

Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software



ENERGY-SAVING SCHEDULING MECHANISM FOR WIRELESS SENSOR NETWORKS

A Dissertation

Submitted to the Council of the College of Information Technology, University
of Babylon in Partial Fulfillment of the Requirements for the Degree of
Doctorate of Philosophy in Information Technology-Software

MAZIN KADHUM HAMEED ALI

Supervised by

Prof. Dr. Ali Kadhum Idrees

2021 A.D.

1443 A.H.

Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Software



بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

وَقُلْ رَبِّ زِدْنِي عِلْمًا

صدق الله العلي العظيم

سورة طه الآية 114

2021 A.D.

1443 A.H.

Supervisor Certification

I certify that the dissertation entitled (**ENERGY-SAVING SCHEDULING MECHANISM FOR WIRELESS SENSOR NETWORKS**) was prepared under my supervision at the department of Software/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Doctor of Philosophy in Information Technology-Software.

Signature:

Supervisor Name: PROF. DR. ALI KADHUM IDREES

Date: / /2021

The Head of the Department Certification

In view of the available recommendations, I forward the dissertation entitled “**ENERGY-SAVING SCHEDULING MECHANISM FOR WIRELESS SENSOR NETWORKS**” for debate by the examination committee.

Signature:

Assist. Ass.Prof. Dr. Ahmed Saleem Abbas

Head of Software Department

Date: / /2021

Certification of the Examination Committee

We hereby certify that we have studied the dissertation entitled (**ENERGY-
SAVING SCHEDULING MECHANISM FOR WIRELESS SENSOR
NETWORKS**) presented by the student (**Mazin Kadhum Hameed**) and
examined him in its content and what is related to it, and that, in our opinion, it
is adequate with (**Excellent**) standing as a dissertation for the degree of Doctor
of Philosophy in Information Technology-Software.

Signature:
Name: Abbas M. Al-Bakry
Title: Prof.Dr.
Date: / / 2021
(Chairman)

Signature:
Name: Samir Jasim
Mohammed AL-muraab
Title: Prof.Dr.
Date: / / 2021
(Member)

Signature:
Name: Ahmed M. Al-Salih
Title: Ass.Prof.Dr.
Date: / / 2021
(Member)

Signature:
Name: Dhiah Eadan Jabor Al-
Shammary
Title: Ass.Prof.Dr.
Date: / / 2021
(Member)

Signature:
Name: Muhammed A. Mahdi
Title: Ass.Prof.Dr.
Date: / / 2021
(Member)

Signature:
Name: Ali Kadhum Idrees
Title: Prof.Dr.
Date: / / 2021
(Member and Supervisor)

**Approved by the Dean of the College of Information Technology, University of
Babylon.**

Signature:
Name:
Title: Prof. Dr. Hussein Attieh
Date: / / 2021
(Dean of Collage of Information Technology)

Dedications

This dissertation is dedicated to the sake of Allah, my Creator and my Master. My humble effort I dedicate to my sweet and loving father and mother. I also dedicate this dissertation to my dearest wife, who leads me through the valley of darkness with light of hope and support.

Mazin Kadhum Hameed

Acknowledgement

First and foremost, I would like to thank my God, Allah Almighty, for giving me endless graces. My deep sense of gratitude goes to the beacon of science, to the master of creatures, to the greatest Prophet, Mohammed (Peace be upon Him and His Family).

I take this opportunity to express my sincere gratitude and greatest appreciation to my supervisor **Prof. Dr. Ali Kadhum Idrees** for his continuous support for my Ph.D. study, his patience, motivation, enthusiasm, and immense knowledge. The words are inadequate and I can't say thank you enough for his tremendous support and help. I feel motivated and encouraged every time I attend his meeting. His tireless guidance has helped me immensely in researching and writing this dissertation.

Also, I would like to show my gratitude to whom weave my happiness from strings woven from her heart, to my dear Father. My thanks and appreciations also go to my Mother, for their encouragement, support and patience.

I wish to express my love and gratitude to my beloved wife for her understanding and endless love through the duration of my study.

Finally, I would also like to express my thanks and gratitude to all those who contributed to making this dissertation possible, and foremost among them all the teaching and staff members at the college of Information Technology at the University of Babylon, headed by **Prof. Dr. Hussein Attieh**.

Abstract

A WSN is a network that consists of a vast number of small, inexpensive, and restricted-power sensors. Sensors are capable of sensing, processing, and transmitting data about their surroundings over a wireless communication to a monitoring center known as the sink (Gateway) for further treatment. Sensors in such area of interest are energy-constrained and their batteries cannot be replaced especially in the remote or hostile environment. Therefore, it is very important to minimize the energy consumption of sensors so as to prolong the network's lifetime as long as possible. For this reason, the scheduling of sensor nodes activities represents an energy-efficient technique for improving the lifetime of the network.

In this dissertation, three different energy-efficient approaches were developed. First, a distributed Density-based spatial clustering of applications with noise(DBSCAN) algorithm for saving the energy of Sensor devices in IoT Networks. This protocol is implemented on each IoT sensor device and the devices apply the DBSCAN algorithm to partition the network into clusters in a distributed way. The efficient periodic and distributed cluster head strategy is proposed.

Then, we propose a Sensor Device Scheduling-based Cuckoo Algorithm (SeDeSCA) for enhancing the lifetime of Cluster-based WSNs. The SeDeSCA technique consists of two phases: clustering and scheduling. For each approach the WSN is clustered into clusters using the DBSCAN algorithm in the first phase. Each cluster choose their cluster head periodically. The elected cluster head executes Cuckoo Algorithm (CA) to solve the scheduling optimization model to select the suitable schedule of sensor nodes that take the mission of sense during

the current period.

Finally, we propose a Hybrid Metaheuristic (CA+GA) scheduling algorithm minimizing the amount of energy consumption and ensuring sufficient coverage for the monitored area while maximizing the network lifespan for WSN. The equilibration of exploration and exploitation abilities was additionally enhanced by incorporating the operators of the genetic algorithm into regular CA, and greater search space was observed during the algorithms' performance. This protocol is based on the same framework of the first scheduling algorithm and the same optimization model.

Extensive simulation experiments were conducted using a custom C++ based simulator to explain the usefulness of the proposed approaches when compared to other existing techniques. The results of these experiments show that the DBSCAN save energy more than other methods. The SeDeSCA seems to provide more favourable coverage efficiency than alternative techniques. It outperforms the other methods in terms of energy consumption and lifespan. The CA+GA tends to result in a more favorable coverage performance than the alternative protocols, maintaining a coverage rate over 80% for many rounds. The network lifespan for Lifetime95 could be enhanced by means of the CA+GA protocol up to 25.1%, 24.7% , 6.9% and 4.5 as compared to the DESK , GAF , PeCO and SeDeSCA protocols respectively. As for Lifetime50, the CA+GA protocol enhances the network lifetime up to 46.8%, 33.3% , 5.6 and 7.4 as compared to the DESK, GAF ,PeCO and SeDeSCA protocols respectively.

Declaration Associated with this Thesis

I hereby declare that this dissertation, submitted to University of Babylon as fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Sciences has not been submitted as an exercise for a similar degree at any other university. I also certify that the work described here is entirely my own except for excerpts and summaries whose sources are appropriately cited in the referenced.

Signature:

Name: Mazin Kadhum Hameed

Date:

Table of Contents

Dedications	iii
Acknowledgement	iv
Abstract	v
Table of Contents	viii
List of Abbreviations	xv
List of Symbols	xvi
CHAPTER ONE.....	XVII
1.1 Overview	1
1.2 Problem Statement	2
1.3 Motivation of the Dissertation	3
1.4 Literature Review	3
1.4.1 Techniques of Clustering	3
1.4.2 Scheduling Mechanisms	7
1.5 Main Objectives of this Dissertation.....	11
1.6 Main Contributions of this Dissertation.....	12
1.7 Organization of the Dissertation	12
CHAPTER TWO	15
2.1 Overview	16
2.2 Wireless Sensor Networks (WSNs)	17
2.3 WSN Applications	18
2.4 Military Applications	19
2.4.1 Health Applications	19
2.4.2 Environmental Applications	19
2.4.3 Applications for Flora and Fauna.....	20
2.4.4 Applications in Industry	20
2.4.5 Urban Applications	21
2.5 WSNs' Challenges.....	21
2.5.1 Energy	21
2.5.2 Self-Management	22
2.5.3 Security	22

2.5.4 Quality of Service (QoS).....	22
2.5.5 Coverage and Connectivity	22
2.5.6 Data Collection and Transmission	23
2.5.7 Deployment.....	23
2.5.8 Heterogeneity	23
2.5.9 Localization Approaches.....	24
2.5.10 Scalability	24
2.6 Energy-efficient techniques in WSNs.....	24
2.6.1 Radio Optimization	25
2.6.2 Data Reduction.....	26
2.6.3 Schemes for sleep and active	26
2.6.3.1 Duty cycling schemes.....	26
2.6.3.2 Passive wake-up radios.....	27
2.6.4 Topology control.....	27
2.6.5 Energy-efficient routing.....	28
2.6.6 Battery repletion.....	28
2.6.7 Clustering techniques.....	28
2.6.7.1 Probabilistic (random or weighted) clustering algorithms	29
2.6.7.1.1 Energy-Efficient Communication Protocol for Wireless Micro sensor Network (LEACH)	29
2.6.7.1.2 An application-specific protocol architecture for wireless microsensor networks (LEACH-C)	30
2.6.7.1.3 A Stale Election Protocol for Clustered Heterogeneous Wireless Sensor Networks (SEP)	30
2.6.7.1.4 Hybrid Energy-Efficient Distributed Clustering (HEED)	31
2.6.7.1.5 Distributed Energy Efficient Hierarchical Clustering for Wireless Sensor Network (DWEHC)	32
2.6.7.1.6 Distributed Energy Efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks (DEEC).....	32
2.6.7.1.7 Distributed Energy Balance Clustering Protocol for Heterogeneous Wireless Sensor Networks (DEBC).....	33

2.6.7.1.8 An unequal cluster-based routing protocol in wireless sensor networks (UCR).....	33
2.6.7.2 Non probabilistic clustering algorithms	34
2.6.7.2.1 Popular Non Probabilistic Clustering Protocols	35
2.6.7.2.2 Highest-Connectivity Cluster Algorithm (HCC).....	35
2.6.7.2.3 Biologically Inspired Clustering Algorithm	36
2.6.7.2.4 Weight-Based Clustering Algorithm	37
2.6.7.3 DBSCAN Clustering	37
2.7 Metaheuristic Methods for Optimization Problem	46
2.7.1 Cuckoo Algorithm	46
2.7.2 Hybrid cuckoo search and genetic algorithm.....	50
2.7.3 Particle swarm optimization (PSO) algorithm.....	54
2.7.4 ANT COLONY OPTIMIZATION	55
2.7.5 Genetic Algorithms	56
2.8 Network Lifetime	65
2.9 Energy Consumption Model	67
2.10 Performance metrics.....	70
2.11 Summary	71
CHAPTER THREE.....	73
3.1 Overview	74
3.2 The Framework of the proposed approaches	74
3.3 Clustering Using DBSCAN Algorithm.....	76
3.4 Scheduling problem Formulation.....	79
3.5 Cluster head selection	80
3.6 Scheduling.....	81
3.6.1 Activity scheduling based Cuckoo Algorithm (CS)	82
3.6.2 Activity scheduling based Hybrid Metaheuristic (CA+GA) Approach for Sensor Node Scheduling.....	85
3.7 Monitoring	87
3.8 Summery	87
CHAPTER FOUR.....	89

4.1 Overview	90
4.2 The Experiment Results of the DBSCAN clustering algorithm	90
4.2.1 Cluster count	91
4.2.2 Average Residual Energy	92
4.2.3 Dead nodes	93
4.2.4 Sent Packets.....	93
4.2.5 Network Lifetime	94
4.2.6 The Experiment Results of the Cuckoo Scheduling Algorithm.....	95
4.2.7 Coverage Ratio.....	96
4.2.8 Active Sensors Ratio.....	97
4.2.9 Energy Consumption	98
4.2.10 Network Lifespan.....	100
4.2.11 Execution Time	101
4.3 The Experiments Result of the Hybrid Metaheuristic (CA+GA) Scheduling Algorithm.	102
4.3.1 Coverage ratio	103
4.3.2 Active Sensor Ratio	104
4.3.3 Energy Consumption	105
4.3.4 Network Lifespan.....	106
4.3.5 Execution Time	107
4.4 Summary	108
CHAPTER FIVE.....	110
5.1 Conclusions	111
5.2 Future Works.....	112
REFERENCES.....	114
Appendix A	134

List of Tables

Table NO.	Table Title	Page NO.
Table 2-1	Main Characteristics of some Clustering Algorithms	39
Table 2-2	Main Characteristics of Some Metaheuristic Algorithms	57
Table 2-3	Commission of Energy	69
Table 4-1	Stimulation Parameters.....	91
Table 4-2	The Different Scenarios for the Network Lifetime	95
Table 4-3	Stimulation Parameters.....	96
Table 4-4	Energy consumption per round for Lifetime ₉₅	99
Table 4-5	Energy consumption per round for Lifetime ₅₀	99
Table 4-6	Network Lifespan for Lifespan ₉₅	100
Table 4-7	Network Lifespan for Lifespan ₅₀	101
Table 4-8	Energy consumption per round for Lifetime ₉₅	106
Table 4-9	Energy consumption per round for Lifetime ₅₀	106
Table 4-10	Network Lifespan for Lifespan ₉₅	107
Table 4-11	Network Lifespan for Lifespan ₅₀	107

List of Figures

Figure NO.	Figure Title	Page NO.
Figure 2-1	Overview of the most popular categories of applications of WSNs[62].....	18
Figure 2-2	Categorization of energy-saving mechanisms [88].	25
Figure 2-3	Example of a network topology control method [88].	29
Figure 2-4	Sigmoid Function[142]	49
Figure 2-5	The model of first-order radio energy for communication [28]	68
Figure 3-1	Flowchart of the proposed approach.	75
Figure 4-1	CH Count (a) LEACH, (b) I-LEACH, (c) DBSCAN	92
Figure 4-2	Network Performance Average Residual Energy	92
Figure 4-3	Lifetime Metrics Dead nodes	93
Figure 4-4	Sent Packets (a) to BS, (b) to CH.....	94
Figure 4-5	Coverage ratio for WSN consisting of 200 deployment nodes.	97
Figure 4-6	Active sensors ratio for WSN consisting of 200 deployed nodes.	98
Figure 4-7	Execution Time (in seconds).....	102
Figure 4-8	Coverage ratio for WSN consisting of 200 deployment nodes.	104
Figure 4-9	Execution Time (in seconds).....	108

List of Algorithms

Algorithm 2-1Cuckoo search algorithm	49
Algorithm 2-2CA-GA algorithm	51
Algorithm 3-1Distributed DBSCAN (sj)	77
Algorithm 3-2Scheduling based CS.....	83
Algorithm 3-3Scheduling based (CA+GA)	86

List of Abbreviations

Abbreviation	Description
WSN	Wireless Sensor Networks
CH	Cluster Head
IOT	Internet Of Thing
DBSCAN	Density-based spatial clustering of applications with noise
minPts	Minimum Points
SeDeSCA	Sensor Device Scheduling based Cuckoo Algorithm
GAF	Geographical Adaptive Fidelity
DESK	Distributed Energy-efficient Scheduling for k-coverage
PeCO	Perimeter based Coverage Optimization protocol
CA	Cuckoo Algorithm
LEACH	Low Energy Adaptive Clustering Hierarchy
LEACH-C	Centralized LEACH
DiLCO	Distributed Lifetime Coverage Optimization
RFID	Radio Frequency Identification
PSO	Particle Swarm Optimization
EACPG	Energy-Aware Clustering Protocol Based Grid
I-LEACH	IOT-LEACH
WPANS	Wireless personal area networks
WLAN	Wireless local area network
MAN	Wireless metropolitan area networks
WAN	Wireless wide area networks
GAN	Global area network

List of Symbols

Symbols	Description
\oplus	Entry-wise multiplication
σ	step-size
λ	Lévy exponent
β and	Lévy distribution
Γ	Gamma function
Max_dis	Maximum ditance
Rs	Range of sensing
Rc	Range of communication
PoP_S	Population Size
$E_{\text{remaining}}$	Remaining energy
E_{initial}	Initial energy
IE _{max}	Maximum initial energy
ϑ	Weight coefficient
δ	Weight coefficient
α	Weight coefficient

CHAPTER ONE

GENERAL

INTRODUCTION

1.1 Overview

Sensors are devices of a smaller size that perform the sensing, processing, and transmission of data within wireless networks, and they are considered to be energy and cost efficient. WSNs are fundamental in many varying types of application, like security, battlefield shriveling, air traffic controlling, bio-detection, environmental monitors, industrial automation, and smart grids[1, 2]. The monitoring of these applications needs multiple sensors to be deployed within the environment. The coverage of the sensor can be defined as the areas that the sensor reaches to be monitored or sensed, as more than one target can be covered at once[3].

Replacing or charging the battery of a sensor is very hard because of the sensor size and that they are used in areas that cannot be reached easily [4]. This explains the importance of the network life span expansion, particularly in WSNs. To do so, a number of factors should be considered, such as deploying sensors optimally[5] [6, 7], sleep scheduling (i.e.) changing the sensor mode from active to sleep [8, 9], and maintaining the balance of load[10].

In a WSN, each distantly deployed sensor node is power-driven by a minor battery. Depending on power necessities of the favorite application, a battery can function for days, months or even years. On the other hand, this energy source is expectedly restricted [11]. There are numerous meta-heuristic methodologies that are broadly implemented for solving several engineering problems such as scheduling, storage and

energy efficiency[12]. the energy-saving protocol suggested by this dissertation concentrates on the sensor activity scheduling problem in sensor networks. The main aim of the scheduling sensor activity is to minimize the number of active nodes to save energy thus increase the network's lifespan while the monitored sensing field maintaining a suitable level of coverage.

1.2 Problem Statement

The sensor nodes are severely restricted by the limited resources such as energy, computation power, and memory. The high-density placement of sensor nodes in a region of interest in a wireless sensor network (WSN) led to increase the transmitted data redundancy to the sink node and increase unnecessary energy consumption since many nodes remain active and they sense the same data. One of the biggest challenges in WSN is the lifetime maximization of the battery. So, it is important in order to reduce the number of active sensors nodes while maintaining a suitable level of coverage for the area of interest. Therefore, we need a protocol for scheduling the sensor activities (sleep/active) in energy efficient way in WSN to minimize the energy consumption while maintaining an acceptable level of coverage, and eventually save the power since the power is the most important constraint in the sensor nodes that must be considered when designing any protocol for WSN.

1.3 Motivation of the Dissertation

We need a protocol for scheduling the sensor activities (sleep/active) in energy efficient way in WSN to minimize the redundant in the data transmitted to the sink while maintaining an acceptable level of data accuracy, and eventually sensor nodes' power is conserved. Since the power is the most important constraint in the sensor nodes that must be considered when designing any protocol for WSN.

1.4 Literature Review

The literature survey can be categorized into two categories clustering techniques and scheduling mechanisms.

1.4.1 Techniques of Clustering

LEACH is regarded as the most classical cluster-based hierarchical routing strategy designed by Heinzelman from MIT[13]. By dividing the entire network into several clusters, LEACH can reduce the number of data transmitted to base station (BS) and save network energy to some extent. However, it utilizes a random way to choose cluster head, which leads the selected cluster head may not be optimal, and even drawbacks like uneven energy consumption, short lifetime, and limited communication range. LEACH-C (centralized LEACH) [14] was a kind of improved LEACH protocol. By minimizing the total sum of squared distances between all cluster members and the closest cluster head, LEACH-C reduced the amount of energy consumption caused by transmitting data from non-cluster-head nodes to their cluster-head.

However, isolated nodes cannot transmit their coordinates and residual energy to the BS, which may result in serious data loss and degrade the network performance. LEACH-F (fuzzy LEACH) [15] was another type of improved LEACH algorithm. Different from LEACH-C, BS offered a cluster-head list for every cluster so that the cluster head was elected in order of the list. Although this method decreased clustering overhead, the unreasonable circumstance that nodes with less remaining power became cluster head may sometimes happen. Arumugam and Ponnuchamy [16] introduced an energy-efficient LEACH (EE-LEACH) protocol for data gathering. It helped to provide better packet delivery ratio with less energy utilization. But it may suffer from security attacks because they only focused on reducing the energy consumption and ignored the confidentiality and integrity of data. Pachlor and Shrimankar [17] presented a centralized routing protocol called Base-station controlled dynamic clustering protocol (BCDCP), which evenly distributed the energy dissipation among all sensor nodes to average energy savings and improve network lifetime. Nevertheless, the design idea and algorithm process of BCDCP were not given in detail. By considering the advantages of channel resource and employing uneven clustering method, Pei et al [18] developed a low-energy adaptive uneven clustering hierarchy (LEAUCH) for cognitive radio sensor network (CRSN). It can not only remarkably balance network load in CRSN but also efficiently extend network life span. However, most of the existing WSNs operate in the crowded public unlicensed spectrum band while LEAUCH was only experimented and evaluated in CRSN; thus, it may not be suitable for real applications in WSNs. LEACH centralized (LEACH-C) has been

proposed also by Heinzelman (2000), [19]. This is a centralized clustering algorithm. It uses the BS to elect CHs. In fact, the BS receives information about the position and energy level of each sensor node in the WSN. Then, BS elects a number of nodes as CH for each round and finally based on minimal power for transmitting, clusters are formed. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) has been elaborated by Lindsey & Raghavendra (2002). This greedy algorithm is based on forming a chain structure from sensor nodes. In fact, each node in the network transmits and receives data only from a neighbor. Only one node is selected from the chain to send data to the sink. It uses data aggregation like LEACH protocol but don't use clustering. The use of chain and the absence of clusters train several threats and attacks and furthermore, communication overhead is increased.

The Hybrid Energy Efficient Distributed (HEED) has been developed by Younis & Fahmy (2004). In this is clustering protocol, probability to elect a CH take into account three factors residual energy, communication cost and average minimum reachability power (AMRP). It uses the same communication method as LEACH protocol but HEED protocol has a well balanced energy and longer network lifetime than LEACH[20]. Reference [21] presents the protocol HEED Non-Probabilistic approach and Fuzzy Logic (HEED-NPF). In this protocol, election of cluster head selection is based on Fuzzy Logic which uses node degree and node centrality as input parameters. The output is the Fuzzy cost. Each node in network chooses the CH with least cost and joins it. This technique is more effective to prolong the lifetime of network than HEED.

The cluster head (CH) has been selected in a random manner, resulting in a similar likeliness for both nodes with high or low energy to become a CH[22]. By integrating grid and density approaches, the authors construct a grid-density clustering algorithm that improves clustering in WSNs [23].

An alternative optimizing method known as Grouped Grey Wolf Search Optimization has been applied in [24] selecting security-aware CH, with the purpose of improving the network lifespan choice. Particle Swarm Optimization(PSO) is discussed in this paper by the authors to create a new coverage control technique. The sensor nodes are first planted at random in a target area and then remain stationary[25].

Jan et al[26]present a new method known as a mutual authentication approach based on payload, as it consists of two steps selecting nodes optimality which act as CH, being allowed to have communication with its neighbors, and the authentication of every CH of its near nodes for forming clusters. Purohit and Bhargava introduced the Multi hop routing scheme [27] where the one hop transmission is transformed into multi hop way, with the purpose of reducing the consumed energy by a sensor.

The work in [28] introduced an improved method of the LEACH named I-LEACH. It is limited the selection of the cluster head using a certain threshold with concurrently changing the level of power between the nodes. The results explain better performance with the original protocol.

1.4.2 Scheduling Mechanisms

Works in [29], [30] and [31] focused on reviewing traditional (not nature-inspired) sleep/active scheduling mechanisms in WSN. Based on our research and investigation the work presented in [32] is the only contribution reviewing nature inspired mechanisms for wake-up scheduling in WSN.

Authors in [33] proposed for randomly distributed WSNs a Huffman coding and Ant Colony Optimization based Lifetime Maximization technique. ACO suggested for multiple paths exploration and Huffman for optimal path selection.

Accordingly, authors in [34] proposed an Ant Colony Optimization (ACO) scheduling approach for network lifetime extension named SAWR that divides sensors into two sets, a first layer set activated when network starts functioning and a successor set scheduled into passive/sleep mode to conserve energy. SAWR defined three phases. The first phase finds the initial active nodes guaranteeing fully coverage. The second phase aims to reach constraints by activating the sensor nodes from successor sensor set.

Authors in [35] suggested for an Industrial WSN context a fuzzy logic-based mechanism that defines the sleeping time of sensor devices according to the battery level and the ratio of throughput to workload. A PSO algorithm is used to find the optimal values and parameters of the proposed Fuzzy Logic Controller.

In order to address the minimization of the message delay issue with energy consideration when a sensor node wake-up after periodic

sleep cycles. Authors in [36] suggested a new approach named DEESL-ABC (Delay Energy Efficient Sleep Scheduling –Artificial Bee Colony).

Authors in [37] proposed an optimization framework combining routing and sleep scheduling algorithms solved by using the evolutionary computing approach called modified semi-adaptive Differential Evolution (MSeDE).

