Sudan University of Science and Technology

Faculty of Computer Science and Information Technology

Extending Lifetime and Optimizing Energy of Wireless Sensor Network using Hybrid Clustering Algorithms

إطالة عمر شبكة الحساس اللاسلكي و تحسين طاقتها باستخدام خوارزميات تجميع مختلطة

Submitted for the degree of Doctor of Philosophy

in Computer Science

By

Basma Fathi Hussain Solaiman

Supervisor

Alaa F. Sheta, Professor

July, 2015

Abstract

The Wireless Sensor Network (WSN) is becoming increasingly popular as it enables sensor nodes to measure the surrounding environment, communicate and process measured data. WSN has been directed from military applications to various civil applications, especially in hostile areas. Medical, industrial and smart energy applications is still in need for extensive research due to various challenges encountered. Energy consumption is one of the vital challenges that face WSNs research. The problem is: nodes are supplied with batteries that cannot be recharged or replaced in the field of operation. Management of WSN's energy helps increasing the network lifetime. Clustering is an efficient technique that is used for enhancing the energy consumed by WSN. However, the dynamic nature of the network made it inappropriate for applying traditional clustering techniques.

In this thesis, we investigate the issue of applying Particle swarm optimization (PSO) as a powerful technique which can handle the WSN clustering problem providing a solution that can prolong the network lifetime. This thesis explores the advantages of hybrid clustering approaches to provide efficient and effective clustering technique that co-op with the dynamic nature of the network. Two problems have to be solved to cluster WSN. They are: the number of clusters to be produced and the cluster head (CH) for each cluster. Three approaches are presented. The first approach is a Hybrid K-means PSO clustering approach, 'KPSO', that clusters the network into predefined number of clusters. K-means searches for the best number of clusters, and then groups the network into the selected clusters. PSO selects the best CH for each cluster. KPSO reduced the complexity on the way we are handling the problem and improved the network lifetime by an order of magnitude compared to the well-known Low Energy Adaptive Clustering Hierarchy protocol, LEACH. In the second approach, PSO task was to solve the whole clustering sub-problems. The second PSO Variable Clustering approach, PSO-VC, provides the optimum number of clusters as well as the best cluster layout. PSO-VC enhanced the network lifetime compared to LEACH and KPSO. The last approach, named KPSO-PSO, is an evolution of the first one. KPSO-PSO added a new PSO phase that perform clustering based on controlling the antenna power and thereby prolong the network lifetime. Experimental results showed that this approach can provide improved WSN lifetime over LEACH, KPSO and PSO-VC. Moreover to investigate the effectiveness of using PSO in the proposed clustering approaches, the same approaches

are re-implemented using Genetic Algorithm (GA) instead of PSO. PSO proved to converge to better fitness values and resulted in an enhanced WSN lifetime over GA. Finally, We were able to develop a WSN Clustering Aided Toolbox (WSN-CAT) which can significantly help in simulating various WSN environments and helps exploring many tuning parameters for the proposed approaches.

