiency, the data delivery reliability, and the scalability requirements.

2.1.1 Applications of Wireless Sensor Networks

WSNs are currently being employed in a variety of applications [26] ranging from medical to military, and from home to industry. Potential WSNs applications are briefly introduced in this section.

Environment Monitoring Systems

Environment monitoring has been an important part of WSN applications [27]. Environment monitoring systems control and monitor environment parameters including temperature, humidity, light and pressure.

The applications of environmental monitoring have grown rapidly in agricultural monitoring, habitat monitoring, indoor monitoring, greenhouse monitoring, climate monitoring and forest monitoring.

There are several studies that focus on environment monitoring applications [28, 29, 30]. The main requirements of environment monitoring applications are scalability, coverage and energy efficiency. Monitored sites can reach several tens of hectares, so the number of nodes deployed varies from dozens to thousands. For this reason, scalability is an important issue when developing protocols to support a large quantity of nodes and to ensure full coverage of the controlled area [31].

The protocols proposed in this thesis are suitable for environment monitoring applications.

Human Body Monitoring

In recent years, research interest in the area of wireless healthcare systems has grown. Increasing numbers of aging population, people who need continuous health monitoring and rising costs of health care have triggered the concept of the novel wireless technology-driven human body monitoring.

Wireless Body Sensor Networks (WBSNs) have great potential to enable a broad variety of assisted living applications such as human biophysical/biochemical control and activity monitoring for health care, e-fitness, emergency detection, emotional recognition for social networking, security, and highly interactive games [32].

Several efforts have been made by researchers to use WBSNs for human body monitoring [33, 34, 35]. Human body monitoring is performed using a network of wireless sensors which may be attached to the body surface or implanted into the body tissue. Recent technology developments have produced small and intelligent medical sensors that can be worn or implanted in the human body. The sensors acquire the data and send it to the center in which the data is aggregated and analyzed.

Health monitoring applications demand high reliability because they involve the life of human beings [36]. The network's energy efficiency is another important requirement to ensure long time operation of the system [35, 36].

Intelligent Buildings

WSNs have been recently adapted for building automation to address the rising cost of energy and the growing green movement. Buildings can optimize their energy consumption, improve safety and security, and reduce operating expenses by using smart sensor nodes.

Several intelligent building management systems, using WSNs, have been proposed in the literature [37, 38, 39].

WSNs used in intelligent building management systems consist of different types of sensor nodes measuring parameters such as temperature, humidity, light, and asphyxiating smoke. In addition, the systems may include actuators, gateways, servers, and communication and application software on different levels as well as different home appliances [40].

Intelligent building management systems require the use of multihop communication to cover entire buildings. Specific data-centric or hierarchical protocols can be used to realize this requirement [38]. The network's energy efficiency is another important requirement for such systems [40].

2.1.2 Limitations, Requirements and Challenges of Wireless Sensor Networks

WSNs consist of a large number of sensor nodes. Sensor nodes are small-scale devices that are very limited in the amount of energy they can store. Hence, *energy efficiency* is a major requirement in WSN design.

WSNs should facilitate the data transmission to the BS and at the same time consider the sensor nodes' limited transmission range. The *data delivery reliability* is another important requirement of WSNs.

Clustering allows sensors to coordinate their local interactions efficiently in order to achieve global goals such as scalability and more efficient resource utilization [41, 20]. Scalability refers to the capability of the system to perform useful work as the size of the system increases [42]. Due to the limited transmission range of the individual nodes relative to the large area of interest, thousands of sensors may have to be deployed for a given task. Any routing scheme must be able to work with this huge number of sensor nodes. Any mechanism employed in WSN should be able to adapt to a wide range of network sizes. Therefore, *scalability* is another critical requirement in the design of the system. In order to measure the performance of the proposed protocols in terms of their scalability, we used the number of un-clustered nodes metric. A higher number of un-clustered nodes indicates a lower performance in terms of the protocol scalability.

However, due to the unreliable nature of wireless communication and the limited resources of sensor nodes, these requirements pose challenges for WSN researchers. The main challenges for the realization of the WSNs can be outlined as follows:

1. Energy poses a big challenge for the WSN designers. Since sensor nodes are battery powered, they have limited energy capacity. When the energy of a sensor reaches a certain threshold, they become faulty and are unable to function properly which

affects the overall network's performance. Consequently, the routing protocols designed for sensors should be as energy efficient as possible to minimize the energy consumption of the nodes and hence prolong the network's lifetime.

2. The main task in WSN is the data transmission from the target area to the BS. However, the sensor nodes have a limited transmission range. Moreover, the link quality between the sensor nodes should be maximized to enhance the network's throughput and hence enhance the data delivery reliability of the network. Clustering techniques coupled with multi-hop routing protocols should be employed to achieve this requirement.

3. WSNs consist of an extremely high numbers of nodes and relatively high node density. It can be a great challenge for WSN designers to design a routing protocol that is scalable enough to work with this huge number of sensor nodes.

4. Finding the best clusters and routes in WSNs is highly impacted by the used link quality metric. Many location-aware or link quality-based clustering protocols assumed that each node is equipped with self-locating hardware such as a Global Positioning System (GPS). Though this is a simple and effective solution, the resulting cost renders such a solution inefficient and unrealistic [43]. In order to design realistic protocols, no assumptions should be made about the nodes' locations.

2.1.3 Link Quality Metrics in Wireless Sensor Networks

Link quality estimation in WSNs has a fundamental impact on the network performance and affects the design of clustering and routing protocols. Many of the previously proposed clustering and routing protocols used the distance between two nodes as a metric of their link's quality. However, this solution suffers from the following problems:

1. In order to calculate the distance between two nodes, each node should be equipped

with self-locating hardware such as a Global Positioning System (GPS). The resulting cost of attaching a GPS to each node renders such a solution inefficient and unrealistic [43].

2. Link asymmetry is an important characteristic of WSN. Using the distance between two nodes as a link quality metric ignores this fact and assumes that links are symmetric.

3. Several studies have shown that link quality in WSN is not necessarily correlated with distance [44, 45, 46, 47].

Two other prominent link-quality metrics are, the Received Signal Strength Indicator (RSSI) and Link Quality Indicator (LQI). These metrics are provided by most of the wireless sensor chips [48].

The RSSI is a parameter that represents the signal strength observed at the receiver at the moment of reception of the packet. The LQI is described as the characterization of the strength and quality of the received packets.

Several studies proved that RSSI can provide a quick and accurate estimate of whether a link is of very good quality [45, 46, 47, 44]. In [47], the authors conducted empirical measurements of the packet delivery performance of various sensor platforms. They found that there was a strong correlation between RSSI and Packet Delivery Rate (PDR). Furthermore, they proved that if the RSSI of a link is -87dBm or stronger, it is almost but not completely set to have a PDR $\geq$ 99%. Below this value, a shift in the RSSI as small as 2 dBm can change a good link to a bad one and vice versa, which means that the link is in the transitional or disconnected region [44].

The symmetry of RSSI and LQI in two directions was studied, and the relation between RSSI and LQI as link quality metrics was analyzed in [48]. Experimental results showed a significant correlation between the two directions of the link in RSSI but a weak correlation

between them in LQI. Moreover, statistical tests on the collected data showed a significant correlation between RSSI and distance in short distance scenarios, which makes RSSI a routing protocol link-quality metric.

In this thesis, the RSSI value for the link between any two nodes is used to assess the quality of that link.

2.2 Evolutionary Algorithms

Evolutionary algorithms (EAs) are powerful optimization techniques that imitate principles of natural selection and survival of the fittest to find near optimal solutions in an unknown search space [18].

In this section, we present a brief overview of three widely used EAs: Genetic Algorithms (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO). These algorithms will be applied to the formulated problem.

2.2.1 Genetic Algorithms

Genetic Algorithms (GA) is a population-based approach for heuristic search in optimization problems. It has been applied to many scientific and engineering problems [49], including optimized routing [50], engineering design [51], job scheduling [52], quality control [53] and so much more.

GAs evolve a set of coded solutions to a specific problem to find the best optimal solution. In GA, each solution is encoded as a vector of binary numbers or floating point numbers, with the same length as the vector of decision variables. The working mechanism of GA is explained below.

Initialization

A vector $(p_1, p_2, ..., p_N)$ is a chromosome to represent a solution to the optimization problem. The GA begins with a randomly initialized population of M chromosomes. Thereafter, the fitness values of all these chromosomes are calculated. A fitness of a chromosome is defined as a value that quantifies the optimality of a solution in such a way that any chromosome in the population can be compared and ranked against all the other chromosomes.

To form a new population, chromosomes are selected according to their fitness. A chromosome is evaluated on its performance with respect to the fitness function. The ones with better fitness values are more likely to survive than the ones with the worse values. Crossover and mutation operators are then applied respectively on the selected chromosomes to find new points in the search space. The result is usually a new generation with better survival abilities. The process is repeated until a stopping criterion is reached.

Selection

GAs use a selection scheme to select individuals from the population to insert into a mating pool. It is desirable that the mating pool comprises good individuals since these individuals will pass their genes on to the next generation [54].

In tournament selection, sets of individuals are chosen randomly from the population and the fitness of individuals from each set is compared with one another. These sets have a fixed size called tournament size. The individual with better fitness is taken to the mating pool. The loss of diversity of the population increases with the increase in the tournament size. In this paper, the tournament size is chosen as two in in order to keep the diversity loss to a minimum.

Crossover

Crossover describes the swapping of fragments between two binary strings at a randomly chosen crossover point (single-point crossover). In other words, it creates two new offspring from two parents. After recombination, two new strings are formed, and these are inserted into the next population.

In this chapter, we use the Simulated Binary Crossover (SBX) [55]. SBX attempts to simulate the offspring distribution of binary-encoded single-point crossover on real-valued decision variables to favour offspring closer to the two parents. The procedure of computing the offspring $p_i^{1,t+1}$ and $p_i^{2,t+1}$ from the parents $p_i^{1,t}$ and $p_i^{2,t}$ is described in [55].

Mutation

The mutation operation is used to prevent the premature convergence of a new generation and to ensure diversity. Mutation alters one or more gene values in a chromosome from their initial state. A mutation probability factor is defined. This probability should be small since it adds some randomness in the population.

In this thesis, we use a polynomial mutation operator. Using this operator, the probability of creating an offspring closer to the parents is higher than the probability of creating one away from them. As the generation t proceeds, this probability of creating an offspring closer to the parents gets higher and higher, and the offspring created are given as follows:

$$y_i^t = p_i^t + (p_{i,max} - p_{i,min}) \times \bar{\delta}_i \tag{2.1}$$

Where $\bar{\delta}_i$ is calculated from the polynomial probability distribution as:

$$\bar{\delta}_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m}+1} - 1, & 0 \leq r_i < 0.5 \\ 1 - [2(1-r_i)]^{\frac{1}{\eta_m}+1}, & 0.5 \leq r_i \leq 1 \end{cases} \tag{2.2}$$

r_i is a random number in the range $[0, 1]$. η_m is called the mutation distribution index and it is a non-negative real number. Smaller values of $\bar{\delta}_i$ are considered a strong mutation and results in new values that are far away from the parents. On the other hand, using bigger values of η_m gives more probability of generating new values that are very close to parents. Strong mutation has better chances of escaping local optima.

2.2.2 Differential Evolution

The differential evolution (DE) algorithm is a simple and powerful population-based stochastic optimization technique that performs well in various standard test functions and real-world optimization problems. The working mechanism of DE is explained below.

Initialization

The DE begins with a randomly defined population of NP vectors. Each vector, also known as genome/chromosome, contains D optimized variables and forms a candidate solution to the optimization problem. The subsequent generations in DE are denoted by $G = 0, 1, ..., G_{max}$. $x_i^G = [x_{i,1}^G, x_{i,2}^G, ..., x_{i,D}^G]$ denotes the ith solution in the Gth generation. The population of the Gth generation is denoted by $X^G = [x_1^G, x_2^G, x_3^G, ..., x_{NP}^G]$.

Mutation

the mutation operation is performed following the creation of the initial population. For each member of the population X^G at generation G, a mutant vector, v_i^G, for $x_i^G, i \in \{1, 2, ..., NP\}$ is formed by adding the weighted difference between two individuals (vectors) chosen randomly from the population to a third one, also chosen randomly, as shown in Eq. 2.3

$$v_i^{G+1} = x_{r1}^G + F \times (x_{r2}^G - x_{r3}^G) \tag{2.3}$$

The indices $r1, r2$ and $r3$ are mutually exclusive integers randomly selected from the range $[1, ..., NP]$ which are also different from the base vector index i. The scale factor F controls the amplification of the differential variation $(x_{r2}^G - x_{r3}^G)$. The step that differentiates one DE approach from another is the mutation process. In the literature, the notation for the mutation scheme in (2.3) is referred to as $DE/rand/1$. Additional mutation schemes are summarized in [56].

Crossover

The following step involves a crossover operation in order to enhance the potential diversity of the population. For each individual x_i^G, a trial individual, u_i^{G+1} , is generated as follows:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1} \; if(R_j \leq CR) or (j = j_{rand}), j = 1, 2, ..., D \\ \\ x_{ij}^G \text{ otherwise} \end{cases} \tag{2.4}$$

where R_j is a uniform random number in the range $[0, 1]$, and j_{rand} is a randomly chosen integer within the range $[1, D]$. The crossover threshold $CR \in [0, 1]$ is used to decide if each dimension of the mutant solution will be brought over to the trial solution.

Selection

Finally, according to the fitness value, selection determines which of the vectors should be chosen for the next generation by implementing one-to-one competition between the generated trail vectors and the corresponding parent vectors. The evaluation operation is expressed as follows, assuming a minimization problem:

$$x_i^{G+1} = \begin{cases} u_i^{G+1}, if\ f(u_i^{G+1}) \leq f(x_i^G) \\ x_i^G, Otherwise \end{cases} \tag{2.5}$$

Where $f(x_i^G)$ is the fitness value for x_i^G.

2.2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a stochastic population based optimization algorithm that was introduced by [57] and inspired by social behaviour of bird flocking or fish schooling.

The basic PSO comprises a swarm of S particles (potential solutions), which fly through a D-dimensional problem search space in search of the global optimum position that produces the best fitness of an objective function [57].

Initially, each particle i is randomly assigned a position x_{id} and a velocity v_{id} ($i = 1, 2, ..., S$), and $d = (1, 2, ..., D)$. In every iteration, each particle adjusts its velocity to follow two best solutions. The first is the cognitive part, where the particle follows its own best solution found so far. This is the solution that produces the lowest cost (has the highest fitness). This value is called $pbest_i$ (particle best). The other best value is the current best solution of the swarm, i.e., the best solution by any particle in the swarm. This value is called *gbest* (global best).

After finding the two best values, particle i then updates both its position and velocity iteratively with the following equations:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times r_1 \times (pbest_i(t) - x_{id}(t)) + c_2 \times r_2 \times (gbest(t) - x_{id}(t)) \tag{2.6a}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2.6b}$$

The parameters, c_1 and c_2 are two positive constants named as learning factors, usually

set as $c_1 = c_2 = c_3 = 2$. r_1 and r_2 are random variables between $[0, 1]$. w is a weight factor that controls the velocity of the particle.

Chapter 3

Literature Review

Clustering techniques have been studied extensively to improve the performance of WSNs [58, 59, 6]. This chapter presents a review of such works based on heuristic and meta-heuristic approaches.

3.1 Heuristic Approaches

3.1.1 Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [60][13] is one of the most common cluster-based routing protocols [58] in WSNs that has been proven to be an effective approach to prolong the network's lifetime. LEACH is a completely distributed approach and does not require a global information of the network. The basic idea of LEACH has been an inspiration for many subsequent clustering protocols. The main objective of LEACH is to equalize the energy load distribution among the CHs.

The operation of LEACH is divided into rounds, and each round is divided into two phases, namely the set-up phase and the steady-state phase. In the set-up phase, the

clusters are organized, while in the steady-state phase, data is delivered to the BS. The Steady-state phase is always longer than the set-up phase to minimize overhead.

During the set-up phase, each node decides whether or not to become a CH for the current round. This decision is based on the suggested percentage of CHs for the network and the number of times the node has been a CH so far. This decision is made by the node by choosing a random number between 0 and 1. The node becomes a CH for the current round if the number is less than the following threshold:

$$T(n) = \begin{cases} \frac{P}{1-P\times(r \mod \frac{1}{P})}, if n \in G \\ 0, Otherwise \end{cases} \tag{3.1}$$

where P is the desired percentage of CHs, r is the current round, and G is the set of nodes that have not been elected as CHs in the last $\frac{1}{P}$ rounds.

When a node is selected as a CH, it broadcasts an advertisement message to the other nodes. Other nodes decide which cluster they will join for this round based on the RSSI of the advertisement. Then, each node sends a membership message to its CH. In order to evenly distribute the energy load among sensor nodes, CHs rotation is performed at each round by generating a new advertisement phase based on equation 3.1. During the steady-state phase, the sensor nodes sense and transmit data to the CHs. The CHs compress the data arriving from nodes that belong to the respective cluster and send an aggregated packet to the BS directly. LEACH uses Time Division Multiple Access (TDMA) to avoid inter-cluster and intra-cluster collisions. After a certain time, which is determined a priori (round length), the network goes back into the set-up phase again and enters another round of CH election.

In LEACH, any node that served as a CH in a certain round cannot be selected as the CH again, so each node can equally share the load imposed upon CHs to some extent. However, LEACH cannot ensure real load-balancing in the case of sensor nodes with dif-

ferent amounts of initial energy because CHs are elected in terms of probabilities without energy considerations. Furthermore, the random mechanism of selecting the CHs does not guarantee even distribution of CHs over the network [7]. LEACH assumes that every node can communicate directly with the BS, which is an unrealistic assumption in many practical situations due to the communication range restrictions of the sensor nodes [14, 3]. We should also take into consideration that increasing the transmission range can cause too much energy consumption.

3.1.2 Hybrid Energy-Efficient Distributed Clustering (HEED)

HEED [61] is another distributed clustering protocol that is an extension of LEACH. The objective of HEED is to provide an energy-efficient clustering protocol with explicit consideration of energy.

HEED is different from LEACH in terms of CH election. HEED does not select nodes as CHs randomly. CHs selection in HEED is primarily based on the residual energy of each node. To increase energy efficiency and further prolong the network's lifetime, a secondary clustering parameter that considers intra-cluster "communication cost" is introduced and can be a function of neighbor proximity or cluster density. Elected CHs in HEED have relatively high average residual energy compared to member nodes. Additionally, one of the main goals of HEED is to get an evenly distributed CHs throughout the networks.

In HEED, CHs selection is achieved with an iterative approach based on two important parameters: residual energy and intra-cluster communication cost of the candidate nodes. Initially, a percentage of CHs among all nodes, C_{prob}, is set to assume that an optimal percentage cannot be computed a priori. The probability that a node becomes a CH is:

$$CH_{prob} = C_{prob} \times \frac{E_{residual}}{E_{max}} \tag{3.2}$$

where $E_{residual}$ the estimated current energy of the node, and E_{max} is a reference maximum energy, which is typically identical for all nodes in the network. The value of CH_{prob}, however, is not allowed to fall below a certain threshold that is selected to be inversely proportional to E_{max}. Afterwards, each node goes through several iterations until it finds the CH. If it hears from no CH, the node elects itself to be a CH and sends an announcement message to its neighbors. Each node doubles its CH_{prob} value and goes to the next iteration until its CH_{prob} reaches 1. Therefore, there are two types of status that a sensor node could announce to its neighbors: tentative status and final status. If its CH_{prob} is less than 1, the node becomes a tentative CH and can change its status to a regular node at a later iteration if it finds a lower cost CH. If its CH_{prob} has reached 1, the node permanently becomes a CH. Every node elects the least communication cost CH in order to join it. On the other hand, CHs send the aggregated data to the BS in a multi-hop fashion rather than the single-hop fashion of LEACH.

The HEED protocol is a fully distributed protocol that creates a uniform CH distribution across the network. However, more CHs are generated than the expected number and this accounts for unbalanced energy consumption [62]. Moreover, some CHs, especially near the BS, may die earlier because these CHs have more workload leading to the hot spot problem [63, 64]. Similar to LEACH, HEED relies on the assumption that CHs can communicate with each other and form a connected graph; realizing this assumption in practical deployments could be tricky due to the previously mentioned communication range restrictions. HEED suffers from a consequent overhead since it needs several iterations to select the CHs.

3.1.3 Energy-Efficient Clustering Scheme (EECS)

EECS [65] is a distributed non-iterative clustering protocol. EECS is a LEACH-like scheme, where the network is partitioned into several clusters, and single-hop communication be-

tween the CH and the BS is performed. EECS extends the LEACH algorithm by dynamic sizing of clusters based on cluster distance from the BS. Unlike LEACH, the CH is elected by localized competition and its no iteration property makes it different from HEED. This competition involves candidates broadcasting their residual energy to neighboring candidates. If a given node does not find a node with more residual energy, it becomes a CH.

In EECS, a node chooses the CH by considering not only saving its energy but also balancing the workload of the CHs. Two distance factors are considered in EECS, the intra-cluster distance, $d(P_j, CH_i)$ and the inter-cluster distance, $d(CH_i, BS)$. A weighted function $cost(j, i)$ is introduced in EECS for the ordinary node P_j to make a decision, which is:

$$cost(j,i) = w \times f(d(P_j, CH_i)) + (1-w) \times g(d(CH_i, BS)) \tag{3.3}$$

Node P_j will join the CH that has the minimum *cost*. In Eq. 3.3, f and g are two normalized functions for the distance $d(P_j, CH_i)$ and $d(CH_i, BS)$, respectively:

$$f(d(P_j, CH_i)) = \frac{d(P_j, CH_i)}{d_{f_{max}}} \tag{3.4}$$

$$g(d(CH_i, BS)) = \frac{d(CH_i, BS) - d_{g_{min}}}{d_{g_{max}} - d_{g_{min}}} \tag{3.5}$$

where $d_{fmax} = \exp(\max d(P_j, CH_i))$, $d_{g_{max}} = \max d(CH_i, BS)$ and $d_{g_{min}} = \min d(CH_i, BS)$. w is a function of P_j as follows:

$$w = c + (1-c) \times \sqrt{\frac{d(CH_i, BS)}{d_{g_{max}} - d_{g_{min}}}} \tag{3.6}$$

Function f in *cost* guarantees that nodes choose the closest CH in order to minimize the intra-cluster communication cost, while function g makes the nodes join the CH with the

smallest $d(CH_i, BS)$ to alleviate the workload of the CHs farther from the BS. Function w is the weighted factor for the tradeoff between f and g. Furthermore, the optimal value of weighted factor c in the function w depends on the specific network scale.

EECS constructs a balancing point between intra-cluster energy consumption and inter-cluster communication load. Clustering in EECS is performed by dynamic sizing based on cluster distance from the BS. This technique addresses the problem that clusters with a larger distance to the BS require more energy for transmission than those with a shorter distance. However, EECS utilizes single-hop communications between the CHs and BS. It also requires more global knowledge about the distances between the CHs and the BS. In EECS, the set of candidate nodes are selected randomly before the competition; this may result in non-optimal CH selection.

3.1.4 Energy Efficient Heterogeneous Clustered (EEHC) Scheme

EEHC [66] was proposed to study the impact of the heterogeneity of nodes in terms of their energy [67] in clustered networks. EEHC assumes the case where a percentage of the population of sensor nodes is equipped with more energy resources than the normal sensor nodes in the network. Three types of sensor nodes equipped with different energy levels are used. Nodes under first level are known as normal nodes, second level nodes are the advanced nodes, and third level nodes are the super nodes. Super nodes have the highest energy among the three types; hence they have the highest chances of selection as a CH. It is assumed that all nodes are distributed uniformly over the network field.

Firstly, EEHC calculates the optimal number of CHs, k_{opt}, based on the size of the sensing field M and the total number of sensor nodes n. EEHC assumes two cases when calculating k_{opt}. The first case assumes that the BS is located in the middle of the sensing area and that the distance of any node to the BS or its CH is $\leq d_0$ (a pre-determined distance threshold). In this case, k_{opt} is calculated as follows:

$$k_{opt} = 0.765 \times \frac{M}{2} \tag{3.7}$$

If the distance of a significant percentage of nodes to the BS is greater than d_0 and given the average distance between a CH and the BS d_{BS}, then k_{opt} is calculated as follows:

$$k_{opt} = \sqrt{\frac{n}{2\pi}} \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \frac{M}{d_{BS}^2} \tag{3.8}$$

The optimal probability of a node to become a CH, p_{opt}, can be computed as follows:

$$p_{opt} = \frac{k_{opt}}{n} \tag{3.9}$$

In the set-up phase, three different kinds of weighted probability formulas are defined for the three kinds of the sensor nodes to elect their own CHs. The election probabilities of CHs are weighted by the initial energy of a node relative to that of other nodes in the network. These probabilities are weighted by the initial energy of a node relative to that of other nodes in the network.

Let m be the fraction of the total number of nodes n, and m_0 the percentage of the total number of super nodes m that are equipped with β times more energy than the normal nodes. The rest $n \times m \times (1 - m_0)$ are the advanced nodes that are equipped with α times more energy than the normal nodes and the remaining $n \times (1 - m)$ are normal nodes. Then the weighted probabilities for normal, advanced and super nodes are, respectively:

$$p_n = \frac{p_{opt}}{1 + m \times (\alpha + m_0 \times \beta)} \tag{3.10}$$

$$p_a = \frac{p_{opt}}{1 + m \times (\alpha + m_0 \times \beta)} \times (1 + \alpha) \tag{3.11}$$

$$p_s = \frac{p_{opt}}{1 + m \times (\alpha + m_0 \times \beta)} \times (1 + \beta) \tag{3.12}$$

The threshold $T(sn)$, which is used to elect the CH in each round, for normal nodes is calculated as follows:

$$T(sn) = \begin{cases} \frac{P_n}{1 - P_n \times (r \mod \frac{1}{P_n})}, if sn \in G \\ 0, Otherwise \end{cases} \tag{3.13}$$

Where r is the current round, G is the set of normal nodes that have not become cluster heads within the last $\frac{1}{P_n}$ rounds. The threshold for advanced and super nodes are calculated in the same way.

Similar to LEACH, after the CHs election ends, the other nodes choose a cluster and join in it according with the RSSI value of the advertisement packet. The CHs take the responsibility to transmit the data packets with a single-hop to the BS.

EEHC is applicable for both homogeneous and heterogeneous WSNs, and it provides a way to calculate the optimal number of CHs based on the network's density. However, if the BS is far from the sensor nodes, calculating k_{opt} usually results in a large number of CHs which will affect the network's energy efficiency. Moreover, it requires more global knowledge about the number of normal, advanced and super nodes, their initial energy and it assumes that the CHs can communicate with the BS directly.

3.1.5 Enhanced Heterogeneous LEACH Protocol for Lifetime Enhancement (EHE-LEACH)

Like EEHC, the Enhanced heterogeneous LEACH protocol for lifetime enhancement of wireless SNs (EHE-LEACH) [68] deals with CH election in heterogeneous networks. There are two main differences. Firstly, EHE-LEACH assumes two level of node heterogeneity

instead of three: nodes under first level are known as normal nodes, at the second level are the advanced nodes. Secondly, a fixed distance based threshold is used by each node to choose between direct communication with the BS or cluster based communication. Sensor nodes that are near the BS send their data directly to the BS and those which are far away from the BS use cluster based communication.

Authors of EHE-LEACH proved that when using the first order energy model, if the distance between the BS and sensor node is sufficiently small then energy consumption is small for direct communication in comparison to cluster-based communication. Therefore, EHE-LEACH introduces a fixed distance based threshold that divides the network field into two partitions. The region near to the BS uses the direct communication approach while the region away from the BS uses the cluster-based approach.

Let m be the fraction of the total number of nodes n, which are equipped with α times more energy than the normal nodes, these nodes are advanced nodes and the remaining $n \times (1 - m)$ are normal nodes. It is also assumed that all nodes are distributed uniformly over the network's field.

Like EEHC, CHs are selected on the basis of weighted probabilities, for the normal and advanced nodes, which can be defined as follows:

$$p_n = \frac{p_{opt}}{1 + \alpha \times m} \tag{3.14}$$

$$p_a = \frac{p_{opt}}{1 + \alpha \times m} \times (1 + \alpha) \tag{3.15}$$

Based on these weighted probabilities, the respective threshold for a normal node is:

$$T(sn) = \begin{cases} \frac{P_n}{1 - P_n \times (r \mod \frac{1}{P_n})}, if sn \in G \\ 0, Otherwise \end{cases} \tag{3.16}$$

The threshold for an advanced node can be calculated in the same way.

EHE-LEACH is applicable for both homogeneous and heterogeneous WSNs. It provides two energy efficient methods for the inter-cluster communication based on the location of the BS. However, each node in EHE-LEACH requires additional global knowledge about the number of normal nodes, advanced nodes, and their initial energy. It relies on the assumption that CHs can communicate with each other and form a connected graph.

3.1.6 Single-hop and Multi-hop Energy-Efficient Clustering Protocols (S-EECP) and (M-EECP)

S-EECP [69] was also proposed to deal with node heterogeneity in WSNs. S-EECP uses the same weighted election probabilities concept as EEHC and the same three levels of node heterogeneity. However, it takes into account the residual energy of nodes in calculating the threshold used to select the CH in each round.

In S-EECP, the weighted probabilities, P_{s_i}, for the nodes are calculated in the same way as EEHC. However, the threshold $T(s_i)$ of each sensor node s_i is calculated as follows:

$$T(s_n) = \begin{cases} \frac{P_{s_i}}{1-P_{s_i}} \times (r \mod \frac{1}{P_{s_i}}) \times \left[\frac{E_i}{E_{avg}} + (rdiv\frac{1}{p_{s_i}})(1 - \frac{E_i}{E_{avg}})\right], if sn \in G \\ 0, Otherwise \end{cases} \tag{3.17}$$

S-EECP is applicable for both homogeneous and heterogeneous WSNs. The CHs are selected based on the ratio between the residual energy of each node and the average energy of the network. However, each node requires additional global knowledge about the number of normal, advanced and super nodes, their initial energy, and their location.

The authors of S-EECP observed that in single-hop communication where data packets are directly transmitted to the BS without any relay nodes, the nodes located far away

from the BS have higher energy consumption because of long range transmission, and these nodes may die out first. They solved this problem in M-EECP by using multi-hop communication to the BS. M-EECP uses a greedy approach to solving the single-source shortest problem to find the shortest path from each CH to the BS. Although S-EECP outperforms EEHC on terms of energy efficiency, the assumption that each node knows all the other nodes' energy level is unrealistic in such a distributed setting. Furthermore, M-EECP suffers from the same problem as S-EECP and assumes that each node knows all other nodes' locations.

3.2 Meta-heuristic Approaches

3.2.1 LEACH-Centralized (LEACH-C)

LEACH-centralized (LEACH-C) [13] is a centralized version of LEACH. Unlike LEACH, where nodes self-configure themselves into clusters, LEACH-C uses the BS for the CHs selection and cluster formations.

Initially, each node sends its information (location and energy level) to the BS, which will use this information and employ a Simulated Annealing (SA) approach to find a predetermined number of CHs and configure the network into clusters. The BS selects the CHs on the basis of location information and residual energy level of all the sensor nodes. To ensure even distribution of load among the sensor nodes, the BS calculates an average energy level and if any nodes' energy level is below that average energy level, then these nodes are not eligible to become CHs.

The clusters are chosen to minimize the amount of energy for the non-CH nodes to transmit their data to the CH, by minimizing the total sum of squared distances between all the non-CH and the closest CH given by:

$$TotalDist = \sum_{n=1}^{N} distance(n, CH_n)^2 \tag{3.18}$$

Where N is the number of nodes in the network and $distance(n, CH_n)$ is the distance from that node to its respective CH.

The steady-state of LEACH-C is identical to that of LEACH [70].