In the same context, authors in [38] conceived new mechanism named CESS-GA (centroid energy-based sleep scheduling algorithm (CESS) using evolutionary algorithms) to select a minimum number of active sensors that provide max network coverage and minimum collected data redundancy with the highest residual power resource.

Reference [39] proposed a two-stage clustering sleep scheduling algorithm with particle swarm optimization (TCSS-PSO) combining sleep scheduling algorithm with a clustering algorithm.

Ren and Meng (2006) have briefly surveyed some existing ACO algorithms, genetic algorithms (GAs), PSO algorithms, reaction – diffusion mechanisms, and other biologically-inspired methodologies proposed for MANETs and wired networks, and have investigated how they could be used in WSNs for routing, clustering, and security[40]. Iyengar et al. (2007) have investigated a couple of algorithms based on GAs as well as various versions of ant-based algorithms, and have considered their use in WSNs[41].

Antoniou et al. [42] have proposed a flock-based congestion control algorithm for WSNs which is able to route data packets over a topological space whilst trying to avoid congested regions.

Particle Swarm Optimisation (PSO) [43] mimics flocking or schooling behaviour in nature and applies it to finding the shortest path in wireless networks.

There are number of works that are related to the present study in some form or another. To start with, The authors proposes a unique sleep-awake mechanism where, firstly, Cluster heads are selected considering the factors like random probabilities, remaining energy of sensor nodes and node degree. Sleep distance is used to decide the SNs, which goes to sleep for a particular round[44].

The authors employ scheduling algorithms to maximize the system's utility. Because the Internet of Things(IOT) hosts a wide range of applications, scheduling algorithms are primarily used to reduce resource consumption and assure effective data collecting. Scheduling algorithms can be applied to maximize the lifespan of wireless sensor networks, which is a key enabler of IoT networks[45].

Sleep scheduling has become a significant mechanism to prolong the lifetime of WSNs and many related methods have been proposed in recent years, which have diverse emphases and application areas. the authors classifies those methods in different taxonomies and provides a deep insight into them[30].

The authors suggest a coverage-control technique (dubbed ESACC) that uses deployment node redundancy to construct a sleep-wake scheduling mechanism[46].

For wireless sensor networks (ESSM), the authors present an energy-efficient sleep scheduling algorithm with a similarity measure, It significantly reduces energy usage by scheduling the sensors into active

or sleep mode [47]. To improve the regional coverage rate and network lifespan of heterogeneous WSN, a sensor node scheduling strategy for heterogeneous wireless sensor networks is proposed [48].

The authors propose a Hybrid Beacon Scheduling (Fast-HyBeS) system to enable periodic reconfiguration of cluster-tree WSNs. The basic idea is to schedule a downstream opportunity window on a regular basis to allow control messages to be distributed more quickly [49]. The goal of this study is to show how to implement a new multi-hop clustering protocol (EACPG) that aims to achieve energy efficiency while maintaining load balancing. The network in EACPG is divided into a large number of virtual square grids. In addition, for cluster head election in a distributed way, certain parameters are examined. [50].

As for [51], an algorithm is suggested for scheduling sensor devices in sub-divided grids in the areas of concern, according to the device's geographical position. In [52], the DESK scheduling algorithm is introduced, which has been applied onto all sensor devices. The decisions that have been made are according to the locally available information from neighbour devices through the perimeter coverage model. The studies conducted in [53] [54] propose two coverage algorithms for the extension of WSN lifespans. The first is DiLCO and makes use of an optimization solver. The coverage optimizing mode depends on essential points when producing the best schedule during the rounds. The other algorithm suggested the maintenance and improvement of coverage and network life span, using the perimeter coverage model.

There are numbers of algorithms introduced for solving scheduling issues in the WSN. They can be divided into the distributed and

centralized scheduling approach. The first one has a rapid performance, as they rely upon locally available information, however no optimal device scheduling can be provided. The second type can suggest an optimal scheduling solution, yet it takes a longer time to be executed when the WS network is larger. A combined type of algorithm involves a global distribution, but is locally centralized and uses an optimization solver. This type improves the WSN lifetime, but their drawback is a rather high time consumption.

1.5 Main Objectives of this Dissertation

The main objectives of this dissertation are summarized as follows:

- Introduce a clustering approach to divide the sensor nodes into many clusters, with one cluster head chosen at the end of each period.
- Suggest an energy-efficient Sensor Scheduling mechanism to schedule sensor nodes (sleep/active) based on spatial correlation between them, reducing redundant sensor nodes and increasing WSN lifetime while maintaining a high level of monitoring on the area of interest
- Propose a new or modified scheduling optimization model to schedule the sensor nodes according to collection of parameters (targets, points, positions). The major goal of this model is to determine which sensors should remain active and which should be turned off depend on the spatial correlation among the sensor nodes.

- Propose a metaheuristic algorithm for dealing with this scheduling issue. The proposed method is to select the appropriate sensor set that is responsible for monitoring throughout the monitoring phase.

1.6 Main Contributions of this Dissertation

The following are the major contributions of this dissertation:

- 1- This dissertation proposes Sensor Device Scheduling-based metaheuristic methods for Enhancing Lifetime of Cluster-based WSNs while maintaining a suitable level of coverage.
- 2- Propose distributed cluster head election approach based DBSCAN clustering algorithm for increasing the lifespan of wireless sensors of IoT Networks.
- 3- Propose a scheduling optimization model for producing the optimal schedule of sensor nodes per period. This model considers three objectives into account such as increase of covering rate over a particular area, minimizing the number of points covering the same node within the sensing field, and the remaining energy of activated sensor nodes.
- 4- Suggest scheduling optimization algorithm based on Cuckoo Scheduling Algorithm and hybrid cuckoo search and genetic algorithm for Enhancing Lifetime of Cluster-based WSNs.

1.7 Organization of the Dissertation

The dissertation is divided into five chapters. Each chapter begins

with a short background that underlines the key contributions and offers an impression of the chapter. The summaries of the chapters are as follows:

- **Chapter 2 - Scientific Background**

In this chapter, an overview of the wireless networks and their types has been presented. This dissertation gives an overview of the applications of WSNs and discusses their major challenges. Also we give some important clustering techniques that used in WSN, and focus on the DBSCAN clustering algorithm. This chapter highlights the data metheuristics optimization methods such as Cuckoo Algorithm and hybrid cuckoo search and genetic algorithm . In addition, the model of energy consumption and the performance metrics that will be used in this dissertation, has been explained.

- **Chapter 3 - Proposed Work**

This chapter presents the proposed a framework for sensor scheduling in WSNs. The design and implementation of the DBSCAN algorithm will be considered. In this chapter we give explicitly explains for the design and implementation of two different metaheuristic approaches which are Cuckoo Algorithm and hybrid cuckoo search and genetic algorithm.

- **Chapter 4 - Results, Analysis, and Discussions**

In this chapter, the performance evaluations for the three proposed approaches as graphs and discussions were presented. To evaluate the performance of proposed approaches several performance parameters were used.

- **Chapter 5 - Conclusion and Future Works**

This chapter presents the conclusion for the dissertation as well as some suggestion for future works.

CHAPTER TWO

SCIENTIFIC

BACHGROUND

2.1 Overview

A computer network that is wireless is referred to as a wireless network in which network nodes communicate via wireless data links [55]. Wireless networking is a means of avoiding the costly procedure of installing cables in a building, or as a link between several equipment sites in homes, businesses, and telecommunications networks [56]. Radio communication is commonly used to establish and administer administrative telecommunications networks[57]. Wireless network types include Wireless personal area networks (WPANs), Wireless local area network (WLAN), Wireless ad hoc network, Wireless metropolitan area networks(MAN) ,Wireless wide area networks (WAN), Cellular network” or mobile network, Global area network (GAN) and Space networks.

For this vast range of connectivity, wireless is the most preferred media. WSNs are a collection of massive sensor nodes spread across a vast area that perceive and gather data and systems from the environment for various applications including weather surveillance, animal tracking, disaster management, bio-medical and IoT applications. As a result, WSN can be regarded as an important component of IoT applications. At the sensor nodes, a lot of processing and calculation is required to glean useful information from a massive volume of data, which are battery-powered and have restricted power. As a result, WSN has various constraints in terms of power, compute ability, and other factors that must be addressed [58]. On the other hand, IoT connects a large number of devices to collect

metadata which can be processed afterwards. This will result in a waste of available power and have an impact on the network's lifespan [58].

2.2 Wireless Sensor Networks (WSNs)

WSNs, which consists of spread sensor and sink nodes, are valuable, effective, and easy data-gathering devices when compared to wired data-gathering devices. Sensor nodes in WSNs perform data perception, data processing, and wireless transmission. They're compact, light, cheap, battery-powered, and simple to set up [59]. Environmental surveillance, Intelligent transport, intelligent healthcare, smart home, and smart manufacturing are some of the industrial and civil domains where WSNs can be used [60]. The energy of nodes limits network coverage and lifetime in WSNs.

Sensor nodes with high network coverage can perceive data from most observing area. Target coverage, regional coverage and barrier coverage are three different types of network coverage based on the observed objects. In sensor nodes that cover all target points are referred to as target coverage. Target coverage can become regional coverage when there are enough target points scattered throughout the monitoring area. Sensor nodes in barrier coverage entirely cover a line. Barrier coverage might become regional coverage when enough lines are available. Regional coverage is an essential form of network coverage. It's used in a variety of fields, including environmental monitoring and intelligent manufacturing. The network lifespan of WSNs is the amount of time they can collect data effectively. The lifespan of sensor nodes

growths as network lifespan increases, and the cost of the application decreases. As a result, the WSN's structure should be maintained the perfect coverage and data gathering for months or years in environmental surveillance and other application fields [61].

2.3 WSN Applications

Various WSN applications are either in the use or in the early phases of development. WSN applications are divided in this study into six broad groups based on their intended usage, The military, health, the environment, flora and fauna, industry and urban as illustrated in the figure 2.1. Various subcategories are taken into account within each category[62].

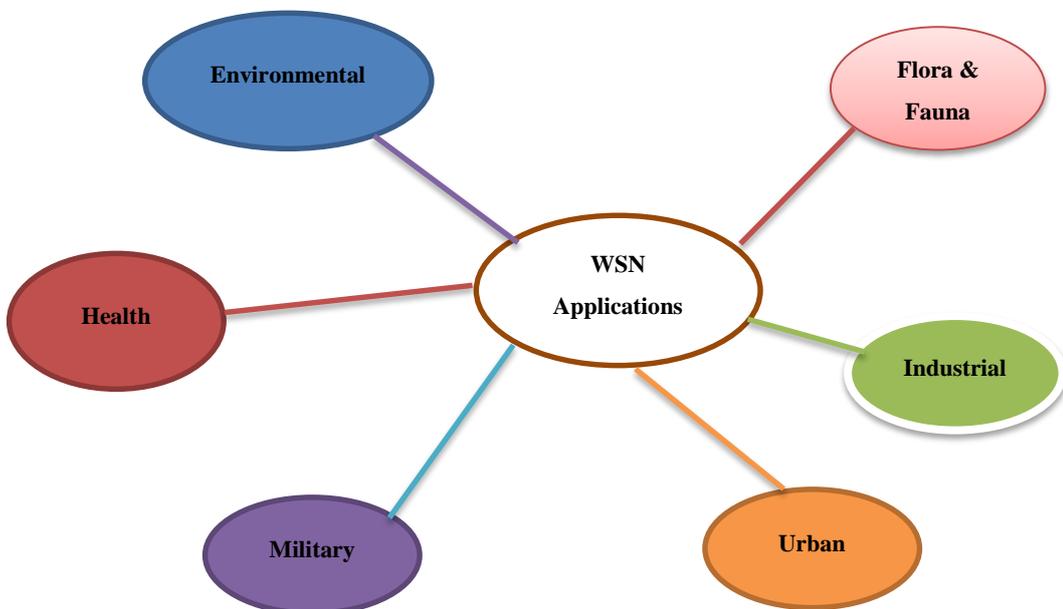


Figure 2-1 Overview of the most popular categories of applications of WSNs[62].

2.4 Military Applications

The military was not just the first area of human activity to deploy WSNs, but it is also said to have sparked sensor network research[62]. Control, Communications, Computing, Intelligence, Monitoring, Recognition and Targeting are highly dependent upon wireless sensor networks. Some military applications are listed below. Intrusion detection, Security, Detection of Nuclear, Biological and Chemical (NBC) threats, Battlefield damage assessment [63].

2.4.1 Health Applications

Patients' vital signs, such as temperature, blood pressure, and heartbeat, are monitored by a WSN in the healthcare industry[64]. The patient is fitted with a wearable or implantable wireless medical sensor that transmits data wirelessly to the base station for healthcare support. Reference [65] constructed the wireless body sensor network in order to communicate with patients in the base station and establish a network. This network also includes four sensors that gather and transfer the patient's physiological signals to the base station. Furthermore, the data is examined by an expert in order to give appropriate health care.

2.4.2 Environmental Applications

Different types of environmental monitoring applications employ WSN. Data collected from the surrounding environment is used to identify floods, earthquakes, volcanic eruptions, and air pollution. For example, By distributing sensor nodes throughout the island, WSN was

deployed on Great Duck Island to detect bird habits and behaviors, as well as activity during climate change [66].

2.4.3 Applications for Flora and Fauna

Every country needs both a flora and a fauna domain. greenhouse monitoring (Kandris, Nakas et al. 2020), crop monitoring (Kandris, Nakas et al. 2020), and animal farming (Kandris, Nakas et al. 2020) are the three primary subcategories of flora and fauna uses of WSNs.

2.4.4 Applications in Industry

Because it is essential for optimizing the performance of global information in real time, in the industrial field, WSN plays an important role, by monitoring the process in the production line, Reduce potential errors, deteriorate device prediction and schedule algorithms for this and other applications [67]. WSN is used in structural monitoring to keep track of flaws in buildings and prevent them from collapsing. WSN has also been established in the United States to monitor the effects of wind and earthquakes on the Golden Gate Bridge by installing multiple sensor nodes to collect monitoring data and compare it to the surrounding environment[68].

In addition, by installing a wireless vibration sensor to the pipes, WSN was adopted by the water supply company to monitor and control water consumption. The user can also examine the sensor data

in order to reduce water consumption. WSN is also capable of detecting leaks in pipelines[68].

2.4.5 Urban Applications

WSNs' wide range of sensing skills allows them to gather unprecedented amounts of data on a goal area, whether it's a room, a construction, or the outdoor. WSNs are a fantastic instrument for determining the geographical and temporal aspects of any phenomena in a city with an almost infinite variety of applications. Intelligent homes, smart cities, transport and structural health surveillance systems are among the most prominent WSN applications in the urban dominion [62].

2.5 WSNs' Challenges

A wireless sensor network's design and performance are influenced by the following major issues:

2.5.1 Energy

Energy efficiency is more crucial for WSN. Sensors require power to perform a variety of tasks. Energy [69, 70] Data collection, processing, and communication all consume energy; additionally, node components (CPUs, radios, and so on) must constantly listen to the medium for reliable operation, even when they are idle.

2.5.2 Self-Management

Once established, wireless sensor networks should be able to function without the assistance of humans. It should be capable of self-configuring, adapting, maintaining, and repairing the network [71, 72].

2.5.3 Security

WSN is used for surveillance, building monitoring, burglar alarms, and essential systems like airports and hospitals, which makes security a difficult issue. In sensor networks, confidentiality is necessary to secure information moving between the network's sensor nodes or between the sensors and the base station; otherwise, the communication could be intercepted.[73-75].

2.5.4 Quality of Service (QoS)

The level of service offered by sensor networks to its consumers is described as quality of service [76, 77]. Because WSNs are utilized in a variety of real-time and mission-critical applications, a high level of QoS is required.

2.5.5 Coverage and Connectivity

Coverage and connectivity are essential in a WSN. The focus of the coverage is on how successfully sensors identify behaviors in the monitored region. If each point in the area is inside a sensing range(s) of the sensor node, it is called completely covered. If at least a path exists between each sensor node and the sink, the network is connected [78].

2.5.6 Data Collection and Transmission

The primary purpose of sensor nodes is to collect data. Sensors collect data from the environment on a regular basis, It will be processed and sent to a base station or sink. Data collection and transmission to the sink node are both part of data gathering. When a sample of data is redundant, there is no need to send it to the sink node because it will waste energy. As a result, care needs to be taken when gathering and transmitting data [79-82].

2.5.7 Deployment

Deployment[83, 84] refers to the installation of a wireless sensor network in a real-world setting. It is a time-consuming and inconvenient task that is dependent on the demographic location of the application and how the network will be installed.

2.5.8 Heterogeneity

It's a collection of nodes that aren't all the same and don't all have the same capabilities, i.e. some nodes are more powerful than others. Cluster design, in which nodes form a cluster head and collect data from less powerful nodes, is an example of heterogeneous grouping. Heterogeneity [85] When two fully distinct WSNs need to communicate with each other, heterogeneity occurs.

2.5.9 Localization Approaches

Localization system must guarantee that maximum nodes in the network must be localized. In some applications, just when known the location of the node, so, the node can be done the monitoring function more effectively. With an unknown environment, how can we get the localization of the node with best permanence. Realizing the localization of the nodes in WSNs becomes a basic challenge of WSNs[86].

2.5.10 Scalability

The majority of applications are required; the number of sensor nodes deployed should be in the hundreds, thousands, or even millions. To respond and function with such a vast number of sensor nodes, the protocols should be scalable[87].

2.6 Energy-efficient techniques in WSNs

This portion discusses the main currently available strategies in which battery-driven nodes can handle their energy consumption problem. Figure 2.3 summarizes the proposed taxonomy of energy-saving methods[88].

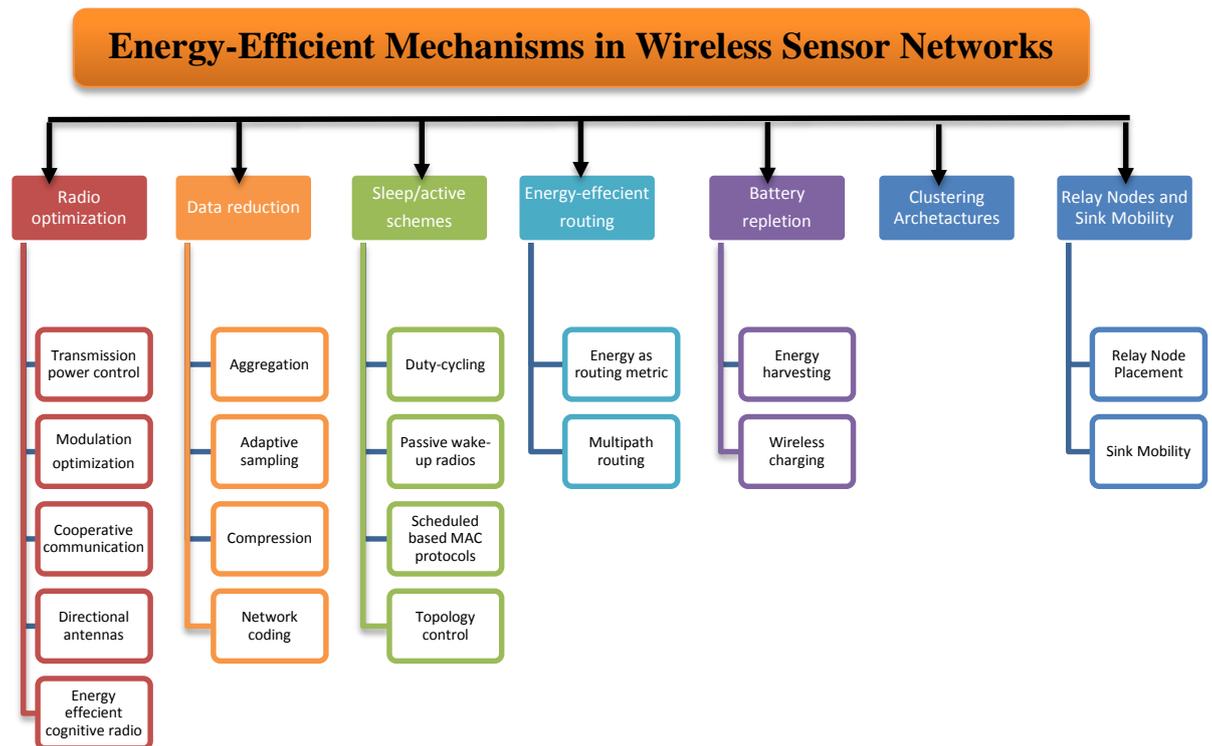


Figure 2-2 Categorization of energy-saving mechanisms [88].

2.6.1 Radio Optimization

The key component that causes sensor node battery depletion is the radio module. Radio characteristics like coding and modulation have been optimized by authors, power transfer and direction of the antenna to reduce energy dissipation by wireless communications (Rault, Bouabdallah et al. 2014). There are some radio optimization methods, which are outlined as below. Modulation optimization (Costa and Ochiai 2010), Cooperative communications schemes (Rault, Bouabdallah et al. 2014), Transmission Power Control (TPC) (Correia, Macedo et al. 2007, Lin, Miao et al. 2016), Directional antennas (Kranakis, Krizanc et al. 2004, Dai 2009) and Energy-efficient cognitive radio [89]

2.6.2 Data Reduction

Another type of solution focuses on the reduction of data sent to the sink. Because data transmission and acquisition are both energy-intensive, two approaches can be used in tandem: the constraint of unnecessary samples and the restriction of sensing jobs [88]. There are some data reduction techniques, which are outlined as below. Aggregation (Rajagopalan and Varshney 2006, Fasolo, Rossi et al. 2007), Adaptive sampling (Anastasi, Conti et al. 2009), Network coding (NC (Wang, Vasilakos et al. 2011) and Data compression (Kimura and Latifi 2005).

2.6.3 Schemes for sleep and active

At the radio component, idle states consume a significant amount of energy. By placing the radio into sleep mode, sleep/active systems try to modify node activity to save energy[88].

2.6.3.1 Duty cycling schemes

To reduce idle listening and to promote sleep mode, Set the radio node state according to the activity of the network. On-demand, asynchronous, and scheduled rendezvous techniques are the three most common types [90] . Although duty cycle-based protocols use the least amount of energy, they hurt from sleep delay because nodes must wait for the receiver to wake up. Furthermore, because not all of a node's neighbors are active at the same time, a node may not always be able to broadcast information to all of them. Finally, It is difficult to configure settings such

as sleep and listening durations, preamble length and slot time because it impacts network performance. [91, 92].

2.6.3.2 Passive wake-up radios

Duty cycling, on the other hand, dispenses energy by needless weakening, low-power radios only use a node if packets are received or transmitted during the transmission process using power-hungry radios. Ba et al. [93] Take a network of WISPMotes and RFID readers, known as passive RFID wake-up radios. A passive RFID waking-up is used to induce a waking interruption the node with the energy distributed by the reader transmitter. All the sensors cannot in practice be fitted with RFID readers because their power consumption is high. This is a big drawback, as it limits their utility to the single-hop scenarios, combined with the slight working range of RFID passive devices.

2.6.4 Topology control

Some nodes can be deactivated while maintenance of network operations and connections if sensors are redundantly installed to provide good space coverage. To decrease the number of active nodes, topology control protocols use redundancies to modify the network architecture dynamically according to the needs of an application. Nodes which do not have to be connected or covered can be switched off in order to extend the network's life. Misra et al. [94] suggest a technique for preserving network coverage while reducing network energy usage by only activating a subgroup of nodes with the least amount of overlap possible.

2.6.5 Energy-efficient routing

The routing is an additional strain that supplies energy. The nodes closest to the sink are more pressured in multi-hop designs because they must route more packets [88]. There are some energy efficient routing methods, which are outlined as below. Cluster architectures [95, 96], Energy as a routing metric [88], Multipath routing [88], Relay node placement [97] and Sink mobility [90, 98].

2.6.6 Battery repletion

Energy harvesting and wireless charging systems have been the subject of several recent research investigations. There are some charging solution methods, which are outlined as below. Energy harvesting [99] and Wireless charging [88]. Both of these systems are promising in that they are designed to charge up sensor batteries without human interaction [88].

2.6.7 Clustering techniques

The architectures of clusters split the network in clusters, each controlled by a node identified as the cluster head (CH). The head of the cluster coordinates and communicates the work of the members [88]. The example of a network topology control procedure is illustrated in Figure 2.3. The various types of clustering techniques in WSNs are described below:

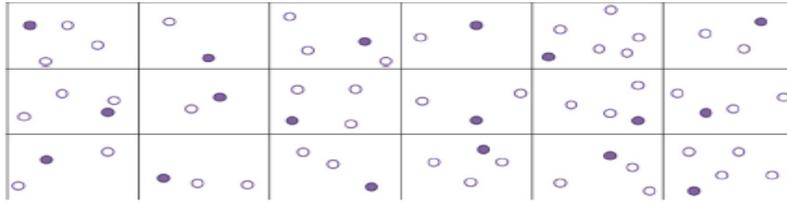


Figure 2-3Example of a network topology control method [88].