الملخص

تزداد شبكة الحساس اللاسلكي شيوعا، فالحساسات تقوم بقياس بيانات البيئة المحيطة وتتصل فيما بينها وتقوم بعمليات اجرائية علي البيانات المقاسة. ظهرت شبكة الحساس اللاسلكي في التطبيقات العسكرية ثم استخدمت في التطبيقات المدنية خاصة في الاماكن النائية. تحتاج التطبيقات الطبية والصناعية والطاقة الذكية الي بحوث مكثفة نظرا لوجود تحديات عديدة تواجههم. تعتبر قضية استهلاك الطاقه واحدة من أهم التحديات التي تواجه بحوث الشبكه حيث أن حساسات الشبكة اللاسلكية مزودة ببطاريات غير قابلة للاستبدال أو إعادة الشحن، لذا فان إدارة استهلاك طاقة الشبكه تساعد علي تحسين عمرها. تقلل عملية التجميع الطاقة المستهلكة في الشبكة بكفاءة، لكن الطبيعة الديناميكية للشبكة تجعل استخدام التقنيات التقليدية في عملية التجميع غير مناسب. نحن تستكشف في هذه الاطروحة استخدام تقنية جزيء السرب الأمثل حيث انها تقنية قوية يمكنها القيام بعملية تجميع شبكة الحساسات اللاسلكية من أجل إطالة عمرها. عملية تجميع الشبكة يجب ان تجد الحل لمشكلتين هما: عدد المجموعات و رأس كل مجموعة. تستكشف هذه الاطروحة استخدام تقنيات مختلطة في عملية التجميع تؤدي الي إصدار نتيجة فعالة وذات كفاءة ومناسبة للطبيعة الديناميكية للشبكة. تم تقديم ثلاث نهج. النهج الأول طور النهج المختلط بين تقنية ك طرق و تقنية جزيء السرب الأمثل. تقوم تقنية ك طرق بتحديد العدد الأمثل للمجموعات و تقسم الشبكة الي المجموعات المطلوبة ثم تقوم تقنية جزيء السرب الأمثل بتحديد الرأس الأمثل لكل مجموعة. قلل هذا النهج من تعقيد استخدام تقنية واحدة كما أطال عمر الشبكة مقارنة بالنهج التقليدي الشهير بروتوكول ليتش. النهج الثاني المطور أخرج العدد الأمثل للمجموعات و حساسات كل مجموعة وكذلك رأس كل مجموعة باستخدام تقنية جزيء السرب الأمثل. أثبت النهج الثاني تفوقا علي النهج الأول من حيث عمر الشبكة. أما النهج الأخير فهو تطوير للنهج الأول عن طريق إضافة مرحلة جديدة للتحكم في طاقة إريـال الحساس اللاسلكي وبالتالي اطالة عمر الشبكة. نتائج النهج الثالث اثبتت إطالة عمر الشبكة مقارنة بالنهج الأول والثاني. بالاضافه الي ما سبق تم اختبار كفاءة استخدام تقنية جزيء السرب الأمثل حيث تم اعادة تطوير استخدام النهج الثلاثة باستخدام تقنية خوارزمية الجينات. تقنية جزيء السرب الأمثل اثبتت أداء أفضل و عمر أطول للشبكة بالمقارنة بتقنية خوارزمية الجينات. وأخيرا تم تطوير برنامج لمحاكاة الشبكة لتقييم نهج التجميع المطورة.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analog-to-Digital Converter
BS	Base Station
CH	Cluster Head
DE	Dissipated Energy
EEUC	Energy Efficient Unequal Clustering
GA	Genetic Algorithm
GUI	Graphic User Interface
HEED	Hybrid Energy-Efficient Distributed clustering
KGA	hybrid K-Means GA clustering
KPSO	hybrid K-Means PSO clustering
LEACH	Low Energy Adaptive Clustering Hierarchy
MCH	Master Cluster Head
MEMS	Micro Electro-Mechanical Systems
MOGA	Multi-Objective Genetic Algorithms
MOPSO	Multi-Objective Particle Swarm Optimization
NN	Neural Network
PSO	Particle Swarm Optimization
PSO-VC	PSO Variable Clustering
QoS	Quality of Service
RF	Radio Frequency
SC	Soft Computing
SHM	Structured Health Monitoring
TVAC	Time Varying Acceleration Coefficient
VCH	Vice Cluster Head
WSN	Wireless Sensor Network
WSN-CAT	Wireless Sensor Network Computed Aided Toolbox

List of Publications

1. Computational Intelligence for Wireless Sensor Networks: Applications and Clustering Algorithms. *International Journal of Computer Applications. Volume 73 No. 15, July 2013*.

2. Evolving a Hybrid K-Means Clustering Algorithm for Wireless Sensor Network Using PSO and GAs. *International Journal of Computer Science Issues, 12, No. 1 (2015): 23-32*.

3. Evolving Clustering Algorithms for Wireless Sensor Networks with Various Radiation Patterns to Reduce Energy Consumption. In *Science and Information Conference (SAI), 2015*, pp. 1037-1045.

4. Energy optimization in wireless sensor networks using a hybrid K-means PSO clustering algorithm. *In: Turkish Journal of Electrical Engineering & Computer Sciences, 2015, Accepted for Publications*.

5. Evolving a Clustering Algorithm for Wireless Sensor Network Using Particle Swarm Optimization. *International Journal of Swarm Intelligence, Inderscience Publisher. Accepted for Publications*

Chapter One

Introduction

Wireless networking is an emerging technology that allows users to access information and services electronically, regardless of their geographic position. The use of wireless communication between devices has become increasingly popular due to recent performance advancements in computer and wireless technologies which led to lower prices and higher data rates. Moreover, the evolution and advance in micro electro-mechanical systems (MEMS) has led to development of reliable, low cost, small size micro sensors [2]. Nowadays, hundreds or thousands of these heterogeneous micro sensors, called nodes, are deployed over a geographical area of interest, and communicate together forming a wireless sensor network, as shown in Figure 1.1. Figure 1.2 shows that the global industrial wireless sensor network market size is estimated to grow from $401 Million in 2013 to $945 Million by 2020, [3].

The benefits of using WSNs are:

- **Ease of Deployment:** Nodes are deployed without cables or wires. Thus, the labor effort for deploying WSN is minimized.
- **Reliability:** Nodes can self-organize to perform the network. Also, broken links can be quickly repaired. They have the ability to dynamically adapt to changing environment.
- **Scalability:** Nodes can join or leave the group without affecting the entire network.