In LEACH-C, the BS utilizes its global knowledge of the network to produce better clusters that require less energy for data transmission. The number of cluster heads in each round of LEACH-C equals a predetermined optimal value. However, LEACH-C shares the same drawback with LEACH in assuming that the CHs can send their aggregated data directly to the BS and it requires GPS or other location tracking methods.

3.2.2 Energy-aware Clustering for WSNs using PSO Algorithm (PSO-C)

PSO-C [71] is a centralized PSO-based clustering protocol that is implemented at the BS. When selecting the optimal set of CHs, PSO-C considers both the residual energy of the sensor nodes and the physical distances between the CHs and their associated cluster members. Authors of PSO-C assume that PSO-C can produce clusters that are evenly positioned throughout the whole network's field.

At the start of each set-up phase, all the sensor nodes send information about their current energy status and locations to the BS. Based on this information, the BS computes the average energy level of all nodes and selects only the sensor nodes with an above average energy level to act as CH candidates. Next, the BS executes the PSO algorithm to determine the best CHs that can minimize the objective function. This objective function tries to minimize both the maximum average Euclidean distance of nodes to their associated CHs and the ratio of the total initial energy of all nodes to the total energy of the CH

candidates. The objective function used to evaluate any particle P_j:

$$Fitness(P_j) = w_1 \times F_1(P_j) + (1 - w_1) \times F_2(P_j), w_1 > 0 \tag{3.19}$$

$$F_1(P_j) = \max_{k=1,2,\ldots,K} \sum_{\forall n_i \in C_{P_j,k}} \frac{d(n_i, CH_{P_j,k})}{|C_{P_j,k}|} \tag{3.20}$$

$$F_2(P_j) = \frac{\sum_{i=1}^{N} E(n_i)}{\sum_{k=1}^{K} E(CH_{P_j,k}} \tag{3.21}$$

Where $F_1(P_j)$ is the maximum average Euclidean distance of the sensor nodes to their respective CHs and $|C_{P_j,k}|$ is the number of nodes that belong to cluster C_k of particle P_j. Function $F_2(P_j)$ is the ratio of total initial energy of all the sensor nodes in the network with the total current energy of the CHs candidates in the current round. w_1 is a user-defined weight used to weight the contribution of each sub-objective. K is the number of clusters.

The objective function, $Fitness_{P_j}$, has the objective of simultaneously minimizing the intra-cluster distance between nodes and their CHs, as quantified by $F_1(P_j)$, and of optimizing the energy efficiency of the network as quantified by $F_2(P_j)$. The objective function $Fitness(P_j)$ was formulated as a minimization function.

Similar to LEACH-C, PSO-C assumes that each CH can send its data directly to the BS.

Authors of [72] showed that PSO-C outperforms GA and K-means-based clustering protocols in terms of convergence time, network energy efficiency and data delivery.

PSO-C takes into consideration the cost of both the inter-cluster communication and the network's energy efficiency. However, PSO-C assumes that the CHs can communicate

with the BS, and it requires a GPS or other location-tracking methods. Moreover, the sub-objectives of Eq. 3.19 are not scaled, hence it is hard to determine the optimal weight coefficient w_1.

3.2.3 Energy Balanced Unequal Clustering Protocol (EBUC)

EBUC [73] is a centralized clustering protocol. Similar to PSO-C, EBUC uses PSO at the BS to find the optimal set of CHs and their associated clusters. EBUC partitions all the nodes into clusters of unequal sizes. The clusters are created such that the ones near the BS have a fewer number of nodes, and so it increases the number of clusters around the BS. The CHs of these clusters can preserve more energy for inter-cluster communication and thus avoid the "hot-spots" problem.

In the first set-up phase, all the sensor nodes send information about their initial energy status and locations to the BS. Based on this information, the BS can estimate the energy level of all nodes in the set-up phase of the following rounds by computing the energy dissipation of the sensor nodes in the last round. Similar to LEACH-C and PSO-C, only the nodes with an above average energy level are eligible to be CH candidates for this round.

The BS uses PSO and defines an objective function to find the best clusters. The objective function takes into consideration minimizing the intra-cluster distance, balancing the energy consumption between the CHs and producing clusters with uneven sizes to balance the energy consumption among the CHs. The objective function used to evaluate any particle P_j is defined as follows:

$$Fitness(P_j) = w_1 \times F_1(P_j) + w_2 \times F_2(P_j) + w_3 \times F_3(P_j) \tag{3.22}$$

$$F_1(P_j) = \max_{k=1,2,...,K} \sum_{\forall n_i \in C_{P_j,k}} \frac{d(n_i, CH_{P_j,k})}{|C_{P_j,k}|} \tag{3.23}$$

$$F_2(P_j) = \frac{\sum_{i=1}^{N} E(n_i)}{\sum_{k=1}^{K} E(CH_{P_j),k}} \tag{3.24}$$

$$F_3(P_j) = \frac{\sum_{i=1}^{K} d(BS, CH_{P_j,k})}{K \times d(BS, NC)} \tag{3.25}$$

Where $F_1(P_j)$ is the maximum average Euclidean distance of the sensor nodes to their respective CHs and $|C_{P_j,k}|$ is the number of nodes that belong to cluster C_k of particle P_j. Function $F_2(P_j)$ is the ratio of the total initial energy of all the sensor nodes in the network with the total current energy of the CH candidates in the current round. Function $F_3(P_j)$ is the ratio of the average Euclidean distance of the CHs to the BS with the Euclidean distance of the network center (NC) to the BS. w_1, w_2 and w_3 are user-defined weights used to weight the contribution of each of the sub-objectives, $w_1 + w_2 + w_3 = 1$.

The objective function $Fitness_{P_j}$ has the objective of simultaneously minimizing the intra-cluster distance between nodes and their CHs, as quantified by $F_1(P_j)$, and of optimizing the energy efficiency of the network as quantified by $F_2(P_j)$; and also of producing clusters with different sizes, as quantified by $F_3(P_j)$. A small value of $F_3(P_j)$ means that there are more CHs in the area closer to the BS, i.e., the size of the clusters located in the area closer to the BS is smaller. The objective function $Fitness(P_j)$ was formulated as a minimization function.

For the inter-cluster communication, EBUC adopts a greedy algorithm to choose a relay node for CH based on the node's residual energy and distance to the BS. Each CH, s_i chooses its relay node rn_i by using a greedy approach. The node rn_i has the least value

of the cost function among all the CHs located between node s_i and the BS. The cost function is defined as:

$$cost(s_i, s_j) = \frac{(d(s_i, s_j))^2 + (d(s_j, BS))^2}{E(s_j)} \tag{3.26}$$

Where $d(s_i, s_j)$ is the distance from node s_i to node s_j, $d(s_j, BS)$ is the distance between node s_j and the BS, and $E(s_j)$ is the residual energy of node s_j.

EBUC provides a method to construct the inter-cluster communication tree and it takes into consideration the cost of both the inter-cluster communication and the intra-cluster communication as well as the network's energy efficiency. However, it assumes that the CHs can communicate with each other regardless of their connectivity, and it requires GPS or other location-tracking methods. Moreover, the sub-objective of Eq. 3.22 are not scaled. Hence, it is hard to determine the optimal weight coefficients.

3.2.4 A Novel Genetic Algorithm in LEACH-C Routing Protocol for Sensor Networks (GA-C)

A genetic algorithm (GA)-based clustering protocol (GA-C) was proposed in [74] to find the optimal set of CHs such that the total network distance is minimized.

In the first set-up phase, all the sensor nodes send information about their residual energy status and locations to the BS. GA-C ensures that only nodes with sufficient energy are selected as CHs. To ensure that, GA-C randomly initializes each chromosome of its population with the IDs of the nodes that have an above average energy level.

The BS uses GA and defines an objective function to find the best clusters. The objective function is defined as the minimization of the total distance from cluster members to their respective CHs in addition to the distance from the CHs to the BS. The objective function used to evaluate any chromosome C_j is defined as follows:

$$Fitness(C_j) = \sum_{k=1}^{K} \sum_{\forall n_i \in C_{C_j,k}} d(n_i, CH_{n_i})^2 + d(CH_{C_j,k}, BS)^2 \tag{3.27}$$

Where K is the number of CHs and $CH_{C_j,k}$ is CH number k in chromosome C_j.

Similar to LEACH-C and PSO-C, GA-C assumes that each CH can send its data directly to the BS.

In GA-C, the BS utilizes its global knowledge of the network to produce better clusters that require less energy for data transmission. However, GA-C assumes that the CHs can communicate with the BS directly and it requires GPS or other location-tracking methods.

3.2.5 A Evolutionary Approach for Load Balanced Clustering Problem for WSN (GA-LBC)

GA-LBC [12] is a centralized GA-based protocol to solve the problem of balancing the load of the CHs. This protocol forms clusters in a way that the maximum load of each CH is minimized. In this protocol, the CHs are determined a priori, and the objective of the protocol is to find the optimal assignments of non-CHs nodes to CHs to form balanced clusters.

The objective function of GA-LBC is constructed on the basis of the standard deviation (σ) of the CH load that gives an even distribution of the load per cluster. If there are m CHs and n sensor nodes, the standard deviation of a CH load is given by:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m} (\mu_{s_j} - W_j)^2}{m}} \tag{3.28}$$

where, μ (average load) $= \frac{\sum_{i=1}^{n} d_i}{m}$, d_i is the load of the sensor node s_j and W_j is the

overall load of the CH g_j. Smaller the standard deviation values produce higher fitness values. Therefore, the objective function to evaluate chromosome C_j was chosen as the reciprocal of the standard deviation of the CH load as given below:

$$Fitness(C_j) = \frac{1}{\sigma_{C_j}} \tag{3.29}$$

Authors of GA-LBC compared the results of applying both GA and DE on the formulated problem. They proved that the GA-based approach achieved faster convergence than the DE-based approach. Another modified DE-based approach was proposed in [10] to solve the same formulated problem.

The GA-LBC objective is to create load-balanced clusters. However, it ignores how the CHs are selected and hence it ignores other network factors like the energy efficiency and the inter-cluster communication method.

3.2.6 Energy-aware Evolutionary Routing Protocol for Dynamic Clustering of WSNs (EAERP)

EAERP [9] is a centralized single-hop clustering protocol where the BS runs an evolutionary-based protocol to optimize the CH election for cluster formation. Each individual of the EAERP population is represented such that it implicitly facilitates the formation of a dynamic number of CHs during the single and throughout the entire rounds of the routing protocol.

The objective function is defined as the minimization of the total dissipated energy in the network, measured as the sum of the total energy dissipated from the non-CHs to send data signals to their CHs, and the total energy spent by CH nodes to aggregate the data signals and send the aggregated signals to the BS. The protocol uses the energy consumption model defined by [13] to compute the energy dissipated during the process

of data transmission and reception. Formally, the objective function used to evaluate individual I_k is defined below:

$$Fitness(I_k) = (\sum_{i=1}^{nc} \sum_{s \in C_i} E_{TX_{s,CH_i}} + E_{RX} + E_{DA}) + \sum_{i=1}^{nc} E_{TX_{CH_i,BS}} \tag{3.30}$$

where nc is the total number of CHs, $s \in C_i$ is a cluster member associated to the i_{th} CH node, $E_{TX_{node1,node2}}$ is the energy dissipated for transmitting data from $node1$ to $node2$. The energy dissipated during the process of transmitting (E_{TX}) and receiving information (E_{RX}) is computed using the first order radio model [13].

After finding the optimal set of CHs, each non-CH determines the cluster to which it belongs by choosing the CH that requires the minimum energy consumption; i.e., the closest CH.

EAERP uses a centralized method that leads to better performance since the BS utilizes its global knowledge of the network to produce better clusters that require less energy for data transmission. However, EAERP assumes that the CHs can communicate with the BS directly and it requires GPS or other location-tracking methods.

3.3 Supplementary Remarks

In addition to the previously mentioned problems, and to the best of our knowledge, all the clustering protocols that were proposed so far use the first order radio model [13] to model the energy consumption of the sensor nodes.

$$E_{TX}(k,d) = \begin{cases} (E_{elec} + \varepsilon_{fs} \times d^2) \times k, d \leq d_0 \\ (E_{elec} + \varepsilon_{mp} \times d^4) \times k, d > d_0 \end{cases} \tag{3.31}$$

$$E_{RX}(k) = E_{elec} \times k \tag{3.32}$$

Where E_{elec} stands for the energy consumption required to run the transmitter or the receiver circuitry. d_0 is the distance threshold. ε_{fs} and ε_{mp} are required energies for amplification of transmitted signals in the open space and the multi-path models respectively.

However, this energy model is very idealized [22, 23] and is fundamentally flawed for modeling radio power consumption in sensor networks. It assumes that all the sensor nodes communicate regardless of the distance between them. Moreover, it ignores the listening energy consumption, which is known to be the largest contributor to expended energy in WSNs.

Another problem is that most of the proposed location-aware or link quality-based clustering protocols assume that each node is equipped with self-locating hardware such as a GPS. Though this is a simple solution, it is considered inefficient and unrealistic for the reasons mentioned previously in Section 2.1.3.

Table 3.1 provides a comparison of the clustering protocols mentioned above with respect to different clustering properties.

3.4 The System Model

In this Section, a detailed explanation of the system model that was used to implement and test the proposed protocols is given. Firstly, we present the WSN model in Section 3.4.1. Then, in Section 3.4.2, we explain the energy consumption model that was used to test the proposed protocols. Section 3.4.3 gives a general overview of the workflow of the proposed protocols. Finally, Section 3.4.4 presents the simulator and the WSN simulation settings that were used to test the proposed protocols.

Table 3.1: Comparison of clustering protocols with respect to clustering attributes

Clustering Protocol	Clustering Method	Clustering Approach	Location Awareness	Number of Cluster Heads	Connectivity to the BS	Network Type	Protocol's Objectives		
							EE [a]	DDR [b]	SC [c]
LEACH	Distributed	Prob [d]./Random	No	Variable	One-hop	Homogeneous	✓	✗	✗
HEED	Distributed	Prob./Energy	No	Variable	Multi-hop	Homogeneous/ Heterogeneous	✓	✓	✗
EECS	Distributed	Prob./Energy	Yes	Variable	One-hop	Homogeneous	✓	✗	✗
EEHC	Distributed	Prob./Energy	No	Fixed (depends on the network density)	One-hop	Homogeneous/ Heterogeneous	✓	✗	✗
EHE-LEACH	Distributed	Prob./Energy	No	Fixed (depends on the network density)	One-hop	Homogeneous/ Heterogeneous	✓	✗	✗
S-EEP	Distributed	Prob./Energy	No	Fixed (depends on the network density)	One-hop	Heterogeneous	✓	✗	✗
M-EEP	Distributed	Prob./Energy	Yes	Fixed (depends on the network density)	One-hop/ Multi-hop	Homogeneous/ Heterogeneous	✓	✓	✗
LEACH-C	Centralized	SA	Yes	Fixed (5% of the network size)	One-hop	Homogeneous	✓	✗	✗
EBUC	Centralized	PSO	Yes	Fixed (5% of the network size)	Multi-hop	Homogeneous	✓	✓	✗
PSO-C	Centralized	PSO	Yes	Fixed (5% of the network size)	One-hop	Homogeneous	✓	✗	✗
GA-C	Centralized	GA	Yes	Fixed (5% of the network size)	One-hop	Homogeneous	✓	✗	✗
GA-LBC	Centralized	GA	Yes	Variable	One-hop	Homogeneous	✓	✗	✗
EAERP	Centralized	EA	Yes	Variable	One-hop	Homogeneous/ Heterogeneous	✓	✗	✗

[a]EE: Energy Efficiency
[b]DDR: Data Delivery Reliability (to the BS)
[c]SC: Scalability
[d]Prob: Probabilistic

3.4.1 The WSN Model

For our model, we consider a two-tiered WSN with N sensor nodes, K cluster heads and one base station. Each sensor node has a unique ID, and the BS ID is 0. In the cluster formation process, each sensor node belongs to only one cluster, and each cluster head node acts as the cluster head of exactly one cluster.

We assume that all nodes are stationary after deployment and that the locations of both the sensor nodes and the cluster heads are unknown. We consider different network densities in our experiments. Furthermore, we consider both homogeneous and heterogeneous network settings.

3.4.2 The Energy Consumption Model

In the proposed protocols, a realistic energy consumption model which is based on the characteristics of the Chipcon CC2420 radio transceiver data sheet [75] is used. The total energy consumed by node i, E_i, is calculated as follows [76]:

$$E_i = \sum_{statej} P_{statej} \times t_{statej} + \sum E_{transitions} \tag{3.33}$$

The index $statej$ refers to the energy states of the sensor: sleep, reception, or transmission. P_{statej} is the power consumed in each $statej$, and t_{statej} is the time spent in the corresponding state. Moreover, the energy spent in transitions between states, $E_{transitions}$, is also added to the node's total energy consumption. The different values of P_{statej} and $E_{transitions}$ can be found in [75].

3.4.3 Overview of the Proposed Protocols

For all the proposed protocols, the network operating time is divided into rounds. Each round consists of two phases, the set-up phase, and the steady-state phase. In the set-up phase, the network is configured. The BS uses an evolutionary algorithm to choose the best set of CHs and to find the optimal configuration of the clusters. The set-up phase consists of the following steps:

1. **Neighbour Discovery**: in this step, each sensor node in the network broadcast a HELLO packet that includes its ID. A sensor node that receives this HELLO packet will update its neighbor table with the ID included in the packet along with the RSSI value in the received packet.

2. **Control Data Broadcasting**: the proposed protocols use the flooding method to transfer the control data to the BS. After the neighbor discovery ends by all the sensor nodes, each node broadcasts the following data about itself: ID, residual energy and its neighbors. A node that receives this packet will rebroadcast it untill it reaches the BS.

3. **Network configuration**: after the BS receives all of the control packets from the network nodes, the BS starts configuring the network. The BS executes the proposed EA-based protocols to find the optimal set of CHs, their associated cluster members, and the inter-cluster communication tree.

4. **Configuration Broadcasting**: after the BS finishes the network configuration, the BS uses flooding again to transfer the configuration to all the nodes. It broadcasts a packet containing that configuration. Each node that receives this packet will modify its status to either a CH, a cluster member or a relay node. A cluster member will update its respective CH and TDMA schedule.

The proposed protocols are explained in more detail in the subsequent chapters.

3.4.4 Simulation Settings

The performance of the proposed protocols was investigated against the well known protocols LEACH, EHE-LEACH, EEHC, the SA-based protocol LEACH-C, the PSO-based protocol PSO-C and the GA-based protocol GA-C. In order to provide a fair comparison, all the competent protocols along with the proposed protocols were implemented under the same WSN simulator.

Simulations were carried on Castalia, which is based on the OMNeT++ platform and can be used to test WSN protocols in realistic wireless channel and radio models [77], with a realistic node behavior. It provides a generic reliable and realistic framework for the first order validation of an algorithm before moving to implementation on a specific sensor platform [78]. The comparisons were used for the purpose of benchmarking the proposed protocols against the well-known protocols cited in the literature.

According to the heterogeneity of the sensors, the simulations were performed on two groups of WSNs ($WSNs\#1, WSNs\#2$), each with 25 different playground topologies. The first case assumes homogeneous sensor networks ($WSNs\#1$) while the second set of experiments ($WSNs\#2$) assumes heterogeneous sensor networks containing advanced nodes forming 10% of the total number of nodes and super nodes also forming 10% of the total number of nodes.

Each WSN group consisted of 5 different network sizes ranging from 100 to 500 sensor nodes. Overall, the simulation results were averaged over five simulation runs for a total of 50 different networks.

The sensor nodes were deployed randomly in an area of $100m \times 100m$ sensor field. The BS was located at the field's corner at position $(0, 0)$. For the medium access control protocol, we used TMAC which is known for its energy efficiency because it adapts a variable sleep schedule that increases the battery utilization [79].

We ran the protocols for $5000s$ and in order to minimize the protocol's overhead, we

Table 3.2: Summery of the WSN simulation settings for the proposed protocols

Parameter	**Value**
BS location	(0,0)
Data transmission rate	1 packet/s
Network Size	(100 - 500) sensor nodes
Field size	$100m \times 100m$
MAC protocol	TMAC
Simulation time	5000 s
Round length	500 s
Slot length	0.4 s
Parameters Settings for $WSN\#1$	
Initial energy	18720 J
Parameters Settings for $WSN\#2$	
Percentage of advanced nodes	10% of Network Size
Percentage of super nodes	10% of Network Size
Initial energy of advanced node	18720 J
Initial energy of super node	12480 J
Initial energy of normal node	6240 J

set the round length to $500s$ with a slot length of $0.4s$. Data packets were generated at a rate of 1 $packet/s$.

In $WSNs\#1$, the initial energy of a standard node is set to 18720 joules, which is the typical energy of two AA batteries [80]. In $WSNs\#2$, the initial energy of a normal node is set to 6240 joules, super node initial energy is set to 12480 joules and advanced node initial energy is set to 18720 joules.

Table 3.2 summarize the configuration of the network's simulation environment.

Chapter 4

Weighted-sum based Optimization Protocols for Clustering in WSN

4.1 Introduction and Motivation

Clustering sensor nodes into groups is an efficient topology control approach in WSNs. The performance of clustering is greatly affected by the selection of Cluster Heads (CHs), which are in charge of creating clusters and controlling member nodes.

The objective of clustering is to search amongst a group of sensor nodes to find a set of nodes that can act as CHs. For a given network topology, it is difficult to find the optimal set of CH nodes. For N sensor nodes and K CHs, there are N^K different combination of solutions. It is straightforward to use the brute-force method for identifying the optimal solution by enumerating all possible combinations. However, the brute-force method has difficulty in solving complex spatial search problems because the solution space is huge. The computational complexity to discover the optimal set of CHs for a large WSN is very high when using a brute force approach [10, 11, 12]. Moreover, finding the set of optimal CHs is a repeated online process that requires quick calculation. Using a brute-force approach

may take days or even months if the search space is big.

Illustration 5.1: In case of a network that has 500 sensor nodes and 25 CHs, there are approximately $500^{25} = 2.98 \times 10^{67}$ different possible solutions for just one round of operation. Enumerating all possible solutions may take days or even months. This is unacceptable in a repeated process such as clustering which is performed in rounds, and each round could take minutes or even seconds.

The clustering problem in WSN has been proved to be a Non-deterministic Polynomial (NP)-hard optimization problem [8, 9, 10, 11, 12, 13]. Solutions to NP-hard problems involve searching through vast spaces of possible solutions. Evolutionary computation approaches have been applied successfully to a variety of problems of that kind.

In this chapter, the problem of CHs selection in WSN is formulated as a single-objective optimization problem. A centralized weighted-sum multi-objective optimization approach is adapted to find the optimal set of CHs. The proposed approach finds a predetermined number of CHs in such a way that they form one-hop clusters. The goal of the proposed approach is to enhance the network's energy efficiency, data delivery reliability and scalability. The formulated problem has been solved using three evolutionary approaches: Genetic Algorithms (GA), Differential Evolution (DE) and Particle Swarm Optimization (PSO). The performance of the three approaches is assessed with respect to the achieved fitness value. Based on the performance assessment results, the best evolutionary algorithm approach is used to evaluate and compare the performance of the proposed protocol against well-known clustering protocols.

Furthermore, in order to study the effect of minimizing the number of CHs on the network's energy efficiency, a hierarchical clustering approach that forms two-hop clusters is proposed. The proposed approach objective is to enhance the network energy efficiency by setting an upper bound on the number of CHs and minimizing the number of CHs compared to that upper bound. Furthermore, it improves the network scalability by using

two-hop communication between the sensor nodes and their respective CHs.

The remainder of this chapter is organized as follows. Section 4.2 describes the multi-objective optimization approach that was adopted to solve the formulated problem. The first proposed protocol, the one-hop clustering protocol, is described in details in Section 4.3, including the experimental results for assessing its performance. The second proposed protocol, the hierarchical clustering protocol, is described in details in Section 4.4, including the experimental results for assessing its performance. Finally, Section 4.5 concludes the chapter.

4.2 Weighted-sum Approach for Multi-objective Optimization

In this chapter, the weighted-sum approach (WSA) was adopted for the construction of the multi-objective fitness function in both protocols. This approach is computationally efficient and straightforward to implement [81, 82, 83] which makes it suitable to apply in WSN.

Since three different EAs will be used in this section, each candidate solution of the population will be referred to as *individual*.

Mathematically, the final objective function for Individual I_i of the population, using the weighted-sum approach, can be expressed as follows:

$$F_{I_i} = \sum_{m=1}^{M} w_m \times F_m(I_i) \tag{4.1}$$

where w_m is a weight coefficient that specifies the contribution of sub-objective F_m in the main objective function F_{I_i}. M is the total number of sub-objective functions.

However, it can be very difficult to precisely and accurately select the final objective

function weights, even for someone familiar with the problem domain [81]. In order to avoid this drawback, each sub-objective $F_i \in F_{I_i}$ is scaled to produce results in a set of values in the range $[0.0, 1.0]$, using the following scaling function:

$$\frac{F_j^{max} - F_j}{F_j^{max} - F_j^{min}} \tag{4.2}$$

where F_j^{max} is the maximum value for function F_j and F_j^{min} is the minimum value for function F_j. Applying this scaling function on every sub-objective $F_m(I_i)$ will result in the scaled value $sF_m(I_i)$. Then, the final objective function to be minimized, assuming each sub-objective is equally important, is expressed as follows:

$$F_{I_i} = \sum_{m=1}^{M} sF_m(I_i) \tag{4.3}$$

4.3 One-hop Clustering Protocol

In this section, a clustering approach that results in forming one-hop clusters is proposed. In this approach, the BS selects a predetermined number of CHs. The objective is to maximize the network energy efficiency, data delivery reliability and scalability. After choosing the optimal set of CHs by the BS, the cluster formation process results in one-hop clusters where each cluster member sends its data directly to its respective CH using a one-hop communication link.

Based on the information at the BS, the BS computes the average energy level of all nodes. Only nodes with an above average energy level are eligible to become CH candidates for this round to ensure that only nodes with sufficient energy are selected as CHs. Then, the BS uses an EA approach to determine the best K CHs. The average energy for all the nodes is computed as follows:

$$AvgEnergy = \frac{\sum_{n=1}^{N} E(n)}{N} \tag{4.4}$$

where N is the number of sensors in the network and $E(n)$ is the residual energy remaining in sensor node number n.

4.3.1 Individual Initialization

The dimension of each individual in the population is same as the number of CH nodes (i.e., K) in the network. Let, $I_i = [X_{i,1}, X_{i,2}, X_{i,3}, ..., X_{i,K}]$ be the ith individual of the population where each component, $X_{i,d}, 1 \leq d \leq K$ denotes CH number d in individual number i. Each component is initialized with a randomly generated number in the range $[1, networksize - 1]$ based on a uniform distribution.

It should be noted that the random initialization and the velocity update by (2.6a) produce non-integer velocity values, which are converted to the nearest integer. In the case, that an individual solution generates duplicate ID's after position update, it is assigned a high penalty value to ensure that the protocol generates the specified predetermined number of CHs.

Illustration 5.2: Consider a WSN with 60 sensor nodes and the number of CHs is 3 ($5\% \times 60$). Therefore, the dimension of each individual in the population is same as the number of CHs, i.e. $K = 3$.

Now, for each $X_{i,d}, 1 \leq d \leq 3$ of individual I_i, a random number is generated to initialize it. Let us assume that an individual $I_i = [31.2, 20.8, 9.4]$, has been randomly generated. The second component of this individual is $X_{i,d} = 20.8$ then the 2nd elected CH ID $= \lfloor 20.8 \rceil = 21$. Hence, the CH candidates IDs that result from this particle are $31, 21$ and 9.

Now, let's consider another individual $I_j = [31.2, 31.4, 9.4]$. The CHs candidates generated are $31, 31$ and 9. Since there are duplicate values in the generated CHs, this particle is assigned a high penalty value to exclude it from further consideration.

Optimal Number of Cluster Heads

Several clustering protocols were proposed in the literature most of which used a fixed number of CHs. The authors in [60, 13] argued that the optimal number of CHs equals 5% of the number of the network nodes. Based on those results, many clustering protocols also used 5% as their ideal setting for the number of CHs. In the proposed one-hop clustering protocol, the percentage of CHs is set to 5% of the total nodes similar to the common protocols.

Cluster Formation

After electing a set of CHs, the clusters are formed by associating each node with exactly one cluster head, based on the RSSI value for the links between the cluster members and their respective CH. The communication link between a sensor node and its respective CH is one-hop.

4.3.2 Individual Evaluations

The next step after initializing each individual in the population, is evaluating them according to an objective function. This helps to periodically converge towards the optimal solution. The optimal set of CHs are selected such that they minimize the cost of the objective function. The goal of that function is to optimize the combined effect of the following WSN properties: energy efficiency, data delivery reliability and scalability.

Energy Efficiency

The residual energy of a sensor node can be a criterion for selecting the best CHs since a node with a better battery life is a better candidate for cluster management and data aggregation. In addition, the consumed energy is distributed among all the sensor nodes. The BS uses the following function to calculate the fitness of individual I_i in terms of energy efficiency:

$$EE_{I_i} = \sum_{k=1}^{K} \frac{initialE(CH_{I_i,k})}{E(CH_{I_i,k})} \tag{4.5}$$

K is the total number of cluster head candidates. $initialE(CH_{I_i,k})$ is the initial energy of CH number k in individual I_i. $E(CH_{I_i,k})$ is the remaining energy for that CH.

Data Delivery Reliability

The aim of this sub-objective is to create clusters such that the link quality between the cluster members and their respective CHs is maximized. This, in turn, will enhance the Packet Delivery Rate (PDR) and hence maximize the data delivery reliability.

Let $RSSI(m, CH_{I_i,k})$ indicate the RSSI value for the link from cluster member m and cluster head number k in individual I_i. Then, the link quality for that link, $LQ_{(m,CH_{I_i,k})}$, can be calculated using:

$$LQ_{(m,CH_{I_i,k})} = \frac{RSSI(m, CH_{I_i,k})}{minRSSI} \tag{4.6}$$

Higher values of LQ indicate worse link quality. To maximize the cluster quality in terms of link quality, the worst cluster quality needs to be minimized. Hence, the following sub-objective needs to be minimized:

$$CQ_{I_i} = \max_{k=1,2,\ldots,K} \frac{\sum_{\forall m \in C_{I_i,k}} LQ_{(m,CH_{I_i,k})}}{|C_{I_i,k}|} \tag{4.7}$$

$minRSSI$ is the worst RSSI value among all communicating pairs and is set to -100. $|C_{I_i,k}|$ is the number of members in cluster k of individual I_i.

Scalability

In order to increase the scalability of the proposed protocol, the number of the clustered nodes should be maximized. To accomplish that, the proposed protocol reduces the number of un-clustered nodes and increases the number of clustered nodes. That can be realized by minimizing the following sub-objective:

$$SC_{I_i} = N - \sum_{k=1}^{K} |C_{I_i,k}| \tag{4.8}$$

where N is the total number of sensor nodes in the network.

After calculating the sub-objectives EE_{I_i}, CQ_{I_i} and SC_{I_i} , they are scaled using Eq. (4.2) to result in the following sub-objectives values sEE_{I_i}, sCQ_{I_i} and sSC_{I_i} respectively. Then, the final objective function $FinalObj_{I_i}$, that needs to be minimized is calculated using:

$$FinalObj_{I_i} = sEE_{I_i} + (1 - sCQ_{I_i}) + sSC_{I_i} \tag{4.9}$$

After a pre-specified number of iterations, the individual with the best fitness (minimum objective value) is considered the optimal solution. The BS then finishes the network configuration by broadcasting a packet that containing the CHs, associated clusters, and each node's TDMA schedule. Each node that receives that packet will modify its status to either CH or CM. A CM node will update its respective CH and TDMA schedule. A node that is not CM or CH is set to sleep to save its energy.