2.6.7.1 Probabilistic (random or weighted) clustering algorithms

In the category of probabilistic selection, clustering algorithm a priori probability assigned to each sensor node is used to determine the initial cluster heads or other type random selection procedure [100]. The probabilities initially assigned to each node often serve as the primary criterion in order to decide individually on their election as cluster heads. However, other secondary criteria may also be considered either during cluster head election process i.e. the residual energy, initial energy, average network energy etc. Beyond the high-energy efficiency, the clustering algorithms of this category usually achieve faster execution or convergence times and reduced volume of exchanged messages.

2.6.7.1.1 Energy-Efficient Communication Protocol for Wireless Micro sensor Network (LEACH)

W. B. Heinzelman et al. [13] proposed first well known clustering protocol LEACH for wireless sensor networks. In this sensors are organized into clusters and randomly select a few nodes as cluster head with a certain probability of becoming a cluster heads per round. The task

of being a cluster head is rotated between nodes. The rotation role balances the energy dissipation of the nodes in the networks. LEACH is a distributed algorithm but cluster count (cluster head) is not fixed in each round per epoch. Due to distributed algorithm, each node is capable to select itself as a cluster head by choosing random number. There is possibility that each node choose same number for cluster head selection, due to randomness property of random number generator. Therefore, cluster head count is varying in each round.

2.6.7.1.2 An application-specific protocol architecture for wireless microsensor networks (LEACH-C)

W. B. Heinzelman et al. [14] this protocol uses a centralized approach where the information of node location and energy level was communicated to base station. The base station decides about the cluster head selection and cluster formation. In this protocol the selection of cluster heads is random and the cluster head number is limited. The base station sure those nodes have less energy than it cannot become a cluster head. This protocol is not suitable for large scale network because there is a problem to send the status of a node which are far from the base station. The cluster head role rotates every time so it is not feasible to send information every time in a quick time. It increases the latency and delay.

2.6.7.1.3 A Stale Election Protocol for Clustered Heterogeneous Wireless Sensor Networks (SEP)

Georgios S. et al. [101] introduces the heterogeneity that prolongs the time interval before the death of first node called stability period. This

protocol is based on the weighted election probabilities of each node to become cluster head according to the remaining energy in each node. In this there are two types of nodes was considered as normal and advanced. This protocol does not require global knowledge of energy at every round to select cluster heads. Authors extended the LEACH protocol except the heterogeneity awareness. Cluster count is variable in this algorithm and unstable period is not good.

2.6.7.1.4 Hybrid Energy-Efficient Distributed Clustering (HEED)

O. Younis et al. [20] improves the LEACH protocol by using residual energy, node degree or density as a main parameters for cluster formation to achieve power balancing. This protocol was proposed with three main parameters: First parameter is to enhance network lifetime by distributing energy consumption, second clustering terminates within a fixed number of iterations third minimum control over head and fourth the cluster heads was well distributed. The algorithms proposed in this protocol periodically selects cluster heads based on the two basic parameters. The first primary parameter is the residual energy of each node; second parameter is the intra-cluster communication cast as a function of cluster density or node degree. The primary parameter selects initial set of cluster heads probabilistically which secondary parameter is breaking ties. HEED is not able to fix the cluster count in each round and it is also not aware of heterogeneity.

2.6.7.1.5 Distributed Energy Efficient Hierarchical Clustering for Wireless Sensor Network (DWEHC)

P Ding et al. [102] proposed a distributed weight based energy efficient hierarchical clustering protocol which aims at high energy efficiency by generating balanced cluster sizes and optimizing the intra cluster topology. Each sensor node calculates its weight after finding the neighboring nodes in its area. The weight is a function of the sensors residual energy and the proximity to the neighbors. In a neighborhood, the node with largest weight would be elected as a cluster head and the remaining nodes become members. At this stage the nodes are considered as first level members because they have a direct link to the cluster head. A node progressively adjusts such membership to reach a cluster head using the least amount of energy. Basically, a node checks with its non-CH neighbors to find out their minimal cost for reaching a cluster head. Given the node's knowledge of the distance to its neighbors, it can reach the cluster head over a two-hop path. The protocol is not performing well in term of stability period due to huge amount of energy consumed in neighbors finding.

2.6.7.1.6 Distributed Energy Efficient Clustering Algorithm for Heterogeneous Wireless Sensor Networks (DEEC)

Li Qing et al. [103] proposed a distributed multilevel clustering algorithm for heterogeneous WSN. In DEEC the cluster heads are selected by a probability based on the ratio between residual energy of each node and the average energy of the network. The approach of being cluster heads for nodes are different according to their initial and residual energy.

The authors have assumed that all the nodes of the sensor network are equipped with different amount of energy. Two levels of heterogeneous nodes are considered in the algorithm and after that a general solution for multi-level heterogeneity is obtained. To avoid that each node needs to know the global knowledge of the networks, DEEC estimates the ideal value of network life-time, which is used to compute the reference energy that each node should expend during a round. Cluster count is variable in this approach lead to uneven clusters.

2.6.7.1.7 Distributed Energy Balance Clustering Protocol for Heterogeneous Wireless Sensor Networks (DEBC)

Changmin D et al. [104] proposed a protocol for heterogeneous wireless sensor network. The selection of cluster heads depends on the probability based on ratio between residual energy of node to the average energy of network. The high initial and residual energy nodes have more chances to become cluster heads then the nodes have low energy. This protocol improves the LEACH and SEP protocol by considering two level heterogeneity and extends up to multihop heterogeneity.

2.6.7.1.8 An unequal cluster-based routing protocol in wireless sensor networks (UCR)

Guihai chen. et al. [105] proposed a protocol for mitigating the hot spot problem in WSNs. It is designed for source driven sensor network applications, such as detection of periodical data from environment. It is a self organized competition based algorithm, where selection of cluster heads based on local information as residual energy of neighboring nodes.

The cluster heads closer to base station are expected to have smaller cluster sizes, than those for their from the base station, thus the cluster heads will consume lower energy during the intra cluster data processing and can presume some more energy for the inter cluster relay traffic. The protocol is accomplish the cluster head selection procedure in two phase, which is time consuming and cluster size is uneven due to variable cluster count.

2.6.7.2 Non probabilistic clustering algorithms

In the category of no probabilistic clustering algorithms, more specific criteria for cluster head election and cluster formation are primarily considered. They are mainly based on the sensor nodes proximity, connectivity, position, location and degree etc. They are also depends on the information received from other closely located nodes. This type of algorithms generally requires more exchanges of messages and probably graphs traversing in some extent, thus leading sometimes to worse time complexity than probabilistic or random clustering algorithms. One the contrary, these algorithms are usually more reliable toward the direction of extracting robust and well-balanced clusters. In addition to node proximity, some algorithms also use a combination of metrics such as the remaining energy transmission power, and mobility (forming corresponding combined weights) to achieve more generalized goals than single-criterion protocols[106].

2.6.7.2.1 Popular Non Probabilistic Clustering Protocols

In Non-Probabilistic Clustering Algorithms, more specific criteria for CH election and cluster formation which are primarily considered are mainly based on the nodes' proximity i.e. connectivity, degree, distance etc. and on the information received from other closely located nodes. In addition to node proximity, some algorithms conjointly use a combination of metrics like the residual energy, transmission power, mobility, etc. to achieve more generalized goals than single-criterion protocols [20, 104]. The cluster formation procedure here is mainly based on the communication of nodes with their neighbour's i.e. in one or multi-hop neighbours and generally requires more intensive exchange of messages and probably graphs traversing in some extent. On the contrary these algorithms are usually more reliable toward the direction of extracting robust and well-balanced clusters.

2.6.7.2.2 Highest-Connectivity Cluster Algorithm (HCC)

HCC is a Non-Probabilistic Clustering Algorithm. It is a distributed multi-hop hierarchical clustering algorithm which also efficiently extends to form a multi-level cluster hierarchy [107, 108] . HCC proceeds in two phases, one is "Tree Discovery" and other one is "Cluster Formation [20, 104]. Any node in the WSN can initiate the cluster formation process. Each node broadcasts the number of neighbor's it have, connectivity of node is considered, the node with highest connectivity is elected as CH, but in the case of a tie, the node with the lowest connectivity persuaded. Node which has already selected a CH withdraws its intention to be a CH. The connectivity based heuristic used in this scheme elect's the sensor

with maximum number of 1-hop neighbours as the CH. The creation of one-hop cluster and clock synchronization requirement limit the practical usage of the algorithm [1, 109].

2.6.7.2.3 Biologically Inspired Clustering Algorithm

In the last few years some new algorithms have also been proposed based on swarm intelligence techniques which model the collective behavior of social insects such as ants [110, 111]. Swarm intelligence clustering algorithm based on the Antclust method which is a model of an ant colonial closure to solve problems in cluster formation. In colonial closure model, once two objects meet along they acknowledge whether or not they belong to identical cluster or not. Within the case of a WSN, in the start the device nodes with a lot of residual energy become CHs severally. Then, every which way chosen nodes meet one another, exchange info, and clusters are created, merged, and discarded through these native conferences and comparison of their info. Every node with less residual energy chooses a cluster supported specific criteria, just like the residual energy of the CH, its distance to the CH, and estimation of the cluster size. Eventually, energy economical clusters are fashioned that end in Associate in Nursing extension of the lifespan of the WSN. Generally, biologically inspired clustering algorithms show that they can dynamically control the CH selection while achieving quite uniform distribution of CHs and energy consumption.

2.6.7.2.4 Weight-Based Clustering Algorithm

WCS is a Non-Probabilistic Clustering Algorithm, which uses distributed scheme for cluster formation in this single hop communication pattern. In this CH is elected non periodically. For purpose of power saving, it invoked a new election every time a sensor loses the connection with any CH. It is invoked on demand, every time a reconfiguration of the network's topology is inescapable [95, 111].

2.6.7.3 DBSCAN Clustering

In 1996, Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu proposed a data clustering algorithm known as Density-based spatial clustering of applications with noise (DBSCAN) [112]. The DBSCAN algorithm is a density-based clustering non-parametric algorithm: given a large number of points in a specific area, it collects composed points that are tightly packed composed (points with numerous close by neighbors), stamping as exceptions focuses that lie alone in low-thickness locales (whose closest neighbors are excessively far away). The DBSCAN is a well-known clustering algorithm that is frequently cited in scientific literature[112] . Considering the focuses on some spaces to be grouped, let ϵ be a boundary indicating the sweep of an area as for some point. In the end, points are delegated core points(thickness) accessible points and anomalies as follows with the DBSCAN clusters:

- If at least minPts points are within separation ϵ of a point D (counting D), it is considered a core point.

- If point k is inside separation from core point d , it can be reached directly from d . Points are only supposed to be reachable directly from the core points.
- A point k can be reached from d when there is a way d_1, d_n with $d_1 = d$ and $d_n = k$, from d_i to each d_{i+1} directly. With the possible exception of k , this indicates that the starting point and all points along the way should be core points.
- All points that cannot be reached from another point are referred to as anomalies or clamour points.

Presently on the off chance that d is a core point; at that point it shapes a group along with all points (core or non-core) that are accessible from it. Every cluster holds at least one core point; non-core points can be a piece of a group, however they structure its "edge", since they can't be utilized to arrive at extra points.

Accessibility is certainly not a symmetrical connection: by description, non-core points can only reachability by the core points. The inverse isn't accurate, so a non-core point might be accessible, however nobody from it can be accessed. In this manner, a additional thought of connectivity is expected to officially characterize the degree of the clusters found by DBSCAN. If there is a point l with the end goal that both d and k are reachable from l , then a points d and k are thickness associated. Thickness connectedness is symmetric. A cluster then fulfills two things:

- i. All points inside the cluster are generally associated with thickness.

- ii. A point is a portion of the cluster if it is thickness-reachable from one of the cluster's purposes [113, 114].

The construction of clusters requires the DBSCAN to draw an unlabeled object d in a random manner, so as to perform the ϵ -range query on d . In case d turns out to be a core object, the ϵ -range query will be executed for all $k \in N(d)$ for expanding the clusters. This will continue until no core objects are found anymore. Furthermore, a cluster label will be assigned to d and its density-connected objects, whereas the unlabeled ones will undergo processing so as to expand novel clusters.

The DBSCAN approach identifies clusters within huge spatial datasets by taking into account the local density of its elements, with the use of a single input parameter. In addition, a suitable parameter value is suggested for the user, so little knowledge about the domain itself is needed. The aim of DBSCAN is categorizing them into clusters apart, eventually defining the differing classifications [112]. The next section suggests a description of the manner in which the DBSCAN algorithm performs. The time requirement of the DBSCAN scheme is $O(n^2)$, where n refers to the size of the dataset. The time complexity will be decreased to the $O(n \log n)$ if the spatial indexing is utilized [115].

The Main Characteristics of some Clustering Algorithms are present in the Table 2.2.

Table 2-1 Main Characteristics of some Clustering Algorithms

Algorithm	Merits	Limitations
-----------	--------	-------------

LEACH[116]]	<ul style="list-style-type: none"> • Each node has equal chance to become cluster head but cannot be selected in subsequent round so load is shared between nodes. • Leach uses TDMA so it keeps CHs from unnecessary collisions. 	<ul style="list-style-type: none"> • It uses single hop communication, so it is not suitable for large scale WSN. • CH are selected on the bases of probability, so uniform distribution cannot be ensured and it does not provide load distribution.
HEED [20]	<ul style="list-style-type: none"> • HEED is fully distributed routing scheme. • Load balancing as it has uniform CH distribution. • High energy efficiency and high scalability by 	<ul style="list-style-type: none"> • Unbalanced energy consumption due to more CHs generation. • Massive overhead due to multiple rounds. • Additional overhead due to several epochs.

	communicating in multi-hop way.	
UCS [117]	<ul style="list-style-type: none"> • SNs in cluster can be variable • It is bi-layered model and two-hop inter cluster communication. 	<ul style="list-style-type: none"> • It is limited by assumptions that CHs are predetermined as well as network is not homogeneous. • Residual energy of SN is not considered and not suitable for large Scale / range networks.
EECS [118]	<ul style="list-style-type: none"> • EECS construct more balanced network in term of energy consumption and communication load. • Use dynamics sizing of cluster. 	<ul style="list-style-type: none"> • Lot of overhead due to global information for communication • Single hop communication consumes lot of energy.

LEACH-VF [119]	<ul style="list-style-type: none"> • Solve the problem of area with overlapped sensing coverage and sensing hole. • In LEACH-VF some nodes can be moved to coverage inside the cluster are. 	<ul style="list-style-type: none"> • Poor energy efficiency • Load balancing is not up to the mark.
TEEN [120]	<ul style="list-style-type: none"> • Data transmission can be controlled by varying two thresholds • Well suited for time critical applications 	<ul style="list-style-type: none"> • Whenever thresholds are not meet the node will not communicate. • Data may be lost if CHs are not able to communicate with each other.
PANEL [121]	<ul style="list-style-type: none"> • It is energy efficient that ensures load balancing and enhanced network lifetime 	<ul style="list-style-type: none"> • Clusters are fixed • To determine geographic position information, special conditions are needed which is always not available.

	<ul style="list-style-type: none"> • Supports asynchronous applications. 	
[122]	<ul style="list-style-type: none"> • It enhances the network lifetime by saving energy. • Routing fidelity is maintained. 	<ul style="list-style-type: none"> • Large traffic injection. • Delay is unpredictable.
TTDD [123]	<ul style="list-style-type: none"> • Resolves the numerous mobile sinks and moving problem of sink in large scale WSNs. • Suitable to event detecting WSNs among irregular data traffic. 	<ul style="list-style-type: none"> • Large latency • Low energy efficiency • It required SNs to be stationary and location aware.
SLGC [124]	<ul style="list-style-type: none"> • Lower energy consumption. 	<ul style="list-style-type: none"> • Huge overheads due to complex

		data communication.
PEGASIS [125]	<ul style="list-style-type: none"> • Energy load is distributed uniformly. • Decrease overhead due to dynamic cluster formation. • Reduce number of data transmission. 	<ul style="list-style-type: none"> • It uses single hop communication, so it is not suitable for large scale WSN. • Network is not very scalable. • Not suitable for time varying topology.
CCS[125]	<ul style="list-style-type: none"> • Energy consumption is reduced. • Reduce data flow from BS. 	<ul style="list-style-type: none"> • Unbalanced energy consumption. • Large delay due to long chain.
TSC [126]	<ul style="list-style-type: none"> • Reduce data transmission in WSN by breaking 	<ul style="list-style-type: none"> • SN distribution is unbalanced.

	long chains into smaller chains.	
EECS [127]	<ul style="list-style-type: none"> • Optimum Coverage • Prolonged Network lifetime 	<ul style="list-style-type: none"> • Unbalanced energy consumption
DBSCN[128]	<ul style="list-style-type: none"> • Reduced Computational Cost & Efficient cluster results for large dataset 	<ul style="list-style-type: none"> • Its only real weakness is that it has some difficulties in distinguishing separated clusters if they are located too close to each other, even though they have different densities.

2.7 Metaheuristic Methods for Optimization Problem

In computer science and mathematics, a metaheuristic is a higher-grade or heuristic procedure that is aimed at finding, generating or selecting a heuristic (partial search algorithm) which can offer the best possible solution to an optimization problem, particularly if the information or computational capacity is incomplete or imperfect, [129, 130]. Metaheuristics chose a subset of alternatives that would otherwise be too big to be listed or fully examined. Metaheuristics can be used for a wide range of issues because they make few presumptions regarding the solution of the optimization problem [131].

2.7.1 Cuckoo Algorithm

CA is classified as a stochastic algorithm whose mechanism is motivated by the remarkable behaviour of certain cuckoo species [132] [133], because they prefer to Put their eggs in other birds' nests. This algorithm contributed to the efficiency of global searching within solution domains in a more significant manner than any alternative optimizing algorithm [134] [135].

A. STANDARD CA

A more simplified form of CA is presented in [136] [137]: eggs represent solutions and the nests are the individuals of the population. Any solution which does not function properly is replaceable by a newer alternative, and the same goes for any abandoned nests. The number of

nests equals the population size, and it is represented by three rules. The standard CA algorithm is mainly determined by three rules [133]:

Every cuckoo bird will put a single egg per time within any randomly chosen nest; Nests that have better-quality eggs will continue to the following generation; The hosting bird (or nest owner) could identify the alien egg with a probability of $P\alpha \in (0,1)$ after which it might dispose the egg or leave the nest itself to create another. The standard CA is considered to be of higher efficiency for its simpler structures, fewer number of variables and relatively easier form of implementing. In terms of mathematics, The following is the definition of a nest position:

$$X_i(k), i \in \{1, 2, \dots, NP\} \quad (2.1)$$

Where NP and k stand for the numbers of cuckoo nests (population size) and generations, respectively.

New solutions $x_i(k+1)$ could be generated through the use of global and local random walk combined in a balanced manner through Lévy flight [133]. The global random walk is provided by:

$$x_i^{(k+1)} = x_i^{(k)} + \alpha \oplus \text{levy}(\lambda) \quad (2.2)$$

Where \oplus denotes entry-wise multiplication, σ represents a step-size in relation to the issue scale. This could often be measured by means of the following calculation [132, 138]:

$$\alpha = \alpha_0 \times (x_i^{(k)} - x_{\text{best}}^{(k)}) \quad (2.3)$$

Where α_0 represents the scaling factor; $x(k)_{best}$ is the most suitable solution at present time, and $levy(\lambda)$ represents a randomly chosen number from a Lévy distribution:

$$levy(\lambda) \sim t^{-1}, (1 \leq \lambda \leq 3) \quad (2.4)$$

In implementation, the calculation of Lévy(λ) could be simply done in the following manner [132, 139, 140]

$$S = \frac{\mu}{|v|^{1/\beta}} \quad (2.5)$$

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \sin(\frac{\pi\beta}{2})}{\Gamma[\frac{(1+\beta)}{2}] 2^{(\beta-1)/2\beta}} \right\}^{1/\beta}, \sigma_v = 1 \quad (2.6)$$

where S forms the simulating value of Lévy (λ); u and v represent two randomly assigned numbers that follow the normal distribution with a zero mean and zero deviations σ_u and σ_v respectively; β represents a Lévy distribution variable, set at 1.5 [139]; and Γ is a Gamma function. As for the CA algorithm, the Lévy flight has been employed in global exploration having properties of a random nature, whereas a cross-over operator is made use of with local exploitation which has a mutation of the present solution. An update of the optimal solution is made after every iterating process. Algorithm 2.1 illustrates the CA algorithm for solving the optimization problems.

B. BINARY CUCKOO SEARCH

In the continuously-valued CA, the nest updates its location to a real value within the possibility search space as limited by any issue constraint. The UC issue represents a direct optimization with 0-1 deciding parameters that represent the ON/OFF unit mode. Thus, the real-valued CA requires a revision so that it could possibly suit a binary issue. BCS acquires its concept from the binary particle swarm optimization (BPSO) algorithm [141], as this usually makes use of a sigmoid function (see Figure 2.5) for restricting any of the novel solutions to a binary value [137, 142].

Algorithm 2-1 Cuckoo search algorithm

```

1  Begin
2  Target Function $f(x)$ ,  $x = (x_1, x_d)^T$ 
3  Create a host nest population as the initial population
4  while ( $t < \text{Maximum Generation}$ ) or (criterion for stopping)
5      Obtain a cuckoo at random via Levy flights and assess its quality/fitness
6       $F_i$ 
7      Pick a nest at random from a list of n (say, j).
8      if ( $F_i > F_j$ ),
9          j should be replaced with the new solution;
10         end if
11     A percentage of the weakest nests (pa) are forgotten, and new ones are created;
12     Save the greatest solutions (or nests with high-quality solutions);
13     Sort and select the best solutions.
14 end while
15 Results and visualization of the post-process
16 end

```

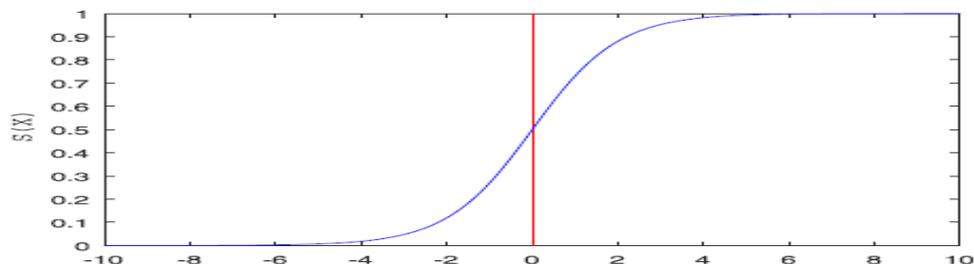


Figure 2-4Sigmoid Function[142]

$$S(x_i^{(k)}) = 1/(1 + e^{-x_i^{(k)}}) \quad (2.7)$$

In other words, a binary solution $x_i^{(k+1)}$ could be obtained by means of a typical updating equation:

$$x_i^{(k+1)} = \begin{cases} 1 & \text{if } S(x_i^{(k)}) > r \\ 0 & \text{otherwise} \end{cases} \quad (2.8)$$

Where r stands for a number of uniform distributions in $(0,1)$.

Whenever new solutions are created, (2.7) and (2.8) are employed in mapping the search processes from continuous to binary spaces, following the global and local searches in CA which took place in advance.

2.7.2 Hybrid cuckoo search and genetic algorithm

The goal of creating a hybrid CA–GA system was to incorporate the benefits of each cuckoo search and genetic algorithms. Depending on the optimization approach chosen and correctly applied, An optimum solution to an optimization problem may be difficult to identify. The CA–GA algorithm is a population algorithm that employs a population of alternative solutions to achieve the best solution, just like the CA and GA algorithms. The initial population is produced randomly with the aim of

ensure that the initial solution is distributed. Algorithm 2.2 illustrates the CA-GA algorithm for handling the optimization problems[143].

Algorithm 2-2CA-GA algorithm

- 1 **Begin**
- 2 **Target function $f(x)$**
- 3 **Initialization. Set the generation counter $t=1$ random population initialization. (Initializing the random number of host nests with each host nest is a solution to the given problem);**
- 4 **Fitness evaluation. Evaluate fitness $f(x)$;**
- 5 **While ($t < \text{Max Generation}$) or (stop criterion); / New population /**
- 6 **Using genetic operators(selection,crossover and mutation), create a new population.**
- 7 **Evaluate fitness (the best individual perform Lévy flight)**
- 8 **Produce a new solution (say x_{new}) via Lévy flights;**
- 9 **Choose a solution (say x_j) randomly among N_{pnew} and evaluate its fitness (F_j);**
- 10 **if($F_{x_{\text{new}}} < F_j$) then**
- 11 **j should be replaced with a new solution;**
- 12 **end if**
- 13 **Store the best solution;**
- 14 **$t = t+1$;**
- 15 **end while**
- 16 **Retrieve the best solution among the current best solution stored in each generation**
- 17 **End**

The population number remains constant throughout the algorithm's execution. In every generation, the population is first crossed and then mutated. The idle solution in the existing population will conduct Lévy flight before the end of each generation to expand the next generation's exploration of the solution space. The program seeks to build a new population with each generation via using genetic principles to replace facts in the existing population with superior facts. By a continuous process of reproduction, the population is directed toward the global optimum (crossover and mutation)[143]. In the CA-GA pseudo code, taking into account the above principles. One cuckoo places an egg

at a time with Lévy flights at the original CA. Like previously stated, the female cuckoo bird will imitate the host bird's eggs. To accommodate for this conduct, a mutation operator has been introduced to the algorithm. This reflects the cuckoo egg genes' behavior being mutated in order to enhance their reproductive potential. Although their genes are mutated, the alien egg is still possible to be revealed by the host bird. The quality of the reproduced eggs determines the discovery of the alien egg. As a result, an elitism approach is employed in which only high-quality eggs are kept, while low-quality eggs are rejected[143].