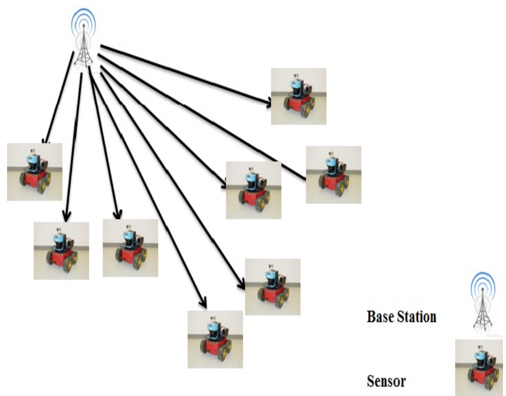


Figure 1.1: Wireless Sensor Network

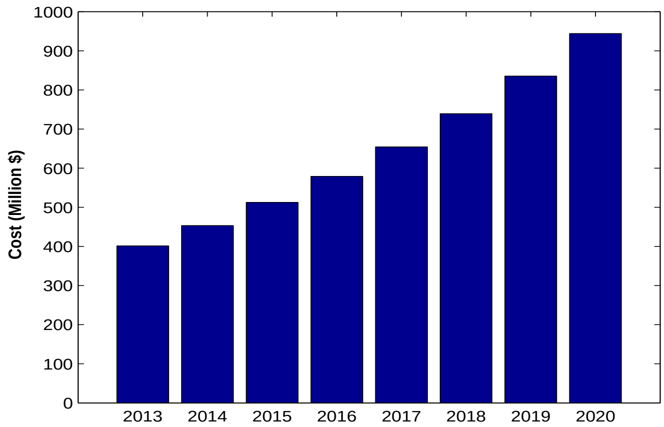


Figure 1.2: Estimated wireless sensor network market size

- **Reduced Cost:** WSN results in saving wire costs, saving installation time, and less labors used.
- **Efficient Use:** Efficiently used for hostiles areas where wired networks are impossible to use, as mountains, forests, oceans ..etc.

WSNs are deployed in land, underground and underwater [4]. It is designed to work for months and years according to the application. The nodes deployment need not be centralized, or with fixed infrastructure. The wireless sensor nodes in the network sense external data from the surrounding environment, process the sensed data locally, and then send the data to a base station for further processing through wireless communication. The nodes may also be stationary or moving. There are two variations of wireless sensor deployment: structured and unstructured. While in Structured WSN the nodes are deployed in a preplanned manner, the deployment of nodes in unstructured WSN is random (e.g. dropped by an airplane).

WSN was first inspired by the US military for enemy surveillance and object tracking. Recently, it is applied to diverse civil applications as: environment monitoring and environmental disaster detection. WSN deployment and operation are application-specific. The massive evolution of sensor nodes encouraged applying it in harsh environments that oppose the presence of humans. However, many applications are not ready for real world. Nowadays, researchers are adopting three important areas: medical health care, smart energy, and industrial automation. For medical health care, WSN is objected to monitor the patient's physical condition and alert the doctor in case of danger. Smart energy WSN is designed to monitor and regulate the energy consumption. Industrial WSN is used for process monitoring rather than process automation because any failure in its functionality could lead to severe circumstances. These new emerging applications motivated for applying extensive research to make it applicable for real world.

1.1 Design Goals of WSN

WSN consists of hundreds or thousands of nodes that are probably deployed in remote areas which makes maintenance and organization of the network not feasible. In order to design a reliable WSN, it is important to understand their challenging parameters facing the design. Some of these challenges are:

- **Deployment:** Deployment is concerned with the design of WSN in many perspectives. It can specify the minimum number of sensor nodes needed to construct the network. It can also specify the sensor placement criteria that satisfy the predetermined lifetime and coverage requirement. An alternative design specifies how large an area can this sensor network cover for a given the number of sensor nodes, and a desired life time of the sensor network. Deployment answers a question like: what is the maximum network lifetime and what is the deployment scheme for a given the number of nodes and coverage area.

- **Security:** Security is vital to the acceptance and use of sensor networks for many applications. Resource constrained wireless sensor nodes cause the network to be highly vulnerable to different kinds of attacks; passive and active. Some of the major attacks includes: denial of service, attacks on data transferred, black-hole attacks and sybil attacks [5], [6]. Therefore, securing the network should consider availability, integrity, authenticity and confidentiality [7].