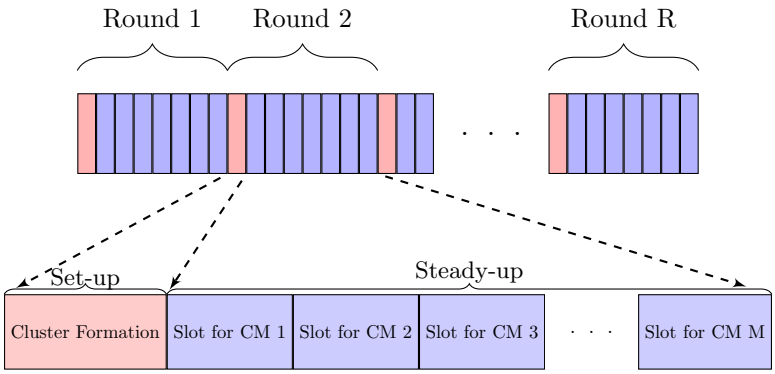


Figure 4.1: Schedule of set-up and steady-state phases in a given round, in the proposed one-hop clustering approach

4.3.3 The Steady-state Phase

In the steady-state phase, each non-CH node uses its TDMA schedule to transmit its data to its respective CH. When a CM node finishes its data transmission slot, it enters the sleep state to save its energy. Fig. 4.1 shows the schedule for the set-up and steady-state phases in a given round in the proposed approach.

4.3.4 Experimental Results

In this section, the results of the experiments employed to evaluate the proposed approach are presented. The goal of the experiments was to:

- Evaluate the performance of applying GA, DE and PSO on the formulated CH selection problem.
- Evaluate the performance of the proposed approach to the well-known clustering approaches LEACH, EHE-LEACH, EEHC, LEACH-C, PSO-C, and GA-C.

The formulated problem has been solved using three EAs: PSO, GA, and DE. These algorithms were applied to one random round of $WSN\#2$, and their performance has been compared in terms of the achieved fitness value.

Parameter	Value
Network Size	[100 - 500]
Problem dimension (Number of CHs)	[5 - 25]
Population size	50
Number of iterations	500
GA	
Tournament size	2
Mutation probability, p_m	1 / Problem dimension
Crossover probability, p_c	0.9
Mutation distribution index, η_m	20
Crossover distribution index, η_n	20
DE	
CR parameter	0.5
F parameter	0.5
Mutation scheme	$DE/rand/1$
PSO	
Learning Factor $c1$	2
Learning Factor $c2$	2
Interia weight w	0.9

Table 4.1: The evolutionary algorithms parameters settings for the proposed one-hop clustering protocol

It should be noted that to solve problems of increasing dimension, it is necessary to increase the population's size and to run additional iterations. However, it is very difficult to predict the population's size and the number of evaluations required to solve a problem of known dimension [84]. Besides, it is of minor importance to tune this parameter based on the problem at hand [85]. Authors in [84, 85] established that a swarm size of 50 is a good choice for PSO if the problem size is above 50. Full analysis and determination of the optimal population size is beyond the scope of this thesis. In this chapter, the population size is set to 50, for all the three EAs. The number of iterations is set to 500. Table 4.1 summarizes the configuration of the EAs parameters.

Table 4.2 includes the mean, and standard deviation of 50 independent runs carried out for the fitness value achieved by the three algorithms, DE, GA, and PSO. In Table 4.2, some cells have two different levels of gray: a darker one, showing the algorithm obtaining

Network Size	PSO	GA	DE
100	$7.21e-01_{5.6e-03}$	$7.27e-01_{5.8e-03}$	$7.36e-01_{4.3e-03}$
200	$6.79e-01_{4.1e-03}$	$6.88e-01_{7.2e-03}$	$6.93e-01_{7.0e-03}$
300	$6.44e-01_{1.1e-02}$	$6.55e-01_{5.3e-03}$	$6.64e-01_{5.4e-03}$
400	$6.19e-01_{5.0e-03}$	$6.28e-01_{6.2e-03}$	$6.41e-01_{4.0e-03}$
500	$5.97e-01_{6.7e-03}$	$6.10e-01_{7.1e-03}$	$6.27e-01_{6.8e-03}$

Table 4.2: Mean and Standard Deviation for the fitness values for the proposed one-hop clustering protocol

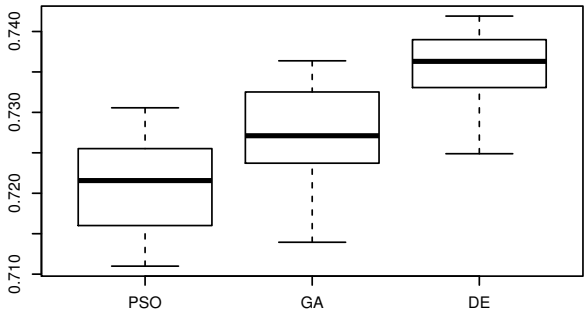
the best fitness value, and a lighter one, highlighting the algorithm obtaining the second best fitness value. The boxplots representing the distribution of fitness function values in the comparison carried out are showed in Fig. 4.2, for different network sizes.

Since the problem was formulated as a minimization problem, the algorithm that achieves the minimum fitness value is considered the best algorithm. Table 4.2 and 4.2 show that, for all the network sizes, PSO obtains the best average fitness value, and GA has obtained the second best average fitness value.

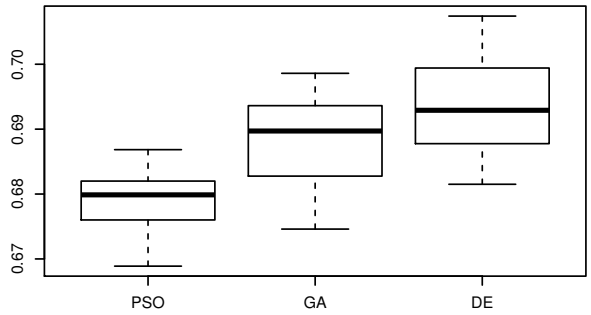
As PSO has proved to have the best average performance among the other EA approaches. The performance of the proposed PSO-based one-hop clustering approach, PSO-OC, is evaluated and compared against the well known protocols LEACH, EHE-LEACH, EEHC, the SA-based protocol LEACH-C, the PSO-based protocol PSO-C and the GA-based protocol GA-C, in terms of the number of un-clustered nodes, the average energy consumed by nodes, and the Packet Delivery Rate (PDR). The values of the PSO parameters are the same values as in Table 4.1.

Fig.4.3 and Fig.4.4 record the average number of un-clustered nodes per round in $WSN\#1$ and $WSN\#2$ respectively. The results presented here represent the average of 5 different runs, for each network size, with a confidence level of 0.99.

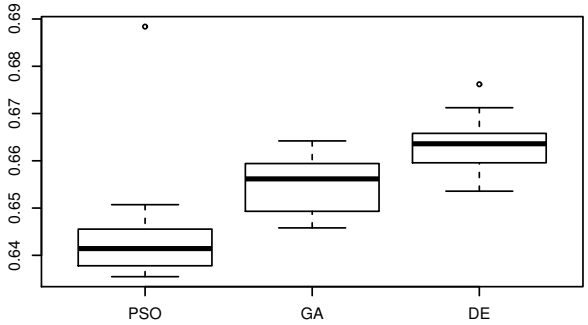
It can be observed from Figs. 4.3 and 4.4 that PSO-OC shows better scalability than the other competent clustering protocols. This is due to the clustering phase of PSO-OC which takes care of minimizing the number of un-clustered nodes as justified by (4.8);



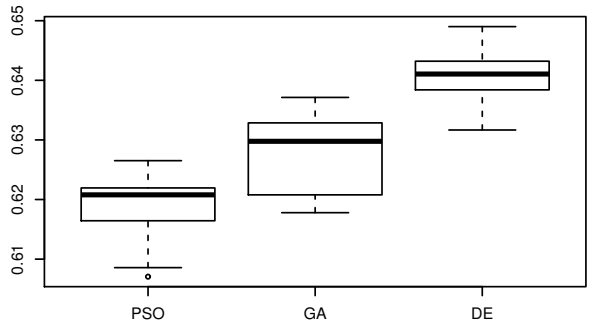
(a) Fitness values for 100 sensor nodes, i.e. number of CHs = 5



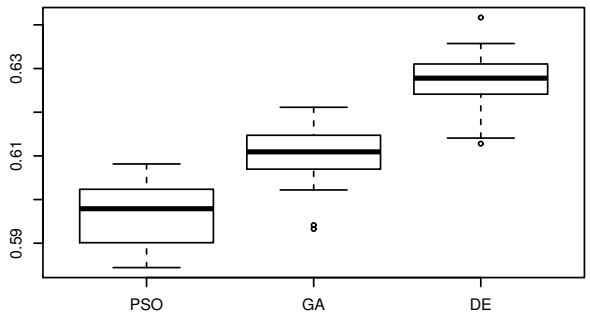
(b) Fitness values for 200 sensor nodes, i.e. number of CHs = 10



(c) Fitness values for 300 sensor nodes, i.e. number of CHs = 15



(d) Fitness values for 400 sensor nodes, i.e. number of CHs = 20



(e) Fitness values for 500 sensor nodes, i.e. number of CHs = 25

Figure 4.2: Comparison of GA, DE and PSO for different network sizes [100 - 500]

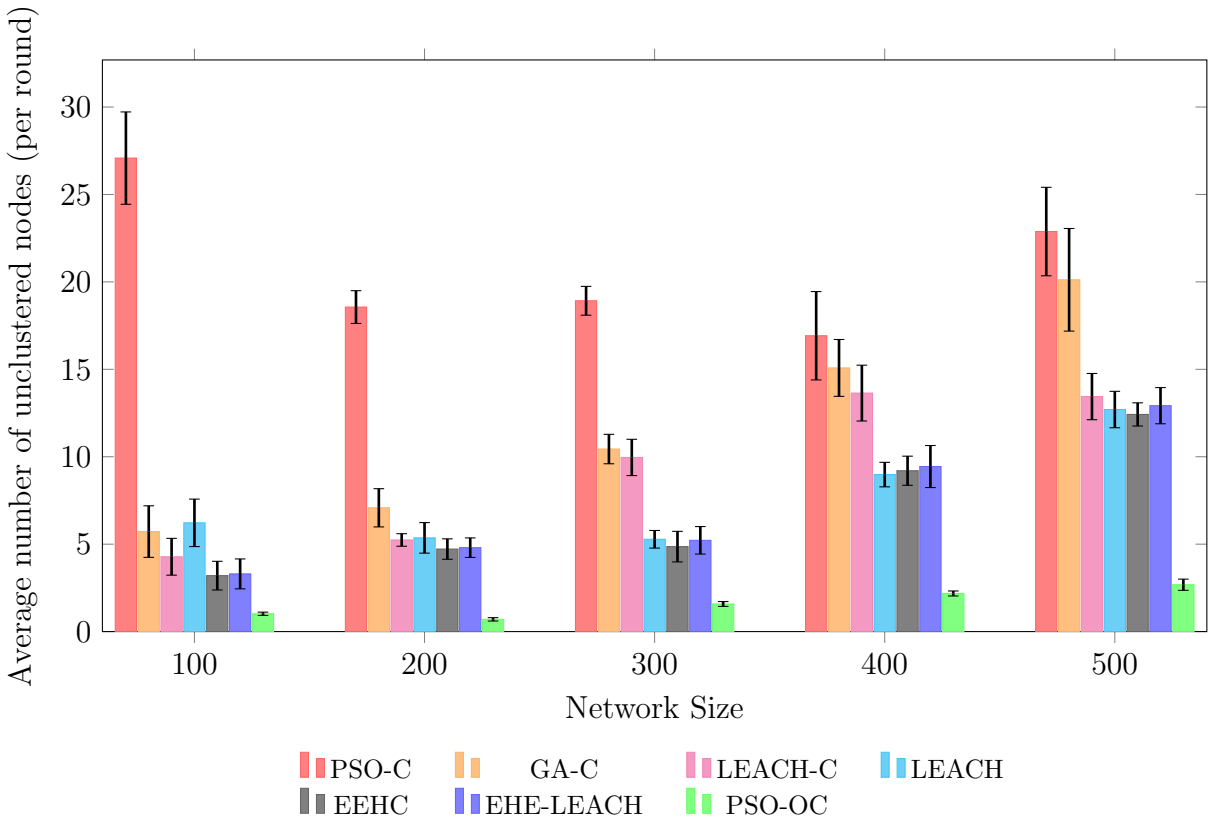


Figure 4.3: Average number of unclustered nodes per round for $WSN\#1$ for PSO-OC

whereas the existing protocols do not deal with that problem.

In order to judge the cluster-based link quality of PSO-OC, the average (mean) PDR for packets received by all the CHs for 5 different runs, for each network size, along with their standard deviations (SD) for both scenarios $WSN\#1$ and $WSN\#2$ are calculated by varying the number of sensor nodes. The results are shown in Tables 4.3 and 4.4 for $WSN\#1$ and $WSN\#2$ respectively. It is clear that the average PDR for PSO-OC is at a maximum with minimum fluctuations in the average PDR. However, GA-C and LEACH-C outperform PSO-C in terms of PDR at the CHs. Also note that LEACH, EHE-LEACH and EEHC protocols have much lower PDR because no link quality measure is taken in either of them. Furthermore, they have more fluctuations around the average due to their probabilistic nature.

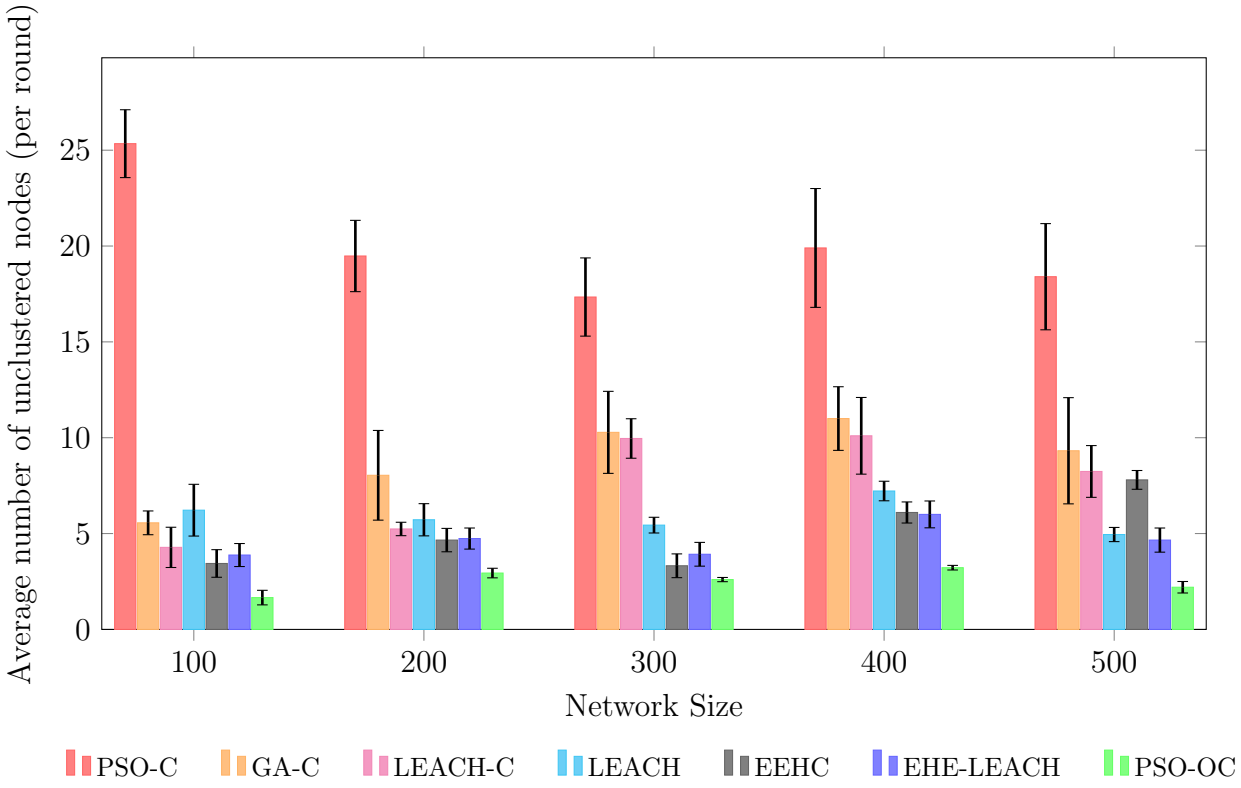


Figure 4.4: Average number of unclustered nodes per round for $WSN\#2$ for PSO-OC

Table 4.3: Mean PDR and standard deviation in $WSN\#1$ for PSO-OC

Protocols	100 Sensor nodes		200 Sensor nodes		300 Sensor nodes		400 Sensor nodes		500 Sensor nodes	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LEACH	0.536	0.024	0.621	0.032	0.674	0.020	0.671	0.017	0.671	0.012
EHE-LEACH	0.572	0.017	0.622	0.023	0.643	0.020	0.634	0.016	0.630	0.010
EEHC	0.589	0.032	0.661	0.037	0.680	0.006	0.669	0.009	0.670	0.015
PSO-C	0.606	0.045	0.775	0.010	0.809	0.012	0.832	0.005	0.835	0.004
GA-C	0.865	0.008	0.873	0.005	0.871	0.003	0.862	0.002	0.861	0.003
LEACH-C	0.861	0.006	0.892	0.001	0.890	0.002	0.890	0.002	0.887	0.002
PSO-OC	**0.877**	0.008	**0.893**	0.002	**0.895**	0.001	**0.891**	0.002	**0.892**	0.001

Table 4.4: Mean PDR and standard deviation in $WSN\#2$ for PSO-OC

Protocols	100 Sensor nodes		200 Sensor nodes		300 Sensor nodes		400 Sensor nodes		500 Sensor nodes	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
LEACH	0.536	0.024	0.617	0.031	0.667	0.015	0.672	0.013	0.680	0.036
EHE-LEACH	0.577	0.042	0.635	0.035	0.625	0.062	0.605	0.016	0.580	0.016
EEHC	0.588	0.029	0.651	0.016	0.577	0.058	0.669	0.020	0.639	0.031
PSO-C	0.615	0.029	0.787	0.016	0.823	0.013	0.826	0.036	0.838	0.020
GA-C	0.861	0.015	0.871	0.006	0.867	0.001	0.866	0.003	0.863	0.013
LEACH-C	0.827	0.035	0.865	0.015	0.871	0.004	0.861	0.021	0.867	0.018
PSO-OC	**0.890**	0.008	**0.893**	0.002	**0.895**	0.001	**0.890**	0.006	**0.895**	0.010

Tables 4.5 and 4.6 show the average (mean) of the average energy consumed by nodes (in joules) for 5 random runs along with their standard deviations (SD) for both scenarios $WSN\#1$ and $WSN\#2$. It is clearly shown that PSO-OC has lower energy consumption compared to the other competing protocols. Furthermore, much higher energy consumption is recorded in LEACH, EHE-LEACH and EEHC. This is the result of the un-clustered nodes which are left unattended without any sleeping schedule. Hence, they are consuming energy even if their nodes are in the idle state. Theoretically, LEACH-C, GA-C and PSO-C cluster all the network nodes and thus give each node a sleep schedule depending on its TDMA turn to transmit. This caused both protocols to have lower energy consumption compared to that of LEACH type protocols. For PSO-OC, any un-clustered node is set to sleep during the whole round. Although this should reduce its energy consumption compared to the other protocols, this is not reflected because the number of un-clustered nodes is already at a minimum in PSO-OC.

4.4 Hierarchical Clustering Protocol

The number of CHs is one of the key factors influencing the performance of clustering in WSNs. Since the CHs are constantly active during the entire round, minimizing the number of CHs will in turn enhance the network's energy efficiency.

PSO has proved to have better performance that GA and DE, as illustrated in the last section. Moreover, PSO has many advantages over other alternatives optimization techniques like GA, which has very high processing demands [86]. PSO advantages include ease of implementation on hardware or software and high-quality solutions because of its ability to escape from local optima and quick convergence [87, 88]. Clustering is a repeated process; therefore, the simpler the optimization algorithm, the better the network's efficiency. That is another reason why PSO is a popular choice for WSN clustering.

In this section, a centralized PSO-based approach for Hierarchical Clustering, PSO-HC,

Table 4.5: Mean for average consumed energy per node and standard deviation in $WSN\#1$ for PSO-OC

Protocols	**100 Sensor nodes**		**200 Sensor nodes**		**300 Sensor nodes**		**400 Sensor nodes**		**500 Sensor nodes**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	175.19	6.545	149.22	10.11	131.07	5.712	132.37	5.500	131.74	4.400
EHE-LEACH	155.19	9.007	140.35	6.245	131.76	4.558	131.64	3.839	130.32	3.571
EEHC	158.85	9.001	137.48	10.35	131.28	1.539	131.32	3.711	130.57	4.673
PSO-C	73.855	0.042	72.208	0.061	71.593	0.028	71.303	0.085	71.102	0.027
GA-C	74.499	0.074	72.660	0.305	71.824	0.304	71.602	0.121	71.336	0.386
LEACH-C	74.549	0.003	73.060	0.004	72.559	0.005	72.308	0.013	72.161	0.006
PSO-OC	**71.271**	0.217	**71.254**	0.175	**71.142**	0.222	**71.129**	0.115	**71.203**	0.086

Table 4.6: Mean for average consumed energy per node and standard deviation in $WSN\#2$ for PSO-OC

Protocols	**100 Sensor nodes**		**200 Sensor nodes**		**300 Sensor nodes**		**400 Sensor nodes**		**500 Sensor nodes**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	175.19	6.545	150.53	10.06	133.62	3.629	130.58	7.248	128.43	5.721
EHE-LEACH	155.68	11.27	137.22	11.34	145.00	10.33	135.09	11.20	145.13	10.62
EEHC	158.98	7.263	140.80	5.602	148.40	9.181	129.70	9.201	139.55	8.120
PSO-C	73.817	0.303	72.086	0.069	71.379	0.222	73.761	0.015	73.529	0.083
GA-C	74.528	0.035	72.752	0.277	71.979	0.249	71.491	0.032	71.357	0.021
LEACH-C	74.549	0.003	73.060	0.004	72.559	0.005	72.311	0.001	72.151	0.002
PSO-OC	**71.367**	0.133	**71.188**	0.144	**71.196**	0.168	**70.986**	0.116	**71.227**	0.192

in WSNs is proposed. The motivation behind proposing this approach is to examine the effect of minimizing the number of CHs on the network energy efficiency. The proposed approach objective is to enhance the network energy efficiency by setting an upper bound on the number of CHs and minimize the number of CHs compared to that upper bound. Furthermore, it improves the network scalability by using two-hop communication between the sensor nodes and their respective CHs.

4.4.1 Particle Initialization

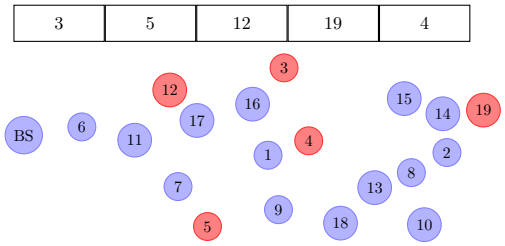
In PSO-HC, each particle's position vector that represents the CH nodes' IDs is initialized with random integer values in the range $[1, networksize - 1]$ where node ID 0 represents the BS. Only nodes with an above average energy level are eligible to be CH candidates for this round to ensure that only nodes with sufficient energy are selected as CHs.

The particle size is equal to the upper bound on the number of CH candidates. It should be noted that the velocity update by (2.6a) gives non-integer velocity values, which are converted to the nearest integer. In the case that a particle generates duplicate IDs during initialization or after position update, the generated unique IDs are used as CH candidates.

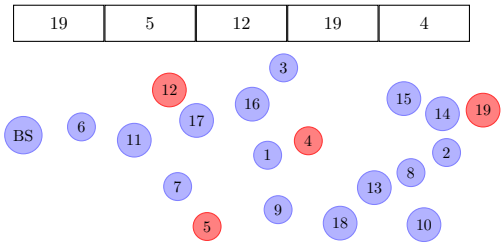
Illustration 5.3: Consider a WSN with 60 sensor nodes and 3 CHs ($5\% \times 60$). The dimension of each individual in the population is same as the number of CHs, i.e. $K = 3$.

Now, let's consider another particle $P_j = [31.2, 31.4, 9.4]$. The CHs candidates generated from this particle are $\lfloor 31.2 \rceil = 31, \lfloor 31.4 \rceil = 31$ and $\lfloor 9.4 \rceil = 9$. Since there is duplication in the generated CHs, the CH candidates IDs that result from this particle are 31 and 9.

Fig. 4.5 shows an example of two different particles that have an upper bound on the number of CHs equal to 5, and the CH candidates generated from them.



(a) A particle position vector that has no duplicates; number of CHs = 5



(b) A particle position vector that has duplicates; number of CHs = 4

Figure 4.5: Example of two different particles and their respective CH candidates, upper bound = 5, red nodes are CHs

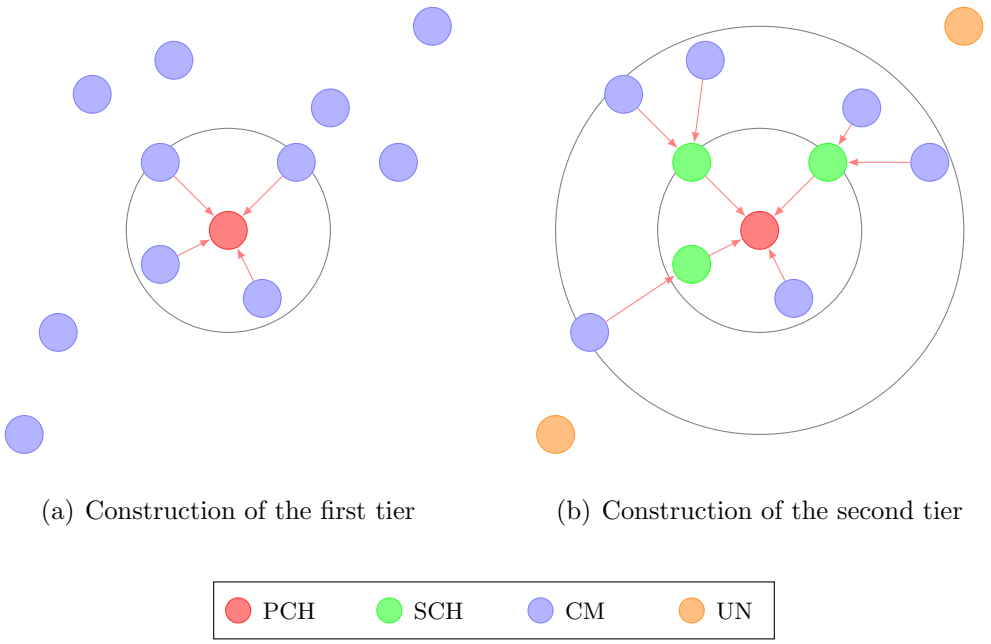
Cluster Formation

The cluster formation phase is done at the BS after generating the CHs from the particle. We aim at designing two-tier clusters with the objective of improving the network scalability and minimizing the number of active CHs during each round.

The BS constructs the first tier clusters by assigning each non-CH node to a CH according to the RSSI value for the link between them. In the case of multiple CHs, the node will become a member of the CH having the maximum RSSI strength. Any CH in the first tier is called Primary CH and has to stay active during the entire round without any sleep schedule.

The second tier is constructed by the BS by clustering all the non-CHs nodes that remained non-clustered from the first tier. The BS assigns each non-CH node in the second tier to a node in the first tier according to the RSSI value for the link between them. A node in the first tier that has members from the second tier is called Secondary CH (SCH) and does not need to be active during the entire round and is set to sleep after it transmits both its own data packets and its members' data packets.

At the end of the cluster formation process, a node in the network can either be a PCH, SCH, a cluster member (CM) or un-clustered node (UN). The cluster formation process is



(a) Construction of the first tier (b) Construction of the second tier

Figure 4.6: Cluster Formation Process for PSO-HC

illustrated in Fig. 4.6.

TDMA Scheduling

The BS creates a schedule based on TDMA to allocate time slots for the cluster members according to the cluster size.

Each CM, whether it is in the first tier or the second tier, is assigned a unique TDMA turn and slot. To illustrate this, Fig. 4.7 depicts an example of TDMA assignment on a given cluster. In this example, each node is labeled in the form Node ID/TDMA turn. Node A is the first node to transmit its data. After its finishes its time slot, it goes into sleep mode to save its energy. The next node to transmit its data is B. Since B is an SCH, it has to wait in RX (Receive) mode till node C transmits its data. After C transmits its data, it goes to sleep mode then node B goes to sleep mode since it has no more members. This process continues until the last node, I, transmits its data and goes to sleep mode. The process is repeated from node A till the round ends.

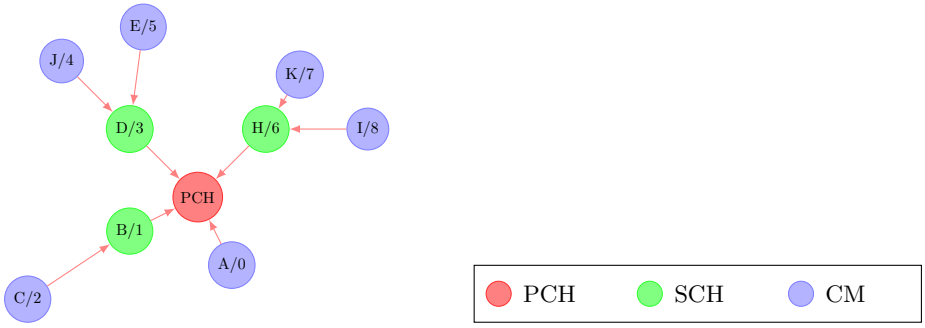


Figure 4.7: TDMA Scheduling in PSO-HC

4.4.2 Particle Evaluations

The best CHs are selected in a way to optimize the combined effect of the following properties: the network's energy efficiency, the network's data delivery reliability and the protocol's scalability. The proposed protocol enhances scalability by using two-hop communication between the sensor nodes and their respective CHs.

Energy Efficiency

To achieve an energy efficient clustering protocol, a smaller number of CHs need to be active during each round. To achieve that, the proposed protocol needs to minimize the number of CHs.

Let S_{P_i} denote the size of particle P_i and vector V_{P_i} denote the vector that represents the CHs generated from decoding particle P_i, after removing duplicate values. Then, minimizing the number of elected CHs is given by the following sub-objective:

$$EE_{P_i} = \frac{|V_{P_i}|}{S_{P_i}} \tag{4.10}$$

Data Delivery Reliability

Let $LQ_{P_i}(m \rightarrow nextHop)$ be an indicator of the link quality between cluster member m of a cluster generated from particle P_i and its next hop (which could be PCH or SCH). It can be calculated using:

$$LQ_{P_i}(m \rightarrow nextHop) = \frac{RSSI(m \rightarrow nextHop)}{minRSSI} \tag{4.11}$$

Now, let t refer to tier number t. In the proposed protocol, the maximum value of t, T, is 2. If $T = 1$, this means that the whole network nodes were clustered using one tier only.

To maximize the cluster quality in terms of the data delivery reliability, the following sub-objective represents the sum of the worst LQ among all the tiers:

$$CQ_{P_i} = \sum_{t=1}^{T} \max_{\forall m \in tier_t} LQ_{P_i}(m \rightarrow nextHop) \tag{4.12}$$

Scalability

To increase the protocol's scalability, it should cluster as much nodes as possible, regardless of the number of the CHs and the network size. The following function represents the number of un-clustered nodes for particle P_i:

$$UN_{P_i} = N - \sum_{k=1}^{|V_i|} |C_{P_i}, k| \tag{4.13}$$

N is the total number of nodes in the network. $|C_{P_i}, k|$ is the number of cluster members in the cluster that corresponds to CH number k generated from decoding particle P_i.