The initial population is created Some host nests are randomly distributed in the initial populations across the search area. Only one egg/solution is available to every nest initially[144].

Creating a new population A new population is established using genetic operators (selection, crossover and mutation). In the population, two parents (cuckoos) are selected using a Roulette Wheel involves the production of two new offspring through recombination or crossbreeding (eggs). Crossover is a recombination technique for the creation of two eggs with similar properties to their parents. The technique of the single point crossover was used. Each cuckoo mutates independently after the crossover procedure. Its goal is to preserve population variety by preventing early convergence. [145].

Elitism strategy A kind of selectivity is used to choice and retain optimal solutions among the population after each crossing and mutation. The new cuckoo egg is taken over from the parents if the cuckoo egg is

greater than the parent; otherwise, the parent cuckoo bird will remain[146]. In terms of mutation, If the newly mutated cuckoo egg is superior to the old cuckoo bird, it will be replaced. If it is poorer, the mutated cuckoo egg is removed as an alien egg. The elitist strategy ensures that the next generation will always maintain the best candidate solution. The following equation can be used to summarize this procedure:

$$x_i^{(n+1)} = \begin{cases} y_i & \text{if } f(y_i) > f(y_i^{(n)}) \\ y_i^{(n)} & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, N \quad (2.9)$$

In the current generation, every cuckoo bird represents one solution and the cuckoo eggs are the new generation solutions (either by crossover or mutation)[146].

Creation of new solutions when a new population has been established the eggs which have become cuckoo birds are being sent through the Lévy flight to a better environment for the rest of their lives. In this article, Lévy Flight will perform the best cuckoo in the present population, to stochastically generate new eggs.

$$x_i^{(k+1)} = x_i^{(k)} + \alpha \oplus \text{levy}(\lambda) \quad (2.10)$$

Where $\alpha > 0$ is a step size to be associated to the interest problem scale [147].

2.7.3 Particle swarm optimization (PSO) algorithm

The general idea of particle swarm optimization (PSO) is based on the way in which bird flocks or fish schools move. In this algorithm, the population is called a swarm and each point is a particle. These swarms represent possible solutions. The particles are randomly initialized in the search space of the objective function. After initialization, each particle during consecutive iterations makes a compromise between its own best position in the history (i.e., pbest), the fittest position of the swarm and a random search (i.e., gbest). In PSO, each particle is associated with two properties (velocity vector V and position vector X) and it moves in the search space with a velocity that is dynamically adjusted according to the particle's experience and the particle companion's experience simultaneously. Mathematically, the velocity and position of the particles are updated according to Equations (2.11) and (2.12):

$$V_{id} = w * v_{id}(t) + C_1 * r_1 * (Pid(t) - x_{id}(t)) + C_2 * r_2 * (Pgd(t) - x_{id}(t)) \quad (2.11)$$

$$X_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \quad (2.12)$$

Where $V_{id}(t+1)$ and V_{id} are the velocities of the particle at the iterations t and $t + 1$. P_{id} is the best position of the particle. P_{gd} is the best position of its neighborhood at the iteration t . c_1 and c_2 are acceleration coefficients reflecting the weight of the stochastic acceleration terms that pull each particle toward pbest and gbest positions, respectively. r_1 and r_2 denote two random numbers uniformly distributed in the range $(0, 1)$. ω is the inertia weight used for balancing the global and local search. In

general, a large inertia weight facilitates the global exploration, while a small inertia weight tends to facilitate the local exploitation, and x_{id} is the position of the particle at the iteration t . PSO starts by generating particles randomly in the search space[148].

2.7.4 ANT COLONY OPTIMIZATION

Ant colony optimization is one of the artificial intelligence metaheuristic techniques that inspired by real ant colonies [149]. The ants that seek for food coordinate with each other by dropping and sensing the paths pheromone level. The ants depend on stochastic or probability theory to select their path [150]. The ACO was originally applied to solve the classical travelling salesman problem and it finds a good solutions[151]. During last few years Ant Colony Optimization approach has been used to solve the complex computational problems and software testing is one of these problems [152]. In our work we use anti-ant Colony Optimization to find the optimal testing paths. The ant moves from start node to other nodes until it reaches the exit node. During its movement from node to another it updates the pheromone value between the two nodes so that the other ants use the updated value to determine their paths. The difference between ant colony algorithm and anti-ant colony algorithm is that in ant colony algorithm the ant selects the path that has maximum value of pheromone while in anti-ant colony algorithm the ant selects the path that has the minimum pheromone value.

2.7.5 Genetic Algorithms

A genetic algorithm is a computational model of biological evolution. Genetic algorithms are useful, both as search methods for solving problems and for modeling evolutionary systems. In genetic algorithms, binary strings are stored in a computer's memory, and over time, the strings are modified in much the same way that populations of individuals evolve under natural selection. Although the computational setting is highly simplified when compared with the natural world, genetic algorithms are capable of evolving surprisingly complex and interesting structures. These structures, called individuals, can represent solutions to problems, strategies for playing games, visual images, or computer programs. Genetic algorithms are loosely based on ideas from population genetics. First, a population of individuals is created randomly. In the simplest case, each individual is a bit string and can be thought of as a candidate solution for some problem of interest[153]. Variations among individuals in the population result in some individuals being more fit than others (e.g., better problem solutions). These differences are used to bias the selection of a new set of candidate solutions at the next time step, referred to as selection. During selection, a new population is created by making copies of more successful individuals and deleting less successful ones. However, the copies are not exact. There is a probability of mutation (random bit flips), crossover (exchange of corresponding substrings between two individuals), or other changes to the bit string during the copy operation. By transforming the previous set of good individuals to a new one, the mutation and crossover operations generate a new set of individuals, or samples, that ideally have a better than average chance of

also being good. When this cycle of evaluation, selection, and genetic operations is iterated for many generations, the overall fitness of the population generally improves, and the individuals in the population represent improved "solutions" to whatever problem was posed in the fitness function[154].

The Main Characteristics of some Metaheuristic Algorithms are present in the Table 2.3.

Table 2-2Main Characteristics of Some Metaheuristic Algorithms

Algorithm	Advantages	Disadvantages
Cuckoo search[155]	<ul style="list-style-type: none"> • CSA has the ability to converge to a true global optimum • CSA can handle local and global search. • It makes use of Levy flight as a strategy for global search 	<ul style="list-style-type: none"> • CSA produces low classification accuracy • CSA has low convergence rate
Differential evolution[156]	<ul style="list-style-type: none"> • DE is good at exploration and diversification 	<ul style="list-style-type: none"> • DE convergence is not stable [59]

	<ul style="list-style-type: none"> • DE has the capacity to deal with cost functions that are non differentiable, multimodal and nonlinear • DE can handle cost functions with high computational complexity DE is easy to use; it requires only few parameters 	<ul style="list-style-type: none"> • DE easily falls into a regional optimum [59] • DE requires parameter tuning
<p>Genetic algorithm[157]</p>	<ul style="list-style-type: none"> • GA is easy to implement • GA has the ability to handle random types of objectives and constraints • GA can be used independently 	<ul style="list-style-type: none"> • In GA, there is no guarantee that the global maxima will be identified. It has high likelihood of getting trapped in the local maxima • GA does not have a standard method for defining a good fitness function. The best

	<p>to solve a given problem. It does not depend on other algorithms or heuristics</p> <ul style="list-style-type: none"> GA can be used to handle problems whose constraints and objective functions are nonlinear or discontinuous GA uses simple operators and can be used to solve problems that have high computational complexity, such as the TSP problem 	<p>solutions majorly depend on a fitness function and hence the fitness function must be very accurate</p> <ul style="list-style-type: none"> In GA, premature convergence seldom occurs, thus losing the population diversity GA does not have standard termination criteria, neither does it have a standard method for adjusting its parameters GA can be time consuming, especially for problems with a large number of variables
<p>Particle</p>	<ul style="list-style-type: none"> Calculation in PSO is simple 	<ul style="list-style-type: none"> PSO suffers from partial optimism

swarm optimisation[158]	<ul style="list-style-type: none"> • It is useful in scientific research and in engineering 	
Symbiotic organisms search[159]	<ul style="list-style-type: none"> • Operation of SOS does not require any specific parameters • SOS has a very fast rate of convergence and reduced computational time • SOS does not require procreation, but adjusts through interactions between individuals in the population • SOS differs from other metaheuristic algorithms by 	<ul style="list-style-type: none"> • Similar to other population-based algorithms, SOS depends on iterative performance, which can be time consuming

	<p>three of its unique operators: parasitism, commensalism and mutualism</p>	
<p>Firefly algorithm[160]</p>	<ul style="list-style-type: none"> • FFA has the ability to automatically divide the population into different groups, hence it is good for diversification 	<ul style="list-style-type: none"> • Similar to some metaheuristic algorithms, FFA performance depends on adequate parameter tuning • Diversification in FFA can lead to reduced speed and reduced convergence rate • FFA is not very suitable for handling complex problems, because it can be trapped in many local optima
<p>Ant colony optimization[161]</p>	<ul style="list-style-type: none"> • The construction process for ACO is inherently parallel, as ant builds solutions independently 	<ul style="list-style-type: none"> • ACO probability distribution changes with iteration • Although convergence in ACO is guaranteed, convergence

	<p>and simultaneously</p> <ul style="list-style-type: none"> • Distributed computation in ACO evades premature convergence • ACO can be used to efficiently handle Travelling Salesman Problem and related problems 	<p>time is undefined</p> <ul style="list-style-type: none"> • It is difficult to theoretically analyse the behaviour of ACO, since ACO is based on sequences of random decisions of different independent artificial ants
<p>Bat algorithm[162]</p>	<ul style="list-style-type: none"> • BA uses frequency tuning to increase the population diversity • BA has the capacity to automatically zoom into areas with good solutions. Automatic zooming helps to balance exploration and 	<ul style="list-style-type: none"> • The standard BA supports continuous values, hence it can be easily applied to nonlinear global optimisation problems. However, BA cannot efficiently handle discrete optimisation problems, hence the need for improved implementations, . In addition, BA requires several parameters that require tuning

	<p>exploitation during search</p> <ul style="list-style-type: none">• BA does not use a fixed parameter, but uses parameter control to vary its parameter values at different iterations. This gives BA the ability to automatically and quickly switch from exploration• stage to exploitation stage• BA is simple, flexible and easy to implement. It can be used to handle a vast	
--	--	--

	number of problems	
Flower pollination algorithm[163]	<ul style="list-style-type: none"> • FPA is flexible, simple, easy to implement, has few parameters and can be used to handle both single and multiple objective optimisation problems. • Flower constancy in FPA may have evolutionary advantages, because the algorithm maximizes the transfer of pollen grain to plants of the same species, which consequently maximizes or enhances the production of the same flower species for evolutionary speciation 	<ul style="list-style-type: none"> • FPA has slow convergence rate and low precision • FPA easily falls into a local optimum

<p>Artificial bee colony algorithm[164]</p>	<ul style="list-style-type: none"> • ABC algorithm is very robust, it converges fast, it requires few parameters and it is flexible 	<ul style="list-style-type: none"> • ABC algorithm has premature convergence in the later stage of its search and the classification accuracy of its best obtained value may not be high enough to meet the requirements
<p>The bee algorithm[165]</p>	<ul style="list-style-type: none"> • The BA is easy to implement and it has the ability to perform both local and global search • It can be combined with other algorithms 	<ul style="list-style-type: none"> • The BA algorithm requires manual parameter settings and it has several parameters for tuning

2.8 Network Lifetime

This section summarizes the most common definitions in a lifetime survey.

A. The number of active nodes determines the network lifespan

The network lifetime T_n^n , according to this concept, terminates when the first node fails thus:

$$T_n^n = \min_{v \in V} \quad (2.13)$$

T_v denotes the lifetime of node v . To represent the assumption that a power plug is provided at the sink nodes, some authors omit the sink nodes from the node set V , T_n^n is a very easy definition. It's easy to calculate and network algorithms have no need to deal with changes in topology[166].

B. Lifespan of the network depends on sensor coverage

Considering the unique characteristics of sensor networks, it is natural to calculate lifetime by measuring network life as the time that the sensor nodes cover the area of interest. The coverage can be defined in different ways depending on the composition of the area of interest and on the redundancy achieved [167].

C. Lifetime of a network based on connectivity

The network connectivity is taken into account in another group of metrics. Because there is no concept of sensor coverage in ad hoc networks, connectivity is a popular metric. so it is most important that the information can be passed on to a specific destination. Blough et al. provided a definition for ad hoc network lifespan[168] The lifetime is

defined as the time period that falls below a certain threshold for the percentage of living nodes or the size of the network's largest connected component. This concept only takes the size of the biggest connecting component of the network into account.

D. Sensor coverage and connectivity determine the network's lifetime

Several writers mix coverage-based measures with connectivity metrics due to the mentioned restrictions. As defined in [169], [170] When coverage or connectivity falls below pre-set threshold, the network lifespan metric is defined. In this case, as previously mentioned, the coverage is measured as α -coverage. Connectivity is measured by the delivery rate of the packet at the sink node.

E. Lifetime network based on service requirements of application quality

Some researchers define the life of the network only in relation to the application of service quality requirements. Kumar et al. [171] define the lifespan of a WSN as the duration of time during which the network consistently meets the application requirement.

2.9 Energy Consumption Model

In the following sub sections we explore the energy consumption models that used in this dissertation:

A. First-order radio energy model

The model of first-order radio energy for communication as shown in

Figure 2.5.

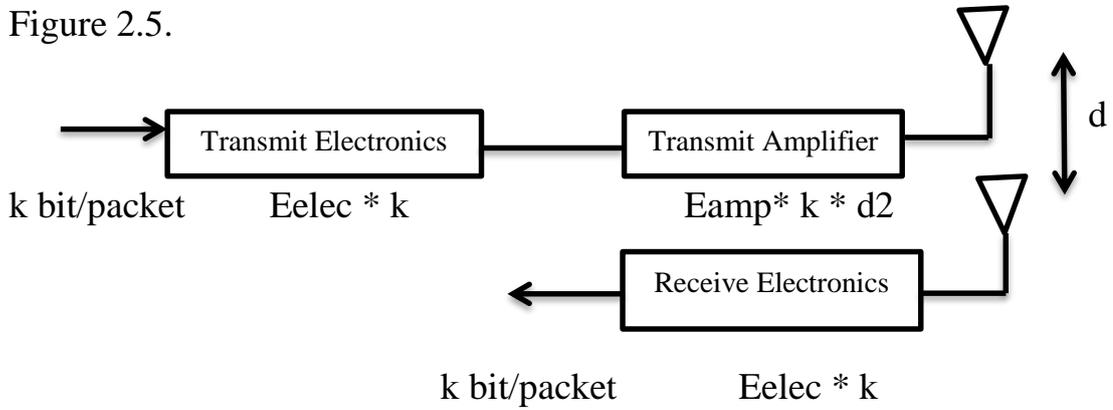


Figure 2-5The model of first-order radio energy for communication [28]

The first-order radio model can be split into a free-space model and a multi-path fading model based on the distance between sending and receiving nodes. The communication channel is supposed to be symmetric and consumes energy from a sensor node to transfer k bits/packet to a node d meters can be written as [28]

$$ET_x(k, d) = ET_{x_elec}(k) + ET_{x_amp}(k, d) \tag{2.14}$$

$$ET_x(k, d) = \begin{cases} E_{elec} * k + E_{fs} * k * d^2, & d \leq d_0 \\ E_{elec} * k + E_{amp} * k * d^4, & d > d_0 \end{cases} \tag{2.15}$$

Similarly, the energy used in the k bits/packet received from a sensor node can be written as

$$ER_x(k) = ER_{x_elec}(k) + kE_{elec} \tag{2.14}$$

Where E_{elec} is the energy the transmitter or the receiver consumes per bit and the transmission amplifier parameters E_{amp} and E_{fs} correspond to the multi-path fading and free-space models, respectively. Every packet contains overhead data bits. The data bits contain useful information, while the overhead contains packet-related information, such as the coding strategy for reliable data transfer[28].

B. Energy Consumption Model based sensor modes

A sensor node usually contains three key energy-consuming units: the microcontroller unit (MCU) that can perform calculations, Communication subsystem that transfers / receives messages and the sensing unit which responsible for data gathering. Dependent on the current sensor status summarized in Table 2.1 each subsystem can be switched on or off in this model[54]:

Table 2-3Commission of Energy

Sensor mode	MCU	Radio	Sensor	Power(mW)
Listening	on	on	on	20.05
Active	on	off	on	9.72

Sleep	off	off	off	0.02
Computation	on	on	on	26.83
Energy required for two bits content messages to be sent or received				0.515

In this work the first model for the results of the clustering is used [28] and the remainder of the work, the second model is used[54].

2.10 Performance metrics

To determinate the efficiency and lifetime of network in sensor network, there are several factors that should be consider like coverage ratio, active sensor ratio and energy consumption.

A. Coverage Ratio (CR): This measures how well the WSN is able to observe the area of interest. Here the sensor range is discretized as a regular circle, which yields the following equation:

$$CR = (\sum_{r=1}^R C_r^{ap})/N \times 100\% \quad (2.16)$$

Where C_r^{ap} is the number of points covered by active sensors in the cluster r during the current period p and N is the total number of all points in the interested area.

A. Active Sensors number (ASN): Is the number of nodes that remain active during the network life time.

$$ASN = \sum_{r=1}^R A_r^p \quad (2.17)$$

Where A_r^p the number of active sensors in cluster r is in period p , R is the number of clusters.

B. Energy Consumption (EC): Energy consumption can be seen as the total energy consumed by the sensors during network lifetime. The value of EC is computed according to the formula

$$EC = \sum_{r=1}^R \left(E_r^{alp} + E_r^{slp} + E_r^{ap} + E_r^{sp} \right) \quad (2.18)$$

Where R corresponds to the number of clusters, p refers to the current period. The total energy consumed by the sensors comes through taking into consideration four main energy factors. The first one, denoted E_r^{alp} refers to the energy needed by all the active leader nodes, E_r^{slp} refers to the energy needed by all the sleep leader nodes, E_r^{ap} and E_r^{sp} indicate the energy consumed active and sleeping nodes.

2.11 Summary

In this chapter, an overview of the wireless networks and their types has been presented and focus on the wireless sensor networks where we show in this dissertation the basic components associated with sensor nodes. This dissertation gives an overview of the applications of WSNs

and discusses their major challenges. The main challenges in WSN is the maximization of their lifetime, therefore several energy efficient mechanisms has been clarified.

The clustering process is very important in the WSN, therefore we give some important clustering techniques that used in WSN, and focus on the DBSCAN clustering algorithm. The network lifetime definitions have been presented in different definitions. This chapter highlights the data metheuristics optimization methods such as Cuckoo Algorithm and hybrid cuckoo search and genetic algorithm to schedule the sensor activates performance metrics. In addition, the model of energy consumption that will be used in this dissertation has been explained.

CHAPTER THREE

PROPOSED APPROACH

3.1 Overview

In this dissertation, new scheduling approaches for maximizing the lifetime of WSN with acceptable coverage's ratio were proposed. First, the sensor nodes are clustered into groups and one cluster head will be elected for each group periodically to achieve a scheduling optimization model. This model is developed to produce the best representative set of sensor nodes per period. Two Metaheuristic algorithms are employed to implement the scheduling optimization model. Cuckoo Algorithm and Hybrid Metaheuristic (CA+GA) Approach are presented to solve this problem for enhancing Lifetime of Cluster-based WSNs.

3.2 The Framework of the proposed approaches

In this section, we present the general framework that will be used to implement the proposed methods to schedule the sensor nodes in WSN. It consists of two phases: Clustering and scheduling. In the clustering phase, the DBSCAN clustering algorithm is employed with slide modification to cluster the sensor nodes in the area of interest into clusters of nodes and the cluster head will be elected every new period. The clustering is only performed at the beginning to group the nodes into clusters, after that the only the cluster head will be elected dynamically every period. In the scheduling phase, we model the scheduling optimization problem and then execute it using two metaheuristic algorithms. Figure 3.1 shows the proposed work.

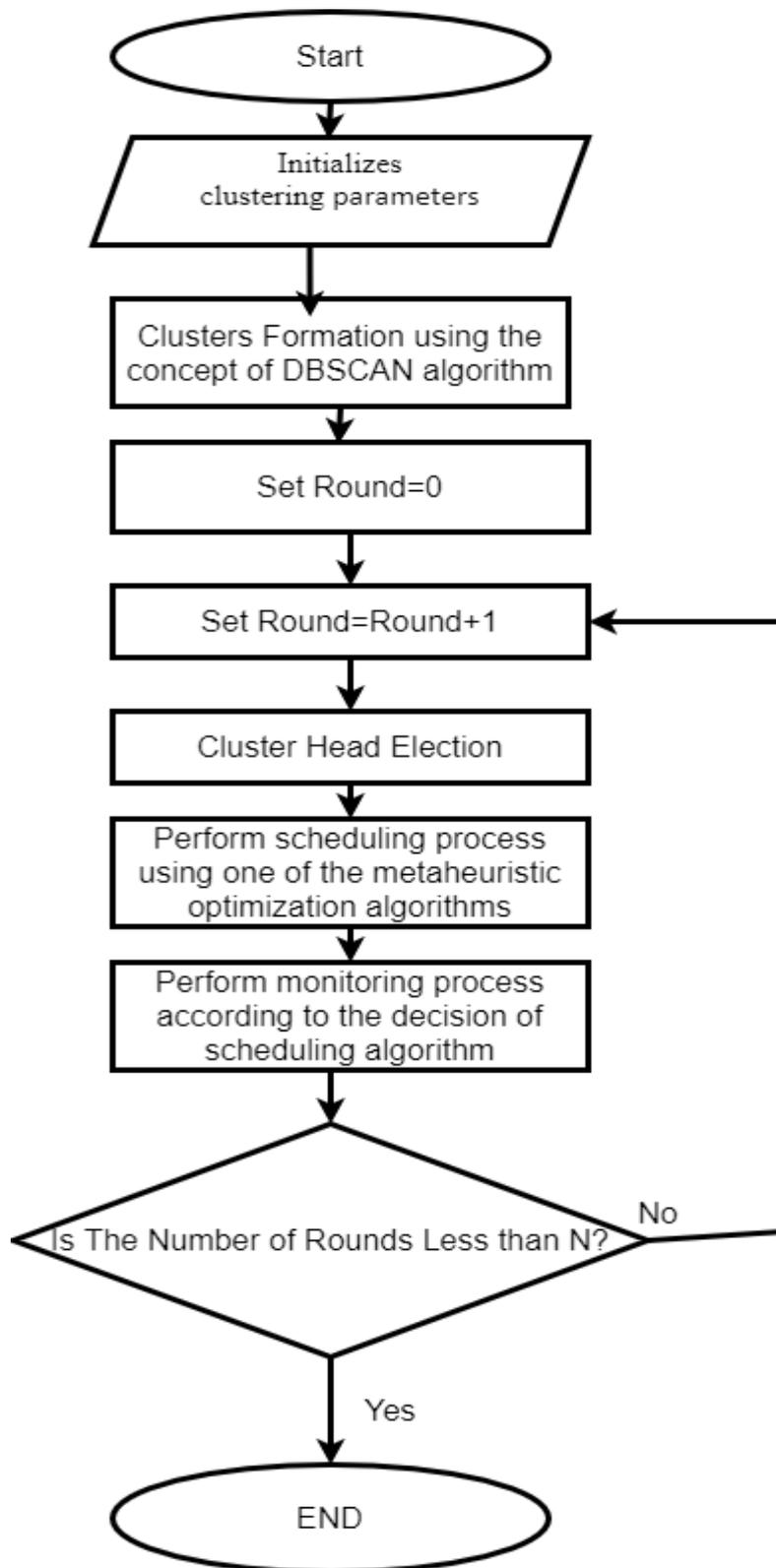


Figure 3-1 Flowchart of the proposed approach.

3.3 Clustering Using DBSCAN Algorithm

The sensor devices of WSN are clustered into small clusters by DBSCAN algorithm. The DBSCAN algorithm has been used for several reasons like it is a density-based clustering, so it is more suitable for dense WSN, number of clusters is determined dynamically, and it can find arbitrarily-shaped clusters.

This distributed DBSCAN clustering algorithm is applied on every sensor device and these devices are cooperated based on the DBSCAN algorithm to form several clusters in the network area of interest.

This work aims at proposing a distributed DBSCAN protocol for maximizing the wireless sensors lifetime. This protocol is distributed at every sensor device deployed in the monitored area. The proposed protocol involving two steps: set-up and steady-state. For the sake of simplicity, the proposed distributed DBSCAN algorithm is named as DBSCAN algorithm in this paper. In the Setup phase, When the sensor devices are deployed in the working area, it is supposed that every sensor device knows its location. According to Dbscan algorithm, each sensor implements the algorithm of DbScan as follow:

1. Each sensor node will perform the same test whether it is Core Point or not, according to the principle of the Dbscan algorithm, as it scans the surrounding area to find out the number of sensors that

are within the sensing range and that must be greater or equal to a specific parameter.