- **Coverage:** One of the challenges is how to cover the monitoring region perfectly [8]. It can be considered as a measurement of quality of service (QoS) that can be provided by a particular network. Coverage specifies how efficiently an event can be detected within a given time frame. Moreover, Coverage includes modifying the deployment scheme due to weakness in some sensor fields.

- **Quality of Service:** Successful QoS solutions have been developed for traditional networks. However, WSN architectures and features differ from traditional networks.

This makes it unfeasible to apply traditional QoS solutions to WSNs. QoS requirements for WSN is mainly application dependent. Common QoS requirements for WSNs are data accuracy, data aggregation, coverage, fault tolerance and network lifetime. The main opposing factor in QoS for WSN is the sensor's limited resources and dynamic topology of the network. This imposes a great challenge on implementing efficient QoS solutions while preserving the energy of the network. A lot of research work is still needed in this issue.

- **Energy and Lifetime:** The expected lifetime is a critical factor in the network deployment. It is requested that the WSN operates for months or years. The primary limiting factor for the lifetime of a sensor network is the energy supply. Each sensor is equipped with a limited battery. Since the nodes are deployed in hostile areas, it may be impossible to replace or recharge the nodes' batteries. Any network operation consumes part of the sensor battery capacity. The network is known to have a certain lifetime. Then designing a network should consider maximizing the network lifetime and minimizing the energy consumption.

1.2 Energy Consumption

Energy consumption is considered the main challenge for WSN operation. The nodes are equipped with limited batteries. They are deployed in hostile areas; making recharging or replacing the battery unfeasible [9]. Recently, some nodes are equipped with renewable energy, or energy harvesting module [10, 11]. However, their expensive cost almost ceased their deployment. The death of the node was preferred economically. Energy consumption is managed at different levels [12]. For example, in technology level research is made to produce low duty cycles, minimize delays, handle data redundancy and implement short range transmission. In network layer, energy efficient routing protocols are developed to prolong network lifetime.

1.3 WSN Lifetime

WSNs are very sensitive to energy consumption and its performance affects the network lifetime. The WSN lifetime is defined as the time when the first node dies [13]. Others consider this definition optimistic because the death of one node does not oppose the functionality of the remaining nodes. Other definitions are introduced in [14]. The most common WSN lifetime definition is when a certain percentage of the nodes die. However, there is no agreement on the percentage that should be used in the definition. The network lifetime is important since it is an indication of the network performance degradation. The death of one node is soon succeeded by the death of others, and node isolation occurs. Therefore, a critical aspect to concern is how to reduce the energy consumption of nodes in WSN to prolong the network lifetime.

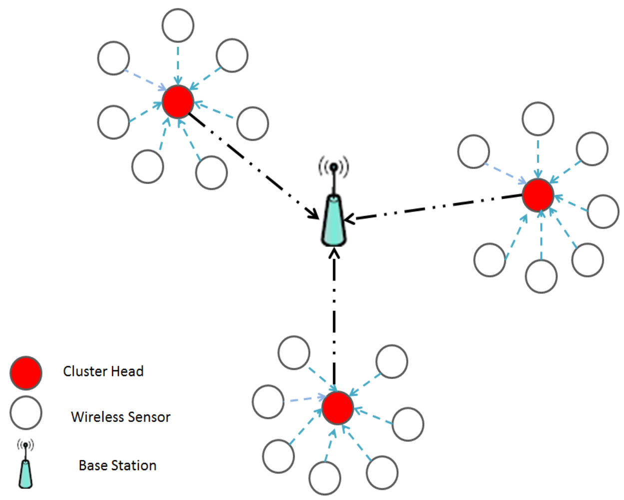


Figure 1.3: A Clustered WSN

1.4 Clustering in WSN

Clustering is found to be an effective technique to solve energy consumption problem for WSN [15]. The nodes are divided into disjoint groups called 'clusters'. The nodes within each cluster can intercommunicate, or communicate with only one node in the group, named Cluster Head (CH). The CH is responsible for gathering data from all nodes in the group, then sending the data to the base station, directly or indirectly, after processing it [16, 17]. Figure 1.3 shows a clustered WSN. Clustering has many advantages such as grouping sensors and saving energy losses. These advantages can be summarized as follows [18]:

- Reduce the number of nodes responsible of sending data.
- Reduce communication overhead.

- Communicate collected data to the base station.
- Increase energy saving.
- Allow scalability by increasing the number of nodes.
- Provide a better use of network resources.

1.5 Thesis Contribution

WSN is highly dynamic and sensitive to energy consumption. The dynamic nature of the network makes the clustering problem a complex task. Although many clustering protocols and algorithms have been implemented, developing a low computational and high performance clustering algorithm is still a challenge. The objective of this thesis is to prolong the WSN lifetime using Clustering to manage energy consumption. Clustering is an NP hard optimization problem that can't be solved effectively by traditional methods [16]. In case of non-hybrid approach, finding the optimal number of clusters and the optimal cluster head is a complex task. For a WSN with n nodes, the search space will consist of $2^n - 1$ solutions to search in.