After calculating EE_{P_i}, CQ_{P_i} and UN_{P_i}, they are scaled using Eq. 4.2 to produce sEE_{P_i}, sCQ_{P_i} and sUN_{P_i}. The final objective function that needs to be minimized is:

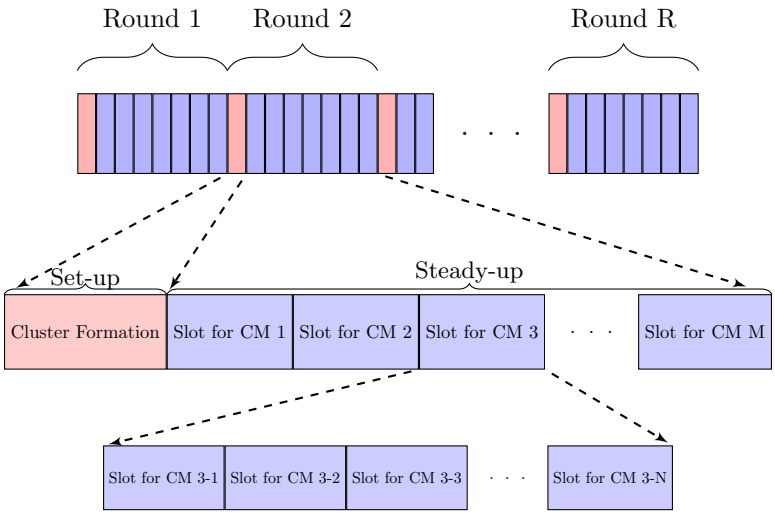


Figure 4.8: Schedule of set-up and steady-state phases in a given round, in PSO-HC

$$Fitness_{P_i} = sEE_{P_i} + (1 - sCQ_{P_i}) + sUN_{P_i} \tag{4.14}$$

After a pre-specified number of iterations, the particle with the best fitness (minimum objective value) is considered the optimal solution.

4.4.3 The Steady-state Phase

The BS then finishes the network configuration by broadcasting a packet containing the PCHs, SCHs, associated clusters and each node's TDMA schedule. Each node that receives that packet will modify its status to either CH or CM. A CM node will update its respective CH and TDMA schedule. A node that is not CM or CH is set to sleep to save its energy.

In the steady-state phase, each member node uses its TDMA schedule to transmit its data to its next hop. When a non-CH node (a CM or an SCH) finishes its data transmission slot, it enters the sleep state to save its energy. Fig. 4.8 shows the schedule of set-up and steady-state phases in a given round, in PSO-HC.

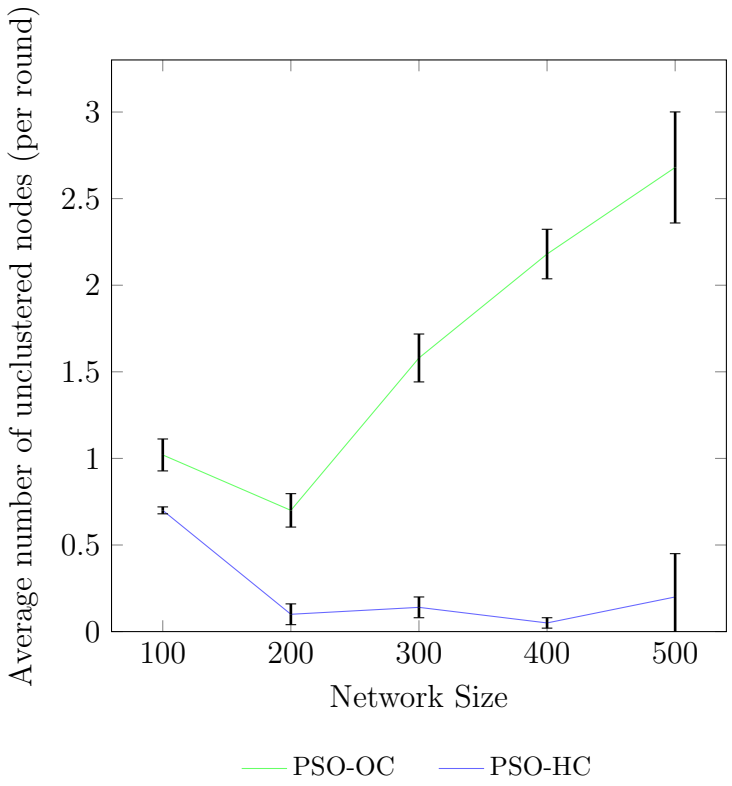


Figure 4.9: Average number of unclustered nodes per round for $WSN\#1$ for PSO-HC

4.4.4 Experimental Results

In this subsection, a performance comparison between PSO-OC and PSO-HC is conducted in terms of the network's energy efficiency, data delivery reliability, and their scalability.

Fig.4.9 and Fig.4.4 record the average number of un-clustered nodes per round in $WSN\#1$ and $WSN\#2$ respectively, for both PSO-OC and PSO-HC. The results presented here represent the average of 5 different runs, for each network size, with a confidence level of 0.99.

It can be observed from Figs. 4.9 and 4.10 that PSO-HC shows better scalability than PSO-OC. This is because PSO-HC tries to cluster all the network nodes using a two-hop clustering approach. If some nodes remain un-clustered in the first tier, PSO-HC will cluster them in the second tier by assigning them to SCHs.

The results displayed in Fig. 4.11 and 4.12 represent the average (mean) of PDR for

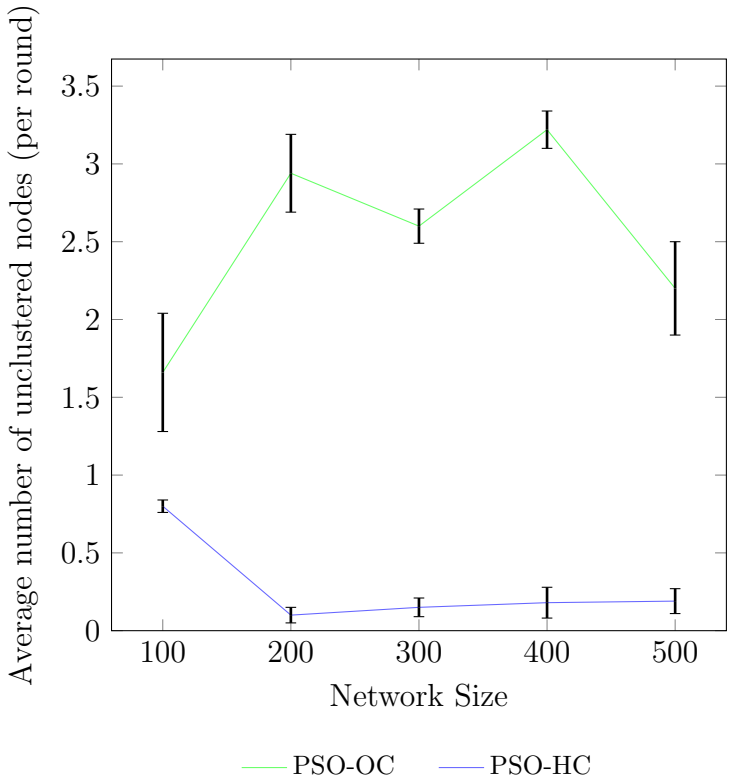


Figure 4.10: Average number of unclustered nodes per round for $WSN\#2$ for PSO-HC

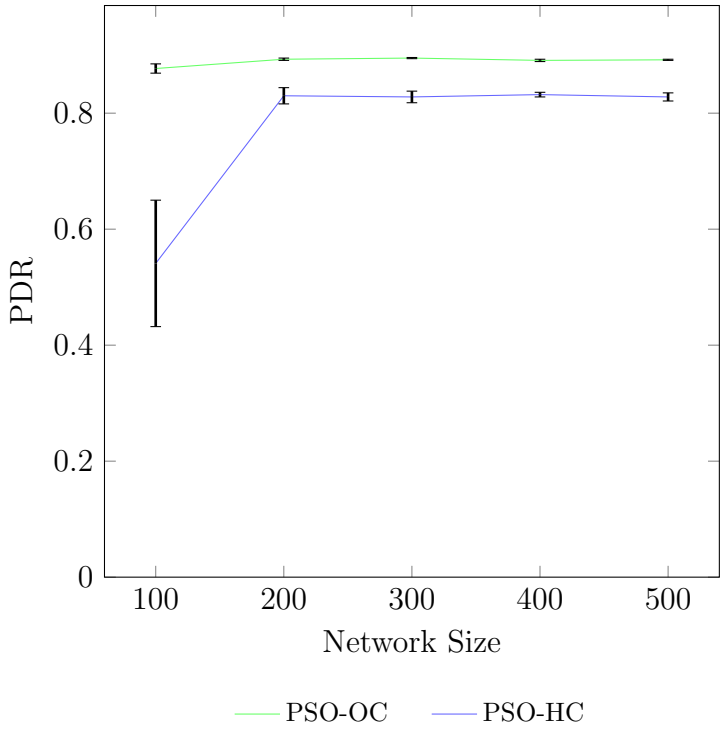


Figure 4.11: PDR for $WSN\#1$ for PSO-HC

packets received by all the CHs for 5 random runs with a confidence level of 0.99, for both $WSN\#1$ and $WSN\#2$ respectively. It is clear that PSO-OC outperforms PSO-HC in terms of the PDR. This is because PSO-OC tends to create more compact clusters and hence this increases the PDR.

Fig. 4.13 and 4.14 show the mean of the average energy consumed per node (in joules) for 5 runs of the protocols, with confidence level of 0.99, for both $WSN\#1$ and $WSN\#2$ respectively. It is clearly shown that PSO-HC has the lowest average consumed energy due to the minimization of the number of active nodes during any given round as justified by (4.10).

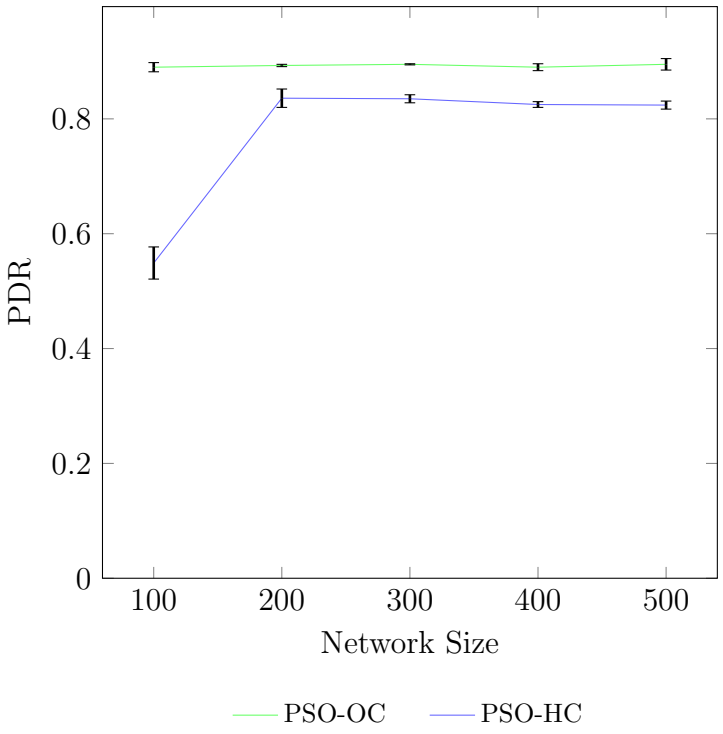


Figure 4.12: PDR for $WSN\#2$ for PSO-HC

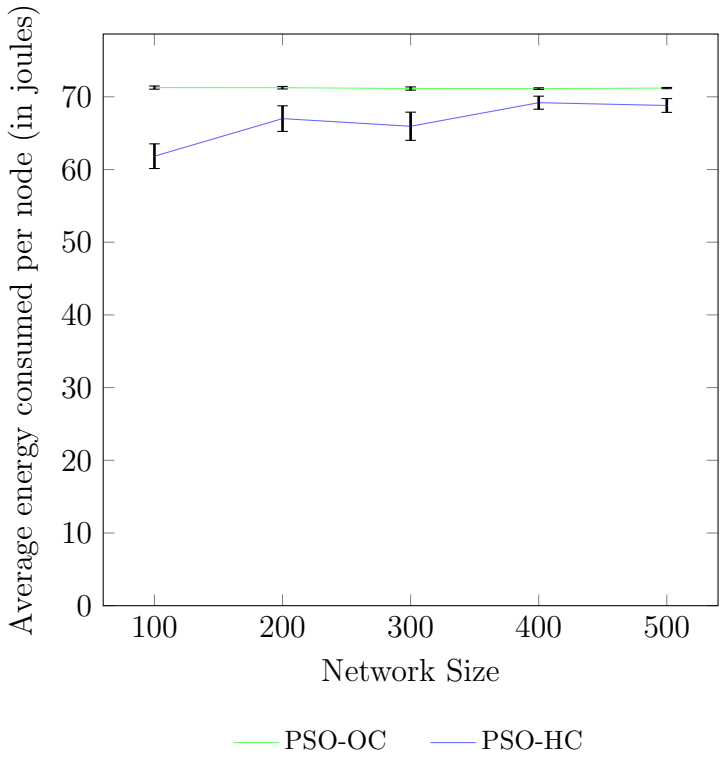


Figure 4.13: Average energy consumed per node in $WSN\#1$ for PSO-HC

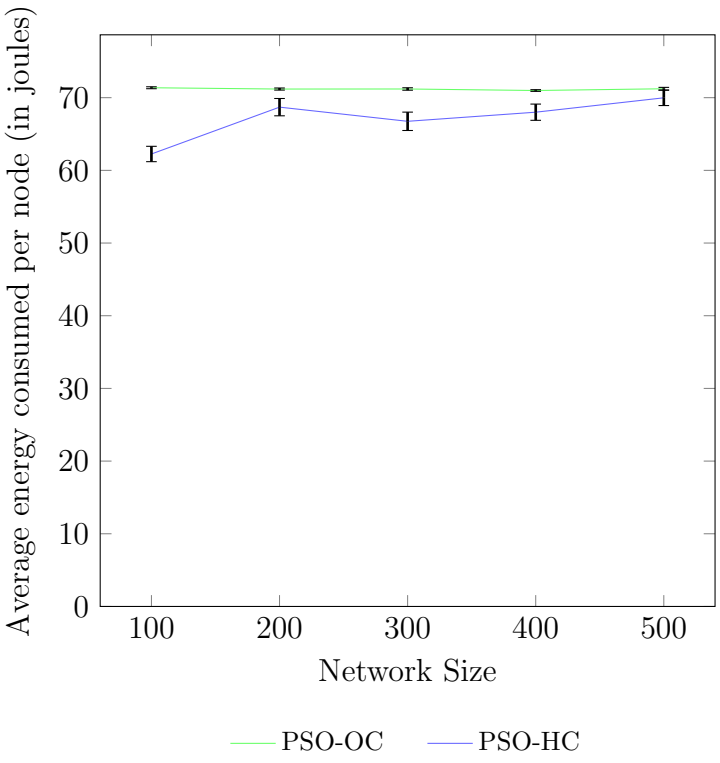


Figure 4.14: Average energy consumed per node in $WSN\#2$ for PSO-HC

4.5 Conclusion

In this chapter, the problem of clustering in WSN was formulated as a single-objective optimization problem, aiming at maximizing the network's energy efficiency, data delivery reliability and scalability. The formulated problem has been solved using three different evolutionary approaches: GA, DE, and PSO. PSO has proved to achieve the best performance in terms of the fitness value.

Based on the results, two PSO-based clustering protocols were proposed. The first, a one-hop clustering protocol (PSO-OC), solves the CH selection problem by electing CHs to form one-hop clusters. The second proposed protocol, the hierarchical clustering protocol (PSO-HC), sets an upper bound on the number of CHs and tries to minimize the number of CHs compared to that upper bound. Furthermore, it allows using two-hop communication links between the sensor nodes and their respective CHs in order to increase its scalability.

Experimental results, under a realistic energy consumption model, showed that the number of active nodes has a great impact on the network's energy efficiency. In PSO-HC, minimizing the number of active CHs led to minimizing the average of energy consumed per node and maximized the network's energy efficiency. However, in PSO-OC, increasing the number of CHs and taking link quality measures into consideration resulted in more compact clusters and hence increased the PDR.

Prior clustering protocols assumed that the key factor in enhancing the WSN energy efficiency is to minimize the distance between the cluster members and their associated CHs. The reason behind their assumption is that they used the first-order radio model which is flawed for modelling radio power consumption in WSNs. This radio model also ignores the listening energy consumption, which is known to be the largest contributor to expended energy in WSNs. Experimental results in this chapter showed that minimizing the number of active nodes is the key factor in achieving energy-efficient networks, assuming a realistic energy consumption model is used. Clustering protocols that ignore minimizing the number of un-clustered nodes, such as the LEACH type protocols, may result in higher energy consumption unless a sleep scheduling mechanism is used.

Chapter 5

PSO-based Routing Protocol for Clustered WSN

5.1 Introduction and Motivation

Several clustering protocols have been proposed in the literature. However, most of these protocols have unrealistic assumption of how the CHs transmit the aggregated data to the BS. They assume that each CH can send its aggregated data directly to the BS using a one-hop approach. However, sensor nodes in WSN have limited communication range, and the BS is usually located far away from the sensing area and is often not directly reachable to all nodes.

A more realistic approach would allow the CHs to form a network among themselves in order to route the data towards the BS (multi-hop approach) [14]. Finding an energy-efficient and reliable routing tree that connects the CHs to the BS is known to be NP-hard problem [15]. Therefore, evolutionary approaches can be employed to solve this problem.

In the last chapter, PSO has proven to have better performance than GA and DE. Moreover, PSO has many advantages over other alternatives optimization techniques like

GA, which has very high processing demands [86]. PSO advantages include ease of implementation on hardware or software, high-quality solutions because of its ability to escape from local optima and quick convergence [87, 88]. Due to its effectiveness in solving NP-hard problems, PSO has been adopted in this chapter to find the optimal inter-cluster routing tree. Clustering and routing are repeated processes; therefore, simpler the optimization algorithm produce more efficient networks. That is another reason why PSO is a popular choice for solving the WSN clustering and routing problems.

In this chapter, a centralized weighted-sum PSO-based protocol is proposed for finding the optimal inter-cluster routing tree. This protocol is appropriate when the CHs are predetermined in advance. The proposed protocol uses a particle encoding scheme and defines an objective function to find the optimal routing tree. The objective function is used to build the trade-off between the energy-efficiency and data delivery reliability of the constructed tree.

In this proposed protocol, it is assumed that the CHs are predetermined in advance using the PSO-OC protocol. The reason for choosing PSO-OC is that it has proven to have higher PDR at the CHs and at the same time maintains reasonable energy consumption. The proposed protocol is named TPSO-CR, from the initials of the words ***T****wo-tier* ***P****article* ***S****warm* ***O****ptimization for* ***C****lustering and* ***R****outing protocol*. The following sections give a detailed description of TPSO-CR.

5.2 Particle Initialization

The method to encode a routing tree into particle is critical for developing the second tier of TPSO-CR. Random encoding can not be used for the following reasons:

- Random encoding results in different particle sizes due to different routes' lengths.

- A random sequence of edges usually does not correspond to a valid tree (that terminates on the destination node without any loop).
- The PSO algorithm involves arithmetic operations such as updating velocity and position which will not be applicable and will increase the number of invalid paths returned.

Authors in [89] have proposed an indirect priority encoding scheme to solve the problems of random encoding. This scheme has been applied successfully in many EA-based protocols like [90, 91, 92]. In this scheme, the particle encodes guiding information about the solution rather than the solution itself. The guiding information used include the priorities of various nodes in the network. In this chapter, a slightly modified scheme is proposed to suit the need of finding the optimal routing tree which connects all the CHs and the BS.

5.2.1 Particle Encoding Process

The dimension of the particle is same as the number of sensor nodes in the network (i.e., N). Let, $P_i = [X_{i,1}, X_{i,2}, X_{i,3}, ..., X_{i,N}]$ be the i_{th} particle of the population where each component, $X_{i,d}, 1 \leq d \leq N$ denotes node N_d priority for selecting it as a relay node. Each component is initialized with a randomly generated number in the range $[-1.0, 1.0]$ based on a uniform distribution.

5.2.2 Particle Decoding Process

A routing tree is built from the encoded particle in a branch growth process. Each branch is a route from a CH to the BS. For example, if there are two CHs in the network, the decoding process will generate two routes, one for each CH. Each route is constructed by appending relay nodes starting from the CH. At each step of the route construction, the

next node with the highest priority is chosen from those which have direct links with the current node. The node that is already included in a growing path will be assigned a large negative priority value, hence that node is highly unlikely to be selected again. In a worst case scenario, if a node is selected again, the concerned route can be treated as an invalid route and can be assigned a high penalty value. The process continues until the BS is reached, and all the CHs are connected to the BS. A routing tree is considered invalid if it has one or more invalid branches (that do not terminate on the destination node or that have loops) and will be assigned a very high fitness value as a penalty. The best particle at the end of a run of the algorithm is the one that contains priorities that lead the decoding procedure to select nodes forming the optimal routing tree.

Illustration 6.1: Consider a WSN with 20 sensor nodes and 2 cluster heads, i.e., N_1, N_8 as shown in Figure 5.1. Therefore, the dimension of the particles is same as the number of sensor nodes, i.e., $N = 20$. Let's consider the graph $G(V, E)$ shown in Fig. 5.1. The edge $u \rightarrow v$ indicates that u can send to v but not necessarily vice versa.

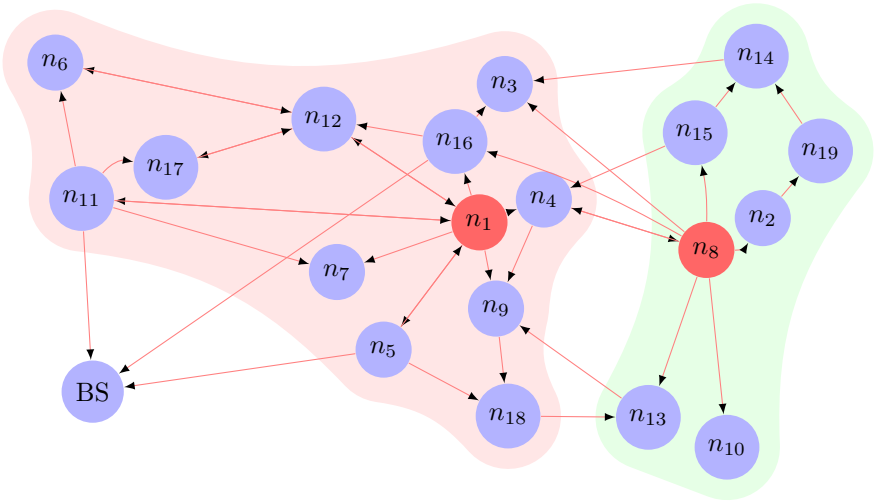


Figure 5.1: A wireless sensor network with 20 sensor nodes and 2 CHs (n_1 and n_8)

Let us assume that a particle P_i has been randomly generated as illustrated in Fig.5.2(a). To find a routing tree from N_1 and N_8 to the BS, the protocol will build a route from N_1

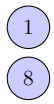
to the BS, and a another route from N_8 to the BS.

To find a branch from N_1 to the BS, a node that is connected to N_1 is identified first. As seen from Figure 5.1, the nodes $[4, 5, 7, 9, 11, 12, 16]$ are the nodes to be considered. Their priorities are $[-0.1, 0.5, -0.1, -0.5, 0.7, -0.3, 0.5]$ respectively. Node 11 has the highest priority and hence it is used as the next relay node to N_1 and its priority is updated into a high negative value $-N$ to avoid selecting it again in the route. The possible nodes from node 11 are nodes $[0, 1, 6, 7, 17]$. The priorities of these nodes are $[1.0, -N, -0.7, -0.1, -0.2]$ respectively. Since node 0 (BS) has the highest priority, it is taken as the next relay node while constructing the route. Since the BS is reached, the route construction from N_1 to BS ends and results in the following route $(8, 1, 0)$. The same procedure is repeated for the branch from N_8 to the BS until a complete route $(8, 16, 0)$ is achieved. Fig.5.2(b - d) demonstrates this process.

Node ID:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Priority:	1.0	0.6	−0.3	0.3	−0.1	0.5	−0.7	−0.1	−0.5	−0.5	−0.1	0.7	−0.3	−0.4	−0.5	0.3	0.5	−0.2	−0.1	−0.2

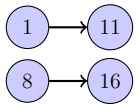
(a) Particle P_i encoding for network in Figure 5.1

Node ID:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Priority:	1.0	$-N$	−0.3	0.3	−0.1	0.5	−0.7	−0.1	−0.5	−0.5	−0.1	0.7	−0.3	−0.4	−0.5	0.3	0.5	−0.2	−0.1	−0.2
	1.0	0.6	−0.3	0.3	−0.1	0.5	−0.7	−0.1	$-N$	−0.5	−0.1	0.7	−0.3	−0.4	−0.5	0.3	0.5	−0.2	−0.1	−0.2



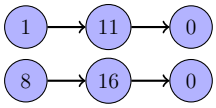
(b) Particle P_i after adding the CHs to the routing tree

Node ID:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Priority:	1.0	$-N$	−0.3	0.3	−0.1	0.5	−0.7	−0.1	−0.5	−0.5	−0.1	$-N$	−0.3	−0.4	−0.5	0.3	0.5	−0.2	−0.1	−0.2
	1.0	0.6	−0.3	0.3	−0.1	0.5	−0.7	−0.1	$-N$	−0.5	−0.1	0.7	−0.3	−0.4	−0.5	0.3	$-N$	−0.2	−0.1	−0.2



(c) Particle P_i after adding nodes 11 and 16 as relay nodes

Node ID:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Priority:	$-N$	$-N$	−0.3	0.3	−0.1	0.5	−0.7	−0.1	−0.5	−0.5	−0.1	$-N$	−0.3	−0.4	−0.5	0.3	0.5	−0.2	−0.1	−0.2
	$-N$	0.6	−0.3	0.3	−0.1	0.5	−0.7	−0.1	$-N$	−0.5	−0.1	0.7	−0.3	−0.4	−0.5	0.3	$-N$	−0.2	−0.1	−0.2



(d) Particle P_i after adding the BS and finishing the routing tree construction

Figure 5.2: Example of priority-based encoding and decoding process for an arbitrary particle P_i

5.3 Particle Evaluation

After particles initialization, the generated routing tree that results from the decoding process is evaluated to determine its fitness value. The optimal routing tree is selected such that it minimizes the cost of the objective function. The goal of the function is to optimize the combined effect of the following properties:

5.3.1 Energy Efficiency

To achieve an energy efficient routing tree, two sub-objectives need to be met:

1. Save energy: fewer sensor nodes need to be active during each round. To achieve that, the protocol needs to minimize the number of relay nodes and favour CHs as better candidates to act as relay nodes.

   Let R_{P_i} represent the vector of relay nodes IDs in the routing tree generated from particle P_i and C_{P_i} represent the set of CHs IDs that act as relay nodes in that tree. Then, the function that represents this sub-objective is formulated as follows:

$$ES_{P_i} = \frac{R_{P_i}}{C_{P_i}} \tag{5.1}$$

2. Balance energy consumption: a relay node with a higher level of energy is a better candidate to include in the routing tree. The following function is used to balance the energy consumption among all the network nodes in terms of routing:

$$EB_{P_i} = \frac{\sum_{i=1}^{N} E(n_i)}{\sum_{r=1}^{|R|} E(RN_{P_i,r})} \tag{5.2}$$

N is the current total number of live nodes in the network. $E(n_i)$ is the remaining energy in node n_i .$|R|$ is the total number of relay nodes. $E(RN_{P_i,r})$ is the remaining energy for relay node number r in particle P_i.

5.3.2 Data Delivery Reliability

To maximize the PDR, the protocol needs to maximize the link quality between the relay nodes in the routing tree. The following function minimizes the worst link quality among all the branches in the routing tree:

$$LQ_{P_i} = \max_{b=1,2,\ldots,B} \sum_{\forall rn_i \in b} \frac{RSSI_{P_i}(rn_i \rightarrow nextHop)}{minRSSI} \tag{5.3}$$

where B is the number of branches (one branch for each CH) in the routing tree. rn_i is relay node number i in branch b

After calculating the sub-objectives ES_{P_i}, EB_{P_i} and LQ_{P_i} , they are scaled using Eq. (4.2) to result in the following sub-objectives values sES_{P_i}, sEB_{P_i} and sLQ_{P_i} respectively. Then, the final objective function $FinalObj_{P_i}$, that needs to be minimized is calculated using:

$$FinalObj_{P_i} = sES_{P_i} + sEB_{P_i} + (1 - sLQ_{P_i}) \tag{5.4}$$

The pseudo-code of the proposed TPSO-CR protocol executed at an arbitrary node u is shown in algorithm (1).

5.4 Experimental Results

The goal of the experiments is to evaluate the effect of using a dedicated routing tree, generated from the TPSO-CR protocol, on both the network's energy efficiency and data

Algorithm 1: Pseudo-code of the proposed TPSO-CR protocol

```
1  begin Procedure startup()
2      setTimer(START − ROUND, 0.0);
3  end
4  begin Procedure timerFiredCalback(index)
5      switch index do
6          case START − ROUND :
7              double timer = uniform(0.0 , r);
8              setTimer(FIND − NBRS, timer);
9              setTimer(BROADCAST − INFO, r);
10             if isBS then
11                 setTimer(RUN − PSO, n);
12             end
13             else
14                 setTimer(RUN − STEADY − PHASE, m);
15             end
16             roundNumber + +;
17             setTimer(START − ROUND, roundLength);          ▷ r, n, and m are random times
18         end
19         case FIND − NBRS :
20             broadcast (ID);
21         end
22         case BROADCAST − INFO :
23             broadcast (ID, residualEnergy, neighbours' IDs and their RSSI);
24         end
25         case RUN − PSO :
26             optimalCHs = runFirstPSO(NetworkInfo);                          ▷ run first tier
27             optimalRoutingTree = runSecondPSO(optimalCHs, NetworkInfo);     ▷ run second tier
28             broadcast(configuration = optimalCHs + optimalRoutingTree);
29         end
30         case RUN − STEADY − PHASE :
31             if (!isCH||!isCM||!isRelayNode) then
32                 setStateSleep();
33             end
34             if (isCH) then
35                 clusterLength = clusterMembers.size();
36                 setTimer(START − SLOT, clusterLength × slotLength);
37             end
38             else
39                 if (!isRelayNode) then
40                     setStateSleep()
41                     setTimer(START − SLOT, myTDMATurn × slotLength);
42                 end
43             end
44         end
45         case START − SLOT :
46             setTimer(START − SLOT, clusterLength × slotLength);
47             if (isCH) then
48                 aggregatePackets();                                  ▷ aggregate packets
49                 processBufferedPackets();                            ▷ send packets to next hop
50             end
51             else
52                 processBufferedPackets();                            ▷ send packets to CH
53                 setTimer(END − SLOT, slotLength);                    ▷ go to sleep mode at end of slot
54             end
55         end
56         case END − SLOT :
57             if (!isCH||!isCM||!isRelayNode) then
58                 setStateSleep();
59             end
60         end
61     endsw
62 end
```

delivery reliability. The simulation settings for TPSO-CR are given in Table 5.1.

Table 5.1: PSO algorithm settings for TPSO-CR

Parameter	**Value**
Network Size	[100 - 500]
Population size	50
Number of iterations	200
Learning Factor $c1$	2
Learning Factor $c2$	2
Interia weight w	0.9

Fig. 5.3 and 5.4 show the comparison of TPSO-CR and the other protocols in term of the network throughput in $WSN\#1$ and $WSN\#2$ respectively, with a confidence level of 0.99. Throughput is defined as the number of data packets successfully received at the BS. Using the number of aggregated packets delivered to the BS is not accurate, since many packets result from the aggregation process of many raw packets collected from the cluster members. In this thesis, the number of the raw packets is used to calculate the throughput at the BS. It can be observed that TPSO-CR protocol outperforms the other protocols in terms of network's throughput as shown in Fig. 5.3 and 5.4.