2. As for the sensor nodes that are within the sensing range for Core Point, it will be its member.
3. If this core point does not belong to any cluster, it forms a new cluster, otherwise it remains with the same cluster.
4. The core point sends a message to all its members to be included in the same its cluster.
5. Repeat steps 1-5 until all sensor nodes are passed.

Algorithm 2 explains the Distributed DBSCAN algorithm that will be executed in every sensor node s_j .

Algorithm 3-1 Distributed DBSCAN (s_j)

Input: N: number of neighbor nodes, Sr: sensing range, minNodes: minimum number of nodes to create cluster.

Output: $s_j.rejon$: the cluster number for node s_j .

```

1:  while  $RE_j \geq Ethr$  do
2:    If  $s_j$  Receive MemberPacket from  $s_i$  then
3:      Mark  $s_j$  as member to the Core  $s_i$  ;
4:      Update  $RE_j$ ;
5:    end
6:     $s_j.rejon \leftarrow 0$ ;
7:    for each node  $s_i$  in N do //  $i \in N$  and  $i \neq j$ 
8:       $nbrNodes \leftarrow nbrNodes + CORE$  Objective Function ( $s_j, s_i, Sr$ );
9:      if CORE Objective Function return 1 then
10:       Send MemberPacket to the sensor node  $i$ ;
11:       Update  $RE_j$ ;
12:     end
13:     if  $nbrNodes \geq minNodes$  then
14:       save the information
15:       if  $((s_j.rejon = 0) \text{ Or } (s_j.rejon \neq 0))$  and  $(r == 0)$  then
16:          $s_j.rejon \leftarrow s_j.rejon + 1$ ;
17:         Call Cluster( $s_j$ );
18:       end
19:     else if  $((s_j.rejon = 0) \text{ Or } (r \neq 0))$  then

```

```

20:          sj.rejon ← sj.rejon +1;
21:          Call Cluster1(sj);
22:          end
23:          else if ((sj.rejon ≠ 0) Or (r ≠ 0)) then
24:              Call Cluster2(sj);
25:          end
26:          end
27:      end for
28:  end while
29: return sj.rejon;

```

The CORE Objective Function will reply with 1 and $r = 0$ in case the sensor node i found in the sensing range (S_r) and does not belong to another cluster. If not, the CORE Objective Function returns 0 and $r = 1$. The cluster functions in putting neighboring nodes within the S_j of the same clusters, and sending the s_j MemberPacket to the sensor node i to inform it that it becomes a member in the same cluster of s_j . The function Cluster1 put any neighbor node within the sensing range of s_j and it has not assigned to any cluster in the same cluster of sensor node j . As for Cluster 2, is also puts neighboring nodes in the S_j . However no clusters are assigned to the node j . When each of the Cluster, Cluster1 and Cluser2 achieve their functions, an update will be made for the energy that remains for the sensor node j by sending a MemberPacket to the sensor node i to inform it that it becomes a member in the same cluster of s_j .

After the stage of creating clusters, the exchange of information between the core points (nodes) is done inside the single cluster, where each core point sends a message to all the core points inside the Cluster, it contains all the necessary information inside like remained power, status, location, number of members, total wireless sensors number in the group (cluster), and etc. Every core sensor node inside each cluster will

include the information of other core nodes in the same cluster, therefore, every core node in the same cluster will execute the Equation (1) for the information of each member inside the core node.

3.4 Scheduling problem Formulation

This stage of procedure involves the formulation of the optimising model that treats the scheduling issues. It is followed by executing the metaheuristic algorithm so as to obtain the best solution. This is done by forming the optimal sensor device scheduling to monitor the current period. A mathematical scheduling model has been employed in the optimization of the network life span and coverage. There are two objectives taken into consideration in the formulation of this optimizing model: the minimization of uncovered zones and number of active sensors after the algorithm decisions, after which a number of alterations have been made in concepts of lowering the number of nodes that are currently active, decreasing the consumption of energy, and improving the network's lifespan. Given that A indicates the coverage of CHs in each cluster, its definition can be stated in the following way:

$$A_{ji} = \begin{cases} 1 & \text{if point center } j \text{ is covered by sensor } i \\ 0 & \text{Otherwise} \end{cases} \quad (3.2)$$

$$T_i = \begin{cases} 1 & \text{if Sensor } i \text{ is Active} \\ 0 & \text{Otherwise.} \end{cases} \quad (3.3)$$

For $1 \leq j \leq N$ and $1 \leq i \leq N$, where N is the number of sensor devices inside the cluster. The (T) refers to the solution parameter that can be either 0 or 1 according to the status of the sensor device. Its definition is stated below:

The coverage probability Q_j of the centre point j can be stated as follows:

$$Q_j = 1 - \prod_{i=1}^N (1 - (A_{ji} * T_i)) \quad (3.4)$$

The initial objective of this model is the increase of covering rate over a particular area, through lowering uncovered ratio $(1 - Q_j)$ in the following way:

$$P_j = 1 - Q_j \quad (3.5)$$

Secondly, the number of points covering the same node within the sensing field is to be minimized, as follows:

$$L_j = \begin{cases} 0 & \text{if point } j \text{ is not covered} \\ (\sum_{i=1}^N (A_{ji} * T_i)) - 1 & \end{cases} \quad (3.6)$$

The issue of the optimization model is formed as follows:

$$\text{minimize } \{ \delta \cdot \sum_{j=1}^N P_j + \vartheta \cdot \sum_{j=1}^N L_j + \alpha \cdot (1 - \frac{RE_j}{IE_{max}}) \cdot T_j \} \quad (3.7)$$

$$\text{Subject to } \sum_{i=1}^N (A_{ji} * T_i) = 1 + L_j - P_j \quad \forall j \in N \quad (3.8)$$

$$T_i \in \{0, 1\} \quad \forall i \in N \quad (3.9)$$

$$P_j, L_j \geq 0 \quad \forall j \in N \quad (3.10)$$

3.5 Cluster head selection

The production of clusters is followed by exchanging information between one core point and another in the clusters. The sensor nodes send messages to all other nodes that share the same cluster. Fundamental pieces of information are involved, such as the rested power, status, position, members' number, and overall number of devices within clusters.

As for this stage, all nodes will carry the same information about the other nodes within the shared cluster. This requires each node to execute Equation (3.1), by means of the information referred to earlier. All devices within this cluster will execute the same equation. The nodes that obtain the most favourable results will be assigned as the CH. The execution of the equation occurs in form of distribution, after which the CH will be identified.

$$FitVal_j = \frac{E_{remaining}}{E_{initial}} + (1 - ((\sum_{i \in N} |S_j(x, y) - S_i(x, y)|) / Max_dis)) \frac{S_j(Members)}{Cluster(Members)} \quad (3.1)$$

Where $E_{remaining}$ is the residual energy of the node j , $E_{initial}$ is the initial energy value of node j , N is the number of nodes in the current cluster, $S_j(x, y)$ and $S_i(x, y)$ refer to the locations of nodes S_j and S_i respectively, Max_dis refers to the maximum summation distance between any point from all points in the cluster. $S_j(Members)$ indicates the number of nodes members of node j , $Cluster(Members)$ indicates the whole number of nodes in the cluster.

All CHs in the network cluster are independently assigned and selected according to distribution.

3.6 Scheduling

The scheduling stage starts with every period, after the clustering phase ends.

3.6.1 Activity scheduling based Cuckoo Algorithm (CS)

This type of CS is described as being evolutionary, as it optimizes the performance globally, according to the special breeding technique of the cuckoo bird [18], besides the Levy flight which is a pattern that birds use when looking for their food. To begin with, the population initialization takes place, consisting of more than one possible solution that is generalized randomly. They are improved by generation, reaching the maximal number or any alternative condition. One solution is the improving procedure, taking place through the application of the Levy flight. This enables the disposal of the solution that is of least benefit, keeping the more favourable ones.

All the CHs are optimized using the energy efficient mechanism based scheduling CS. The CS optimization obtains the best cover sets of active nodes according to their centres only. They are in charge of the sensing process when being monitored. The local random walk is stated as follows:

$$x_i^{(k+1)} = \begin{cases} x_i^{(k)} + ra & \text{if } r'a > Pa \\ x_i^{(k)} & \text{otherwise.} \end{cases} \quad (3.11)$$

Where ra and $r'a$ are two random numbers in range (0, 1).

As for the suggested algorithm, the following assumptions stay true:

- Sensor coverage are circular formed, and all sensors share the same covering with radius R_s .

- No sensing occurs across walls (boundaries or obstacles).
- No change occurs in sensing quality within R_s , and its value stays zero when outside it, based on a binary model.

The following algorithm represents the scheduling CS used to obtain the best node scheduling, which remains active throughout the monitoring of the present period.

Algorithm 3-2 Scheduling based CS

Input: POP_S: is the population size

Output: Gbest: is the best solution(nest)

```

1: Initialize population of POP_S solutions(nests);
2: Population transformed into 0 or 1 by (2.6);
3: Evaluate each solution(nest) using (3.7);
4: Update best solution Gbest;
5: While Stopping criteria is not satisfied do
6:   For i = 1 to POP_S
7:     Generate new solution(nest)  $y_i(\text{new})$  via (2.2);
8:     Evaluate  $y_i(\text{new})$  via (3.7);
9:     If  $\text{fit}(y(\text{new})) < \text{fit}(y_i(\mathbf{k}))$ ;
10:       $y_i(\mathbf{k}+1) = y_i(\text{new})$ ;
11:     Else
12:       $y_i(\mathbf{k}+1) = y_i(\mathbf{k})$ ;
13:     End If;
14:   End For;
15:   For i = 1 to POP_S
16:     Generate a new solution  $y_i(\text{new})$  via (3.11);
17:     Evaluate Individual  $y_i(\text{new})$  via (3.7);
18:     If  $\text{fit}(y_i(\text{new})) < \text{fit}(y_i(\mathbf{k}))$ ;
19:       $y_i(\mathbf{k}+1) = y_i(\text{new})$ ;
20:     Else
21:       $y_i(\mathbf{k}+1) = y_i(\mathbf{k})$ ;
22:     End If;
23:   End For;
24:   Update the best solution Gbest;
25: End While;
26: return Gbest;

```

There exists a matrix of k rows and n columns in the initial population, which represents the number of nests and sensors in a respective order. The CS can be outlined in the following way.

- a) **Generation of Initial Cuckoo Population:** The nest in the initial population presents a probable solution for the sensor scheduling, so as to cover the clusters overall. The values associated with the initial population are real values, based on the CS algorithm.
- b) **Representation of Solution:** The Cs aims to identify the optimization scheduling to monitor the regions in the steps that follow. Nests are sensor nodes schedules, involving a number of eggs. The eggs can have one of two values: (1), which implies that the device is active, or (0), which represents the state of the device being idle. The population is converted from continuous to discrete, using the sigmoid function.
- c) **Fitness function:** After evaluating all entities, the fitness function is applied using Eq. (3.7), so as to assign fitness values. The suggested algorithm implies that lower fitness values are more favorable, as they indicate that the entity is more likely to remain active and survive.
- d) **Generation of new nests $x_i^{(k+1)}$:** The standard CS deploys each of the global and local random walk with a random combination. They are given Eq. 2.2 and Eq. 3.11, respectively.

- e) **Evaluation of new nests:** The evaluation of the new individuals is obtained through Eq. 3.7.
- f) **Immigration:** After evaluation, the algorithm is applied in two stages: the replacement of nests by a new solution obtained through the random walk and Levy flight, and the calculation of the $P\alpha$ fraction for the least suitable nest in order to replace them.
- g) **Updating the Gbest:** The global best will be changed based on the newer best solution that has been obtained using the Cs.

3.6.2 Activity scheduling based Hybrid Metaheuristic (CA+GA)

Approach for Sensor Node Scheduling

In this dissertation, the cuckoo search (CS) meta-heuristic optimization method is combined with the well-known genetic algorithm (GA) to form the CS–GA hybrid algorithm is used to minimizing the amount of energy consumption and ensuring sufficient coverage for the monitored area while maximizing the network lifespan for WSN. The balance between exploration and exploitation abilities was further enhanced by embedding the genetic operators in standard CS, and larger search space was seen during the algorithms' performance. This protocol consists of two phases as in the previous protocol: Clustering using DBSCAN protocol. Scheduling using Hybrid Metaheuristic (CA+GA) Approach and based on the same proposed scheduling optimization model that explained previously.

Algorithm 3-3Scheduling based (CA+GA)

Input : POP_S: is the population size

Output : Gbest:is the best solution(nest)

```

1:  Initialize population of POP_S solutions(nests);
2:  Population transformed into 0 or 1 by (2.6);
3:  Evaluate each solution(nest) using (3.7);
4:  Update best solution Gbest;
5:  While Stopping criteria is not satisfied do
6    Generate a new population via employing the operators of the
      genetic algorithm which is selection, crossover and mutation
7:    Generate new solution(nest) yi(new) via (2.2);
8:    Evaluate yi(new) via (3.7);
9:    If fit(y(new) < fit (yi(k));
10:     yi(k+1) = yi(new);
11:   Else
12:     yi(k+1) = yi(k);
13:   End If;
14:   Update the best solution Gbest;
15: End While;
16: return Gbest;

```

The CA-GA can be explained in more details as follows.

- The initial population's creation: The initial population is made up of a set of host nests that are randomly distributed around the search space. A solution is represented by an egg, which is kept in a nest.
- Creating a new population: Generate a new population via employing the operators of the genetic algorithm which is selection, crossover and mutation .Tournament selection includes recombination or crossbreeding to create two new offspring(eggs) from two parents (cuckoos) chosen from the population. Crossover is a method of recombining the genetic material of two parent cuckoos to generate two eggs with features similar to their parents.

The technique of uniform crossover was used. Each cuckoo mutates independently after the crossover procedure. Its goal is to preserve population variety by preventing early convergence. The flip mutation technique is employed.

- Strategy of elitism: Every time, both after the crossover and after the mutation, The best solutions are chosen and kept in the population through a type of elitism.
- Creation of new solutions: Following the establishment of a new population, the eggs that have grown into cuckoo birds migrate to a better environment for the rest of their lives via Lévy flight and so on.

3.7 Monitoring

- After the production of the ideal sensor scheduling using metaheuristic algorithm, the CH will inform all devices about their status during the next round (monitoring). The message will be either (0), indicating that it should remain idle, or (1), which implies that the sensor devices has to stay active during the following stage.

3.8 Summery

The design of a WSN requires special attention when it comes to the enhancement of the network lifespan. This dissertation proposed a framework for sensor scheduling in WSNs. First the DBSCAN is implemented to cluster the sensor nodes in the sensing field and then one

cluster head will be elected for each period. A scheduling optimization model is developed to schedule the sensor nodes according to certain criteria. After that, one of the two metaheuristics approaches is employed (Cuckoo Algorithm and Hybrid Metaheuristic (CA+GA)) to execute the scheduling optimization approach to find the best set of active nodes per period.

CHAPTER FOUR

RESULT, ANALYSIS, AND DISCUSSIONS

4.1 Overview

Here, the evaluation of the performance and the results of simulation are displayed as a graphs and discussion for the proposed approaches presented in chapter Three. The goal is twofold: firstly, to assess approaches performance using differ performance metrics, secondly, to compare the proposed approaches with competitive protocols belong to the same field.

4.2 The Experiment Results of the DBSCAN clustering algorithm

This section focuses on evaluating the proposed distributed clustering-based DBSCAN protocol using different performance metrics like cluster count, remaining energy, dead nodes number, packets number transmitted to cluster head, number of packets sent to cluster sink, and network lifetime. The conduction of the simulation results is performed by means of a C++ custom simulator for 2500 iterating round, so that several plots are obtained. The sensor nodes are deployed in the monitored area in a random manner. The location of the sink is in the central of the monitored area, with no limitation on energy, normal nodes would have its limitations in terms of energy, memory, and processing capabilities. The suggested protocol is applied for generating the results with regards to the parameters referred to in Table (4.1).

Table 4-1 Stimulation Parameters

Symbol	Description	The Value
$X1_m$	distance at X-axis	400 m
$Y1_m$	distance at Y-axis	400m
N_s	WSN size	100 nodes
P_{Tx}, P_{Rx}	The initial energy	0.5 Jule
Eng_{mp}	Consumption of energy for receiving	0.0013/pJ/bit/m ⁴
Eng_{fs}	Free space model of energy dispersion	10/pJ/bit/m ²
Eng_{amp}	Dispersion of Energy:power amplifier	100/pJ/bit/m ²
$Energy_{DA}$	Consumption of energy for collection	5/nJ/bit
d_0	Reference distance	87 m
I	Packet size	4000 bits

As for this work, the packet size tends to be relatively larger, namely be 4000 bits. The proposed protocol in this paper is named as DBSCAN. The DBSCAN protocol is applied the same energy consumption model that employed in[28]. An obvious result of stimulation is that the DBSCAN outperforms the I-LEACH [28]and LEACH [14]protocols in light of several performance metrics.

4.2.1 Cluster count

The number of CHs is found to extend up to 2000 rounds for the DBSCAN, whereas it reaches only 1750 and 850 rounds for I-LEACH and LEACH respectively. Figure 4.1 shows the cluster count for the LEACH, I-LEACH, and DBSCAN.

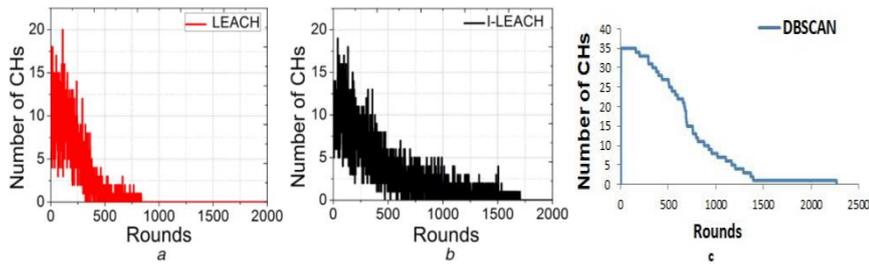


Figure 4-1CH Count (a) LEACH, (b) I-LEACH, (c) DBSCAN

4.2.2 Average Residual Energy

Similarly, the stimulation results in Figure 4.2 present the fact that at just “500 rounds”, the average power of LEACH reaches 0, while I-LEACH goes on to ~“1250 rounds”, the DBSCAN continues to 1750. The rise in rounds leads to the energy depletion of sensor nodes, until they die.

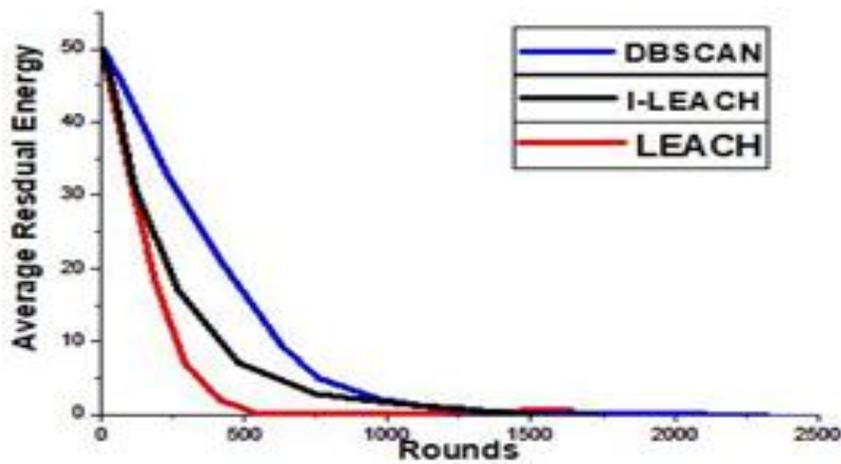


Figure 4-2Network Performance Average Residual Energy

4.2.3 Dead nodes

The network Lifetime through the representation of dead nodes illustrates in Figure 4.3. After 750 rounds, the number of nodes alive levels out at 0 for LEACH, whereas a few nodes remain active till 1500 rounds with I-LEACH. The DBSCAN, on the other hand, has several nodes that remain active till 2000 rounds.

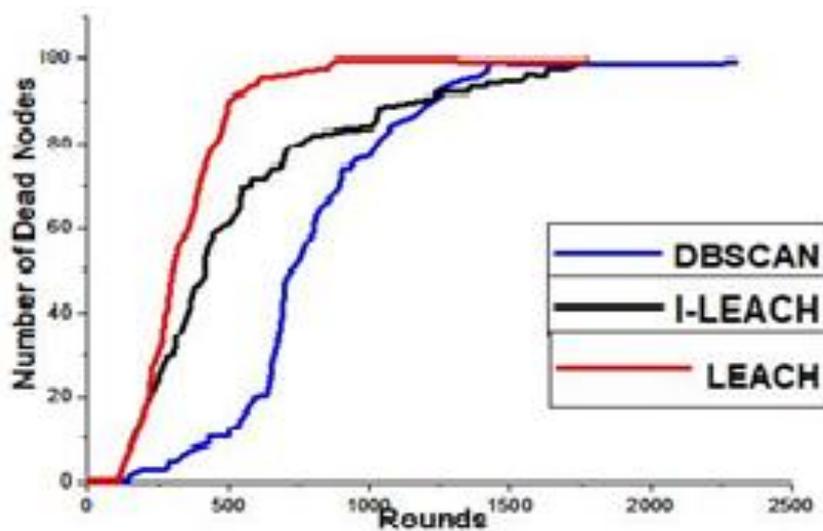


Figure 4-3 Lifetime Metrics Dead nodes

4.2.4 Sent Packets

The amount of data packets transmitted to BS within LEACH and I-LEACH reaches maximally (0.5×10^4) and (1.75×10^4) respectively. As for the DBSCAN, the value may reach (2×10^4), as is shown in Figure 4.4a. A similar increase in data packets transmitted to Cluster Head (CH) is noticed in Figure 4.5b, proving the effectiveness of the suggested protocol. This illustrates the whole situation for the maximization of the

network lifespan, merely due to assigning various energy levels for differing communicating modes in the network.

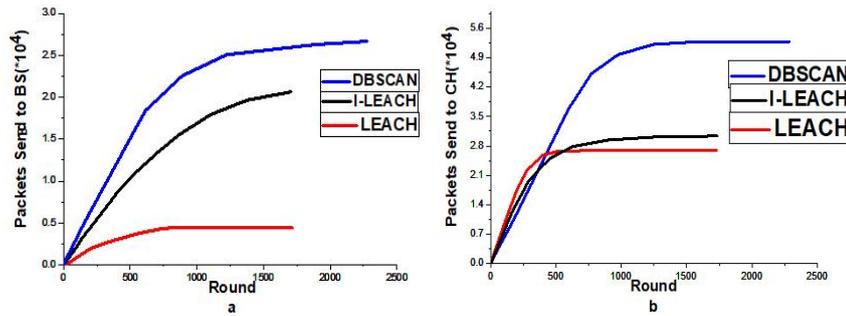


Figure 4-4 Sent Packets (a) to BS, (b) to CH

4.2.5 Network Lifetime

Studying the algorithm behavior in different uses requires the comparison of lifespan metrics with regards to the “first node dead (FND)” and “last node dead (LND)”, as illustrated in Table 4.2. Simulation is performed with regards to three differing areas (100, 200, and 400) m² with a network that is poorly to richly deployed with sensors. Keeping the initiated energy at (0.5 J) on a poorly deployed area of 100 m² improves the network timespan to 1.16, 1.36, and 1.3 for I-LEACH, LEACH, and CPCHSA respectively. There is a positive relation between the lifespan and number of nodes (keeping the area and energy constant), as it increases to 1.04, 1.87, and 1.19 times the value for I-LEACH, EECS and LEACH, respectively. Doubling the area to 100 nodes, results in a lifespan of 1.16, 1.3, and 1.47 times the value for I-LEACH, ModLeach

and LEACH respectively. At 400 nodes, the lifespan is 1.23, 1.4, and 1.61 times the value for I-LEACH, ModLeach, and LEACH subsequently. With the initiated energy of 1 J for 1000 nodes, the lifespan is 1.4, 1.4 and 3.5 times the value for I-LEACH, LEACH and EECS. Comparing the DBSCAN to I-LEACH and LEACH over a wider area of 400 m² shows the increase of the stability period by (1.23, 2.6) and (1.28, 1.7) and (1.58, 1.7) times for 100, 400 and 1000 nodes, respectively. One can therefore draw the conclusion that the suggested protocol proves a more favorable performance with both smaller and larger areas, regardless whether the networks were poorly or richly covered with nodes.