Soft Computing paradigms are suitable to adapt for WSN dynamic nature. This can be achieved by adopting hybrid clustering approach for WSN. The hybrid approach integrates two or more technique to solve the problem efficiently. This can be done using a two-phase approach. The first phase performs clustering the nodes into groups. Then the second phase selects the optimal cluster head for each cluster. With hybrid approach, the problem will be easily managed and less computation will be applied. By knowing the number of clusters required, the first phase will simply divide the search space into, say, k clusters. If we assume uniform cluster size, then each cluster will have $\frac{n}{k}$ nodes. Now the search space is divided into $\frac{n}{k}$ solutions for each cluster, which is considerably lower than the non-hybrid scheme.

In this thesis three solutions are proposed for clustering problem for WSN to prolong the network lifetime. They are: 'KPSO' hybrid K-means and PSO approach, 'PSO-VC' PSO Variable Clustering approach and 'KPSO-PSO' hybrid K-means and PSO approach. The results are analyzed and compared with the famous LEACH protocol. Moreover, performance of PSO is explored by comparing results of the same solutions but with using GA instead.

1.5.1 Solution 1: 'KPSO' Hybrid K-Means PSO Clustering Approach

This proposed solution consists of two-phases. In the first phase we use K-means clustering to search for the best number of clusters and then cluster the network. The next phase selects the best CH for each cluster using PSO having a new proposed fitness. Simulation results are recorded and analyzed.

1.5.2 Solution 2: 'PSO-VC' PSO Variable Clustering Approach

In this solution, PSO is developed to perform complete WSN clustering. It outputs the optimal number of clusters, optimal CHs and members of each cluster. A new fitness is proposed. Simulation results showed improvement over the KPSO solution.

1.5.3 Solution 3: 'KPSO-PSO' Hybrid K-Means PSO Antenna Pattern Based Clustering Approach

This proposed solution is an evolution to the hybrid KPSO approach. The first phase applies our first proposed solution to produce the best CHs and proposed clusters. A new extended phase developed a PSO model that obtains the best antenna pattern dimension and the optimal cluster members for each cluster based on the wireless antenna pattern. The results showed improvement in the WSN lifetime.

1.5.4 WSN-CAT Toolbox

A MATLAB Wireless Sensor Network Clustering Aided Toolbox (WSN-CAT) is designed and implemented to develop and simulate the WSN clustering solutions. The toolbox is provided with a simple GUI that is user friendly.

1.6 Thesis Structure

This thesis consists of nine chapters. Chapter 2 gives a detailed discussion about the wireless sensor network; its technology, applications, types, and previous WSN clustering work is described and commented. Chapter 3 explains the SC techniques used in the proposed solutions. Chapter 4 explains in details the proposed solutions. Chapters 5,6 and 7 shows and discusses the simulation results of the three proposed solutions, respectively. Chapter 8 describes the developed toolbox. Finally, we state our conclusion and recommended future work in Chapter 9.

Chapter Two

Wireless Sensor Networks

WSN is a group of wireless nodes deployed in a geographic area. The deployment and operation of WSN are application dependent. The operation of WSN is very sensitive to the energy consumed by the nodes. This chapter states some WSN applications and WSN types. The components of the wireless node are described. A mathematical model for energy consumption is derived. Finally, a literature review on WSN is discussed.

2.1 Applications of WSN

In [19], the applications were grouped according to their objectives. With respect to the type of WSN operation, the applications were classified into two categories: event detection and periodic sensing [20]. In event detection applications, the network is designed to warn about the occurrence of a specific event. The area of interest is monitored by sensors that communicate with each other, process data and reach important conclusions. This category includes: fire-forest detection, earthquake detection. In the second category, data is sensed periodically and sent to the base station for further analysis. Education monitoring, inventory, industry, traffic are examples of periodic sensing applications. This section describes some WSN applications. Figure 2.1 shows example of WSN applications.