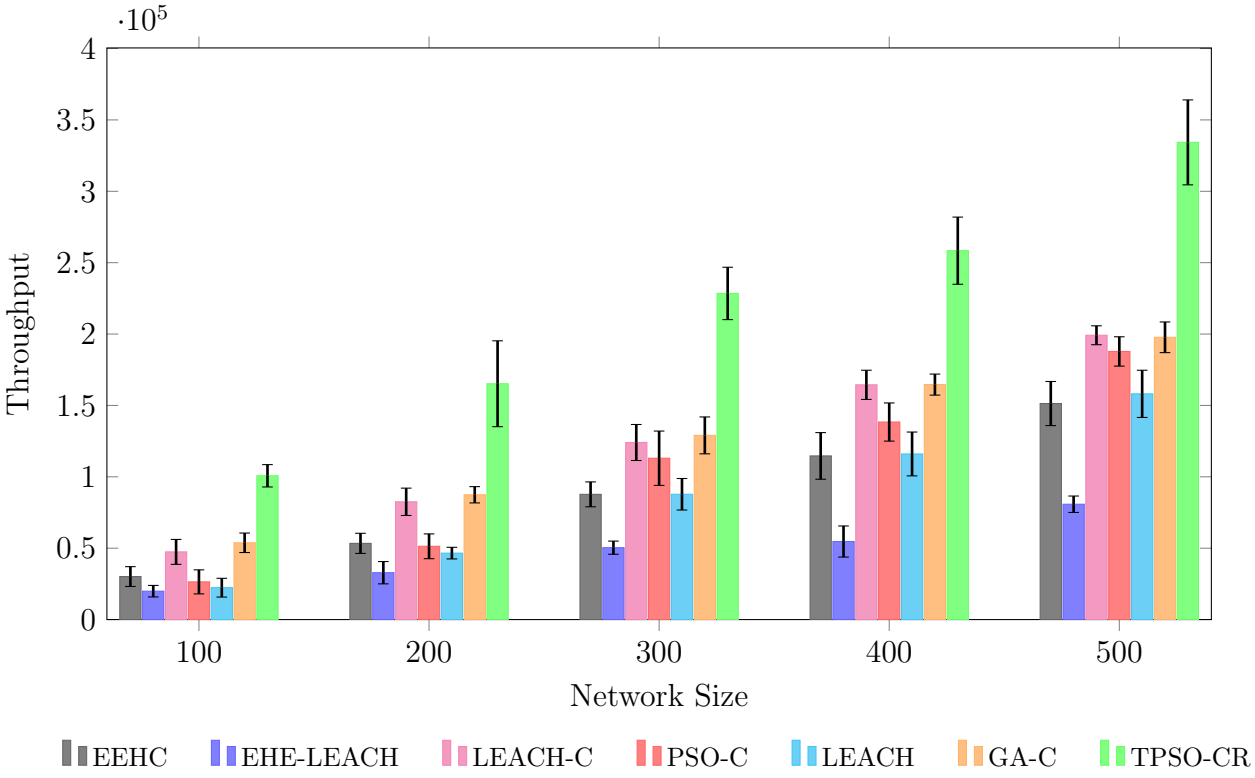


Figure 5.3: Throughput for $WSN\#1$, for TPSO-CR

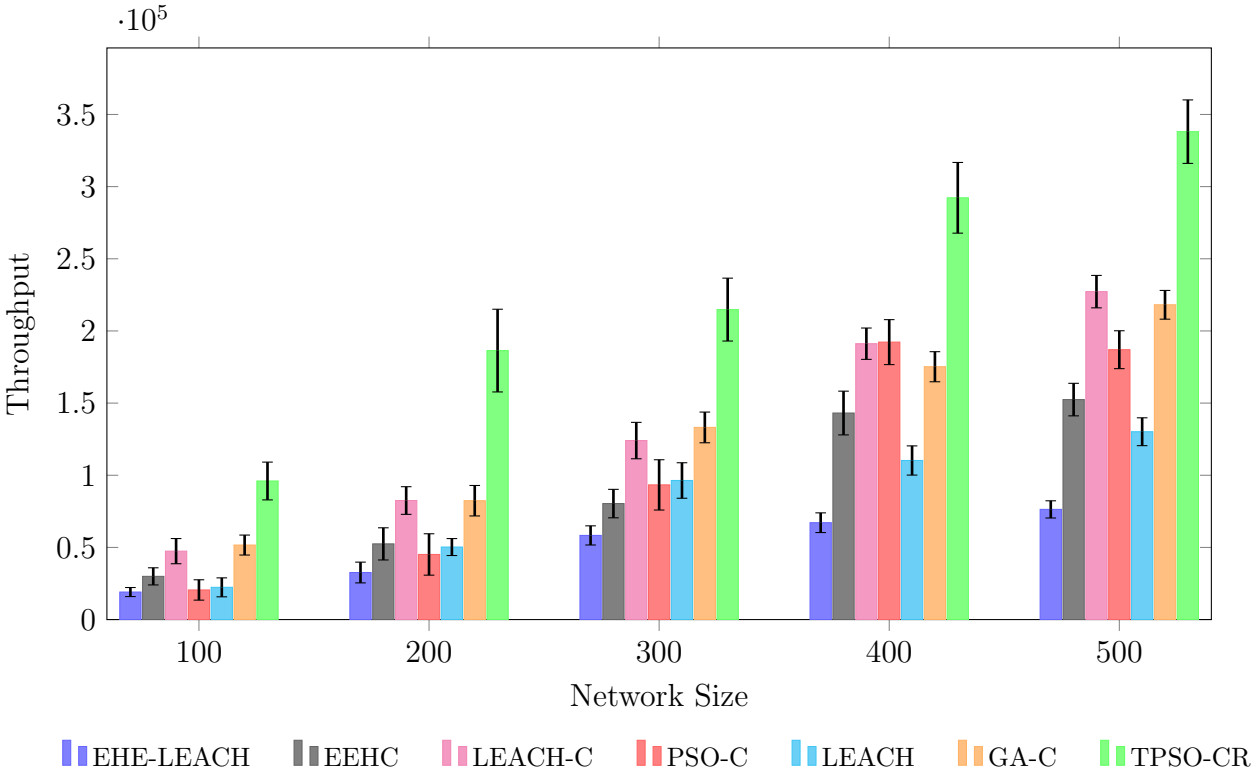


Figure 5.4: Throughput for $WSN\#2$, for TPSO-CR

We also studied the effect of using relay nodes for multi-hop data transmission on the network's energy efficiency. Table. 5.2 and Table 5.3 show the comparison of the TPSO-CR protocol and other protocols in term of the average energy consumed by a node (in joules) in $WSN\#1$ and $WSN\#2$ respectively. It was noted that, in the case of sparsely deployed WSN, the average energy consumed per node in TPSO-CR is higher than LEACH-C, GA-C and PSO-C. This is mainly due to an increase in the number of active nodes during any round. This increase is caused by adding more nodes to act as relay nodes since the number of CHs is small, and their transmission range is limited. As the sensor density increases, the number of CHs that cover the same area increases. At the same time, the routing algorithm favours the inter-cluster communication between the CHs. This caused the average consumed energy for TPSO-CR to be closer to that of LEACH-C, GA-C and PSO-C in densely deployed WSN.

Table 5.2: Mean for average consumed energy per node and standard deviation in $WSN\#1$, for TPSO-CR

Protocols	**100 Sensor nodes**		**200 Sensor nodes**		**300 Sensor nodes**		**400 Sensor nodes**		**500 Sensor nodes**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	175.1	6.54	149.2	10.1	131.0	5.71	132.3	5.50	131.7	4.40
EHE-LEACH	155.1	9.00	140.3	6.24	131.7	4.55	131.6	3.83	130.3	3.57
EEHC	158.8	9.00	137.4	10.3	131.2	1.53	131.3	3.71	130.5	4.67
PSO-C	73.8	0.04	72.2	0.06	71.5	0.02	71.3	0.08	71.1	0.02
GA-C	74.4	0.07	72.6	0.30	71.8	0.30	71.6	0.12	71.3	0.38
LEACH-C	74.5	0.003	73.0	0.004	72.5	0.005	72.3	0.01	72.1	0.006
TPSO-CR	80.5	0.35	76.2	0.20	74.6	0.09	74.0	0.05	73.4	0.05

Table 5.3: Mean for average consumed energy per node and standard deviation in $WSN\#2$, for TPSO-CR

Protocols	**100 Sensor nodes**		**200 Sensor nodes**		**300 Sensor nodes**		**400 Sensor nodes**		**500 Sensor nodes**	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	175.1	6.54	150.5	10.0	133.6	3.62	130.5	7.24	128.4	5.72
EHE-LEACH	155.6	11.2	137.2	11.3	145.0	10.3	135.0	11.2	145.1	10.6
EEHC	158.9	7.26	140.8	5.60	148.4	9.18	129.7	9.20	139.5	8.12
PSO-C	73.8	0.30	72.0	0.06	71.3	0.22	73.7	0.01	73.5	0.08
GA-C	74.5	0.03	72.7	0.27	71.9	0.24	71.4	0.03	71.3	0.02
LEACH-C	74.5	0.003	73.0	0.004	72.5	0.005	72.3	0.001	72.1	0.002
TPSO-CR	80.5	0.35	76.2	0.20	74.6	0.09	74.0	0.05	73.4	0.05

5.5 Conclusion

In this chapter, A PSO-inspired protocol was proposed to solve the routing tree construction problem for clustered WSN. The protocol runs in two tiers: the first tier finds the best CHs and their associative clusters using PSO-OC while the second tier solves the problem of the inter-cluster communication by finding the optimal routing tree.

Prior clustering protocols assumed that the CHs can send their data to the BS directly by maximizing their transmission power. However, this solution is considered unrealistic in many practical situations due to the communication range restrictions of the sensor nodes. Furthermore, maximizing the transmission range will result in high level of energy consumption and will minimize the network's lifetime.

Experimental results of TPSO-CR proved that using a dedicated routing tree results in higher network throughput. Moreover, limiting the inter-cluster communication to the CHs only will result in a smaller number of active nodes, and this will in turn minimize the average consumed energy per node.

Chapter 6

Pareto-based Optimization Protocol for Clustering and Routing in WSN

6.1 Introduction and Motivation

Using the conventional weight sum approach in multi-objective optimization is computationally efficient and straightforward to implement [81, 82, 83]. It has been widely used because of its simplicity. However, it is known that this approach has the following problems [93, 94, 95, 96]:

- Only one optimal solution can be obtained from one single run.
- This approach can not find the optimal solution when the feasible solution set in the objective domain is not convex.
- The choice of the weight vector can highly affect the obtained solutions.

These problems are particularly critical if the objectives are conflicting or must be handled simultaneously. In these cases, the concept of optimal solution changes because the goal is to find a set of good trade-off solutions from which the decision maker wants to

select one. To achieve that, Pareto-based optimization techniques, which make direct use of the dominance relation for ranking different solutions in terms of the objective functions, can be used to find the set of optimal solutions.

The clustering problem in WSN consists of multiple conflicting objectives that need to be optimized simultaneously. Pareto-based optimization techniques can be used to solve the CHs election problem especially if the number of CHs is not fixed. For example, clustering can provide an energy-efficient solution if only a few number of CHs are involved in doing the main operations in the network such as routing, management, and data aggregation. However, minimizing the number of CHs may lead to minimizing the number of clustered nodes and hence minimize the clustering protocol scalability. Another objective to consider concurrently is the inter-cluster communication cost that affects the data delivery reliability.

In this chapter, a centralized multi-objective Pareto-optimization approach for determining an energy efficient, scalable and reliable clustering protocol is adapted. A new individual encoding scheme that represents a joint solution for both the clustering and routing problems in WSN is proposed.

The proposed approach uses a variable number of CHs, and its objective is to assign each network node to its respective CH and each CH to its respective next hop. The joint problem of clustering and routing in WSN is formulated as a multi-objective minimization problem with a variable number of CHs, aiming at determining an energy efficient, reliable and scalable clustering and routing scheme.

The formulated problem has been solved by two state-of-the-art Multi-Objective Evolutionary Algorithms (MOEA), and their performance has been compared using some quality indicators. Furthermore, a performance comparison between the proposed approach against the other well-known clustering approaches is conducted.

6.2 Pareto-based Multi-objective Optimization

A Multi-objective Optimization Problem (MOP) involves optimizing a number of objectives (usually conflicting) simultaneously [97]. Due to having multiple conflicting objectives in MOP, there is no single solution that can be described as an optimal solution. Therefore, we are interested in finding a number of optimal solutions. Evolutionary algorithms (EAs) are well suited to solve multi-objective optimization problems due to their population-based nature [98].

6.2.1 Basic Concepts

Assuming a minimization problem for convenience, a MOP with n decision variables and M objective functions can be expressed as follows: given an n-dimensional decision variable vector $x = \{x_1, ..., x_n\}$ in the solution space X find a vector x^* which yields the optimum value for a given set of M objective functions $z(x^*) = \{z_1(x^*), ..., z_M(x^*)\}$ where $M \geq 2$.

However, due to the conflicting nature of the objective functions, it is rare that the global optimum for all of the individual objective functions occurs simultaneously at one single point of search space. Instead, we are interested in finding a set of trade-off solutions. The most commonly adopted notion of optimality is the so-called Pareto optimality.

Pareto-dominance Principle

A feasible solution x is said to dominate another feasible solution y if and only if the following two conditions are true:

- Solution x is no worse than a solution y in all objectives.
- Solution x is strictly better than a solution y in at least one objective.

Formally speaking, x dominates y (denoted by $(x \succ y)$), if and only if:

$$z_i(x) \leq z_i(y), \qquad \forall i \in 1, ..., M \tag{6.1a}$$

$$z_i(x) < z_i(y), \qquad \exists i \in 1, ..., M \tag{6.1b}$$

If any of the conditions mentioned above is false, then solution x does not dominate the solution y. If solution x dominates solution y, then solution x is better than solution y.

Pareto Optimality

Solution x^* is a **Pareto optimal solution** if there exists no feasible vector of decision variables $x \in X$, which would decrease some objective value without causing a simultaneous increase in at least one other objective value. There are no superior solutions to the problem than x^*, although there may be other equally good solutions. Formally speaking, $x \in X$ is Pareto optimal if and only if,

$$z(y) \prec z(x), \qquad \forall y \in X \tag{6.2}$$

The set of solutions that satisfy Equation (6.2) is known as the **Pareto optimal set**. A Pareto optimal set is a set of solutions that are non-dominated with respect to each other. The vector corresponding to the solutions included in the Pareto optimal set is called **non-dominated vector**. The plot of the objective functions whose non-dominated solutions are in the Pareto optimal set is called the **Pareto optimal front** [99] which corresponds to the trade-off surface in objective space.

The literature hosts several interesting approaches for tackling MOPs, with Multi-Objective Evolutionary Algorithms (MOEAs), posing all the desired characteristics for obtaining a set of non-dominated solutions, in a single run. These approaches work with two main goals:

- **Convergence:** find a set of Pareto-optimal solutions, and
- **Diversity:** find a set of diverse solutions in order to prevent premature convergence and achieve a well-distributed trade-off Pareto front.

The first goal guides the solutions towards the Pareto-optimal region and the second goal guides along the Pareto-optimal front.

In this thesis, two different types of MOEAs are considered as the optimization tools to solve the joint problem of clustering and routing in WSN:

1. **The Non-dominated Sorting Genetic Algorithm II (NSGA-II)** and
2. **Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO)**

These two algorithms have found extensive applications in different fields of WSNs [100, 101, 102, 103, 104, 105]. The literature also reveals that these two algorithms have provided the most to the needs of practical optimization problems known to date [106]. These algorithms are also popular because of their ease of hardware implementation [106].

6.2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

NSGA-II [107] is a popular non-domination based genetic algorithm for multi-objective optimization. It has demonstrated better performance than the Strength Pareto Evolutionary Algorithm (SPEA) [108] and Pareto Archived Evolution Strategy (PAES) [109], in terms of convergence and diversity of the obtained Pareto front [107, 110].

NSGA-II starts with producing a population that consists of $nPop$ random solutions (chromosomes). In each generation, the population in NSGA-II is sorted into several non-dominated fronts using a ranking algorithm first (non-dominated sorting). Then, individual solutions are selected from these non-dominated fronts by calculating the crowding

distance. The crowding distance measures the distance between the individual solutions and the rest of the solutions in the population. If two individual solutions are in the same non-dominated front, the solution with a higher value of crowding distance will be selected. The crowding distance calculation is used to preserve the diversity among non-dominated solutions in the later stage of the run in order to obtain a good spread of solutions. After that, the algorithm applies the standard crossover and polynomial operators to combine the current population and its offspring generated as next generation. At last, the best individuals in terms of non-dominance and diversity are selected as the solutions. The steps of the NSGA-II algorithm is presented in Algorithm (2).

Algorithm 2: The main steps of the NSGA-II Algorithm

```
Create a random population of nPop chromosomes (candidate solutions)
while Stopping condition is not met do
    Evaluate the multi-objective fitness of each chromosome in the population.
    Rank population by following steps:
    begin
        Rank population by using Algorithm (3).
        Calculate the crowding distance by using Algorithm (4).
    end
    Choose two parent chromosomes from a population based on the crowding
    selection operator described by Algorithm (5).
    With a crossover probability, crossover the parents to form new offspring
    (children). If no crossover was performed, offspring is the exact copy of parents.
    With a mutation probability, mutate new offspring at each gene.
    Place new offspring in the new population.
end
Return the set of the non-dominated Pareto-optimal solutions in current population.
```

6.2.3 Speed-constrained Multi-objective PSO (SMPSO)

Recently, PSO has been playing a very important role in MOPs because of its convergence speed and simple operators. Speed-constrained Multi-objective Particle Swarm Optimization (SMPSO) algorithm [111] is based on the PSO theory.

Algorithm 3: Non-dominated Sorting

Let rank number, $r = 0$
while *population is not empty* **do**
    $r = r + 1$
    Find the non-dominated individuals from population P based on the definition of domination.
    Assign rank r to these individuals.
    Remove these individuals from population P.
end

Algorithm 4: Crowding distance calculation

Let $d_i = 0$ for $i = 1, 2, ..., Z$.
For each objective function f_k, $k = 1, 2, ..., M$, sort the population ascending.
Let $d_1 = d_Z = INF$.
for $j = 2$ ***to*** $(Z1)$ **do**
    set $d_j = d_j + (f_{kj+1} - f_{kj1})$.
end

An experimental comparison was conducted in [112] to assess the performance of SMPSO against six of the state of the art Pareto-based MOPSO representatives namely, Non-dominated Sorting PSO (NSPSO) [113], Sigma MOPSO (Sigma MOPSO) [114], Optimized MOPSO (OMOPSO) [115], Another MOPSO (AMOPSO) [116], Pareto Dominance MOPSO (MOPSOpd) [117] and Comprehensive Learning MOPSO (CLMOPSO) [118]. SMPSO has outperformed the other protocols in terms of the quality of results. Furthermore, SMPSO has shown a remarkable performance in terms of other different assessment criteria [119]: convergence towards the optimum solutions [120], and scalability with the problem size [121].

Similar to NSGA-II, SMPSO selects best solutions by calculating crowding distance and also stores the selected individual solutions in an archive. SMPSO applies a polynomial mutation operator [122] to 15% of the population to accelerate the speed of convergence.

Algorithm 5: Crowding Selection Operator

$x > y$ *iff* $r_x < r_y$ *or* $r_x = r_y$ *and* $d_x > d_y$

In addition, SMPSO incorporates a velocity constriction procedure [123] to produce new effective particle positions in those cases in which the velocity becomes too high and hence avoid the swarm explosion problem [123]. In this procedure, each particle velocity is calculated according to (Eq. 2.6a). The resulting velocity is then multiplied by the constriction factor, χ, given by the following equation:

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \tag{6.3}$$

where

$$\varphi = \begin{cases} C_1 + C_2, & if\ C_1 + C_2 > 4 \\ 1, & if\ C_1 + C_2 \leq 4 \end{cases} \tag{6.4}$$

Then, the accumulated velocity of each variable j, in each particle i in iteration t, is further bounded by means of the following velocity constriction equation:

$$v_{i,j}(t) = \begin{cases} delta_j, & if\ v_{i,j}(t) > delta_j \\ -delta_j, & if\ v_{i,j}(t) \leq -delta_j \\ v_{i,j}(t), & Otherwise \end{cases} \tag{6.5}$$

where

$$delta_j = \frac{UpperLimit_j - LowerLimit_j}{2} \tag{6.6}$$

The steps of the SMPSO algorithm is shown in Algorithm (6).

6.2.4 Performance assessment of different MOEAs

With the existence of different MOEAs, it is necessary to quantify the performance of each algorithm. A number of quality indicators have been proposed in the literature for

Algorithm 6: The main steps of the SMPSO Algorithm

```
Initialize a swarm of N particles (candidate solutions) randomly
Evaluate the particles
Determine non-dominated solutions and store them in the leader archive
while Stopping condition is not met do
    Compute the particles' velocities, according to (Eq. 6.5)
    Find the best global particle by randomly taking two solutions from the leaders
    archive and select the one that has the largest crowding distance
    Update particles' positions
    Apply the polynomial mutation operator with a given probability
    Evaluate the particles according to the objective functions
    Update the leader archive. If the leaders archive becomes full, Use the crowding
    distance to decide which particles must remain in it
end
Return the set of the non-dominated Pareto-optimal solutions in the current leader
archive.
```

measuring both the convergence and the diversity of the obtained set of non-dominated solutions.

The quality indicator method is the dominant method in the literature to assess the performance of different MOEAs [124]. It maps each Pareto set approximation to a number and performs statistics on the resulting distributions of numbers [124].

Some quality indicators require the knowledge of the true Pareto-optimal front that is unknown in this application. Instead, an approximation set to the optimal Pareto-optimal front of the problem is computed. Taking this into account, the hypervolume indicator and the Epsilon indicator are adopted to access the performance of SMPSO and NSGA-II in this thesis. The Epsilon indicator takes into account measuring the convergence properties of the obtained Pareto-optimal front [125]. The hypervolume indicator measures both the convergence and diversity of the obtained Pareto-optimal front solutions simultaneously [126, 125].

The Hypervolume Indicator

The Hypervolume (HV) indicator was introduced in [127]. It has gained increasing interest in recent years and has become a popular indicator of the performance of different MOEAs [128, 129].

If solutions are considered as points in objective space, hypervolume is the n-dimensional space that is contained within a solution set, i.e. the n-dimensional volume of the set relative to some reference point, usually the anti-optimal point or worst possible point for the space. In other words, the hypervolume of a set is the total size of the space dominated by the solutions in the set. A set with a larger hypervolume is likely to represent a better set of trade-offs than sets with lower hypervolume.

Given a set of non-dominated solutions Q, for each solution $i \in Q$, a hypercube v_i is constructed with a reference point W and the solution i as the diagonal corners of the hypercube. Accordingly, a union of all hypercubes is found and its hypervolume is calculated by:

$$HV = \cup_{i=1}^{|Q|} v_i \tag{6.7}$$

Figure 6.2.4 shows an example of a HV for a 2-dimensional minimization problem with set of non-dominated solutions $Q = \{A, B, C\}$ and reference point W.

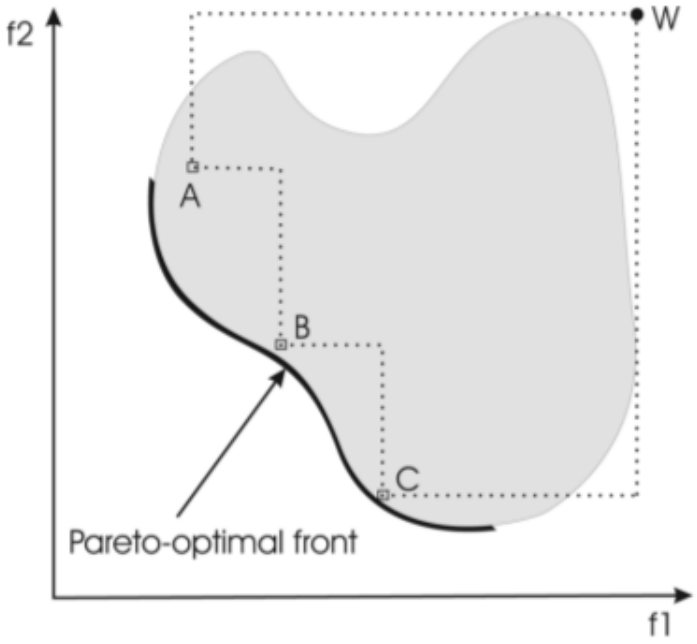


Figure 6.1: HV enclosed by the non-dominated solutions A,B, and C [130].

Algorithms with larger values of HV are desirable [130].

The Epsilon Indicator

The Epsilon indicator was proposed in [131]. Given two sets of non-dominated solutions A and B, this indicator computes the minimum factor by which objectives of solutions in B can be multiplied so that the transformed set of non-dominated solutions is still weakly dominated by A.

More formally, given $z^1 = \{z_1^1, ..., z_n^1\}$ and $z^2 = \{z_1^2, ..., z_n^2\}$, where n is the number of objectives:

$$I_{\epsilon+}(A, B) = inf_{\epsilon \in R}\{\forall z^2 \in B, \exists z^1 \in A : z^1 \prec_\epsilon z^2\} \tag{6.8}$$

where $z^1 \prec_\epsilon z^2$ if and only if $\exists 1 \leq i \leq n : z_i^1 < \epsilon + z_i^2$ [125].

For the Epsilon indicator, the lower the value the better the computed fronts [125].

6.3 Joint Clustering and Routing Approach for WSN

This section gives a detailed description of the proposed approach that the BS uses to find the optimal set of CHs and the routing tree that connects them. The BS adopts a Pareto-based multi-objective approach for determining the optimal set of CHs.

In the proposed approach, a new individual encoding scheme that represents a joint solution for both the clustering and routing problems in WSN is proposed. The inter-cluster communication is limited to the CHs. The proposed approach assigns each network node to its respective CH, and each CH to its respective next hop, which should be a CH also. Fig. 6.2 provides an overview of the proposed approach workflow. The next subsections give a detailed description of the proposed approach.

6.3.1 Individual Initialization

The individuals are presented in such a way that each individual provides the optimal set of CHs and the route from each CH to the BS. The dimension of an individual is same as the number of sensor nodes in the network (i.e., N). Let, $I_i = [X_{i,1}, X_{i,2}, X_{i,3}, ..., X_{i,N}]$ be the i_{th} individual of the population where each component, $X_{i,d}, 1 \leq d \leq N$ maps the assignment of the sensor node n_d to a CH. Each component is initialized with a randomly generated number in the range $[0.0, 1.0]$ based on a uniform distribution. Let $Nbrs(n_d)$ be the list of all n_d neighbours. Then, the CH of node K is encoded as follows: $CH_K = \lceil (X_{i,d} \times |Nbrs(n_d)|) \rceil$

Illustration 7.1 Consider a WSN with 20 sensor nodes, i.e., $S = \{n_0, n_2, ..., n_{19}\}$ where n_0 is the BS as shown in Fig. 6.3. Therefore, the dimension of the individual is same as the number of sensor nodes minus the BS, i.e., $N = 19$. The edge $u \rightarrow v$ indicates that

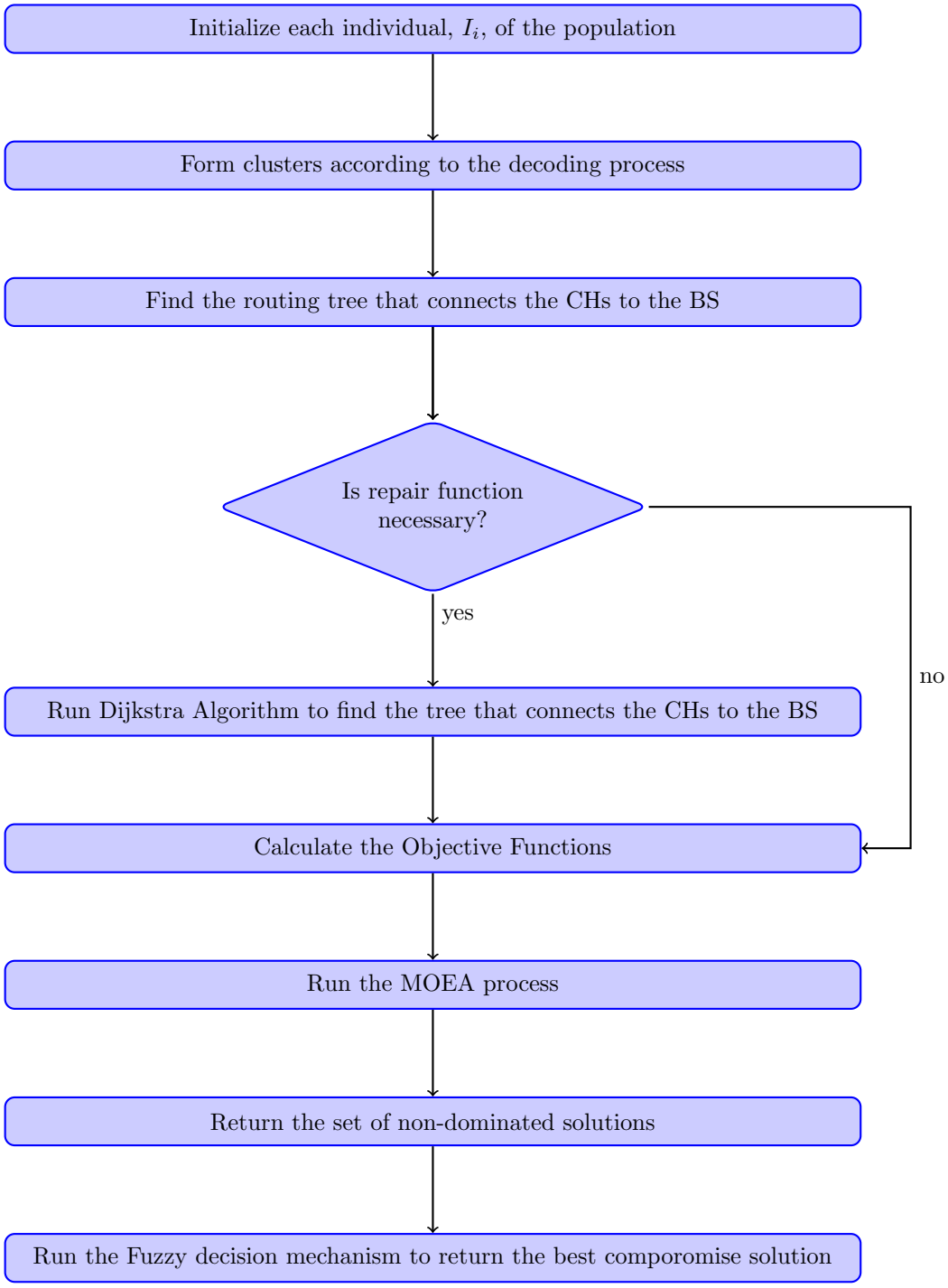


Figure 6.2: Workflow of the Proposed Approach

that node v within communication range of node u hence node u can send to node v but not necessarily vice versa.

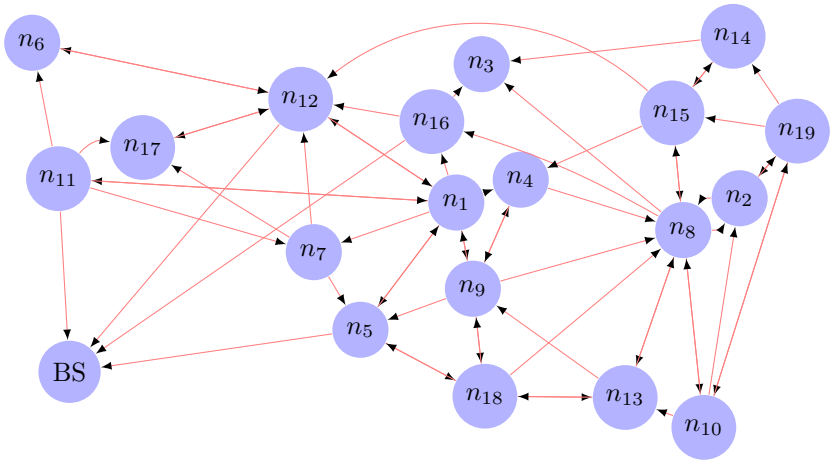


Figure 6.3: A wireless sensor network with 20 sensor nodes

Now, for each $X_{i,d}, 1 \leq d \leq 19$ of individual I_i , a random number is generated to initialize it. Let us assume that an individual $I_i = [1.00, 0.79, 0.20, 0.43, 0.71, 0.62, 0.61, 0.74, 0.11, 0.29, 0.29, 0.33, 0.18, 0.60, 0.46, 0.47, 0.24, 0.57, 0.77]$ has been randomly generated as shown in the second column (i.e., $X_{i,d}$) of Table 6.1. We show that this individual actually represents a candidate solution to both the clustering and routing problems as follows.