Table 4-2 The Different Scenarios for the Network Lifetime

Nodes	Energy	Area	LEACH		ILEACH		DBSCAN		Other protocols	
			FDN	LDN	FDN	LDN	FD	LD	FDN	LDN
100	0.5	100	980	1450	1050	1700	1150	2292	600 [172]	1500 [172]
		200	780	1150	850	1450	676	1919	200 [28]	1300 [28]
		400	100	800	98	1700	194	1812	-	-
400	0.5	100	1000	1500	1100	1700	1171	2141	820 [118]	950 [118]
		200	850	1300	900	1700	681	1980	190 [28]	1500[28]
		400	100	1200	100	1600	316	1807	-	-
1000	1	100	2000	2700	2000	2700	2005	4441	-	-
		200	1700	2600	1700	2650	1121	3661	810 [118]	1050 [118]
		400	300	2500	300	2700	842	3554	-	-

4.2.6 The Experiment Results of the Cuckoo Scheduling Algorithm

In this section, the evaluation is made regarding the efficiency of the Sensor Device Scheduling based Cuckoo Algorithm (SeDeSCA) through executing multiple experiments with the use of C++ custom simulator. Table 4.3 presents the parameters applied in the simulating process. Fifty executions have been performed by means of different

WSN topologies. The presented results indicate the average rate of these executions. Five network sizes from 100 to 300 nodes have been used in the simulation process, deploying nodes in a controlled manner over a sensing area of $(50 \times 25) \text{ m}^2$ for ensuring a full coverage for the present area of interest. The suggested protocol uses the energy model discussed in [54].

Table 4-3 Stimulation Parameters

Parameter	Value
Field of the Sensing	$(50 \times 25) \text{ m}^2$
WSN size	100, 150, 200, 250 and 300 nodes
Range of the initial Energy	500-700 joules
R_s	5m
R_c	10m
POP_S	60
δ	0.02
θ	0.03
σ	0.95

The range between [500-700] involves a random initialization of sensor node energy. The evaluation of the SeDeSCA protocol is based on the same performance indicators that were employed in [54], including Coverage Ratio, Active Sensors Ratio, Network Lifetime, and Energy Consumption. There are three additional methods applied in this comparison: the DESK [51], GAF [52], and PeCO [54].

4.2.7 Coverage Ratio

The average values of covering ratio for each of the SeDeSCA, DESK, GAF, and PeCO with 200 nodes are illustrated in Figure 4.5. During initial stages, the DESK, GAF, and PeCO result in a slightly better coverage rates (99.99%, 99.96%, and 98.76% respectively), as compared

to the (97.5%) provided by SeDeSCA. This can be traced back to the fact that SeDeSCA shuts down relatively more redundant nodes than DESK, GAF, and PeCO. After the 71th period, the SeDeSCA seems to provide more favourable coverage efficiency than alternative techniques, with the maintenance of a coverage rate of about 80% for several rounds. Such an increase in efficiency is the result of the large quantity of energy that has been saved by SeDeSCA during the initial rounds.

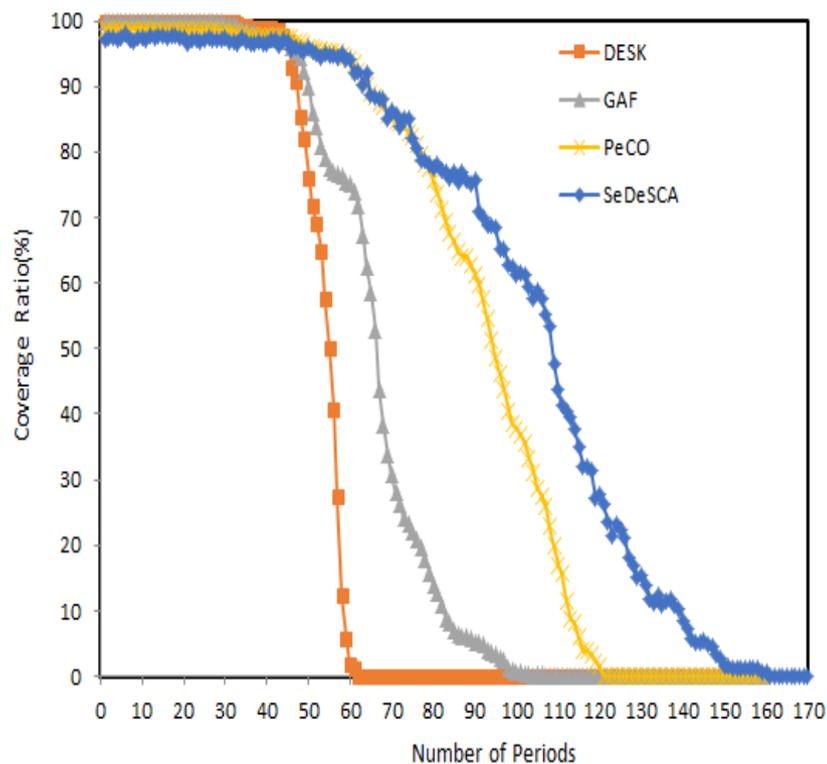


Figure 4-5 Coverage ratio for WSN consisting of 200 deployment nodes.

4.2.8 Active Sensors Ratio

The life span of WSNs could be maximized by conserving energy. This is realized by limiting the number of nodes that remain active during

the stages. As presented in the diagram below, period one to 15 activated 30.68 %, 34.5 % and 20.18 nodes for each of DESK, GAF and PeCO respectively. The SeDeSCA, on the other hand, required the activation of only 19.8% sensor nodes. A gradual increased in activated nodes is observed over time in SeDeSCA, in order to expand the coverage rates, as shown in Figure 4.6.

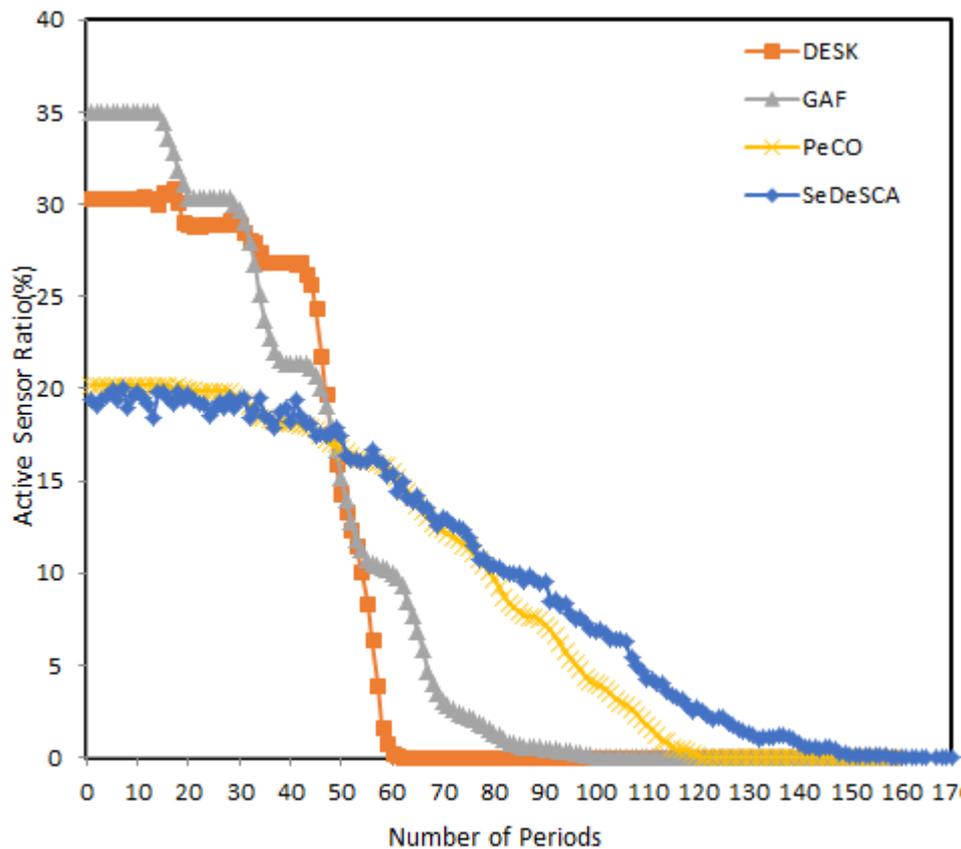


Figure 4-6 Active sensors ratio for WSN consisting of 200 deployed nodes.

4.2.9 Energy Consumption

The current subsection introduces the effect of energy consumption on behalf of the network throughout several statuses of the sensor node (like during the modes of communicating, computing, and listening, as

well as the active and sleep statuses), for different WSN sizes. An investigation is made of other approaches being compared. Each of Table 4.4 and 4.5 illustrate the amount of energy consumption for different WSN sizes, in addition to Lifespan₉₅ and Lifespan₅₀. The relative superiority of SeDeSCA in terms of economizing could be concluded from the illustration. Both Tables indicate the reduction in the amount of energy consumed by SeDeSCA in comparison with other methods. The rate of energy consumption hits relatively lower for Lifespan₉₅ and Lifespan₅₀.

Table 4-4 Energy consumption per round for Lifetime95

WSN-Size	DESK	GAF	PeCO	SeDeSCA
100	1838.643862	1563.75989	1100.102555	955.5551782
150	2049.984991	1651.88042	1155.023859	1069.753857
200	2286.129699	1736.76608	1242.260565	1214.561651
250	2703.422153	1830.1949	1331.442999	1361.093725
300	3080.548003	1932.54577	1486.527288	1465.981447

Table 4-5 Energy consumption per round for Lifetime50

WSN-Size	DESK	GAF	PeCO	SeDeSCA
100	1923.436152	2012.71794	1319.914893	1299.193461
150	2294.227782	2067.8668	1412.642551	1267.272026
200	2562.795067	2110.78426	1538.27953	1399.149407

250	2898.722815	2232.7196	1706.155994	1587.945125
300	3327.300428	2341.87753	1930.953805	1715.481298

4.2.10 Network Lifespan

By comparing the resulting data, the conclusion can be drawn that SeDeSCA provides a relatively more efficient Lifespan improvement for Lifespan₅₀, while SeDeSCA functions optimally for Lifespan₉₅. Tables 4.6 and 4.7 show the exhibition of Lifespan₉₅ and Lifespan₅₀ at different WSN densities, indicating that SeDeSCA enhances the network lifespan remarkably along with the increase in WSN size. The network lifespan for Lifespan₉₅ (see Table 4.6) could be enhanced by means of the SeDeSCA protocol up to 21.6%, 21.2% and 2.5 for each of DESK, GAF and PeCO respectively. As for Lifespan₅₀ (see Table 4.7), the SeDeSCA technique enhances the network Lifespan up to 42.5%, 27.9% as in comparison to the DESK and GAF protocols, respectively.

Table 4-6 Network Lifespan for Lifespan95

WSN-Size	DESK	GAF	PeCO	SeDeSCA
100	29	19	30	28
150	33	33	45	39
200	45	47	57	59
250	53	55	65	68
300	58	65	74	84

Table 4-7 Network Lifespan for Lifespan50

WSN-Size	DESK	GAF	PeCO	SeDeSCA
100	32	35	50	55
150	44	51	74	75
200	55	66	94	90
250	60	79	113	102
300	66	91	125	125

4.2.11 Execution Time

The viability of our approach in real-world wireless sensor networks with limited resources is investigated in this part, as well as why the suggested technique is superior to other distributed techniques. For numerous methodologies and varied network sizes, The average time it takes to solve our suggested scheduling optimization problem is displayed. Using a Lenovo computer with an Intel Core i5 2520 M processor and a "MIPS (Million Instructions per Second)" rate of 50350, we calculate the original execution time. To execute the optimization resolution on a sensor node containing an "Atmel AVR ATmega103L microcontroller (6 MHz)" and a "MIPS" rate of 6, this time is multiplied by $2097.9 \left(\frac{50350}{4} \times \frac{1}{6} \right)$ and reported on Figure 4.9.

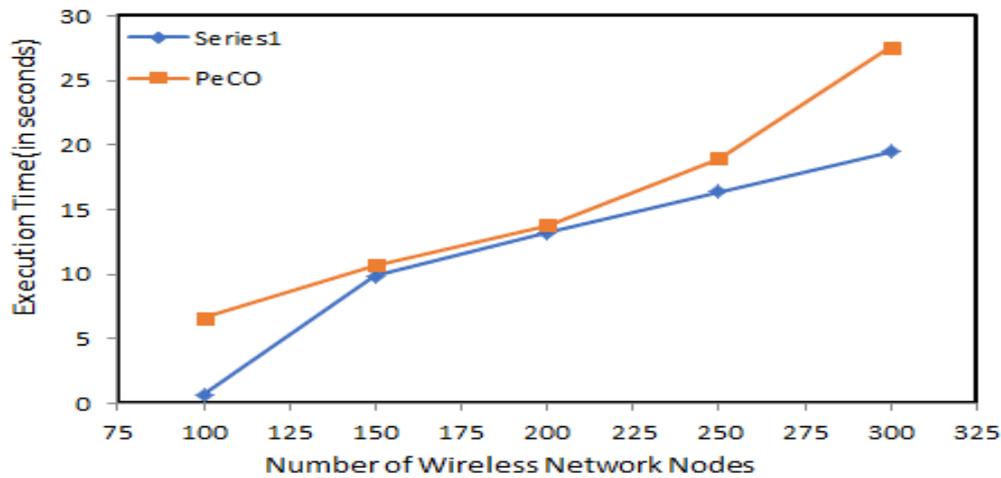


Figure 4-7 Execution Time (in seconds)

Figure 4.9 shows that due to the use of a metaheuristic method (Cuckoo Algorithm), the SeDeSCA technique has a faster execution time than other methods rather than using the optimization solver, to discover effective approximation solutions for the scheduling optimization problem.

4.3 The Experiments Result of the Hybrid Metaheuristic (CA+GA) Scheduling Algorithm.

In this section, the evaluation is made regarding the efficiency of the Hybrid Metaheuristic (CA+GA) through executing multiple experiments with the use of C++ custom simulator. The parameters applied in the simulating process are present in the Table 4.3. Fifty executions have been performed by means of different WSN topologies. The presented results indicate the average rate of these executions. Five network sizes from 100 to 300 nodes have been used in the simulation process, deploying nodes in a controlled manner over a sensing area of (50×25) m² for ensuring a

full coverage for the present area of interest. The suggested protocol uses the energy model discussed in[54].

The range between [500-700] involves a random initialization of sensor node energy. The evaluation of the CA+GA protocol is based on the same performance measures as [54], including Coverage Ratio, Active Sensors Ratio, Network Lifetime, and Energy Consumption. There are three additional methods applied in this comparison: the DESK[51], GAF[52], and PeCO[54].

4.3.1 Coverage ratio

Each of these 5 protocols' average coverage ratios for 200 nodes are as follows: presented in the Figure 4.10 . During the first rounds, DESK, GAF, PeCO and SeDeSCA result in a slightly better coverage rates (99.99%, 99.96%, 98.76% and 97.52 respectively), as compared to the (96.4%) provided by CA+GA. The reason behind this is that CGA turns off relatively more redundant nodes than DESK, GAF, PeCO and SeDeSCA. After the 70th round, the CA+GA tends to result in a more favourable coverage performance than the alternative protocols, maintaining a coverage rate over 80% for many rounds.

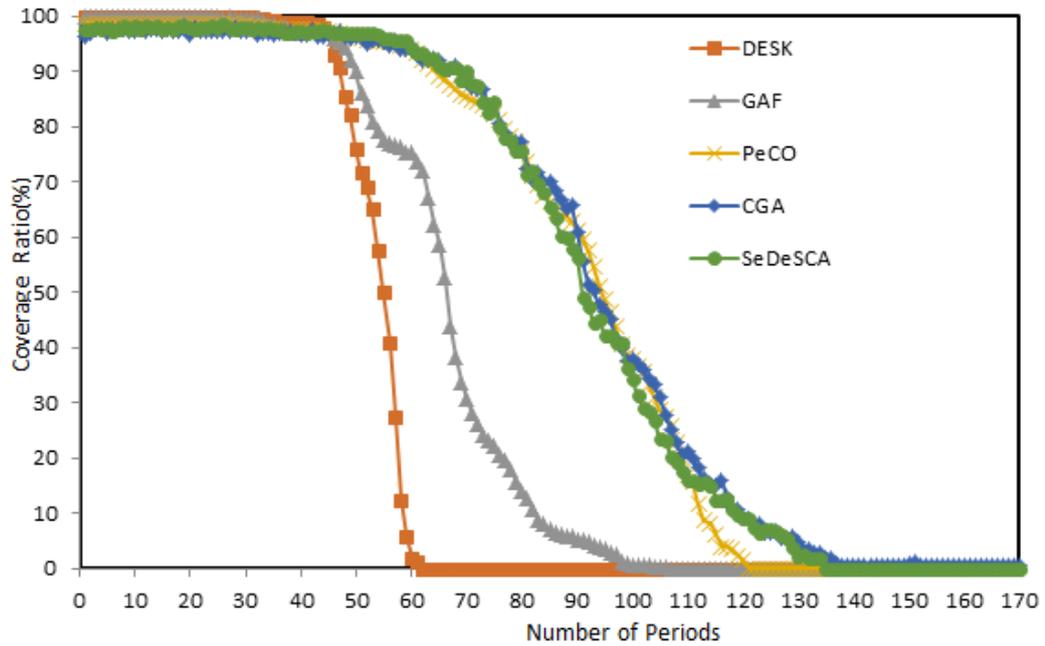


Figure 4-8 Coverage ratio for WSN consisting of 200 deployment nodes.

4.3.2 Active Sensor Ratio

The average ratio of active nodes for every 200 deployed ones illustrates in Figure 4.9. During the first fifteen rounds, DESK, GAF, PeCO and SeDeSCA activated 30.68%, 34.5%, 20.18 and 19.75 nodes respectively, whereas CA+GA activated only 17.5% sensor nodes.

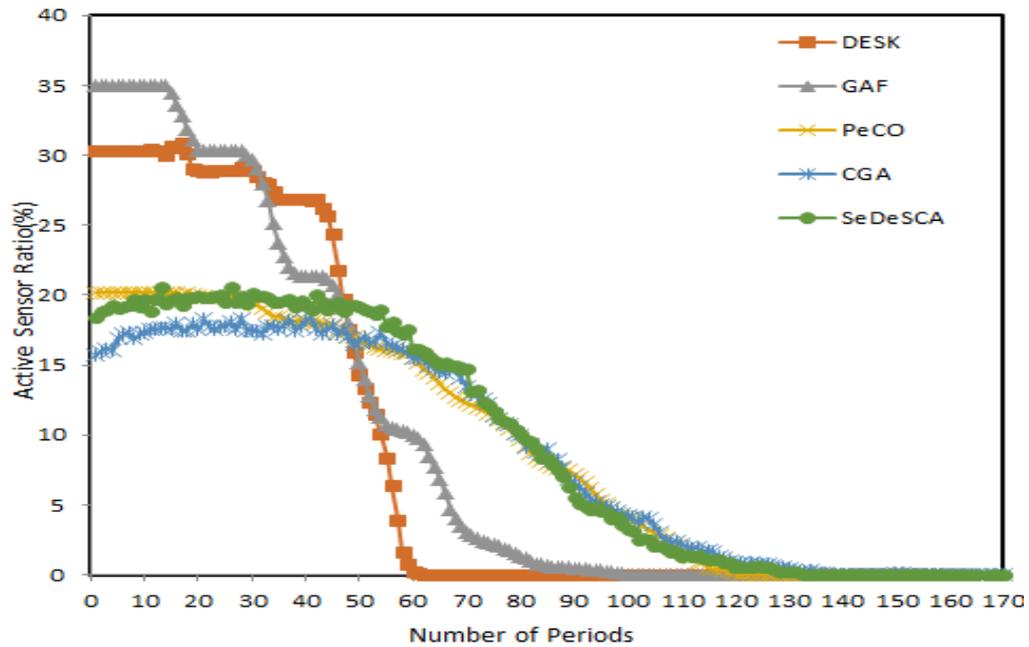


Figure 4-9 Active sensors ratio for WSN consisting of 200 deployed nodes.

4.3.3 Energy Consumption

The amount of energy consumption for different WSN sizes, in addition to $Lifetime_{95}$ and $Lifetime_{50}$ illustrates in the Table 4.8 and 4.9. The relative superiority of CA+GA in terms of economizing could be concluded from the illustration. Both figures indicate the reduction in the amount of energy consumed by CA+GA in comparison with alternative protocols. The rate of energy consumption hits relatively lower for $Lifetime_{95}$ and $Lifetime_{50}$.

Table 4-8 Energy consumption per round for Lifetime₉₅

WSN-Size	DESK	GAF	PeCO	CA-GA
100	1838.643862	1563.75989	1100.102555	883.4268115
150	2049.984991	1651.88042	1155.023859	1018.908618
200	2286.129699	1736.76608	1242.260565	1128.266371
250	2703.422153	1830.1949	1331.442999	1208.782245
300	3080.548003	1932.54577	1486.527288	1328.854812

Table 4-9 Energy consumption per round for Lifetime₅₀

WSN-Size	DESK	GAF	PeCO	CA-GA
100	1923.436152	2012.71794	1319.914893	1043.432066
150	2294.227782	2067.8668	1412.642551	1206.008452
200	2562.795067	2110.78426	1538.27953	1300.133591
250	2898.722815	2232.7196	1706.155994	1414.483264
300	3327.300428	2341.87753	1930.953805	1531.909934

4.3.4 Network Lifespan

The exhibition of Lifetime₉₅ and Lifetime₅₀ at different WSN densities illustrates in the Table 4.10 and 4.11, indicating that CA+GA enhances the network lifespan remarkably along with the increase in WSN size. The network lifespan for Lifetime₉₅ (see Table 4.8 (a)) could be enhanced by means of the CA+GA protocol up to 25.1%, 24.7% , 6.9% and 4.5 as compared to the DESK , GAF , PeCO and SeDeSCA protocols respectively. As for Lifetime₅₀ (see Table 4.8 (b)), the CA+GA protocol enhances the network lifetime up to 46.8%, 33.3% , 5.6 and 7.4 for each of DESK, GAF , PeCO and SeDeSCA respectively .

Table 4-10 Network Lifespan for Lifespan₉₅

WSN-Size	DESK	GAF	PeCO	CA-GA
100	29	19	30	29
150	33	33	45	44
200	45	47	57	57
250	53	55	65	72
300	58	65	74	89

Table 4-11 Network Lifespan for Lifespan₅₀

WSN-Size	DESK	GAF	PeCO	CA-GA
100	32	35	50	61
150	44	51	74	79
200	55	66	94	93
250	60	79	113	120
300	66	91	125	130

4.3.5 Execution Time

The viability of our approach in WSNs with limited resources is investigated in this part, as well as why the suggested technique is superior to other distributed techniques. For numerous methodologies and varied network sizes, The average time it takes to solve our suggested scheduling optimization problem is displayed. Using a Lenovo computer with an Intel Core i5 2520 M processor and a “MIPS (Million Instructions per Second)” rate of 50350, we calculate the original execution time. To execute the optimization resolution on a sensor node containing an “Atmel AVR

ATmega103L microcontroller (6 MHz)” and a “MIPS” rate of 6, this time is multiplied by $2097.9(\frac{50350}{4} \times \frac{1}{6})$ and reported on Figure 4.10.

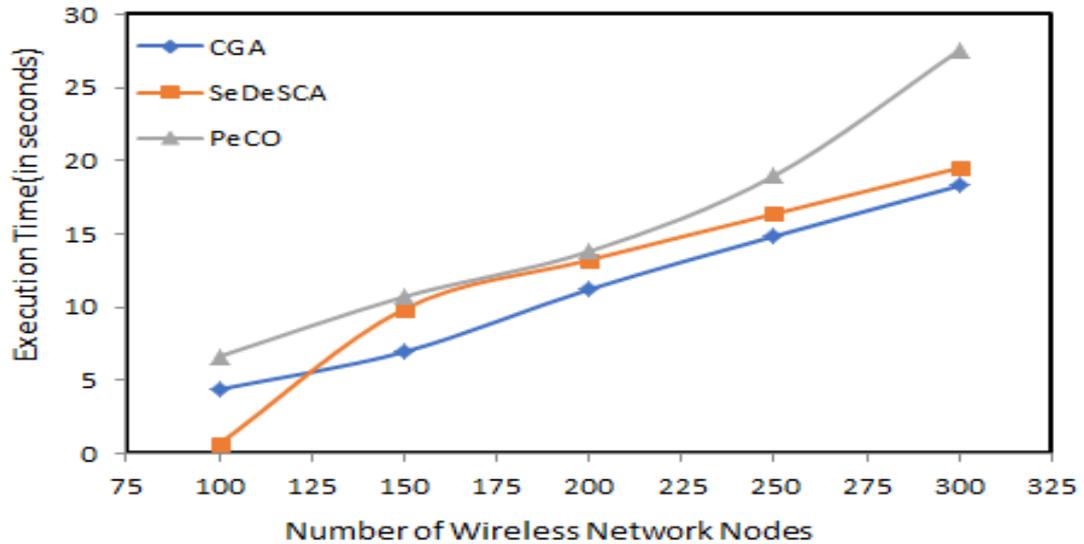


Figure 4-9 Execution Time (in seconds)

Figure 4.14 illustrates that the CA+GA methodology takes less time to execute than other methods because it uses a hybrid metaheuristic (CA+GA) to obtain good approximation solutions to the scheduling optimization problem rather than an optimization solver.

4.4 Summary

As a conclusion for this chapter, the performance evaluations for the three proposed approaches as graphs and discussions were presented. The effectiveness and efficiency of the proposed approaches are proven through using several performance parameters such as cluster count, remaining energy, dead nodes number, packets number transmitted to cluster head, number of packets sent to cluster sink, coverage ratio, energy

consumption, active sensor ratio and network lifetime. A comparison with some existing related works were done in this chapter too.