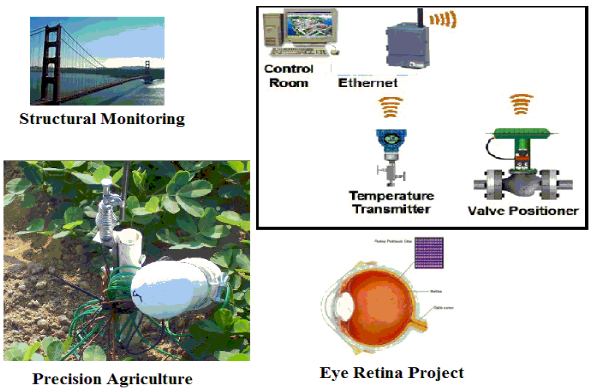


Figure 2.1: Wireless Sensor Network Applications

Table 2.1: Wireless Sensor Network Applications

Deployed Applications	Potential Applications
Military	Medical Healthcare
Environmental	Smart Energy
Agricultural	Industrial Automation
Home and Office Buildings	
Structural Health monitoring	

In this thesis, the applications are categorized according to its current state. Table 2.1 lists the categories and the applications. Two main categories are proposed:

- **Deployed:** the deployed category contains the actual applications that are implemented in real-world.
- **Potential:** this category contains the applications that are not yet mature, and are viewed as vital and promising.

2.1.1 Military Applications

It was the first motivation for development of WSN. In 1980, the Defense Advanced Research Projects Agency (DARPA) adopted the Sensor Information Technology (SENSIT) and National Science Foundation (NSF) Programs on WSN for more tracking capabilities [21]. Other applications included battle field surveillance, and intrusion detection. The nodes were programmed to take measurements, communicate with each other, and send notification in case of object movement detection. More recent military projects aimed to detect nuclear, chemical and biological toxins as well as calculation of their concentration levels [22].

2.1.2 Environmental Applications

As nodes are deployed in a natural hostile area, long term environmental data is gathered either for future research, monitoring, or disaster detection (as fire, flood or earthquake forecast, etc.) [23]. In 1970s, the earliest real world project founded was the Automated Local Evaluation in Real Time, (ALERT). It was designed to detect the existence of flood using sensors that take measurements as: temperature, humidity, rain, and water level. The data was transmitted to a station using Laser technology.

In 2002, Intel Research Laboratory and University of California founded the Great Duck Island project, North Atlanta [22], to monitor the behavior of Petrel bird. The project started with 32 nodes that collected millions of data, and till now there are about hundred well equipped nodes, some of them have cameras for video monitoring. The project was not only beneficiary to monitoring, but it also reported the network operation and functionality problems that needed more research.

2.1.3 Home and Office Buildings

Started in late 1980s, smart buildings are those equipped with systems that do some intelligent actions, as door opening. Wireless nodes are used to monitor the employees, students, etc. In 1990s, research has been adopted to use smart buildings to disabled people. Smart Kindergarten deploys wireless nodes for childhood education and monitoring [24]. WSN is recently incorporated in smart building for more quality of life.

2.1.4 Agriculture Applications

Precision agriculture is applying the right amount of input (water, fertilizer, etc.) at the right location and at the right time to enhance production and improve quality, while protecting the environment [25]. It is accomplished with WSN that monitors parameters as: soil moisture and air temperature, and calculates the amount of water and fertilizers needed. Also, irrigation management is another application adopted by WSNs that help farmers to prevent damages to their crops and increasing crop production. WSN is also used to control the green house temperature and humidity levels starting from messaging to using controller [26].

2.1.5 Structure Health Monitoring

Structured Health Monitoring (SHM) systems have been proposed in 1990s to assess civil buildings as dams, bridges, hydroelectric power plants and pipes. Its main objective is to extend the building lifetime by detecting and localizing damages. However, the deployment of wired SHM is rare due to high installation cost. Recently, WSN is expected to be the next generation for SHM that will avoid the high cost wired-installation [27].

2.1.6 Health Applications

Health care is considered a very potential application whose research is dominant [28]. Most medical applications are tele-monitoring physiological data, tracking patient locations and patient drug management [29]. WSN will allow the patient to be under constant supervision without hospital admission. The Code Blue project is an example of patient monitoring ongoing projects [30]. Gluco Watch G2 project focuses on diabetic patients. Two promising applications are being investigated: glucose level and artificial retina [31, 32]. The diabetic patient can be implanted with glucose meter that monitors the sugar level and alerts the patient in case of serious condition detection. The second project under investigation considers implanting a chip of micro-sensors in the human eye to enhance vision. Reliability, communication, and safety are challenging issues. This field is a great motivation to enhance the quality of life and decrease medical cost.

2.1.7 Smart Energy

Energy production and consumption is an extremely critical problem worldwide. Research on producing smart building has gained great interest. Energy improvement solution incorporated the use of wireless nodes for improving home utilities, such as lighting, water and gas [33]. Studies are in-process to design the network to monitor the energy consumption parameters, analyze them and finally regulate consumption. Recent studies are working on controlling the devices automatically. WSN is expected to be the next generation for smart home and buildings by improving energy distribution and consumption. In the United States, it is expected that WSN will result in saving about 50 billion dollars yearly and reduce 35 million metric tons of carbon emissions [34].