Let's consider the generated random number for the first component, 1.00, i.e., $X_{i,1} = 1.00$ as shown in first column of Table 6.1. Hence, $\lceil (X_{i,1} \times |Nbrs(n_1)|) \rceil = 7$, therefore the 7th neighbour from $Nbrs(n_1)$, i.e., n_{12} is selected as a CH for n_1 as shown in Table 6.1. In the same way, each sensor node is assigned to a CH using the randomly generated particle. Then, the CH candidates that result from decoding I_i is $V_i = \{n_{12}, n_{15}, n_8, n_0\}$. Table 6.1 summarize the decoding process for individual I_i.

The final assignment of each node to its next hop for individual I_i and the generated clusters and are given in Fig. 6.4 and Fig. 6.5 respectively.

Table 6.1: Individual decoding process to assign a CH to each node

n_i	$Nbrs(n_i)$	$\lvert Nbrs(n_i)\rvert$	$X_{i,d}$	$\lceil (X_{i,d} \times \lvert Nbrs(n_i)\rvert) \rceil$	CH
s_1	$\{n_9, n_7, n_{16}, n_4, n_{11}, n_5, n_{12}\}$	7	1.00	7	n_{12}
s_2	$\{n_{19}, n_8\}$	2	0.79	2	n_8
s_3	$\{Null\}$	0	0.20	0	None
s_4	$\{n_8, n_9\}$	2	0.43	1	n_8
s_5	$\{n_{18}, n_0, n_1\}$	2	0.71	2	n_0
s_6	$\{n_{12}\}$	1	0.62	1	n_{12}
s_7	$\{n_5, n_{12}, n_{17}\}$	3	0.61	2	n_{12}
s_8	$\{n_3, n_2, n_{10}, n_{16}, n_{15}, n_{13}\}$	6	0.74	5	n_{15}
s_9	$\{n_8, n_1, n_4, n_5, n_{18}\}$	5	0.11	1	n_8
s_{10}	$\{n_2, n_8, n_{13}, n_{19}\}$	4	0.29	2	n_8
s_{11}	$\{n_6, n_{17}, n_7, n_0\}$	4	0.92	4	n_0
s_{12}	$\{n_1, n_0, n_{17}, n_6\}$	4	0.33	2	n_0
s_{13}	$\{n_8, n_9, n_{18}\}$	3	0.18	1	n_8
s_{14}	$\{n_3, n_{15}\}$	2	0.60	2	n_{15}
s_{15}	$\{n_{14}, n_8, n_4, n_{12}\}$	4	0.46	2	n_8
s_{16}	$\{n_3, n_{12}, n_0\}$	3	0.47	2	n_{12}
s_{17}	$\{n_{12}\}$	1	0.24	1	n_{12}
s_{18}	$\{n_{13}, n_9, n_8, n_5\}$	4	0.57	3	n_8
s_{19}	$\{n_{14}, n_{10}, n_2, n_{15}\}$	4	0.77	4	n_{15}

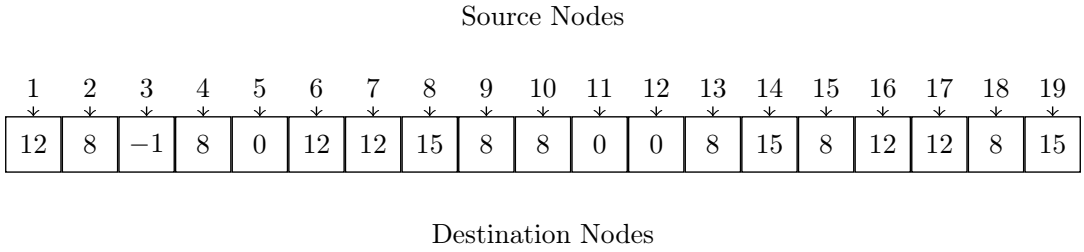


Figure 6.4: Final assignment of the sensor nodes to their respective next hop

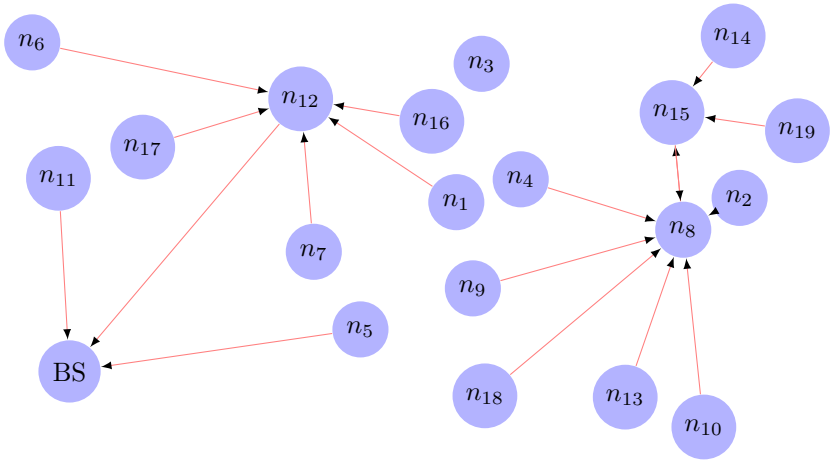


Figure 6.5: The generated clusters that correspond to the final assignments

6.3.2 Routing Tree Construction

The inter-cluster communication is used to carry data from the CHs to the BS. In the proposed approach, a multi-hop model where the CHs form a network among themselves, with each CH node using a multi-hop route for routing data towards the BS.

It should be noted that using the proposed individual encoding scheme also results in the routing tree construction by assigning each CH to its next hop. However, the constructed routing tree is considered not valid if any condition of the following is violated:

- Loop-free routing tree.
- Each route from each CH should terminate at the BS.

Otherwise, the constructed routing tree is valid and can be used for the inter-cluster communication.

Illustration 7.2: Let's consider the final nodes assignments in Fig. 6.4 and the generated CHs set, $V_i = \{n_{12}, n_{15}, n_8, n_0\}$. We notice that CH n_{12} can send to the BS node directly.

However, there is loop between CH n_8 and CH n_{15} and hence the constructed routing tree will be considered not valid and is assigned a high penalty fitness value to exclude this solution from further consideration.

Now, let's suppose that the generated random number for n_{15} was 0.96, i.e. $X_{i,15} = 0.96$ in Table 6.1, then node n_{15} will be assigned to node n_{12} instead of n_8. In this case, the final nodes assignment to their next hop is given in Fig. 6.6. For the CH set $V_i = \{n_{12}, n_{15}, n_8, n_0\}$, using Fig. 6.6 will result in the following routes from each CH, $n_{12} \rightarrow n_0$, $n_{15} \rightarrow n_{12} \rightarrow n_0$ and $n_8 \rightarrow n_{15} \rightarrow n_{12} \rightarrow n_0$.

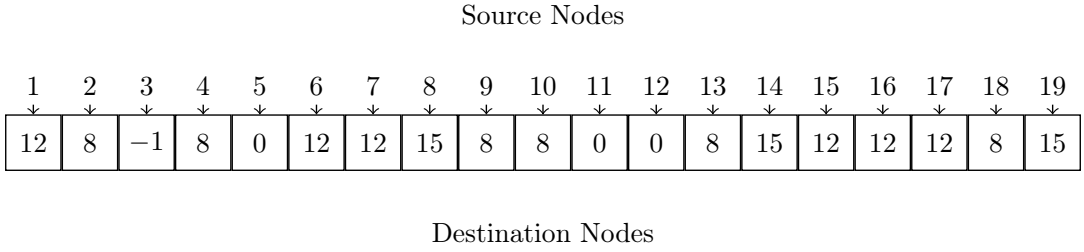


Figure 6.6: Final assignment of the sensor nodes to their respective next hop

This routing tree is considered valid since there is no loops between the CHs and each route from each CH terminates at the BS. The corresponding routing tree is illustrated in Fig.6.7.

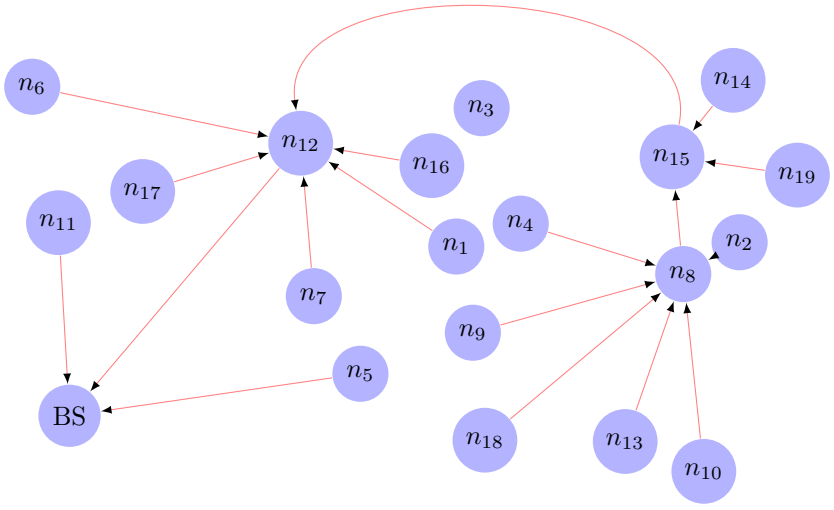


Figure 6.7: Final assignment of the sensor nodes to their respective next hop

Throughout different experimentation, It was found that, regardless of the network density, the proposed approach results in a large number of non-valid routing trees due to existing loops. Hence, there was a need for a repair function to repair the constructed routing tree.

Repair Function based on Dijkstra algorithm

In order to construct a loop-free routing tree, the Dijkstra algorithm was used to find the Shortest Path Tree (SPT) that connects the CHs to the BS.

The network is presented as a weighted directed graph, $G = (V, E)$, where V represents the set of CHs in addition to the BS and E represents the set of edges between them. It is assumed that the best link quality between two nodes represents the shortest path. An edge e from node u to v has weight $w_{u \to v}$, given by:

$$w_{u\to v} = \begin{cases} \frac{RSSI_{u\to v}}{worstRSSI} & \text{if v is neighbor of u.} \\ 0 & \text{if u=v} \\ INF & \text{otherwise} \end{cases} \tag{6.9}$$

Where $RSSI_{u\to v}$ represents the $RSSI$ for the link from u to v. $worstRSSI$ represent the worst $RSSI$ value between any to nodes, and is set to -100. INF represents a very high weight value.

The link quality for the route R from CH ch to the BS is calculated as follows:

$$LQ_{ch\to BS} = \begin{cases} \sum_{e\in R} w_e & \text{If BS is reachable from } ch \\ INF & \text{otherwise} \end{cases} \tag{6.10}$$

It should be noted that the link quality from the sensor u to sensor v is different from the link quality from the sensor v to sensor u. Therefore, at iteration t, the BS generates a dynamic Adjacency Matrix D_t as follows:

$$D_t = \begin{pmatrix} 0 & w_{1\to 2} & \cdots & w_{1\to n} \\ w_{2\to 1} & 0 & \cdots & w_{2\to n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n\to 1} & w_{n\to 2} & \cdots & 0 \end{pmatrix}$$

The BS uses the adjacency matrix D_i to find an SPT that connects the BS and all the CH candidates. The Dijkstra algorithm to find the SPT that connects the CHs to the BS is shown in Algorithm 7.

6.3.3 Individual Evaluations

The next step after initializing the individuals is evaluating the generated clusters and the constructed routing tree according to some objective functions. This step helps periodically

Algorithm 7: The Dijkstra algorithm to find the SPT (and its cost) that connects the CHs to the BS

Input : The directed graph $G = (V, E)$ and the positive edge lengths $\{w_e : e \in E\}$ given by D_t.
Output: The SPT and its associated cost
// For each the CH, $ch \in V$, $lq[ch]$ is the link quality for the route from ch to the BS and calculated by 6.10. The SPT cost is calculated using 6.17.
// Q: Set of unvisited vertices

$lq[BS] = 0$
$prev[BS] = null$
foreach $ch \in V$ **do**
    if $ch! = BS$ **then**
        $lq[ch] = \infty$
        $prev[ch] = null$
    end
    add ch to Q
end
while Q *is not empty* **do**
    $u \leftarrow$ vertex in Q with minimum $rssi(u)$ value
    remove u from Q
    foreach *neighbour* v *of* u **do**
        $alt \leftarrow lq[u] + w_{v \to u}$
        if $alt < lq[v]$ **then**
            $lq[v] \leftarrow alt$
            $prev[v] \leftarrow u$
        end
    end
end
return $prev[]$, $\sum_{ch \in V} lq[ch]$

to update the set of Pareto-optimal solutions and the Pareto front. A detailed description of how the objective functions are calculated is given in section 6.4.

6.3.4 Determining the Best Compromise Individual

Upon obtaining a set of Pareto optimal solutions using MOEAs provide, a mechanism is needed to determine the best compromise solution. Due to the imprecise nature of the decision makers judgment, it is assumed that there is fuzziness in the goal for each objective. This fuzziness is defined by membership functions that represent the degree of fuzziness of some fuzzy sets using values in the range $[0, 1]$.

The fuzzy mechanism looks at the way the solutions are contributing to each objective and assigns a fuzzy variable. It shows a possible way of finding a compromise solution in case solutions are very close to each other. In this thesis, a fuzzy based mechanism [132] is used to find out a compromise solution on the Pareto front. This mechanism has been successfully used in many different applications of MOEAs [106, 102, 133, 134].

In the fuzzy-based mechanism, a membership value for ith objective of jth solution in the Pareto-front is calculated using the membership function as:

$$\mu_i^j = \begin{cases} 1 & \text{if } F_i \leq F_i^{min}. \\ \frac{F_i^{max} - F_i}{F_i^{max} - F_i^{min}} & \text{if } F_i^{min} < F_i < F_i^{max} \\ 0 & \text{if } F_i \geq F_i^{max}. \end{cases} \tag{6.11}$$

μ_i^j indicates how well the jth solution in the Pareto optimal set can satisfy the ith objective. The sum of membership values for all objectives of the jth solution suggests how well it satisfies all the objectives.

Given N solutions in the Pareto-optimal set and M objective functions for each solution, the achievement of each non-dominated solution with respect to all non-dominated

solutions can be calculated using:

$$\mu^j = \frac{\sum_{i=1}^{M} \mu_i^j}{\sum_{j=1}^{N} \sum_{i=1}^{M} \mu_i^j} \tag{6.12}$$

The solution with the maximum value of μ^j is a compromise solution that can be accepted by the decision maker.

6.4 Calculation of the Objective Functions

In this section, the objective functions' formulation for the joint clustering and routing problem in WSN is presented. The main goal of the protocol is to find the optimal set of CHs such that the following objectives are achieved concurrently:

- Minimize the average consumed energy per node in order to maximize the network lifetime.
- Maximize the protocol's scalability.
- Maximize the network throughput.

The joint clustering and routing problem is formulated as a multi-objective minimization problem. The objective functions are constructed to evaluate each candidate solution I_i depending on the following parameters described as follows.

6.4.1 Energy Efficiency

In order to save more energy, fewer sensor nodes need to be active during each round. Our main approach to achieving that is to minimize the number of elected CHs, K. Let vector

V_i denotes the vector that represents the CHs generated from decoding individual I_i, after removing duplicate values. Then, the number of elected CHs is given by:

$$K_{I_i} = |V_i| \tag{6.13}$$

Furthermore, a sensor node with a higher level of energy is a better CH candidate to both aggregate the data and to act as a relay node towards another CH or BS. The objective function is chosen as the reciprocal of the average remaining energy for the CH candidates and is given by:

$$EE_{I_i} = \frac{|V_i|}{\sum_{k=1}^{|V_i|} E(CH_{I_i,k})} \tag{6.14}$$

$E(CH_{I_i,k})$ is the remaining energy of CH number k generated from decoding individual I_i.

6.4.2 Scalability

To increase the protocol's scalability, the clustering process should cluster as much sensor nodes as possible. This, in turn, will avoid creating clusters with one node only. To achieve that, the protocol minimizes the number of un-clustered nodes UN given by:

$$UN_{I_i} = N - \sum_{k=1}^{|V_i|} |C_{I_i}, k| \tag{6.15}$$

N is the total number of nodes in the network. $|C_{I_i}, k|$ is the number of cluster members in the cluster that corresponds to CH number k generated from decoding I_i.

6.4.3 Data Delivery Reliability

In order to increase the network throughput and hence increase the data delivery reliability, two objectives need to be considered simultaneously:

- Minimize the cost of the intra-cluster communication.
- Minimize the cost of the inter-cluster communication.

It should be noted that the cost of the link between any two nodes was given previously as link weights in the Adjacency Matrix D_t.

The intra-cluster communication cost is defined as the total cost of the links between all the cluster members and their correspondent CHs and is given by:

$$CC_{I_i} = \sum_{k=1}^{|V_i|} \sum_{m=1}^{|C_{I_i},k|} w_{cm_m \to CH_{I_i},k} \tag{6.16}$$

The total cost of the constructed tree, the inter-cluster communication cost, is defined as the sum of the costs of links between the CHs forming that tree. In the case that any two CHs are not connected, the constructed tree is assigned a high penalty value to narrow the search to optimal valid tree solutions only. Therefore, the total cost of the constructed tree is calculated as follows:

$$TC_{I_i} = \begin{cases} \sum_{k=1}^{K} \sum_{e=1}^{E} w_e & \text{If all nodes in V are connected} \\ INF & \text{otherwise} \end{cases} \tag{6.17}$$

Where K is the number of CH candidates. E is the number of edges in path number k. w_e is the weight of edge e.

Finally, The protocol objective is to simultaneously minimize K_{I_i}, EE_{I_i}, UN_{I_i}, CC_{I_i}, and TC_{I_i} for individual I_i.

6.5 Experimental Results

In this section, the results of the experiments that are employed to evaluate the proposed approach are presented. The goal of the experiments is to:

- Evaluate the performance of applying both NSGA-II and SMPSO on the formulated joint clustering and routing problem.
- Evaluate the performance of the proposed protocol against the well-known protocols LEACH, EHE-LEACH, EEHC, LEACH-C, PSO-C, and GA-C.
- Evaluate the performance of the proposed protocol against the previously proposed approach TPSO-CR.

This section is divided into two subsections. Firstly, the simulation parameters for both NSGA-II and SMPSO are introduced, and the performance comparison results between them are presented. Secondly, the performance of the proposed clustering approach is compared to the well-known clustering approaches, LEACH, EHE-LEACH, EEHC, LEACH-C, PSO-C, and GA-C. In addition, the performance of the proposed protocol is evaluated against the previously proposed approach TPSO-CR.

6.5.1 Performance Evaluation of NSGA-II and SMPSO

In this subsection, the performance results of applying both NSGA-II and SMPSO, on the formulated joint clustering and routing problem, are compared.

To evaluate the performance of both algorithms, fifty independent runs using different random seeds are performed for a random round of $WSN\#2$. The parameters setting of NSGA-II and SMPSO is given in Table 6.2.

The capability of NSGA-II and SMPSO in comparison to each other is measured using two quality indicators, namely, the hypervolume indicator (HV) and the Epsilon indicator.

Table 6.2: Parameters setting of NSGA-II and SMPSO

Parameter	**Value**
Problem dimension	$NetworkSize - 1$
NSGA-II Parameters Settings	
Population size	100
Number of iterations	250
Crossover probability	0.9
Crossover distribution index	20
Mutation probability	1.0/Problem dimension
Mutation distribution index	20
SMPSO Parameters Settings	
Swarm Size	100
Archive Size	100
Number of iterations	250
Mutation probability	1.0/Problem dimension
Mutation distribution index	20

Table 6.3: Mean and standard deviation for the HV Indicator

Network Size	**NSGA-II**	**SMPSO**
100	$4.92e-02_{3.4e-01}$	$2.41e+02_{9.3e+02}$
200	$1.18e-02_{5.0e-02}$	$1.71e+01_{5.2e+00}$
300	$1.80e-02_{9.1e-02}$	$2.08e+01_{1.1e+01}$
400	$3.07e-03_{3.4e-04}$	$2.07e+01_{1.2e+01}$
500	$1.58e-02_{9.4e-02}$	$2.56e+01_{1.5e+01}$

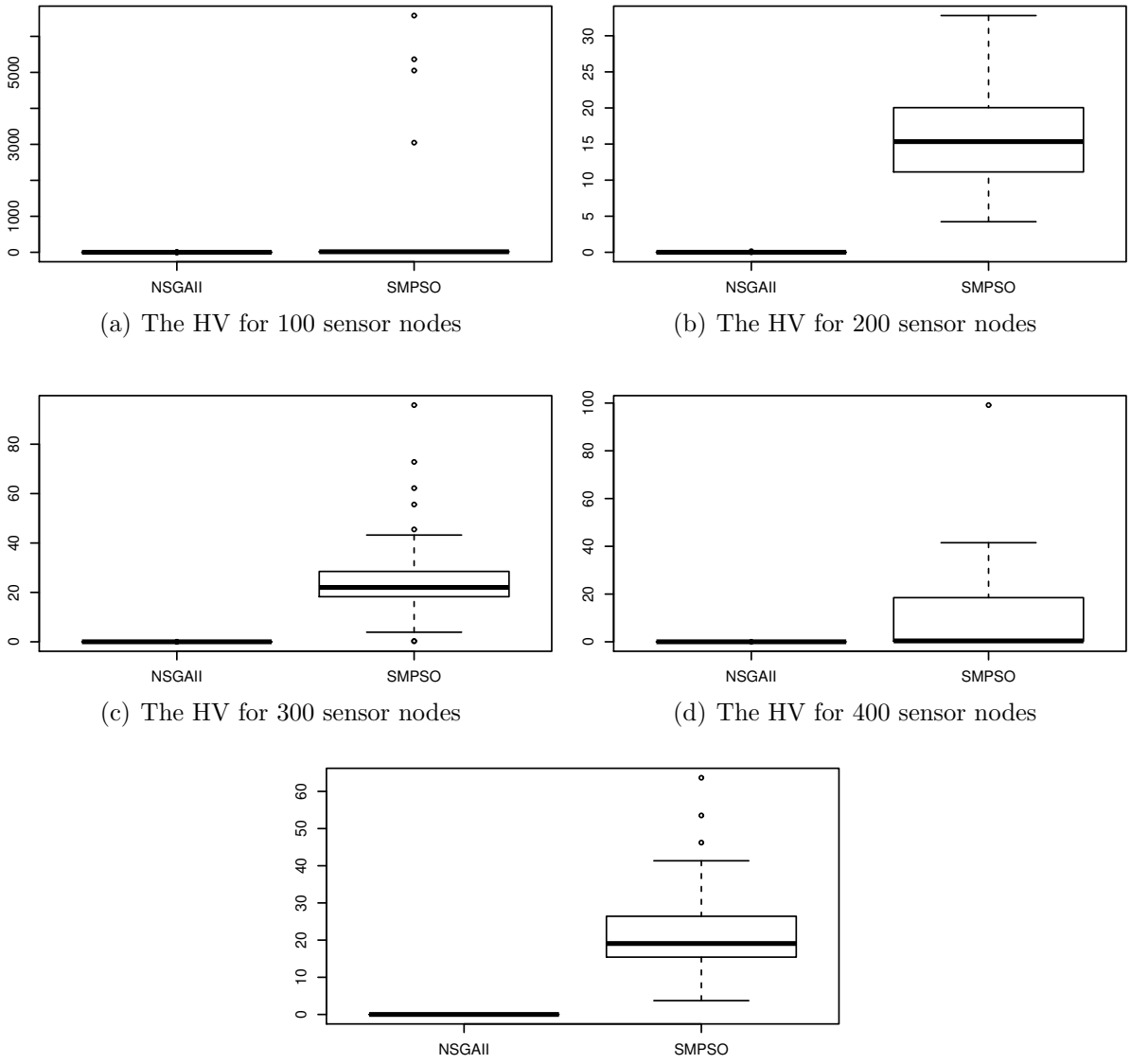
Table 6.3 and 6.4 show the comparisons of the (HV) and Epsilon indicators respectively, for different network sizes. To ease the analysis of these tables, some cells have a gray colored background in each row; particularly, there are two different gray levels: a darker one, pointing out the algorithm obtaining the best value of the indicator, and a lighter one, highlighting the algorithm obtaining the second best value of the indicator.

The boxplots representing the distribution of values for the HV and Epsilon Indicators in the comparison carried out are showed in Fig. 6.9 and Fig. 6.8 respectively, for different network sizes.

It is clearly observed that SMPSO has clearly outperformed NSGA-II, in terms of the HV and Epsilon indicators, for all the network sizes. Hence, it is concluded that SMPSO

Table 6.4: Mean and standard deviation for the Epsilon Indicator

Network Size	NSGA-II	SMPSO
100	$9.27e+01_{2.5e+00}$	$6.65e+00_{2.7e+00}$
200	$2.04e+02_{4.0e+00}$	$7.67e+00_{2.6e+00}$
300	$3.16e+02_{4.6e+00}$	$5.34e+00_{1.7e+00}$
400	$4.32e+02_{5.7e+00}$	$6.20e+00_{2.0e+00}$
500	$5.47e+02_{6.0e+00}$	$6.02e+00_{1.8e+00}$



(a) The HV for 100 sensor nodes

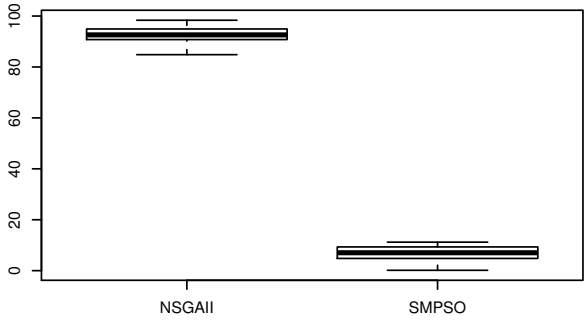
(b) The HV for 200 sensor nodes

(c) The HV for 300 sensor nodes

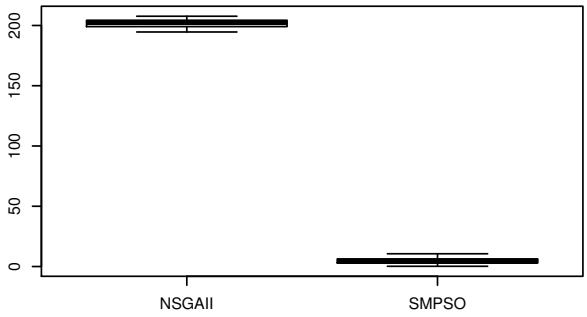
(d) The HV for 400 sensor nodes

(e) The HV for 500 sensor nodes

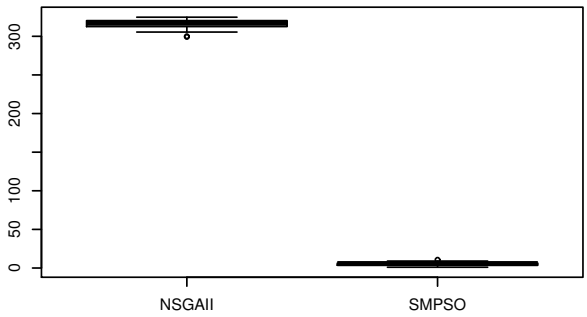
Figure 6.8: Boxplots of the HV obtained by NSGA-II and SMPSO in the evaluated problem, for different network sizes [100 - 500]



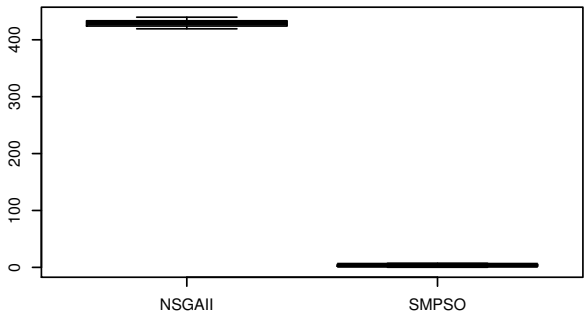
(a) The Epsilon for 100 sensor nodes



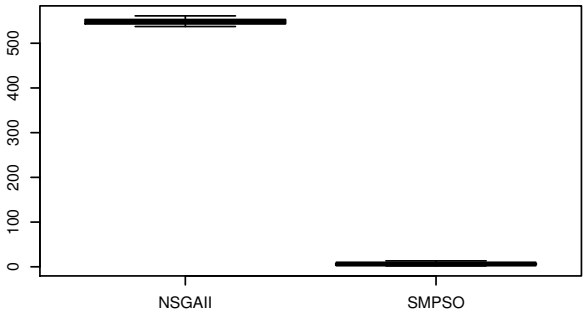
(b) The Epsilon for 200 sensor nodes



(c) The Epsilon for 300 sensor nodes



(d) The Epsilon for 400 sensor nodes



(e) The Epsilon for 500 sensor nodes

Figure 6.9: Boxplots of the Epsilon obtained by NSGA-II and SMPSO in the evaluated problem, for different network sizes [100 - 500]

Table 6.5: The Average number of non-dominated solutions per run

Network Size	NSGA-II	SMPSO
100	24.22	95.76
200	26.26	94.5
300	25.52	85.1
400	24.52	93.62
500	25.56	94.26

outperforms NSGA-II in terms of the diversity of the non-dominated solutions and the convergence towards the true approximated Pareto-front.

The number of non-dominated solutions (NNDS) is another widely used performance metric with larger value representing better performance [135, 136]. Table 6.5 shows the average number of non-dominated solutions per run for both NSGA-II and SMPSO. The computational results show that for the NNDS metric, SMPSO algorithm significantly outperform NSGA-II algorithm.

Table 6.6 and Table 6.7 respectively illustrate the average and minimum values, among all the simulation runs, for the different objective functions. It is clearly shown that SMPSO has obtained the best values for all the objective functions. Both algorithms were able to cluster all the sensor nodes.

6.5.2 Performance Evaluation of the Proposed Protocol

In the previous subsection, SMPSO has proved to have better performance than NSGA-II. Therefore, the performance of the proposed SMPSO-based approach, SMPSO-CR,indicator is evaluated and compared to the well-known protocols LEACH, EHE-LEACH, EEHC, LEACH-C, PSO-C, and GA-C. In addition, the performance of the proposed protocol is evaluated against the previously proposed approach TPSO-CR.

According to the heterogeneity of the sensors, the simulations were performed on two groups of WSNs ($WSNs\#1, WSNs\#2$), each with 25 different playground topologies. The first case assumes homogeneous sensor networks ($WSNs\#1$) while the second experiments

Table 6.6: Average objective functions values for NSGA-II and SMPSO

Network Size	**NSGA-II**					**SMPSO**				
	CH	**SC**	**LQ**	**EE**	**TC**	**CH**	**SC**	**LQ**	**EE**	**TC**
100	58.518	0	0.873	2.62	173.01	**18.486**	0	**0.869**	**2.007**	**30.20**
200	119.87	0	0.873	2.61	255.65	**25.556**	0	**0.868**	**2.019**	**56.495**
300	181.81	0	0.871	2.63	382.7	**37.138**	0	**0.868**	**2.112**	**82.190**
400	243.60	0	0.870	2.62	500.22	**37.824**	0	**0.873**	**2.014**	**79.865**
500	305.42	0	0.871	2.63	624.70	**41.205**	0	**0.870**	**2.097**	**90.468**

Table 6.7: Minimum objective functions values for NSGA-II and SMPSO

Network Size	**NSGA-II**					**SMPSO**				
	CH	**SC**	**LQ**	**EE**	**TC**	**CH**	**SC**	**LQ**	**EE**	**TC**
100	50	0	0.853	2.438	109.71	**10**	0	**0.843**	**1.652**	**24.11**
200	107	0	0.861	2.521	218.36	**15**	0	**0.851**	**1.672**	34.55
300	167	0	0.858	2.545	341.03	**18**	0	**0.854**	**1.797**	**42.69**
400	227	0	0.861	2.544	458.91	**21**	0	**0.856**	**1.674**	**43.54**
500	285	0	0.862	2.562	571.815	**23**	0	**0.855**	**1.813**	**48.89**

($WSNs\#2$) assume heterogeneous sensor networks with advanced nodes of 10% and super nodes of 10%.