CHAPTER FIVE

**CONCLUSION AND FUTURE
WORKS**

5.1 Conclusions

1. In this dissertation, we have focused on designing distributed optimization approaches to extend the lifespan of WSNs. We have addressed the sensor scheduling problem and lifespan optimization in WSNs. The ultimate goal is to prolong the WSN lifespan and coverage preservation continuously and efficiently when observing a certain area of interest.
2. The area of interest is clustered using DBSCAN method, and we have proposed a distributed cluster head election to select the cluster head for each cluster. The experimental results have validated our and explained its efficiency in the electing the cluster head after the cluster formation compared with LEACH and I-LEACH.
3. We proposed a sensor node scheduling optimization model that considered three important objectives into account such as increase of covering rate over a particular area, minimizing the number of points covering the same node within the sensing field, and the remaining energy of activated sensor nodes.
4. The sensor node scheduling optimization model is first solved using Cuckoo Algorithm and the proposed approach called SeDeSCA. The results show that the proposed work reduce the active sensor nodes, save energy, extend the lifespan, and maintain a suitable level of coverage compared with DESK, GAF and PeCO.
5. We applied a hybrid metaheuristic approach: Cuckoo Algorithm and Genetic Algorithm to solve the sensor node scheduling

optimization model to provide the best schedule of sensor nodes per period. The conducted results show that the proposed hybrid approach reduce the active sensor nodes, save energy, extend the lifespan, and maintain a suitable level of coverage compared with DESK, GAF and PeCO.

6. Due to the use of metaheuristic approaches that require less time than optimization approach-based branch and bound methods, the SeDeSCA and hybrid approach have less execution times than other methods. Furthermore, hybrid approach is faster than SeDeSCA method.

5.2 Future Works

1. We plan to propose a multiperiod protocol to provide more than one schedule during one execution.
2. The scheduling optimization model can be enhanced to include the heterogeneity of the nodes into account.
3. It is interested to consider a network of mobile sensor nodes to study the effect of the proposed techniques on this type of networks.
4. The problem of target coverage to provide the schedules for certain important targets instead of the whole sensing field is one of our interests in future.
5. The real experimentations represent one of the most our focus in future.

REFERENCES

REFERENCES

REFERENCES

- .1 Akyildiz, I.F., et al., *Wireless sensor networks: a survey*. Computer networks, 2002. **38**(4): p. 393-422.
- .2 Gungor, V.C., B. Lu, and G.P. Hancke, *Opportunities and challenges of wireless sensor networks in smart grid*. IEEE transactions on industrial electronics, 2010. **57**(10): p. 3557-3564.
- .3 Cardei, M. and J. Wu, *Energy-efficient coverage problems in wireless ad-hoc sensor networks*. Computer communications, 2006. **29**(4): p. 413-420.
- .4 Shih, K.-P., et al., *On target coverage in wireless heterogeneous sensor networks with multiple sensing units*. Journal of Network and Computer Applications, 2009. **32**(4): p. 866-877.
- .5 Han, X., et al., *Fault-tolerant relay node placement in heterogeneous wireless sensor networks*. IEEE Transactions on Mobile computing, 2009. **8**(5): p. 643-656.
- .6 Krause, A., et al., *Simultaneous optimization of sensor placements and balanced schedules*. IEEE Transactions on Automatic Control, 2011. **56**(10): p. 2390-2405.
- .7 Yang, D., et al., *Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations*. IEEE Transactions on Mobile Computing, 2011. **11**(8): p. 1399-1411.
- .8 Ashouri, M., et al., *New optimal solution to disjoint set K-coverage for lifetime extension in wireless sensor networks*. IET Wireless Sensor Systems, 2012. **2**(1): p. 31-39.
- .9 Jia, S.G., et al. *An efficient sleeping scheduling for save energy consumption in wireless sensor networks*. in *Advanced Materials Research*. 2013. Trans Tech Publ.

REFERENCES

- .10 Chen, C.-P., et al., *Efficient coverage and connectivity preservation with load balance for wireless sensor networks*. IEEE sensors journal, 2014. **15**(1): p. 48-62.
- .11 Nahar, K.M., et al., *MPF-LEACH: modified probability function for cluster head election in LEACH protocol*. International Journal of Computer Applications in Technology, 2019. **60**(3): p. 267-280.
- .12 Prakash, B. and V. Viswanathan, *A comparative study of meta-heuristic optimisation techniques for prioritisation of risks in agile software development*. International Journal of Computer Applications in Technology, 2020. **62**(2): p. 175-188.
- .13 Heinzelman, W.R., 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*. 2000. IEEE.
- .14 Heinzelman, W.B., A.P. Chandrakasan, and H. Balakrishnan, *An application-specific protocol architecture for wireless microsensor networks*. IEEE Transactions on wireless communications, 2002. **1**(4): p . 670.-660
- .15 Nayak, P. and A. Devulapalli, *A fuzzy logic-based clustering algorithm for WSN to extend the network lifetime*. IEEE sensors journal, 2015. **16**(1): p. 137-144.
- .16 Arumugam, G.S. and T. Ponnuchamy, *EE-LEACH: development of energy-efficient LEACH Protocol for data gathering in WSN*. EURASIP Journal on Wireless Communications and Networking, 2015. **2015**(1): p. 1-9.
- .17 Pachlor, R. and D. Shrimankar, *VCH-ECCR: A centralized routing protocol for wireless sensor networks*. Journal of Sensors, 2017. **2017**.
- .18 Pei, E., et al., *LEAUCH: low-energy adaptive uneven clustering hierarchy for cognitive radio sensor network*.

REFERENCES

- EURASIP Journal on Wireless Communications and Networking, 2015. **2015**(1): p. 1-8.
- .19 Geetha, V., P.V. Kallapur, and S. Tellajeera, *Clustering in wireless sensor networks: Performance comparison of leach & leach-c protocols using ns2*. Procedia Technology, 2012. **4**: p. 163-170.
 - .20 Younis, O. and S. Fahmy, *HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks*. IEEE Transactions on mobile computing, 2004. **3**(4): p. 366-379.
 - .21 Taheri, H., et al. *Improving on HEED protocol of wireless sensor networks using non probabilistic approach and fuzzy logic (HEED-NPF)*. in *2010 5th International Symposium on Telecommunications*. 2010. IEEE.
 - .22 Singh, S.K., P. Kumar, and J.P. Singh, *A survey on successors of LEACH protocol*. Ieee Access, 2017. **5**: p. 4298-4328.
 - .23 Alghamdi, Y. and M. Abdullah, *Improving K-means algorithm by grid-density clustering for distributed WSN data stream*. Int. J. Adv. Comput. Sci. Appl., 2018. **9**(11): p. 583-588.
 - .24 Shankar, A., et al., *Hybrid model for security-aware cluster head selection in wireless sensor networks*. IET Wireless Sensor Systems, 2019. **9**(2): p. 68-76.
 - .25 Wang, J., et al., *A PSO based energy efficient coverage control algorithm for wireless sensor networks*. Comput. Mater. Contin, 2018. **56**(3): p. 433-446.
 - .26 Jan, M., et al., *PAWN: a payload-based mutual authentication scheme for wireless sensor networks*. Concurrency and Computation: Practice and Experience, 2017. **29**(17): p. e3986.
 - .27 Purohit, R. and D. Bhargava, *An illustration to secured way of data mining using privacy preserving data mining*.

REFERENCES

- Journal of Statistics and Management Systems, 2017. **20**(4): p. 637-645.
- .28 Behera ,T.M., U.C. Samal, and S.K. Mohapatra, *Energy-efficient modified LEACH protocol for IoT application*. IET Wireless Sensor Systems, 2018. **8**(5): p. 223-228.
- .29 Djiroun, F.Z. and D. Djenouri, *MAC protocols with wake-up radio for wireless sensor networks: A review*. IEEE Communications surveys & tutorials, 2016. **19**(1): p. 587-618.
- .30 Zhang, Z., et al. *A short review on sleep scheduling mechanism in wireless sensor networks*. in *International conference on heterogeneous networking for quality, reliability, security and robustness*. 2017. Springer.
- .31 Valera, A.C., W.-S. Soh, and H.-P. Tan, *Survey on wakeup scheduling for environmentally-powered wireless sensor networks*. Computer Communications, 2014. **52**: p. 21-36.
- .32 Musilek, P., P. Krömer, and T. Bartoň, *Review of nature-inspired methods for wake-up scheduling in wireless sensor networks*. Swarm and evolutionary computation, 2015. **25**: p. 100-118.
- .33 Kumar, S., O. Kaiwartya, and A.H. Abdullah, *Green computing for wireless sensor networks: Optimization and Huffman coding approach*. Peer-to-Peer Networking and Applications, 2017. **10**(3): p. 592-609.
- .34 Jukuntla, A., K. Pendam, and N. Arjun. *SAWR: Scheduling algorithm for wireless sensor networks with rendezvous nodes*. in *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICIC)*. 2015. IEEE.
- .35 Collotta, M., G. Pau, and V. Maniscalco, *A fuzzy logic approach by using particle swarm optimization for effective energy management in IWSNs*. IEEE Transactions on Industrial Electronics, 2017. **64**(12): p. 9496-9506.

REFERENCES

- .36 Seeniselvi, T. and I. Shanmugapriya, *Artificial Bee Colony Algorithm based delay energy efficient sleep scheduling in wsn*. International Journal, 2014. **3**.(2)
- .37 Kundu, S., et al., *A modified differential evolution-based combined routing and sleep scheduling scheme for lifetime maximization of wireless sensor networks*. Soft Computing, 2015. **19**(3): p. 637-659.
- .38 Hosseinirad, S.M., *Centroid Energy-Based Sleep Scheduling Design in Dense Wireless Sensor Networks using Evolutionary Algorithms*. International Journal Of Computer Science And Network Security, 2017. **17**(6): p. 171-184.
- .39 Guo, W., et al., *A Two-stage Clustering Sleep Scheduling Algorithm with Particle Swarm Optimization in Wireless Sensor Networks*. Adhoc & Sensor Wireless Networks, 2015. **27**.
- .40 Ren, H. and M.Q.-H. Meng. *Biologically inspired approaches for wireless sensor networks*. in *2006 International Conference on Mechatronics and Automation*. 2006. IEEE.
- .41 Iyengar, S.S., et al., *Biologically inspired cooperative routing for wireless mobile sensor networks*. IEEE Systems Journal, 2007. **1**(1): p. 29-37.
- .42 Antoniou, P., et al., *Congestion control in wireless sensor networks based on bird flocking behavior*. Computer Networks, 2013. **57**(5): p. 1167-1191.
- .43 Kennedy, J., R. Eberhart, and R. Swarm, *Intelligence*. 2001, San Francisco: Morgan Kaufman Publishers.
- .44 Indora, S. and D. Singh, *Sleep-Distance Based Sleep-Awake Mechanisms in Wireless Sensor Network*. 2018.
- .45 MUHAMMAD, Z. and N. SAXENA. *Survey on scheduling mechanisms for wireless sensors in IoT scenarios*. in *Proceedings of 102nd IASTEM International Conference, Seoul, South Korea*. 2018.

REFERENCES

- .46 Wang, H., et al., *An effective scheduling algorithm for coverage control in underwater acoustic sensor network*. Sensors, 2018. **18**(8): p. 2512.
- .47 Wan, R. and N. Xiong, *An energy-efficient sleep scheduling mechanism with similarity measure for wireless sensor networks*. Human-centric Computing and Information Sciences, 2018. **8**(1): p. 1-22.
- .48 Wang, Z., et al., *A sensor node scheduling algorithm for heterogeneous wireless sensor networks*. International Journal of Distributed Sensor Networks, 2019. **15**(1): p. 1550147719826311.
- .49 Leão, E., et al. *A hybrid beacon scheduling scheme to allow the periodic reconfiguration of large-scale cluster-tree wsns*. in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. 2018. IEEE.
- .50 Yousid, Y.K., et al., *An Energy-Aware Clustering Protocol Based Grid For WSN*. JOIV: International Journal on Informatics Visualization, 2018. **2**(4): p. 239-244.
- .51 Yu, J., et al., *A stochastic k-coverage scheduling algorithm in wireless sensor networks*. International Journal of Distributed Sensor Networks, 2012. **8**(11): p. 746501.
- .52 Xu, Y., J. Heidemann, and D. Estrin. *Geography-informed energy conservation for ad hoc routing*. in *Proceedings of the 7th annual international conference on Mobile computing and networking*. 2001.
- .53 Idrees, A.K., et al., *Distributed lifetime coverage optimization protocol in wireless sensor networks*. The journal of supercomputing, 2015. **71**(12): p. 4578-4593.
- .54 Idrees, A.K., et al., *Perimeter-based coverage optimization to improve lifetime in wireless sensor networks*. Engineering Optimization, 2016. **48**(11): p. 1951-1972.
- .55 Park, B., *Future directions of information and telecommunication systems through the technological*

REFERENCES

- advancement convergence*. International Journal of Multimedia and Ubiquitous Engineering, 2013. **8**.(1)
- .56 Sangve, S.M. and U.V. Kulkarni, *ANOMALY BASED IMPROVED NETWORK INTRUSION DETECTION SYSTEM USING CLUSTERING TECHNIQUES*. International Journal of Advanced Research in Computer Science, 2017. **8**.(7)
- .57 Miao, G., et al., *Fundamentals of mobile data networks*. 2016: Cambridge University Press.
- .58 Patrick, B., *The Effect of Smartphone on the Society Beau Patrick, Dylan Austin and Charles O. Connor* Computer Science Murdoch University, Australia. 2018.
- .59 Del Castillo, I., et al., *Hardware platform for wide-area vehicular sensor networks with mobile nodes*. Vehicular Communications, 2 :3 .016p. 21-30.
- .60 Antolín, D., N. Medrano, and B. Calvo, *Reliable lifespan evaluation of a remote environment monitoring system based on wireless sensor networks and global system for mobile communications*. Journal of Sensors, 2016. **2016**.
- .61 Zheng, Y ,et al. *Learning transportation mode from raw gps data for geographic applications on the web*. in *Proceedings of the 17th international conference on World Wide Web*. 2008.
- .62 Kandris, D., et al., *Applications of wireless sensor networks: an up-to-date survey*. Applied System Innovation, 2020. **3**(1): p. 14.
- .63 Azzabi, T., H. Farhat, and N. Sahli. *A survey on wireless sensor networks security issues and military specificities*. in *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. 2017. IEEE.
- .64 Saini, R.K., R. Ritika, and S. Vijay, *Data flow in wireless sensor network protocol stack by using bellman-ford routing algorithm*. Bulletin of Electrical Engineering and Informatics, 2017. **6**(1): p. 81-87.

REFERENCES

- .65 Aminian, M. and H.R .Naji, *A hospital healthcare monitoring system using wireless sensor networks*. J. Health Med. Inform, 2013. **4**(02): p. 121.
- .66 Bathiya, B., S. Srivastava, and B. Mishra. *Air pollution monitoring using wireless sensor network*. in *2016 IEEE international WIE conference on electrical and computer engineering (WIECON-ECE)*. 2016. IEEE.
- .67 Al-Ani, K., A. Abdalkafor, and A. Nassar, *An overview of wireless sensor network and its applications*. Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), 2019. **17**(3): p. 1480-1486.
- .68 Erdelj, M., N. Mitton, and E. Natalizio, *Applications of industrial wireless sensor networks*. Industrial wireless sensor networks: applications, protocols, and standards, 2013: p. 1-22.
- .69 Sharma, S., R.K. Bansal, and S. Bansal, *Energy-efficient data collection techniques in wireless sensor networks*. Emerging Communication Technologies Based on Wireless Sensor Networks: Current Research and Future Applications, CRC Press, Taylor & Francis, USA, 2016: p. 275-296.
- .70 Dargie, W. and C. Poellabauer, *Fundamentals of wireless sensor networks: theory and practice*. 2010: John Wiley & Sons.
- .71 Sohrabi, K., et al., *Protocols for self-organization of a wireless sensor network*. IEEE personal communications, 2000. **7**(5): p. 16-27.
- .72 Manjaly, J.J. and J. Sandeep, *An Authentication Protocol for Clustered Wireless Sensor Networks*. International Journal of Advanced Research in Computer Science, 2017. **8**(3)
- .73 Mohanty, P., et al., *SECURITY ISSUES IN WIRELESS SENSOR NETWORK DATA GATHERING PROTOCOLS:*

REFERENCES

- A *SURVEY*. Journal of Theoretical & Applied Information Technology, 2010. **13**.
- .74 Saleh, A.A. and A.S. Abdullah, *International Journal of Wireless and Microwave Technologies (IJWMT)*.
- .75 Kindy, D., *Prof. Al-Sakib Khan PATHAN International Islamic University Malaysia (IIUM), Malaysia Selected Scholarly Contributions SCOPUS*. Muslim World, 2013. **2013**: p. 5th.
- .76 Troubleyn, E., I. Moerman, and P. Demeester, *QoS challenges in wireless sensor networked robotics*. Wireless personal communications, 2013. **70**(3): p. 1059-1075.
- .77 Chen, D. and P.K. Varshney. *QoS Support in Wireless Sensor Networks: A Survey*. in *International conference on wireless networks*. 2004.
- .78 Farsi, M., et al., *Deployment techniques in wireless sensor networks, coverage and connectivity: A survey*. IEEE Access, 2019. **7**: p. 28940-28954.
- .79 Feng, J., S. Megerian, and M. Potkonjak. *Model-based calibration for sensor networks*. in *SENSORS, 2003 IEEE*. 2003. IEEE.
- .80 Xue, Y., Y. Cui, and K. Nahrstedt, *Maximizing lifetime for data aggregation in wireless sensor networks*. Mobile Networks and Applications, 2005. **10**(6): p. 853-864.
- .81 Rajagopalan, R. and P.K. Varshney, *Data aggregation techniques in sensor networks: A survey*. 2006.
- .82 Boulis, A., S. Ganeriwal, and M.B. Srivastava, *Aggregation in sensor networks: an energy-accuracy trade-off*. Ad hoc networks, 2003. **1**(2-3): p. 317-331.
- .83 Ahmed, A., et al. *Wired vs wireless deployment support for wireless sensor networks*. in *TENCON 2006-2006 IEEE Region 10 Conference*. 2006. IEEE.
- .84 Li, J., et al. *Power: Planning and deployment platform for wireless sensor networks*. in *2006 Fifth International*

REFERENCES

- Conference on Grid and Cooperative Computing Workshops*. 2006. IEEE.
- .85 Yadav, J. and S. Mann, *Coverage in wireless sensor networks: A survey*. Int. J. Electron. Comput. Sci. Eng, 2013. **2**: p. 465-471.
- .86 Chen, X., et al. *WSNs Node Localization Algorithm Based on Mobile Robot Assistance*. in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2018IEEE.
- .87 Eisa, I., et al., *Challenges in Wireless Sensor Networks*. International Journal of Advanced Research in Computer Science & Technology, 2016. **4**.(4)
- .88 Rault, T., A. Bouabdallah, and Y. Challal, *Energy efficiency in wireless sensor networks :A top-down survey*. Computer networks, 2014. **67**: p. 104-122.
- .89 Masonta, M., et al. *Energy efficiency in future wireless networks: Cognitive radio standardization requirements*. in *2012 IEEE 17th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2012. IEEE.
- .90 Anastasi, G., et al., *Energy conservation in wireless sensor networks: A survey*. Ad hoc networks, 2009. **7**(3): p. 537-568.
- .91 Hsu, J., et al. *Adaptive duty cycling for energy harvesting systems* .in *Proceedings of the 2006 international symposium on Low power electronics and design*. 2006.
- .92 de Paz Alberola, R. and D. Pesch, *Duty cycle learning algorithm (DCLA) for IEEE 802.15. 4 beacon-enabled wireless sensor networks*. Ad Hoc Networks, 2012. **1** :(4)0 p. 664-679.
- .93 Ba, H., I. Demirkol, and W. Heinzelman, *Passive wake-up radios: From devices to applications*. Ad hoc networks, 2013. **11**(8): p. 2605-2621.

REFERENCES

- .94 Misra, S., M.P. Kumar, and M.S. Obaidat, *Connectivity preserving localized coverage algorithm for area monitoring using wireless sensor networks*. Computer communications, 2011. **34**(12): p. 1484-1496.
- .95 Kumar, D., T.C. Aseri, and R. Patel, *EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor networks*. computer communications, 2009. **32**(4): p. 662-667.
- .96 Li, H., et al., *COCA: Constructing optimal clustering architecture to maximize sensor network lifetime*. Computer Communications, 2013. **36**(3): p. 256-268.
- .97 Younis, M. and K. Akkaya, *Strategies and techniques for node placement in wireless sensor networks: A survey*. Ad Hoc Networks, 2008. **6**(4): p. 621-655.
- .98 Di Francesco, M., S.K. Das, and G. Anastasi, *Data collection in wireless sensor networks with mobile elements: A survey*. ACM Transactions on Sensor Networks (TOSN) :(1)8 .2011 ,p. 1-31.
- .99 Sudevalayam, S. and P. Kulkarni, *Energy harvesting sensor nodes: Survey and implications*. IEEE communications surveys & tutorials, 2010. **13**(3): p. 443-461.
- .100 Zhang, Y., L.T. Yang, and J. Chen, *RFID and Sensor Networks, auerbach publication*. International Standard Book, 2010(978-1): p. 4200-7777.
- .101 Smaragdakis, G., I. Matta, and A. Bestavros. *SEP: A stable election protocol for clustered heterogeneous wireless sensor networks*. in *Second international workshop on sensor and actor network protocols and applications (SANPA 2004)*. 2004. Boston, MA.;
- .102 Ding, P., J. Holliday, and A. Celik. *Distributed energy-efficient hierarchical clustering for wireless sensor networks*. in *International conference on distributed computing in sensor systems*. 2005. Springer.

REFERENCES

- .103 Qing, L., Q. Zhu, and M. Wang, *Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks*. Computer communications, 2006. **29**(12): p. 2230-2237.
- .104 Duan, C. and H. Fan. *A distributed energy balance clustering protocol for heterogeneous wireless sensor networks*. in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*. 2007. IEEE.
- .105 Chen, G., et al., *An unequal cluster-based routing protocol in wireless sensor networks*. Wireless Networks, 2009. **15**(2): p. 193-207.
- .106 Jain, N., P. Sinha, and S. Gupta, *Clustering protocols in wireless sensor networks: A survey*. International Journal of Applied Information System (IJ AIS), 2013. **5**(2)
- .107 Saeidmanesh, M., M. Hajimohammadi, and A. Movaghar, *Energy and distance based clustering: An energy efficient clustering method for wireless sensor networks*. World Academy of Science, Engineering and Technology, 2009. **55**: p. 555-559.
- .108 Cook, D. and S.K .Das, *Smart environments: technology, protocols, and applications*. Vol. 43. 2004: John Wiley & Sons.
- .109 Min, R., et al. *Low-power wireless sensor networks*. in *VLSI Design 2001. Fourteenth International Conference on VLSI Design*. 2001. IEEE.
- .110 Wang ,C., et al. *An improved LEACH protocol for application-specific wireless sensor networks*. in *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. 2009. IEEE.
- .111 Mehta, E.K., *Analysis and exploration of probabilistic and non-probabilistic clustering approaches with topology control mechanisms*. International Journal of Advanced Science and Research, 2016. **1**(5): p. 41-45.

REFERENCES

- .112 Ester, M., et al. *A density-based algorithm for discovering clusters in large spatial databases with noise.* in *kdd*. 1996.
- .113 Schubert, E., et al., *DBSCAN revisited, revisited: why and how you should (still) use DBSCAN.* *ACM Transactions on Database Systems (TODS)*, 2017. **42**(3): p. 1-21.
- .114 Sander, J., et al., *Density-based clustering in spatial databases: The algorithm gbscan and its applications.* *Data mining and knowledge discovery*, 1998. **2**(2): p. 169-194.
- .115 Han, D., et al. *A novel scalable DBSCAN algorithm with Spark.* in *2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2016. IEEE.
- .116 Li, L.-Y. and C.-D. Liu. *An improved algorithm of LEACH routing protocol in wireless sensor networks.* in *2014 8th International Conference on Future Generation Communication and Networking*. 2014. IEEE.
- .117 Soro ,S. and W.B. Heinzelman. *Prolonging the lifetime of wireless sensor networks via unequal clustering.* in *19th IEEE international parallel and distributed processing symposium*. 2005. IEEE.
- .118 Ye, M., et al. *EECS: an energy efficient clustering scheme in wireless sensor networks.* in *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005*. 2005. IEEE.
- .119 Awad, F., E. Taqieddin, and A. Seyam, *Energy-efficient and coverage-aware clustering in wireless sensor networks.* 2012.
- .120 Manjeshwar, A. and D.P. Agrawal. *TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks.* in *ipdps*. 2001.
- .121 Buttyán, L. and P. Schaffer, *Position-based aggregator node election in wireless sensor networks.* *International*

REFERENCES

- Journal of Distributed Sensor Networks, 2010. **6**(1): p. 679205.
- .122 Zu-jue, C., O. Ye-long, and L. Shao-qing, *A gaf algorithm of dislocated grid for wireless sensor network*, in *Advanced Technology in Teaching*. 2012, Springer. p. 721-728.
- .123 Ye, F., et al. *A two-tier data dissemination model for large-scale wireless sensor networks*. in *Proceedings of the 8th annual international conference on Mobile computing and networking*. 2002.
- .124 Delavar, A.G., et al., *SLGC: A new Cluster Routing Algorithm in wireless sensor network for decrease energy consumption*. International Journal of Computer Science, Engineering and Applications, 2012. **2**(3): p. 39.
- .125 Jung, S.-M., Y.-J. Han, and T.-M. Chung. *The concentric clustering scheme for efficient energy consumption in the PEGASIS*. in *The 9th international conference on advanced communication technology*. 2007. IEEE.
- .126 Gautam, N., W.-I. Lee, and J.-Y. Pyun. *Track-sector clustering for energy efficient routing in wireless sensor networks*. in *2009 Ninth IEEE international conference on computer and information technology*. 2009. IEEE.
- .127 Agarwal, A., K. Gupta, and K. Yadav. *A novel energy efficiency protocol for WSN based on optimal chain routing*. in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. 2016. IEEE.
- .128 Backlund, H., A. Hedblom, and N. Neijman, *DBSCAN: A Density-Based Spatial Clustering of Application with Noise*. Linköping University–ITN, Rep. TNM033, 2011.
- .129 Balamurugan, R., A. Natarajan, and K. Premalatha , *Stellar-mass black hole optimization for biclustering microarray gene expression data*. Applied Artificial Intelligence, 2015. **29**(4): p. 353-381.