Each WSN group consists of 5 different network sizes ranging from 100 to 500 sensor nodes. Overall, the experimental results presented here have been averaged over five simulation runs for each network size, for a total of 50 different networks.

The sensor nodes were deployed randomly in an area of $100m \times 100m$ sensor field. The BS was located at the field's corner at position $(0, 0)$. TMAC that is known for its energy efficiency was used as a medium access control because it adapts a variable sleep schedule that increases the battery utilization [79].

To execute SMPSO-CR, an initial population of 100 particles is considered, and they evolve for 250 iterations. The values of the other SMPSO parameters are taken to be the same as in Table 6.2 and are re-listed in Table 6.8 for convenience. Table 6.8 summarizes the configuration of the network simulation environment.

The results in Table 6.9 and Table 6.10 record the average number of CHs per round for both $WSN\#1$ and $WSN\#2$ respectively, for different network sizes. It can be observed that as the network density increases, SMPSO-CR achieves a lower number of CHs per round. LEACH-C, GA-C, and PSO-C always use a fixed number of CHs (which is equal to 5% of network size) regardless of the network density. As for the EEHC and EHE-LEACH protocols, they showed better performances in the case of $WSN\#2$ because the CHs selection process takes into consideration selecting only nodes with higher residual energy.

Next, the protocols are compared in terms of their scalability by varying the sensor nodes from 100 to 500 on both of the network scenarios, $WSN\#1$ and $WSN\#2$. Figure 6.10 and Figure 6.11 show the comparison of SMPSO-CR against the other competent protocols in terms of the number of non-clustered nodes per round in $WSN\#1$ and $WSN\#2$ respectively. The produced results represent the average of 5 different runs, for each net-

Table 6.8: Simulation settings for SMPSO-CR

Parameter	**Value**
BS location	(0,0)
Data transmission rate	1 packet/s
Network Size	(100 - 500) sensor nodes
Field size	$100m \times 100m$
MAC protocol	TMAC
Simulation time	5000 s
Round length	500 s
Slot length	0.4 s
Parameters Settings for *WSN*#1	
Initial energy	18720 J
Parameters Settings for *WSN*#2	
Percentage of advanced nodes	10% of Network Size
Percentage of super nodes	10% of Network Size
Initial energy of advanced node	18720 J
Initial energy of super node	12480 J
Initial energy of normal node	6240 J
Parameters Settings for SMPSO	
Swarm Size	100
Archive Size	100
Number of iterations	250
Mutation probability	1.0/Problem dimension
Mutation distribution index	20

Table 6.9: Average number of cluster heads per round for $WSN\#1$, for SMPSO-CR

Network Size	EEHC	EHE-LEACH	LEACH	LEACH-C	GA-C	PSO-C	SMPSO-CR
100	18.4	17.22	**4.84**	5	5	5	5.6
200	28.08	28.56	**9.6**	10	10	10	9.72
300	36.72	35.8	15.3	15	15	15	**14.58**
400	42.14	41.18	20.08	20	20	20	**19.16**
500	48.86	46.02	24.46	25	25	25	**24.06**

Table 6.10: Average number of cluster heads per round for $WSN\#2$, for SMPSO-CR

Network Size	EEHC	EHE-LEACH	LEACH	LEACH-C	GA-C	PSO-C	SMPSO-CR
100	6.6	6.26	**4.84**	5	5	5	5.8
200	9.42	**8.9**	9.6	10	10	10	9.7
300	12.86	12.26	15.3	15	15	15	**14.6**
400	15.1	14.18	20.08	20	20	20	**19.14**
500	17.74	16.24	24.46	25	25	25	**24.24**

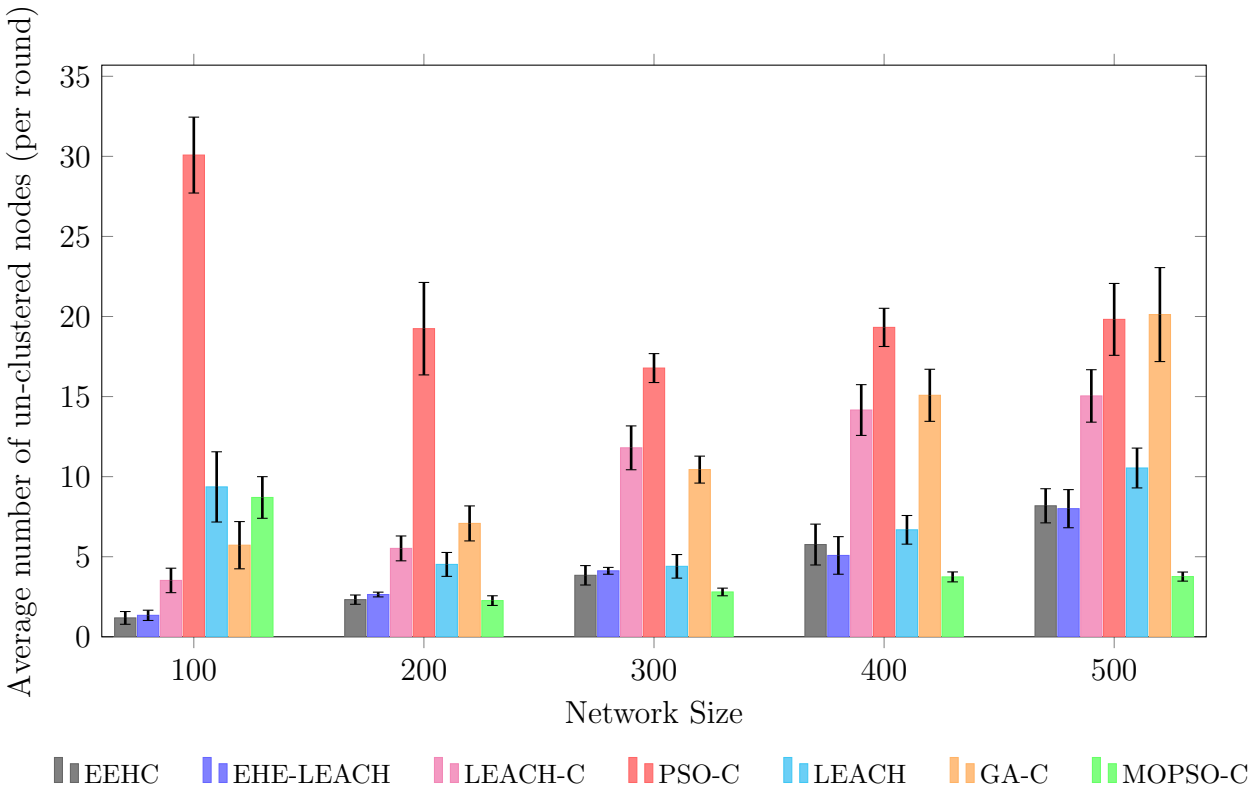


Figure 6.10: Average number of unclustered nodes per round for $WSN\#1$, for SMPSO-CR

work size, with a confidence level of 0.99.

It can be observed from Figs. 6.10 and 6.11 that SMPSO-CR has better scalability than the other competent protocols, especially in the case of densely deployed networks. This result is due to the clustering phase of SMPSO which takes care of minimizing the number of non-clustered nodes (Eq. 6.15). Whereas the other protocols do not deal with that problem.

In order to judge the energy efficiency of SMPSO-CR, Table 6.11 and Table 6.12 record the mean and standard deviation for the average consumed energy per node for $WSN\#1$ and $WSN\#2$ respectively, for different network sizes. It was noted that as the network density increases, SMPSO-CR records lower energy consumption. This is because it also used less number of CHs (and hence less number of active nodes), as illustrated in Table 6.9 and Table 6.10. On the other side, LEACH, EHE-LEACH and EEHC protocols recorded

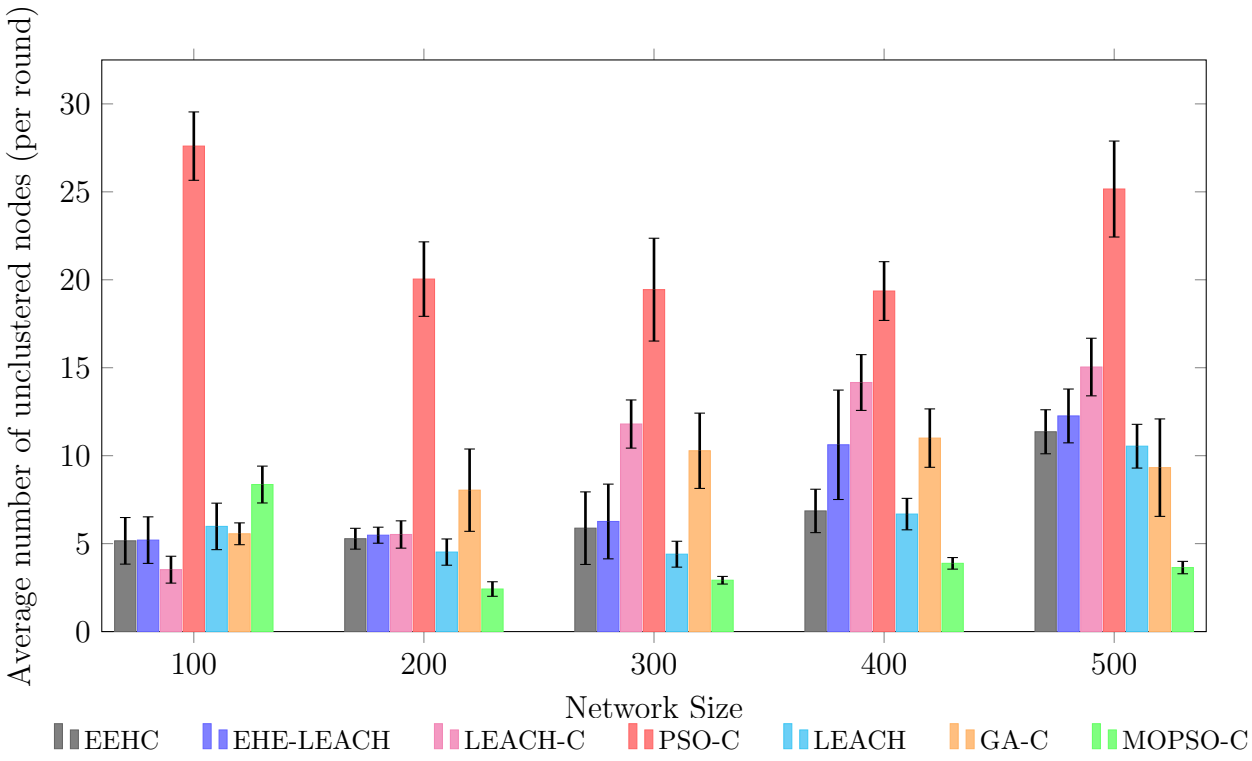


Figure 6.11: Average number of unclustered nodes per round for $WSN\#2$, for SMPSO-CR

higher levels of energy consumption because there are many non-clustered nodes that are left unattended without any sleeping schedule. Although, PSO-C have the worst performance in terms of the number of unclustered nodes; it showed lower energy consumption in comparison to LEACH, EHE-LEACH and EEHC protocols. This is because PSO-C virtually clusters all the network nodes and hence it gives each node a sleeping schedule.

Figures 6.12 and 6.13 show the comparison of SMPSO-CR and other protocols, in term of the network throughput, for $WSN\#1$ and $WSN\#2$ respectively. Throughput is defined as the number of data packets successfully received at the BS. Using the number of aggregated packets delivered to the BS is not accurate, since many packets result from the aggregation process of many raw packets collected from the cluster members. In this thesis, the number of the raw packets is used to calculate the throughput at the BS. The produced results represent the average of 5 different runs, for each network size, with a confidence level of 0.99. It can be observed that SMPSO-CR outperforms the

Protocols	100 Sensor nodes		200 Sensor nodes		300 Sensor nodes		400 Sensor nodes		500 Sensor nodes	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	271.87	5.425	140.56	5.933	122.35	3.914	124.19	5.747	120.66	3.748
EHE-LEACH	176.87	7.484	160.15	2.556	146.38	3.725	139.68	1.602	138.01	2.764
EEHC	179.05	7.393	160.98	2.654	148.19	3.240	141.45	1.490	141.55	2.660
PSO-C	**71.492**	0.131	**71.469**	0.085	71.509	0.038	71.394	0.083	71.460	0.045
GA-C	74.499	0.074	72.660	0.305	71.824	0.304	71.602	0.121	71.336	0.386
LEACH-C	74.554	0.008	73.056	0.005	72.558	0.002	72.309	0.004	72.161	0.002
SMPSO-CR	76.478	3.921	71.521	0.337	**71.026**	0.519	**70.398**	0.348	**70.386**	0.330

Table 6.11: Mean and standard deviation for the average consumed energy per node and standard deviation in $WSN\#1$, for SMPSO-CR

Protocols	100 Sensor nodes		200 Sensor nodes		300 Sensor nodes		400 Sensor nodes		500 Sensor nodes	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
LEACH	170.57	6.035	140.56	5.933	122.35	3.914	124.19	5.747	120.66	3.748
EHE-LEACH	159.53	6.975	137.08	6.594	125.34	6.612	123.64	7.185	124.56	5.516
EEHC	157.85	6.955	134.71	6.029	122.32	7.405	121.78	6.182	124.83	4.085
PSO-C	**71.560**	0.031	**71.441**	0.107	71.506	0.056	71.451	0.084	71.476	0.067
GA-C	74.528	0.035	72.752	0.277	71.979	0.249	71.491	0.032	71.357	0.021
LEACH-C	74.554	0.008	73.056	0.005	72.558	0.002	72.309	0.004	72.160	0.002
MOPSO-C	77.591	5.505	71.550	0.250	**71.063**	0.557	**70.404**	0.368	**70.610**	0.299

Table 6.12: Mean and standard deviation for the average consumed energy per node in $WSN\#2$, for SMPSO-CR

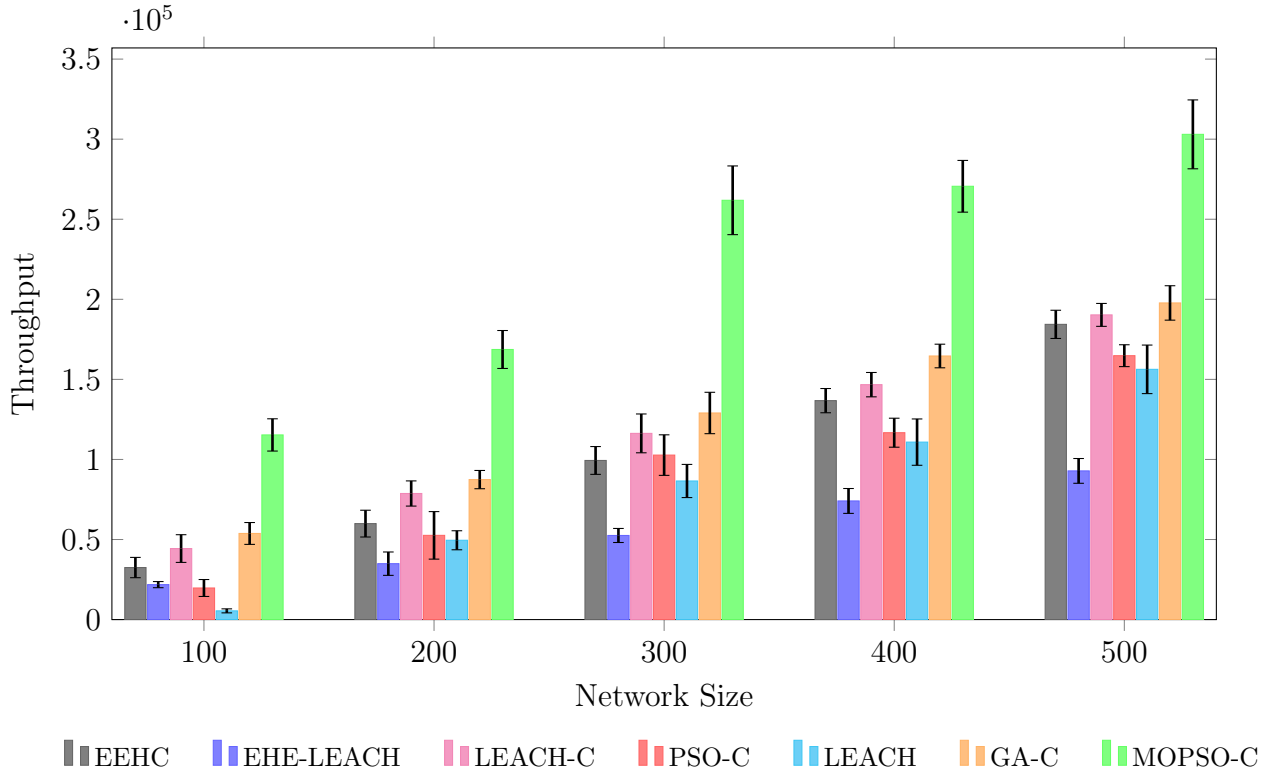


Figure 6.12: Throughput for $WSN\#1$, for SMPSO-CR

other competent protocols in terms of network throughput. This is mainly due to using a dedicated routing tree for the inter-cluster communication.

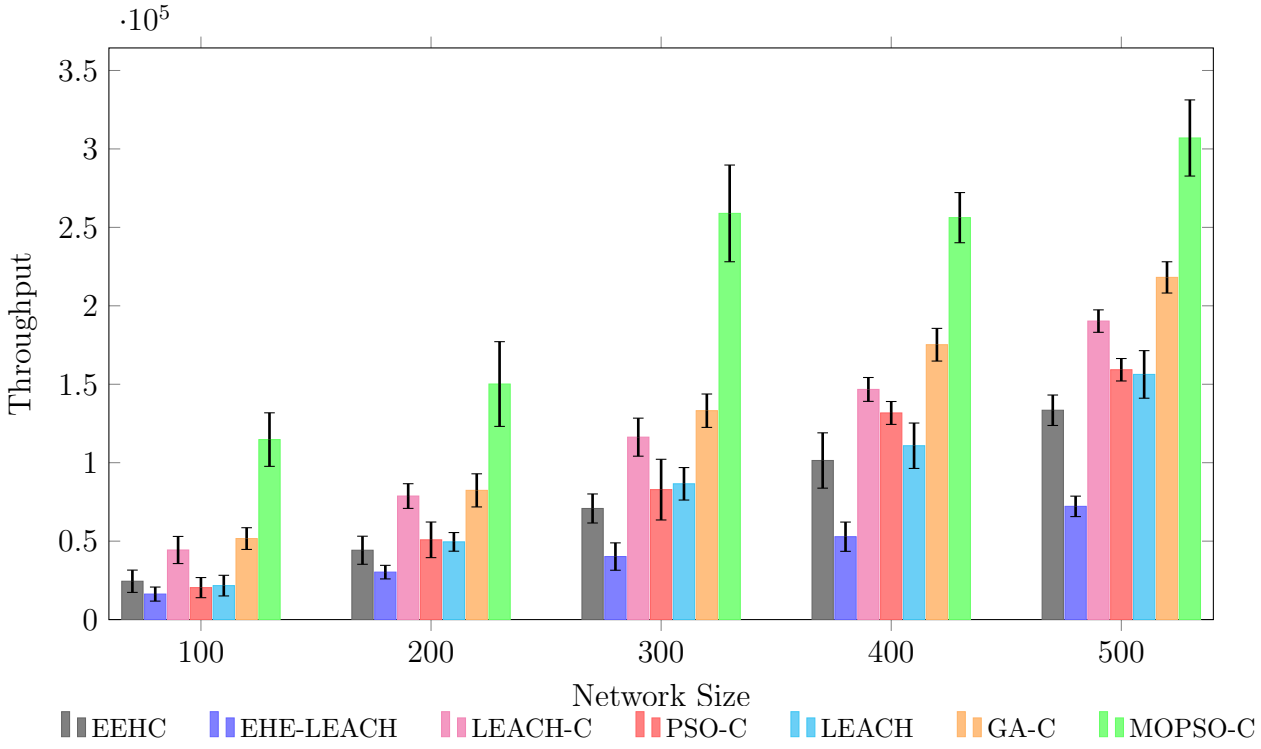


Figure 6.13: Throughput for $WSN\#2$, for SMPSO-CR

In addition to the previous experiments, a comparison between SMPSO-CR and TPSO-CR, in terms of their scalability, energy efficiency and data delivery reliability has been conducted. All the produced results represent the average of 5 different runs, for each network size, with a confidence level of 0.99.

Figure 6.14 and Figure 6.15 show the average number of CHs per round for both $WSN\#1$ and $WSN\#2$ respectively, for different network sizes. The results show that TPSO-CR outperformed SMPSO-CR for most of the cases. TPSO-CR showed better scalability in more than 90% of the networks under test. This is because TPSO-CR uses a larger number of CHs that cover the network.

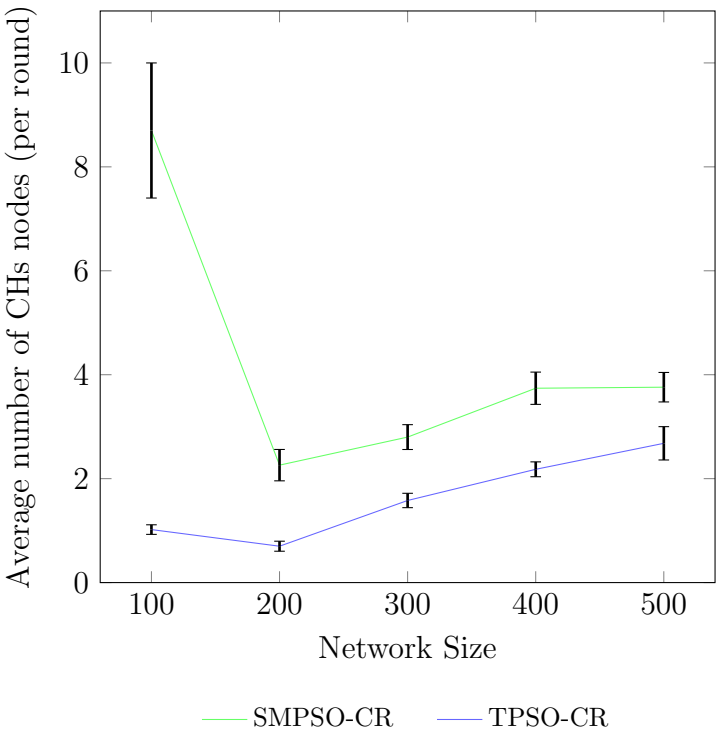


Figure 6.14: Average number of CHs nodes per round for $WSN\#1$, for SMPSO-CR

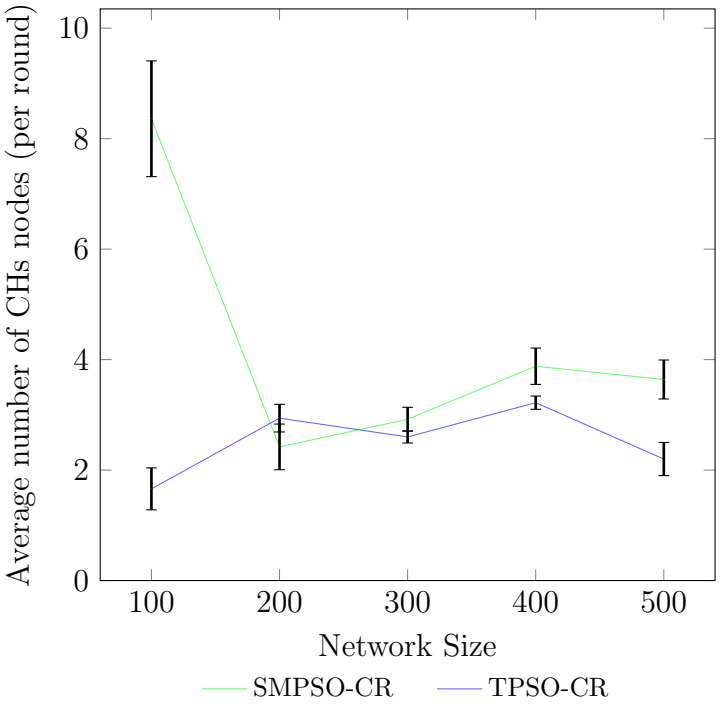


Figure 6.15: Average number of CHs per round for for $WSN\#2$, for SMPSO-CR

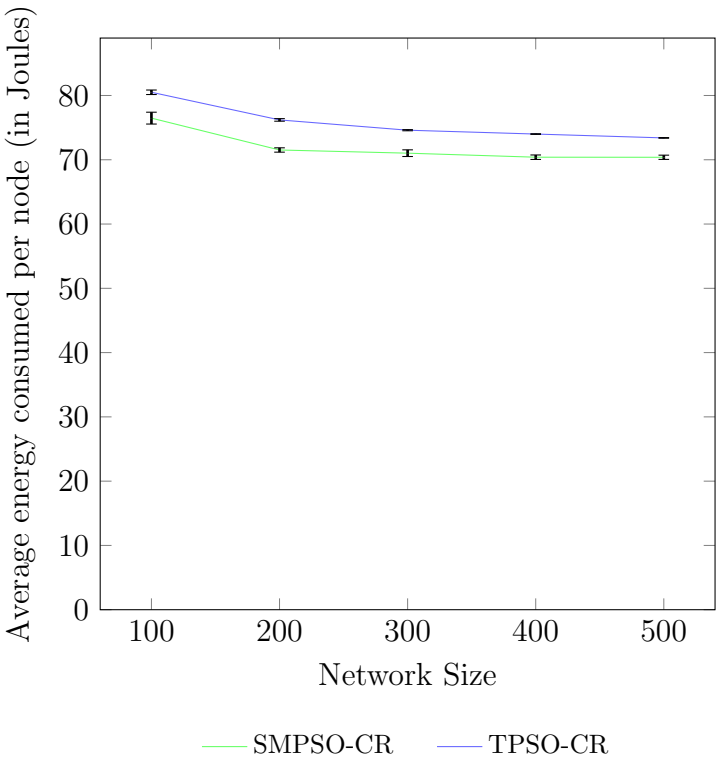


Figure 6.16: Average consumed energy per node for $WSN\#1$, for SMPSO-CR

Figure 6.16 and Figure 6.17 show the average energy consumed per node and their 99% confidence intervals, for both $WSN\#1$ and $WSN\#2$ respectively. It is clearly shown that SMPSO-CR has lower energy consumption than TPSO-CR. This is because SMPSO-CR uses a smaller number of active node per round and it limits the inter-cluster communication to the CHs only. While in TPSO-CR, extra relay nodes can be added in addition to the CHs in order to construct the inter-cluster communication tree.

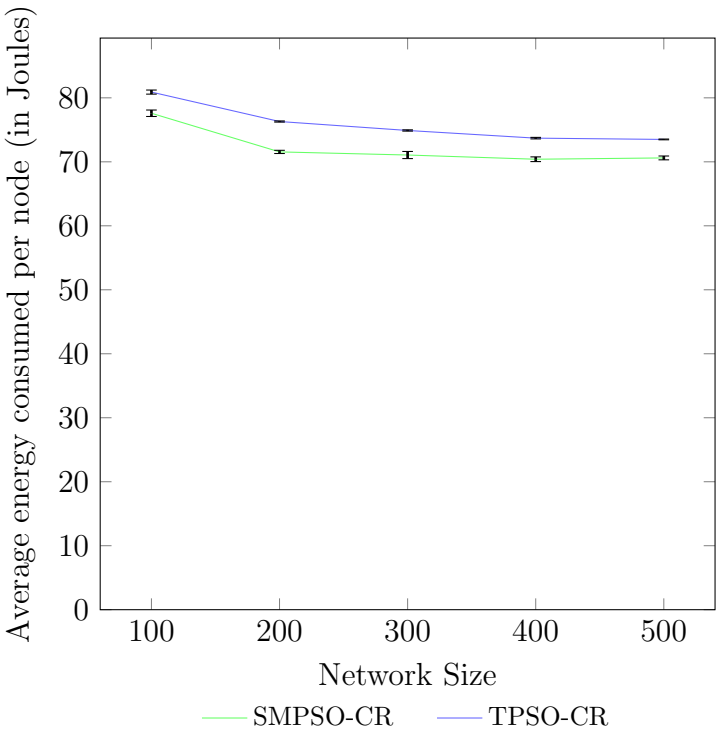


Figure 6.17: Average consumed energy per node for $WSN\#2$, for SMPSO-CR

Fig. 6.18 and Fig. 6.19 show the average network throughput and the 99% confidence interval for these results, for both $WSN\#1$ and $WSN\#2$ respectively. While SMPSO-CR has a higher throughput average for 60% of the cases, the confidence intervals in Fig. 6.18 and Fig. 6.19 show that these results are not statistically significant.

6.6 Conclusion

In this chapter, a centralized multi-objective Pareto-optimization approach was adapted to find a joint solution to both the clustering and routing problems in WSN. A new individual encoding scheme that represents a complete solution for both the clustering and routing problems in WSN was proposed. The problem was formulated as a multi-objective minimization problem aiming at determining an energy efficient, reliable and scalable clustering and routing scheme.

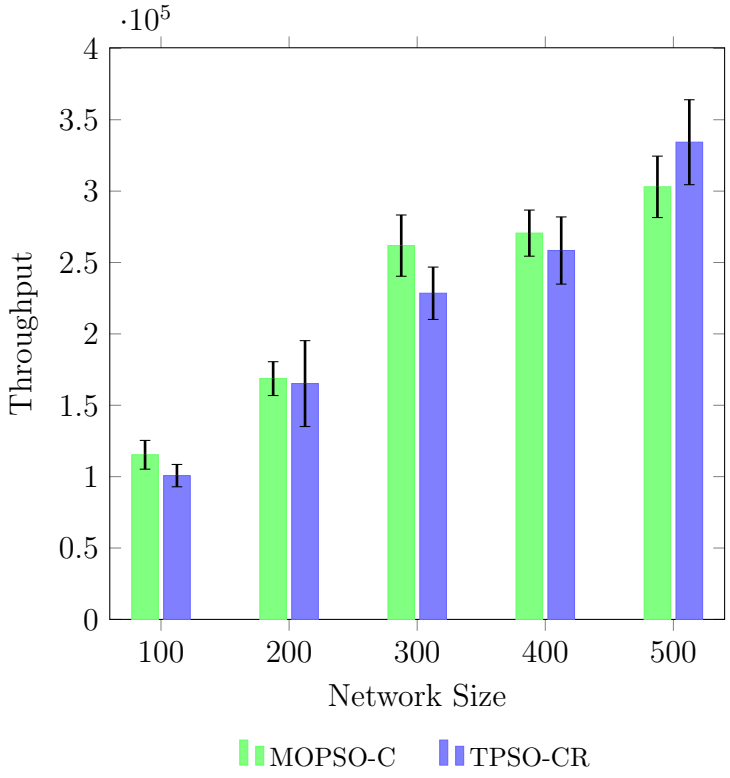


Figure 6.18: Throughput for $WSN\#1$, for SMPSO-CR

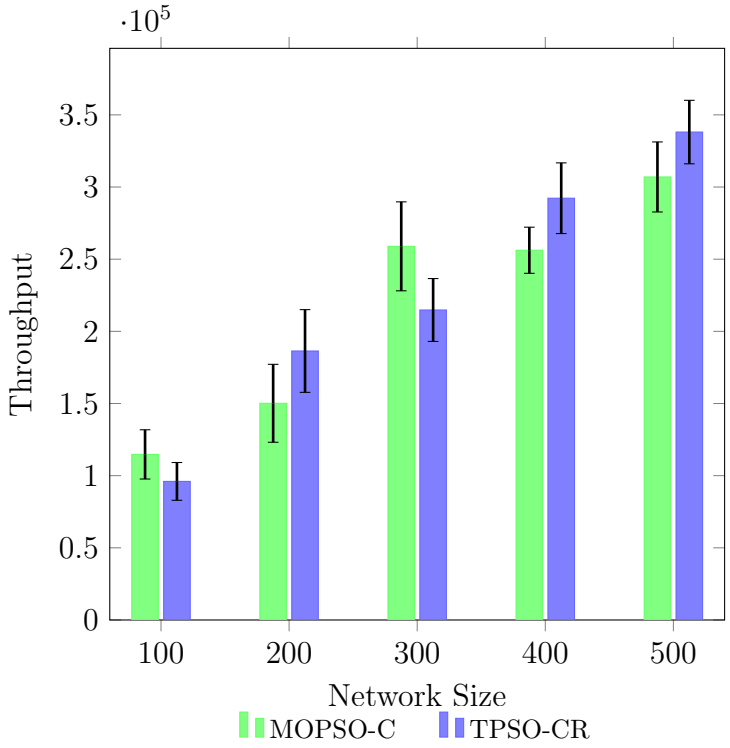


Figure 6.19: Throughput for $WSN\#2$, for SMPSO-CR

The formulated problem has been solved by SMPSO and NSGA-II in order to compare their performance. Simulation results showed that SMPSO outperformed NSGA-II in terms of the number of non-dominated solutions, the objective functions values, the convergence toward the true Pareto-front and the diversity of the obtained solutions.

Furthermore, the performance of the SMPSO-based approach (SMPSO-CR) was evaluated and compared to the well-known protocols LEACH, EHE-LEACH, EEHC, LEACH-C, PSO-C, and GA-C. Experimental results showed that SMPSO-CR protocol outperformed the other protocols in terms of the average consumed energy per node, number of clustered nodes and the throughput at the BS. The experimental results also confirmed that using a smaller number of active nodes (CHs) and restricting the inter-cluster communication to the CHs only enhances the energy efficiency of WSN. Moreover, using a dedicated routing tree enhances the data delivery reliability by maximizing the throughput at the BS.

In addition, the performance of the proposed protocol was evaluated and compared to the previously proposed approach, TPSO-CR. Performance results showed that TPSO-CR has better scalability than SMPSO-CR because TPSO-CR uses a larger number of CHs (5% of the network size). However, SMPSO-CR showed better energy efficiency than TPSO-CR because SMPSO-CR tends to minimize the number of CHs per round, and it limits the inter-cluster communication to the CHs only. However, in TPSO-CR, more nodes in addition to the CHs may be added to construct the routing tree. As for the throughput, SMPSO-CR had a higher throughput average for almost 60% of the cases. However, statistical analysis showed no statistical significance in the obtained throughput results.

Chapter 7

Conclusions and Future Research Directions

7.1 Conclusions

In recent years, wireless sensor networks have been attracting the attention of the research community due to their potential applications in several areas. We have observed that, a flat sensor network architecture poses serious issues on the performance of the network. Under this architecture, the unattended low-powered sensor nodes can deplete their energy quickly resulting in a short network lifetime. Routing protocols that are based on clustering can be used to solve these problems.

Cluster-based routing provide an efficient approach to reduce the energy consumption of the sensor nodes and maximize the lifetime and scalability of WSNs. In WSNs, it is essential to use a routing protocol that is energy efficient, scalable and robust in terms of reliable packet delivery.

Many clustering and routing protocols have been proposed for WSNs. However, the performance of those protocols is limited by problems related to determining an accurate

radio model for the sensor nodes in the network. A discrete radio model should be used for more accurate and realistic calculation of the power consumption.

Energy efficiency, data delivery reliability and scalability are key requirements in WSNs. In this thesis, we have developed a set of clustering and routing protocols to address these requirements.

Clustering and routing in WSNs are two well-known optimization problems and are known to be non-deterministic polynomial (NP)-hard problems. The results of this research show that evolutionary approaches can be applied successfully to these problems. Moreover, experimental results have shown that the PSO algorithm outperforms both the GA and the DE algorithms in terms of the fitness value. Due to its effectiveness in solving NP-hard problems, PSO can be adapted to solve the clustering and routing problems in WSNs.

Experimental results, under a realistic energy consumption model, showed that the number of active nodes has a great impact on the network's energy efficiency. Minimizing the number of active CHs led to minimizing the average of energy consumed per node and in turn maximized the network's energy efficiency. However, increasing the number of CHs and taking link quality measures into consideration resulted in more compact clusters and hence increased the PDR.

Clustering protocols that ignore minimizing the number of un-clustered nodes lead to leaving those nodes unattended, and hence deplete their energy quickly. A sleep scheduling mechanism should be employed to minimize the energy consumption of such nodes.

The main task in clustered WSNs is the data transmission from the CHs to the BS. Many of the prior clustering protocols assumed that the CHs can send their data to the BS directly by maximizing their transmission power. However, this solution is considered an unrealistic assumption in many practical situations due to the communication range restrictions of the sensor nodes. Furthermore, maximizing the transmission range will result

in a high level of energy consumption and will minimize the network's energy efficiency.

Experimental results in this thesis have showed that using a dedicated routing tree results in higher network throughput and hence enhance the network's data delivery reliability. Moreover, limiting the inter-cluster communication to the CHs results in fewer active nodes, and this minimizes the average consumed energy per node and hence enhances the network's energy efficiency.

The clustering problem in WSN consists of multiple conflicting objectives. Furthermore, it can be viewed as a problem that is divided into two sub-problems: finding the optimal set of CHs and finding the inter-cluster communication tree that connects them to the BS. Pareto-optimization approaches can be adapted to find a joint solution to both the clustering and routing problems in WSNs. The SMPSO algorithm and the NSGA-II algorithm are two popular Pareto-optimization techniques.

Experimental results showed that the SMPSO algorithm outperforms NSGA-II in terms of the number of non-dominated solutions, the objective functions values, the convergence toward the true Pareto-front and the diversity of the obtained solutions, when applying them to the joint problem of clustering and routing in WSNs. The experimental results also confirmed that limiting the inter-cluster communication to the CHs only results in fewer active nodes which minimizes the average consumed energy per node and hence enhances the network's energy efficiency. Moreover, using a dedicated routing tree enhances the data delivery reliability by maximizing the throughput at the BS.

7.2 Future Research Directions

During our research work, we have identified several future research directions that can add to or enhance the proposed protocols:

1. A method to significantly reduce the energy consumption in WSNs is applying Trans-

mission Power Control (TPC) techniques to adjust the transmission power [137, 31] dynamically. In the proposed protocols in this thesis, each node transmits packets at the same power level that is normally the maximum possible power level. However, if a node transmits packets at the high power level, it may generate too much interference in the network and consume more energy than necessary. In the case of two nodes that are close to each other, low transmission power is sufficient to communicate with each other. The power level should be high enough to guarantee the transmission and should be low enough to save energy. TPC techniques can be embedded into any existing Medium Access Control (MAC) protocol [138]. As a future research direction, a cross-layer clustering protocol can be proposed such that it takes into consideration finding the optimal CHs and finding the optimal transmission power for each sensor node.

2. The WSNs contains a large number of sensor nodes. As a result, many nodes share the same monitored regions, some of the nodes are redundant and can be turned off to preserve energy while the others still work to offer full coverage [139]. Activating only the necessary sensor nodes at any particular moment can save energy. The Optimal Coverage Problem (OCP) in WSN is defined as finding the smallest set of nodes to monitor an area in order to save energy while meeting the full coverage and connectivity requirements. Sensor scheduling selects only a subset of sensor nodes to be sensing active, such that the area covered by these active nodes can still be the same as the one covered by all nodes. Both network clustering and sensor scheduling can help to conserve energy. As a future research direction, an integrated solution for both problems can be proposed to enhance the network's energy efficiency.

References

[1] S. H. Yang, "Introduction," in *Wireless Sensor Networks*, Signals and Communication Technology, pp. 1–6, Springer London, 2014.

[2] Y. Yu, V. K. Prasanna, and B. Krishnamachari, *Information Processing and Routing in Wireless Sensor Networks*. World Scientific Pub., 2006.

[3] A. M. Zungeru, L. M. Ang, and K. P. Seng, "Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison," *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508–1536, 2012.

[4] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[5] A. K. Dwivedi and O. P. Vyas, "Network layer protocols for wireless sensor networks: Existing classifications and design challenges," *International Journal of Computer Applications*, vol. 8, no. 12, pp. 30–34, 2010.

[6] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[7] L. M. Arboleda and N. Nasser, "Comparison of clustering algorithms and protocols for wireless sensor networks," in *IEEE Canadian Conference on Electrical and Computer Engineering*, pp. 1787–1792, 2006.

[8] A. Dabirmoghaddam, M. Ghaderi, and C. Williamson, "On the optimal randomized clustering in distributed sensor networks," *Computer Networks*, vol. 59, no. 0, pp. 17 – 32, 2014.

[9] E. A. Khalil and B. A. Attea, "Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 195–203, 2011.

[10] P. Kuila and P. K. Jana, "A novel differential evolution based clustering algorithm for wireless sensor networks," *Applied Soft Computing*, vol. 25, no. 0, pp. 414 – 425, 2014.

[11] P. Kuila and P. Jana, "Approximation schemes for load balanced clustering in wireless sensor networks," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 87–105, 2014.

[12] P. Kuila, S. K. Gupta, and P. K. Jana, "A novel evolutionary approach for load balanced clustering problem for wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 12, no. 0, pp. 48 – 56, 2013.

[13] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[14] M. Saleem, G. A. D. Caro, and M. Farooq, "Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions," *Information Sciences*, vol. 181, no. 20, pp. 4597–4624, 2011.

[15] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[16] D. Li, Q. Liu, X. Hu, and X. Jia, "Energy efficient multicast routing in ad hoc wireless networks," *Computer Communications*, vol. 30, no. 18, pp. 3746 – 3756, 2007.

[17] D. Li and Q. Zhu, "Approximation algorithms for multicast routing in ad hoc wireless networks," *Journal of Combinatorial Optimization*, vol. 21, no. 3, pp. 293–305, 2011.

[18] B. Dorronsoro, P. Ruiz, G. Danoy, Y. Pign, and P. Bouvry, *Introduction to Evolutionary Algorithms*, pp. 27–47. John Wiley & Sons, 2014.

[19] Y. Xinjie and G. Mitsuo, "Introduction," in *Introduction to Evolutionary Algorithms*, vol. 0 of *Decision Engineering*, pp. 3–10, Springer London, 2010.

[20] S. M. Zin, N. B. Anuar, M. L. M. Kiah, and A.-S. K. Pathan, "Routing protocol design for secure wsn: Review and open research issues," *Journal of Network and Computer Applications*, vol. 41, no. 0, pp. 517 – 530, 2014.

[21] M. A. Mahmood, W. K. Seah, and I. Welch, "Reliability in wireless sensor networks: A survey and challenges ahead," *Computer Networks*, vol. 79, no. 0, pp. 166 – 187, 2015.

[22] K. Xu and I. Howitt, "Realistic energy model based energy balanced optimization for low rate wpan network," in *IEEE Southeastcon*, pp. 261–266, 2009.

[23] M. Mallinson, S. Hussain, and J. H. Park, "Investigating wireless sensor network lifetime using a realistic radio communication model," in *International Conference on Multimedia and Ubiquitous Engineering*, pp. 433–437, 2008.

[24] S. K. Mitra and M. K. askar, "Comparative study of radio models for data gathering in wireless sensor network," *International Journal of Computer Applications*, vol. 27, no. 4, pp. 49–57, 2008.

[25] F. L. Lewis, *Wireless Sensor Networks*. John Wiley & Sons, 2005.

[26] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.

[27] M. F. Othman and K. Shazali, "Wireless sensor network applications: A study in environment monitoring system," *Procedia Engineering*, vol. 41, no. 0, pp. 1204 – 1210, 2012.

[28] M. Deng and Q. Chen, "Coal and gas outburst monitoring system based on {WSN}," *Procedia Engineering*, vol. 7, no. 0, pp. 387–391, 2010.

[29] M. Lazarescu, "Design of a wsn platform for long-term environmental monitoring for iot applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.

[30] W. D. Chang, T. S. Lin, J. A. Jiang, C. W. Liu, C. P. Chen, D. W. Lai, H. C. Lu, C. W. Yen, and P. L. Yen, "An implementation of a wsn-based medical monitoring system: A pilot study of the blood pressure monitoring of hemodialysis patients," *Engineering in Agriculture, Environment and Food*, vol. 5, no. 3, pp. 83–89, 2012.

[31] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks*, vol. 67, no. 0, pp. 104 – 122, 2014.

[32] F. Aiello, F. Bellifemine, G. Fortino, S. Galzarano, and R. Gravina, "An agent-based signal processing in-node environment for real-time human activity monitoring based on wireless body sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 7, pp. 1147–1161, 2011.

[33] N. F. Gulcharan, H. Daud, N. M. Nor, T. Ibrahim, and M. Z. Shamsudin, "Investigation of stability and reliability of the patient's wireless temperature monitoring device," *Procedia Computer Science*, vol. 40, no. 0, pp. 151 – 159, 2014.

[34] C. Kateretse, G.-W. Lee, and E.-N. Huh, "A practical traffic scheduling scheme for differentiated services of healthcare systems on wireless sensor networks," *Wireless Personal Communications*, vol. 71, no. 2, pp. 909–927, 2013.

[35] R. .G.K and K. Baskaran, "A survey on futuristic health care system: WBANs," *Procedia Engineering*, vol. 30, no. 0, pp. 889–896, 2012.

[36] M. Souil and A. Bouabdallah, "On QoS provisioning in context-aware wireless sensor networks for healthcare," in *Proceedings of 20th International Conference on Computer Communications and Networks*, pp. 1–6, 2011.

[37] N. Suryadevara, S. Mukhopadhyay, S. Kelly, and S. Gill, "Wsn-based smart sensors and actuator for power management in intelligent buildings," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 564–571, 2015.

[38] G. Fortino, A. Guerrieri, G. O'Hare, and A. Ruzzelli, "A flexible building management framework based on wireless sensor and actuator networks," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1934 – 1952, 2012.

[39] A. I. Dounis, "Artificial intelligence for energy conservation in buildings," *Advances in Building Energy Research*, vol. 4, no. 1, pp. 267–299, 2010.

[40] K. Jaafar and M. Watfa, "Sensor networks in future smart rotating buildings," in *IEEE Consumer Communications and Networking Conference*, pp. 962–967, 2013.

[41] M. Tubaishat and S. Madria, "Sensor networks: an overview," *IEEE Potentials*, vol. 22, no. 2, pp. 20–23, 2003.

[42] L. Lee, H. Nwana, D. Ndumu, and P. De Wilde, "The stability, scalability and performance of multi-agent systems," *BT Technology Journal*, vol. 16, no. 3, pp. 94–103, 1998.

[43] G. Molina and E. Alba, "Location discovery in wireless sensor networks using metaheuristics," *Applied Soft Computing*, vol. 11, no. 1, pp. 1223 – 1240, 2011.

[44] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 16:1–16:49, 2010.

[45] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 4, pp. 34:1–34:33, 2012.

[46] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the Third Workshop on Embedded Networked Sensors*, 2006.

[47] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pp. 419–420, 2006.

[48] F. Entezami, M. Tunicliffe, and C. Politis, "Find the weakest link: Statistical analysis on wireless sensor network link-quality metrics," *IEEE Vehicular Technology Magazine*, vol. 9, no. 3, pp. 28–38, 2014.

[49] S. Forrest, "Genetic algorithms," *ACM Computing Surveys (CSUR)*, vol. 28, no. 1, pp. 77–80, 1996.

[50] I. Apetroaei, I. Oprea, B. Proca, and L. Gheorghe, "Genetic algorithms applied in routing protocols for wireless sensor networks," in *10th Roedunet International Conference*, pp. 1–6, 2011.

[51] S. W. Hsiao, F. Y. Chiu, and S. H. Lu, "Product-form design model based on genetic algorithms," *International Journal of Industrial Ergonomics*, vol. 40, no. 3, pp. 237 – 246, 2010.

[52] F. T. S. Chan, T. C. Wong, and L. Y. Chan, "The application of genetic algorithms to lot streaming in a job-shop scheduling problem.," *International Journal of Production Research*, vol. 47, no. 12, pp. 3387 – 3412, 2009.

[53] Y.-R. Wang and S.-L. Kong, "Applying genetic algorithms for construction quality auditor assignment in public construction projects," *Automation in Construction*, vol. 22, no. 0, pp. 459 – 467, 2012.

[54] B. L. Miller and D. F. Goldberg, "Genetic algorithms, selection schemes and the varying effects of noise.," *Evolutionary Computation*, vol. 4, no. 2, p. 113, 1996.

[55] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems*, vol. 9, pp. 115–148, 1995.

[56] S. P. Torres and C. A. Castro, "Differential evolution: A survey of the state of the art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[57] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[58] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon {LEACH} protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 2, pp. 623 – 645, 2013.

[59] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, 2006.

[60] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2, 2000.

[61] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[62] N. Aslam, W. Phillips, W. Robertson, and S. Sivakumar, "A multi-criterion optimization technique for energy efficient cluster formation in wireless sensor networks," *Information Fusion*, vol. 12, no. 3, pp. 202–212, 2011.

[63] P. Poonguzhali, "Energy efficient realization of clustering patch routing protocol in wireless sensors network," in *2012 International Conference on Computer Communication and Informatics*, pp. 1–6, 2012.

[64] D. Wei, Y. Jin, S. Vural, K. Moessner, and R. Tafazolli, "An energy-efficient clustering solution for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 11, pp. 3973–3983, 2011.

[65] M. Ye, C. Li, G. Chen, and J. Wu, "EECS: an energy efficient clustering scheme in wireless sensor networks," in *24th IEEE International Performance, Computing, and Communications Conference*, pp. 535–540, 2005.

[66] D. Kumar, T. C. Aseri, and R. Patel, "Eehc: Energy efficient heterogeneous clustered scheme for wireless sensor networks," *Computer Communications*, vol. 32, no. 4, pp. 662 – 667, 2009.

[67] S. Tanwar, N. Kumar, and J. J. Rodrigues, "A systematic review on heterogeneous routing protocols for wireless sensor network," *Journal of Network and Computer Applications*, vol. 53, no. 0, pp. 39 – 56, 2015.

[68] S. Tyagi, S. Gupta, S. Tanwar, and N. Kumar, "EHE-LEACH: enhanced heterogeneous LEACH protocol for lifetime enhancement of wireless sns," in *2013 International Conference on Advances in Computing, Communications and Informatics*, pp. 1485–1490, 2013.

[69] D. Kumar, "Performance analysis of energy efficient clustering protocols for maximising lifetime of wireless sensor networks," *IET Wireless Sensor Systems*, vol. 4, no. 1, pp. 9–16, 2014.

[70] V. Geetha, P. Kallapur, and S. Tellajeera, "Clustering in wireless sensor networks: Performance comparison of LEACH; LEACH-C protocols using NS2," *Procedia Technology*, vol. 4, no. 0, pp. 163 – 170, 2012. 2nd International Conference on Computer, Communication, Control and Information Technology(C3IT-2012) on February 25 - 26, 2012.

[71] N. Latiff, C. Tsimenidis, and B. Sharif, "Energy-aware clustering for wireless sensor networks using particle swarm optimization," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5, 2007.

[72] N. Abdul Latiff, C. Tsimenidis, and B. Sharif, "Performance comparison of optimization algorithms for clustering in wireless sensor networks," in *IEEE Internatonal Conference on Mobile Adhoc and Sensor Systems*, pp. 1–4, 2007.

[73] C.-J. Jiang, W. ren Shi, M. Xiang, and X. lun Tang, "Energy-balanced unequal clustering protocol for wireless sensor networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 4, pp. 94–99, 2010.

[74] A. Rahmanian, H. Omranpour, M. Akbari, and K. Raahemifar, "A novel genetic algorithm in leach-c routing protocol for sensor networks," in *24th Canadian Conference on Electrical and Computer Engineering*, pp. 001096–001100, 2011.

[75] "Chipcon CC2420 radio transceiver data sheet." http://www.ti.com/lit/ds/symlink/cc2420.pdf. Last access: September 25, 2014.

[76] A. Barberis, L. Barboni, and M. Valle, "Evaluating energy consumption in wireless sensor networks applications," in *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, pp. 455–462, 2007.

[77] A. Rastegarnia and V. Solouk, "Performance evaluation of castalia wireless sensor network simulator," in *34th International Conference on Telecommunications and Signal Processing*, pp. 111 – 115, 2011.

[78] A. Patil and D. P. M. .Hadalgi, "Evaluation of discrete event wireless sensor network simulators," *International Journal of Computer Science and Network*, 2012.

[79] S. Khatarkar and R. Kamble, "Wireless sensor network mac protocol: Smac and tmac," *Indian Journal of Computer Science and Engineering*, vol. 4, no. 4, pp. 304–310, 2013.

[80] A. Boulis, "Castalia user's manual." https://forge.nicta.com.au/docman/view.php/301/592/Castalia+-+User+Manual.pdf, 2011. Last access: April 2, 2014.

[81] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992 – 1007, 2006.

[82] H. L. Liu, Y. Wang, and Y. M. Cheung, "A multi-objective evolutionary algorithm using min-max strategy and sphere coordinate transformation," *Intelligent Automation and Soft Computing*, vol. 15, no. 3, pp. 361–384, 2009.

[83] L. T. Bui and S. Alam, *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*. IGI Global, 2008.

[84] S. Piccand, M. O'Neill, and J. Walker, "On the scalability of particle swarm optimisation," in *IEEE Congress on Evolutionary Computation*, pp. 2505–2512, 2008.

[85] D. Bratton and J. Kennedy, "Defining a standard for particle swarm optimization," in *IEEE Swarm Intelligence Symposium*, pp. 120–127, 2007.

[86] W. Guo and W. Zhang, "A survey on intelligent routing protocols in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 38, no. 0, pp. 185 – 201, 2014.

[87] R. Kulkarni and G. Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 2, pp. 262–267, 2011.

[88] Y. Del Valle, G. Venayagamoorthy, S. Mohagheghi, J.-C. Hernandez, and R. Harley, "Particle swarm optimization: Basic concepts, variants and applications in power systems," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 171–195, 2008.

[89] A. W. Mohemmed, N. C. Sahoo, and T. K. Geok, "Solving shortest path problem using particle swarm optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1643 – 1653, 2008.

[90] C. Chitra and P. Subbaraj, "A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1518 – 1525, 2012.

[91] J. Yao, B. Yang, M. Zhang, and Y. Kong, "Pso with predatory escaping behavior and its application on shortest path routing problems," in *2011 3rd International Workshop on Intelligent Systems and Applications*, pp. 1 – 4, 2011.

[92] A. W. Mohemmed and N. C. Sahoo, "Efficient computation of shortest paths in networks using particle swarm optimization and noising metaheuristics," *Discrete Dynamics in Nature and Society*, vol. 2007, pp. 1 – 25, 2007.

[93] W. Zhao, S. Alam, and H. A. Abbass, "MOCCA-II: a multi-objective co-operative co-evolutionary algorithm," *Applied Soft Computing*, vol. 23, no. 0, pp. 407 – 416, 2014.

[94] M. Gobbi, P. Guarneri, L. Scala, and L. Scotti, "A local approximation based multi-objective optimization algorithm with applications," *Optimization and Engineering*, vol. 15, no. 3, pp. 619–641, 2014.

[95] G. Chiandussi, M. Codegone, S. Ferrero, and F. Varesio, "Comparison of multi-objective optimization methodologies for engineering applications," *Computers and Mathematics with Applications*, vol. 63, no. 5, pp. 912 – 942, 2012.

[96] A. Setmaa-Krkkinen, K. Miettinen, and J. Vuori, "Best compromise solution for a new multiobjective scheduling problem," *Computers and Operations Research*, vol. 33, no. 8, pp. 2353 – 2368, 2006.

[97] K. Deb, "Multi-objective optimisation using evolutionary algorithms: An introduction," in *Multi-objective Evolutionary Optimisation for Product Design and Manufacturing*, pp. 3–34, Springer London, 2011.

[98] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multi-objective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32 – 49, 2011.

[99] C. Coello, G. Pulido, and M. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[100] S. B. Chaudhry, V. C. Hung, R. K. Guha, and K. O. Stanley, "Pareto-based evolutionary computational approach for wireless sensor placement," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 3, pp. 409 – 425, 2011.

[101] J. Jia, J. Chen, G. Chang, Y. Wen, and J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius," *Computers and Mathematics with Applications*, vol. 57, no. 1112, pp. 1767 – 1775, 2009.

[102] A. Mahapatro and P. M. Khilar, "Detection and diagnosis of node failure in wireless sensor networks: A multiobjective optimization approach," *Swarm and Evolutionary Computation*, vol. 13, no. 0, pp. 74 – 84, 2013.

[103] A. Ibrahim, S. Rahnamayan, M. V. Martin, and B. Yilbas, "Multi-objective thermal analysis of a thermoelectric device: Influence of geometric features on device characteristics," *Energy*, vol. 77, no. 0, pp. 305 – 317, 2014.

[104] S. Garca, D. Quintana, I. M. Galvn, and P. Isasi, "Time-stamped resampling for robust evolutionary portfolio optimization," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10722 – 10730, 2012.

[105] S. Wang, S. Ali, and A. Gotlieb, "Cost-effective test suite minimization in product lines using search techniques," *Journal of Systems and Software*, vol. 103, no. 0, pp. 370 – 391, 2015.

[106] P. M. Pradhan and G. Panda, "Connectivity constrained wireless sensor deployment using multiobjective evolutionary algorithms and fuzzy decision making," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1134–1145, 2012.

[107] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[108] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[109] J. Knowles and D. Corne, "The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation," in *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 1, pp. 98–105, 1999.

[110] S. Ramesh, S. Kannan, and S. Baskar, "Application of modified nsga-ii algorithm to multi-objective reactive power planning," *Applied Soft Computing*, vol. 12, no. 2, pp. 741 – 753, 2012.

[111] A. Nebro, J. Durillo, J. Garcia-Nieto, C. Coello Coello, F. Luna, and E. Alba, "SMPSO: a new pso-based metaheuristic for multi-objective optimization," in *IEEE symposium on Computational intelligence in multi-criteria decision-making*, pp. 66–73, 2009.

[112] J. Durillo, J. Garca-Nieto, A. Nebro, C. Coello, F. Luna, and E. Alba, "Multi-objective particle swarm optimizers: An experimental comparison," in *Evolutionary Multi-Criterion Optimization*, vol. 5467 of *Lecture Notes in Computer Science*, pp. 495–509, 2009.

[113] X. Li, "A non-dominated sorting particle swarm optimizer for multiobjective optimization," in *Genetic and Evolutionary Computation*, vol. 2723 of *Lecture Notes in Computer Science*, pp. 37–48, 2003.

[114] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (mopso)," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pp. 26–33, 2003.

[115] M. Sierra and C. Coello Coello, "Improving pso-based multi-objective optimization using crowding, mutation and -dominance," in *Evolutionary Multi-Criterion Optimization*, vol. 3410 of *Lecture Notes in Computer Science*, pp. 505–519, 2005.

[116] G. Pulido and C. Coello Coello, "Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer," in *Genetic and Evolutionary Computation*, vol. 3102 of *Lecture Notes in Computer Science*, pp. 225–237, 2004.

[117] J. E. Alvarez-Benitez, R. M. Everson, and J. E. Fieldsend, "A mopso algorithm based exclusively on pareto dominance concepts," in *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, EMO'05, (Berlin, Heidelberg), pp. 459–473, Springer-Verlag, 2005.

[118] V. Huang, P. Suganthan, and J. Liang, "Comprehensive learning particle swarm optimizer for solving multiobjective optimization problems," *International Journal of Intelligent Systems*, vol. 21, no. 2, pp. 209–226, 2006.

[119] A. Nebro, J. Durillo, and C. Coello, "Analysis of leader selection strategies in a multi-objective particle swarm optimizer," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3153–3160, 2013.

[120] J. J. Durillo, A. J. Nebro, F. Luna, C. A. Coello Coello, and E. Alba, "Convergence speed in multi-objective metaheuristics: Efficiency criteria and empirical study," *International Journal for Numerical Methods in Engineering*, vol. 84, no. 11, pp. 1344–1375, 2010.

[121] J. Durillo, A. Nebro, C. A. Coello Coello, J. Garcia-Nieto, F. Luna, and E. Alba, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 4, pp. 618–635, 2010.

[122] D. Kalyanmoy, *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, 1 ed., 2001.

[123] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[124] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of pareto set approximations," in *Multiobjective Optimization*, vol. 5252 of *Lecture Notes in Computer Science*, pp. 373–404, 2008.

[125] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760 – 771, 2011.

[126] H.-C. Cheng, T.-C. Chiang, and L.-C. Fu, "A two-stage hybrid memetic algorithm for multiobjective job shop scheduling," *Expert Systems with Applications*, vol. 38, no. 9, pp. 10983 – 10998, 2011.

[127] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms a comparative case study," in *Parallel Problem Solving from Nature*, vol. 1498 of *Lecture Notes in Computer Science*, pp. 292–301, 1998.

[128] M. Li, S. Yang, and X. Liu, "Diversity comparison of pareto front approximations in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2568–2584, 2014.

[129] Y. Cao, B. J. Smucker, and T. J. Robinson, "On using the hypervolume indicator to compare pareto fronts: Applications to multi-criteria optimal experimental design," *Journal of Statistical Planning and Inference*, vol. 160, no. 0, pp. 60–74, 2015.

[130] A. Nebro, F. Luna, E. Alba, B. Dorronsoro, J. Durillo, and A. Beham, "Abyss: Adapting scatter search to multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 4, pp. 439–457, 2008.

[131] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[132] B. Panigrahi, V. R. Pandi, R. Sharma, S. Das, and S. Das, "Multiobjective bacteria foraging algorithm for electrical load dispatch problem," *Energy Conversion and Management*, vol. 52, no. 2, pp. 1334 – 1342, 2011.

[133] P. M. Pradhan and G. Panda, "Pareto optimization of cognitive radio parameters using multiobjective evolutionary algorithms and fuzzy decision making," *Swarm and Evolutionary Computation*, vol. 7, no. 0, pp. 7 – 20, 2012.

[134] P. M. Pradhan and G. Panda, "Cooperative spectrum sensing in cognitive radio network using multiobjective evolutionary algorithms and fuzzy decision making," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1022 – 1036, 2013.

[135] M. Rabiee, M. Zandieh, and P. Ramezani, "Bi-objective partial flexible job shop scheduling problem: Nsga-ii, nrga, moga and paes approaches," *International Journal of Production Research*, vol. 50, no. 24, pp. 7327–7342, 2012.

[136] K. Gao, P. Suganthan, Q. Pan, T. Chua, T. Cai, and C. Chong, "Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling," *Information Sciences*, vol. 289, no. 0, pp. 76 – 90, 2014.

[137] "Transmission power control techniques for wireless sensor networks," *Computer Networks*, vol. 51, no. 17, pp. 4765 – 4779, 2007.

[138] J. Vales-Alonso, E. Egea-Lpez, A. Martnez-Sala, P. Pavn-Mario, M. V. Bueno-Delgado, and J. Garca-Haro, "Performance evaluation of MAC transmission power control in wireless sensor networks," *Computer Networks*, vol. 51, no. 6, pp. 1483 – 1498, 2007.

[139] Z.-H. Zhan and J. Zhang, "Bio-inspired computation for solving the optimal coverage problem in wireless sensor networks: A binary particle swarm optimization approach," in *Bio-Inspired Computation in Telecommunications*, pp. 263 – 285, 2015.