REFERENCES

- .130 Bianchi, L., et al., *A survey on metaheuristics for stochastic combinatorial optimization*. Natural Computing, 2009. **8**(2): p. 239-287.
- .131 Blum, C. and A. Roli, *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*. ACM computing surveys (CSUR), 2003. **35**(3): p. 268-308.
- .132 Yang, X.-S. and S. Deb, *Engineering optimisation by cuckoo search*. International Journal of Mathematical Modelling and Numerical Optimisation, 2010. **1**(4): p. 330-343.
- .133 Yang, X.-S. and S. Deb. *Cuckoo search via Lévy flights*. in *2009 World congress on nature & biologically inspired computing (NaBIC)*. 2009. Ieee.
- .134 Mlakar, U., I. Fister Jr, and I. Fister, *Hybrid self-adaptive cuckoo search for global optimization*. Swarm and Evolutionary Computation, 2016. **29**: p. 47-72.
- .135 Mlakar, U., et al., *Modified binary cuckoo search for association rule mining*. Journal of Intelligent & Fuzzy Systems, 2017. **32**(6): p. 4319-4330.
- .136 Ouaarab, A., B. Ahiod, and X.-S. Yang, *Improved and discrete cuckoo search for solving the travelling salesman problem*, in *Cuckoo search and firefly algorithm*. 2014, Springer. p. 63-84.
- .137 Ouaarab, A., B. Ahiod, and X.-S. Yang, *Discrete cuckoo search algorithm for the travelling salesman problem*. Neural Computing and Applications, 2014. **24**(7): p. 1659-1669.
- .138 Wang, L. and Y. Zhong, *Cuckoo search algorithm with chaotic maps*. Mathematical Problems in Engineering, 2015. **2015**.
- .139 Mantegna, R.N., *Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes*. Physical Review E, 1994. **49**(5): p. 4677.

REFERENCES

- .140 Li, S., et al., *An improved information security risk assessments method for cyber-physical-social computing and networking*. IEEE Access, 2018. **6**: p. 10311-10319.
- .141 Kennedy, J. and R. Everhart. *A discrete binary version of the particle swarm algorithm*. in *Proceedings 1997 Conference on Systems, Man and Cybernetics*. 1997.
- .142 Gherboudj, A., A. Layeb, and S. Chikhi, *Solving 0-1 knapsack problems by a discrete binary version of cuckoo search algorithm*. International Journal of Bio-Inspired Computation, 2012. **4**(4): p. 229-236.
- .143 Kanagaraj, G., S. Ponnambalam, and N. Jawahar, *A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems*. Computers & Industrial Engineering, 2013. **66**(4): p. 1115-1124.
- .144 Singh, A. and M. Kaur, *A Cuckoo Search based Approach for Solving Standard Cell Placement Problem*. Journal of VLSI Design Tools & Technology, 2014. **4**(2): p. 1-7.
- .145 Kanagaraj, G., S. Ponnambalam, and N. Jawahar, *Reliability-based total cost of ownership approach for supplier selection using cuckoo-inspired hybrid algorithm*. The International Journal of Advanced Manufacturing Technology, 2016. **84**(5-8): p. 801-816.
- .146 Kanagaraj, G., et al., *An effective hybrid cuckoo search and genetic algorithm for constrained engineering design optimization*. Engineering Optimization, 2014. **46** :(10)p. 1331-1351.
- .147 Mohamad, A.B., A.M. Zain, and N.E. Nazira Bazin, *Cuckoo search algorithm for optimization problems—a literature review and its applications*. Applied Artificial Intelligence, 2014. **28**(5): p. 419-448.
- .148 Fakhouri, H.N., A. Hudaib, and A. Sleit, *Multivector particle swarm optimization algorithm*. Soft Computing, 2019: p. 1-19.

REFERENCES

- .149 Wang, Y. and J. Xie. *Ant colony optimization for multicast routing*. in *IEEE APCCAS 2000. 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems.(Cat. No. 00EX394)*. 2000. IEEE.
- .150 Dorigo, M. and C. Blum, *Ant colony optimization theory: A survey*. Theoretical computer science, 2005. **344**(2-3): p. 243-278.
- .151 Dorigo, M., V. Maniezzo, and A. Colorni, *Ant system: optimization by a colony of cooperating agents*. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 1996. **26**(1): p. 29-41.
- .152 Gold, R., *Control flowgraphs and code coverage*. 2010.
- .153 Holland, J.H., *Genetic algorithms*. Scientific american, 1979:(1)267 .92p. 66-73.
- .154 Holland, J.H., *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. 1992: MIT press.
- .155 Fister Jr, I., D. Fister, and I. Fister, *A comprehensive review of cuckoo search: variants and hybrids*. International Journal of Mathematical Modelling and Numerical Optimisation, 2013. **4**(4): p. 387-409.
- .156 Wu, Y.-C., W.-P. Lee, and C.-W. Chien. *Modified the performance of differential evolution algorithm with dual evolution strategy*. in *International conference on machine learning and computing*. 2011.
- .157 Ruan, C., *Genetic Algorithms*. 2017.
- .158 Bai, Q., *Analysis of particle swarm optimization algorithm*. Computer and information science, 2010. **3**(1): p180. .
- .159 Vickers, N.J., *Animal communication: when i'm calling you, will you answer too?* Current biology, 2017. **27**(14): p. R713-R715.
- .160 Ali, N., et al., *A review of firefly algorithm*. ARPN Journal of Engineering and Applied Sciences, 2014. **9**(10): p. 1736.-1732

REFERENCES

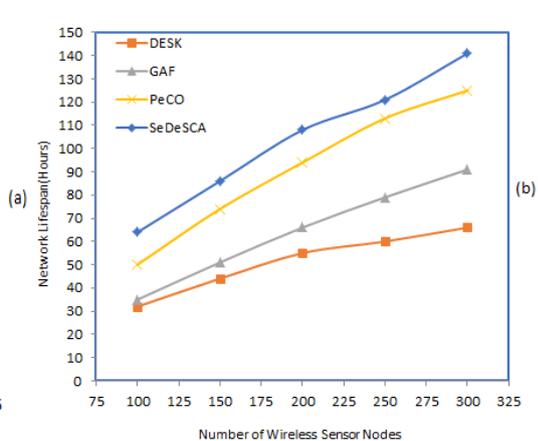
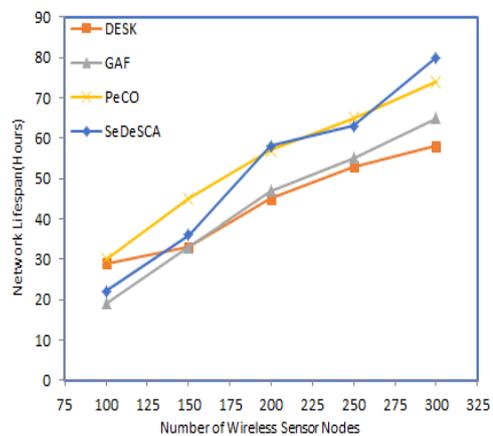
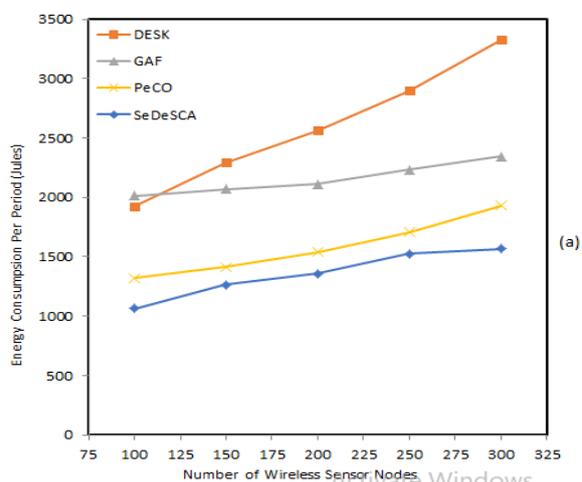
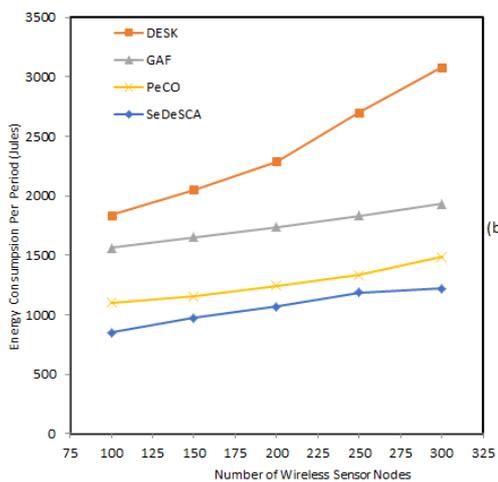
- .161 Selvi, V. and R. Umarani, *Comparative analysis of ant colony and particle swarm optimization techniques*. International Journal of Computer Applications, 2010. **5**(4): p. 1-6.
- .162 Yang, X.-S. and X. He, *Bat algorithm: literature review and applications*. International Journal of Bio-inspired computation, 2013. **5**(3): p. 141-149.
- .163 Wang, Y., et al. *Improved flower pollination algorithm based on mutation strategy*. in *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. 2017. IEEE.
- .164 Yan, G. and C. Li, *An effective refinement artificial bee colony optimization algorithm based on chaotic search and application for pid control tuning*. Journal of Computational Information Systems, 2011. **7**(9): p. 3309-3316.
- .165 Ezugwu, A.E., et al., *A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems*. Neural Computing and Applications, 2020. **32**(10): p. 6207-6251.
- .166 Dietrich, I. and F. Dressler, *On the lifetime of wireless sensor networks*. ACM Transactions on Sensor Networks (TOSN), 2009. **5**(1): p. 1-39.
- .167 Cardei, M. and J. Wu, *Coverage in wireless sensor networks*. Handbook of sensor networks, 2004. **21**: p. 201-202.
- .168 Blough, D.M. and P. Santi. *Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks*. in *Proceedings of the 8th annual international conference on Mobile computing and networking*. 2002.
- .169 Wang, X., et al. *Integrated coverage and connectivity configuration in wireless sensor networks*. in *Proceedings*

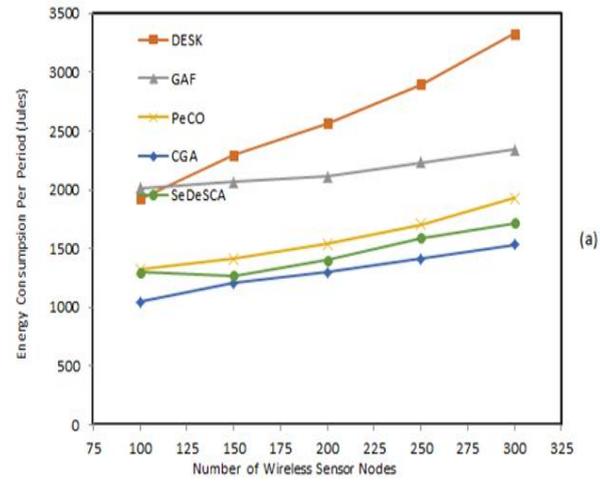
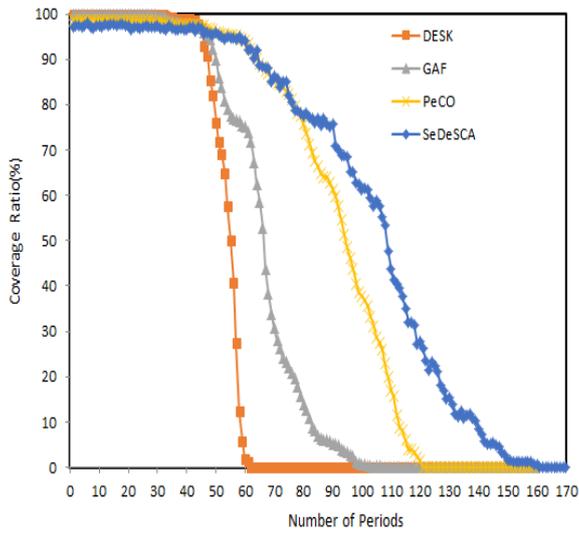
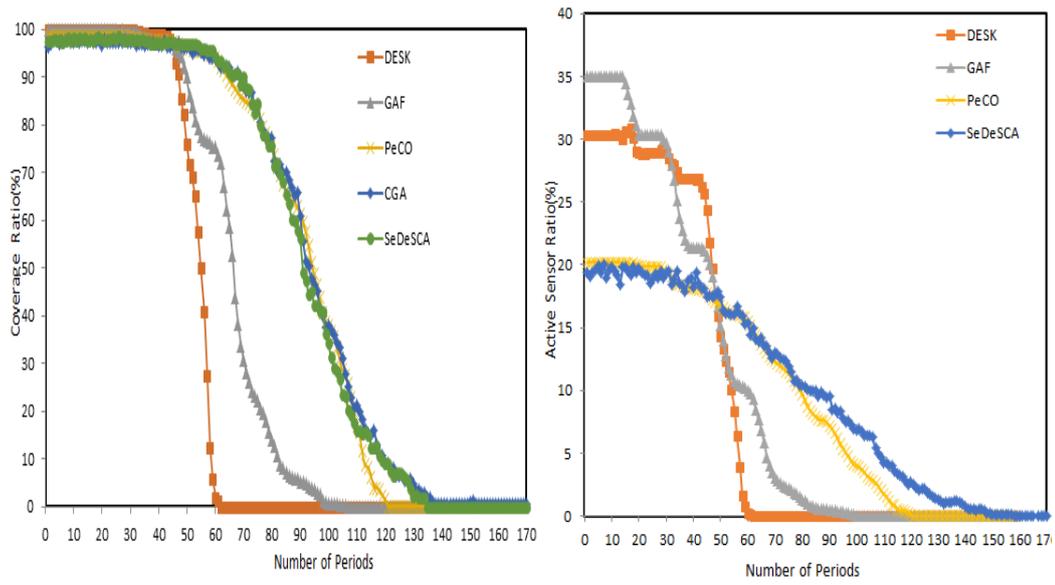
REFERENCES

- of the 1st international conference on Embedded networked sensor systems*. 2003.
- .170 Xing, G., et al., *Integrated coverage and connectivity configuration for energy conservation in sensor networks*. ACM Transactions on Sensor Networks (TOSN), 2005. **1**(1): p. 36-72.
- .171 Kumar, S., A. Arora, and T.-H. Lai. *On the lifetime analysis of always-on wireless sensor network applications*. in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005*. 2005. IEEE.
- .172 Tsai, Y.-R., *Coverage-preserving routing protocols for randomly distributed wireless sensor networks*. IEEE Transactions on wireless communications, 2007. **6**(4): p. 1240-1245.

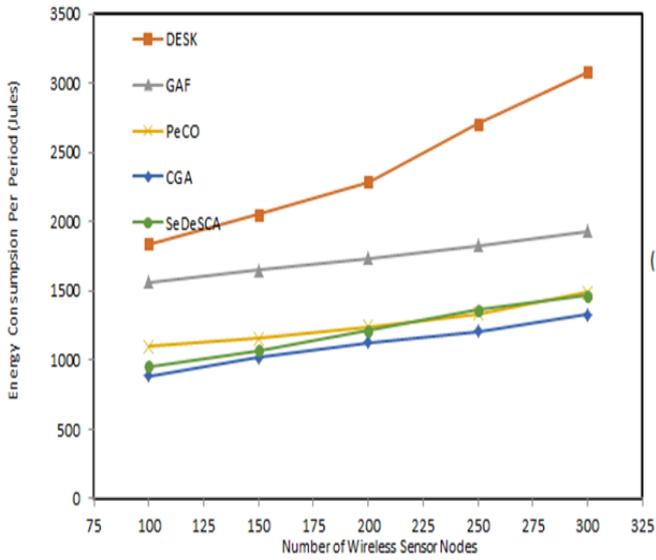
APPENDICES

Appendix A

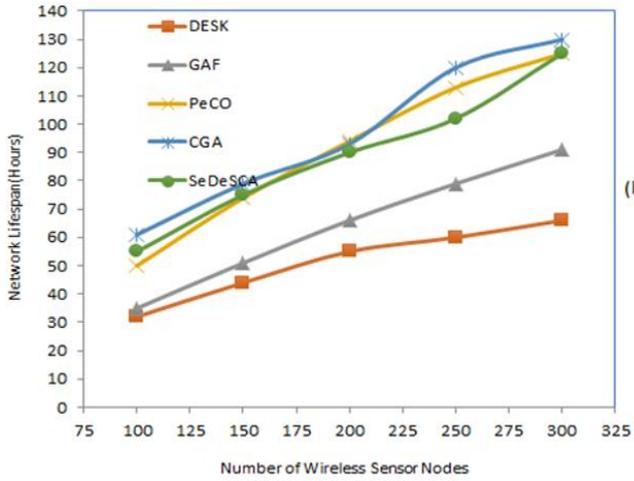
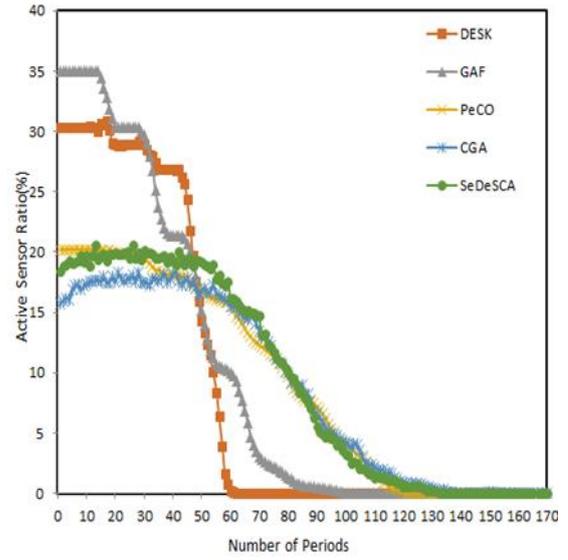




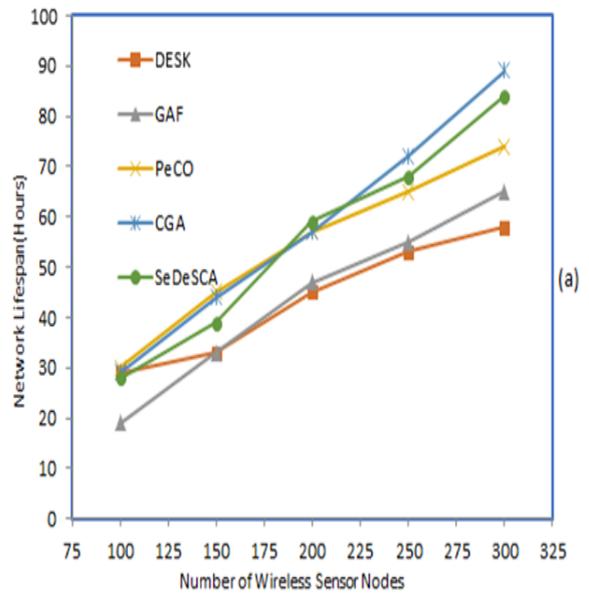
(a)



(b)



(b)



(a)

الخلاصة

تتكون شبكة الاستشعار اللاسلكية (WSN) من عدد كبير من أجهزة الاستشعار الصغيرة وغير المكلفة والمقيدة بالطاقة. تعتبر أجهزة الاستشعار من الاجهزة التي لها القدرة على استشعار ومعالجة ونقل البيانات التي تكون حول محيطها عبر اتصال لاسلكي إلى مركز مراقبة يعرف بـ Gate (بوابة) ليجري عليها عمليات معالجة اضافية. تكون أجهزة الاستشعار في مجال الذي تهتم فيه مقيدة بالطاقة ولا يمكن استبدال بطارياتها خاصة في البيئة البعيدة أو المعادية. لذلك ، من المهم جداً تقليل استهلاك الطاقة لأجهزة الاستشعار من أجل إطالة عمر الشبكة لأطول فترة ممكنة. لهذا السبب ، تمثل جدولة أنشطة عقد الاستشعار تقنية موفرة للطاقة لتحسين عمر الشبكة.

في هذه الاطروحة ،تم تطوير ثلاثة طرق لكفاءة الطاقة. أولاً ، بروتوكول DBSCAN وهو بروتوكول موزع يتم استخدامه لتوفير طاقة لأجهزة الاستشعار في شبكات إنترنت الأشياء. حيث يتم تنفيذ هذا البروتوكول على كل جهاز استشعار لإنترنت الأشياء وتقوم الأجهزة بتطبيق خوارزمية DBSCAN لتقسيم الشبكة إلى عناقيد بطريقة موزعة. وفي هذا الصدد تم اقتراح استراتيجية فعالة وتكون بشكل دوري وموزعة لانتخاب مسؤول للعنقود.

بعد ذلك ، تم اقتراح خوارزمية الوقواق المستندة إلى جدولة جهاز الاستشعار (SeDeSCA) لتحسين عمر شبكات WSN التي تم تقسيمها الى عناقيد. تتكون تقنية SeDeSCA من مرحلتين: العنقدة والجدولة. في المرحلة الاولى يتم عنقدة WSN الى عناقيد باستخدام خوارزمية DBSCAN. يختار كل عنقود مسؤول خاص بها ويكون ذلك بشكل دوري. يقوم رئيس كل عنقود منتخب بتنفيذ خوارزمية الوقواق (CA) لحل النموذج الرياضي الخاص الخاص بتحديد الجدولة المناسبة والتي من خلاله تأخذ عقد المستشعرات مهمة التحسس خلال الفترة الحالية.

أخيراً ، تم اقتراح خوارزمية جدولة Hybrid Metaheuristic(CA+GA) لتقليل كمية استهلاك الطاقة وضمان تغطية مناسبة للمنطقة التي يتم مراقبتها مع زيادة عمر الشبكة الشبكة WSN. من خلال دمج الخوارزمية الجينية في خوارزمية الوقواق العادية تم تحسين التوازن بين قدرات الاستكشاف والاستغلال بشكل إضافي ، ولوحظ وجود مساحة بحث أكبر أثناء

أداء الخوارزميات. يعتمد هذا البروتوكول على نفس إطار عمل خوارزمية الجدولة الأولى ونفس نموذج التحسين.

لشرح فائدة الأساليب المقترحة عند مقارنتها بالتقنيات الأخرى الموجودة تم إجراء تجارب محاكاة مكثفة باستخدام محاكي مخصص يعتمد على ++C. تظهر نتائج هذه التجارب أن بروتوكول ال DBSCAN يوفر الطاقة أكثر من الطرق الأخرى. كما أن SeDeSCA يوفر نسبة تغطية أكثر ملاءمة من التقنيات البديلة. وكذلك كما ويتفوق على الطرق الأخرى من حيث استهلاك الطاقة وإطالة عمر الشبكة. تميل CA + GA إلى تحقيق أداء تغطية أكثر ملاءمة من البروتوكولات البديلة ، مع الحفاظ على معدل تغطية يزيد عن 80% للعديد من الدورات. يمكن تحسين عمر الشبكة لـ Lifetime95 عن طريق بروتوكول CA + GA حتى 25.1% ، 24.7% و 6.9% و 4.5 مقارنة ببروتوكولات DESK و GAF و PeCO و SeDeSCA على التوالي. أما بالنسبة إلى Lifetime50 ، فإن بروتوكول CA + GA يعزز عمر الشبكة حتى 46.8% و 33.3% و 5.6 و 7.4 مقارنة ببروتوكولات DESK و GAF و PeCO و SEDSCA على التوالي.



وزارة التعليم العالي و البحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات

الية جدولة حافظة للطاقة لشبكات الاستشعار اللاسلكية

أطروحة مقدمة الى مجلس كلية تكنولوجيا المعلومات في جامعة بابل وهي
جزء من متطلبات الحصول على درجة الدكتوراه فلسفة في علوم
الحاسبات/برمجيات

مقدمة من قبل

مازن كاظم حميد علي

بإشراف

الاستاذ الدكتور

علي كاظم ادريس