2.1.8 Industrial Automation

WSN is promising in replacing wired industrial control process. The process of cooling a reactor is an example of a process control that can use WSN technology [35]. However, any miscellaneous WSN functionality could lead into severe catastrophic circumstances. WSN industrial process automation is not yet mature to be applied due to many challenges that still need extensive research work. Challenges include: energy-limitation, QoS, and security [36]. Recent commercial industrial WSN products are only used for process monitoring.

Table 2.2: Wireless Sensor Network Types

	Terrestrial WSN	Underground WSN	Underwater WSN	Multimedia WSN	Mobile WSN
Cost	inexpensive	expensive	expensive	inexpensive	expensive
Deployment	structured/ unstructured	unstructured	structured/ unstructured	structured	initial spreading
Node density	high (100-1000s)	low	low	application specific	application specific
Challenges	Energy	Energy, signal loss, attenuation	Energy, bandwidth, delay, signal fading	Energy high bandwidth, high data rate, QoS	Energy, deployment, localization navigation

2.2 Types of WSN

There are five types of WSNs: Terrestrial, Underground, Underwater, Multimedia, and Mobile WSNs [4]. Each type has its architecture, characteristics and challenges. Table 2.2 summarizes the major difference between the network types.

- **Terrestrial WSN:** The Terrestrial WSN deploys 100-1000s of inexpensive nodes that must reliably communicate with the base station. The nodes can be randomly deployed , or deployed in structured manner. Its main challenge is to conserve energy because the nodes have limited battery power. Energy can be saved by reducing the communication consumption.

- **Underground WSN:** Underground WSN nodes are more expensive than terrestrial WSN. The nodes are highly equipped and deployed in mines, caves or buried in ground. Sink nodes over the ground collect the measured data from the sensor nodes and send them to the base station. The nodes are carefully deployed in a structured manner. This network faces challenges of signal loss and attenuation. Also, the batteries cannot be recharged or replaced. This makes energy consumption a great challenge.

- **Underwater WSN:** Underwater WSN, like Underground WSN, consists of expensive nodes, and vehicle nodes that collect data from the nodes and connect to the base station. Due to its high cost, underwater applications use deployment of few nodes. Wireless communication is performed by means of acoustic waves. Different challenges faced are: limited bandwidth, delay, signal fading and again energy conservation.

- **Multimedia WSN:** In Multimedia WSN, nodes are equipped with cameras and microphones. Multimedia WSN is less expensive than underground and underwater WSNs. The nodes communicate with each other to process, correlate, and compress data. This network demands high bandwidth, high data rate and QoS. The challenge here is to support for both high bandwidth and low energy consumption, since both are competing.

- **Mobile WSN:** Mobile WSN differs than the above in that the nodes are initially deployed, and then move with the surrounding to gather and explore information. Mobile nodes communicate with others within its range. Mobile nodes are self-organizing; they perform dynamic routing techniques and achieve better coverage than static network. Mobile WSN faces challenges like deployment, localization, self-organizing, navigation and also energy.

2.3 Components of a Wireless Sensor Node

The wireless sensor node is mainly composed of: sensing unit, processing unit, transceiver, and a power supply [19]. Famous manufactured sensor nodes are (Figure 2.2): Smart Dust sensor, and MOTE (abbreviation of remote) [37, 38]. Figure 2.3 shows the block diagram of the wireless sensor node.

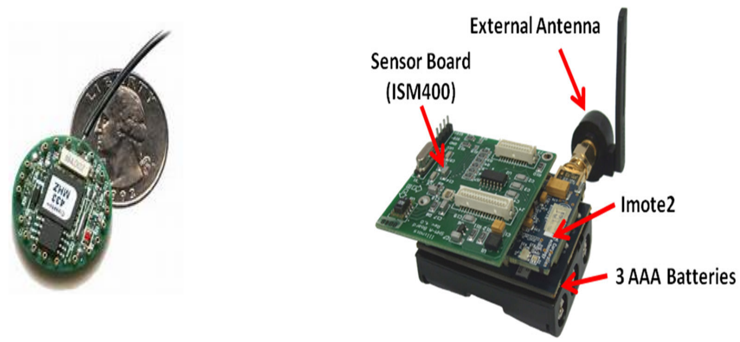


Figure 2.2: (a) Smart Dust Wireless Sensor Node (b) Imote2 Wireless Sensor Node

Each node in a WSN consists of:

- **Sensing Unit:** It contains at least one sensor that measures data from its surrounding. Different sensing units exist according to the application deployed. For example:

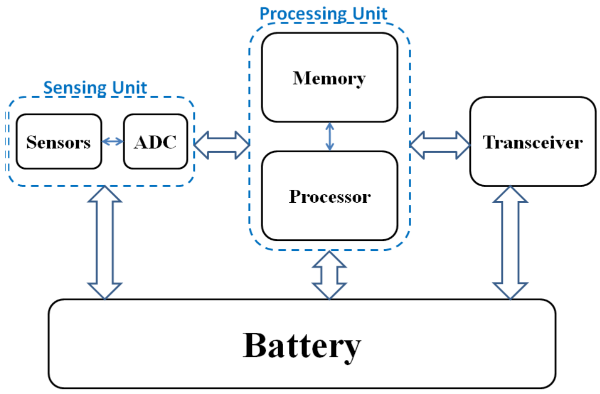


Figure 2.3: Block diagram of a wireless sensor node

environmental monitoring node is equipped with temperature, wind and humidity sensors. The analog signal sensed by the sensors is then digitized by an analog-to-digital converter (ADC) adapted to the processing unit. The attached sensors should also be small in size and consume extremely low energy.

- **Processing Unit:** With the help of embedded memory, the processing unit processes the data measured by the sensing unit as well as data gathered from neighbor nodes. Also, the processor schedules tasks, and controls the functionality of other hardware components. The processor and all hardware components are controlled by an operating system that is specially designed for WSNs. One of the earliest operating systems designed for WSN is TinyOS [39]. WSN operating systems are simpler than general-purpose operating systems to overcome energy and resource constraints.

- **Transeiver:** The data is sent and received by means of a transceiver. The nodes communicate by laser, infrared, or radio waves. WSN communication using radio frequency (RF) is the most commonly used. The operational states of a transceiver are Transmit, Receive, Idle and Sleep. The transceiver is equipped with an antenna that is responsible for transmitting and receiving messages. Antennas types include: omnidirectional, and directional. Every antenna is described by its antenna pattern. In case of omni-directional antennas, the antenna pattern is a circle; the energy is radiated and received equally in all directions. Antenna patterns for directional antennas have various shapes depending on the antenna design. They usually focus energy in a particular direction.

- **Battery:** Batteries are the main source of power supply for sensor nodes. For example, Mica2 Mote runs on 2 AA batteries [39]. The battery has limited lifetime, and this makes energy consumption a key concern during WSN operations. The energy that the node consumes could be useful (sensing, receiving, transmitting, processing) or unuseful (idle listening, overhearing).

2.4 Major Sources of Energy Consumption

Although different challenges face WSN, they all share one main challenge, 'Energy'. Energy consumption is considered the main challenge for WSN operation. The sensor nodes are equipped with limited batteries. They are deployed in hostile areas; making recharging or replacing the battery unfeasible [9].

Every node operation consumes energy. Energy is consumed in sensing, processing and communicating. Sources of energy consumption include:

- **Idle:** It reflects the time during which the node keeps listening to the channel waiting to receive data. The idle process consumes energy which we call passive. The node could be designed to sleep during passive time and wake-up to receive data. Designing node's duty cycle to sleep and wake-up at the right time is still a challenge.

- **Data Aggregation:** Sending data messages from all sensors to the base station overheads the traffic. Aggregating data can reduce communication traffic. This is done by combining data messages into one. Data aggregation requires the node to have sufficient memory, processor capabilities and energy for processing.

- **Communication:** Most of the node's energy is consumed during communication [40]. The consumed energy during communication is affected exponentially by the distance between the communicating nodes; the more communication distance between two nodes the more energy consumed. In order to save energy, communication should be minimized. Moreover, designing a suitable pattern for the antenna help reducing energy waste. It was reported that the energy required for an antenna pattern to reach all hosts is proportional to the area it covered [41].

2.5 Energy Consumption Model

A wireless sensor node consists of: sensing unit, processing unit, transceiver and power supply. The power supply provides energy to all other sensor components. The sensed measurements are converted to a digital signal by means of the analog-to-digital converter (ADC) of the sensing unit. The processing unit aggregates the digitized data into one single message to be sent by the transceiver. Typical operations of the transceiver are: sleep, idle, transmit and receive. The energy consumed by a wireless sensor node can be modeled as:

$$E_{Wsens} = E_{SU} + E_{aggreg} + E_{Trans} \tag{2.1}$$

$$E_{Wsens} = E_{SU} + E_{aggreg} + E_{sleep} + E_{idle} + E_{Tx} + E_{Tr} \tag{2.2}$$

where:

- E_{Wsens} is the total energy consumed by a wireless node,
- E_{SU} is the energy consumed by the sensing unit,
- E_{aggreg} is the energy consumed in aggregating measured data,
- E_{Trans} is the total energy consumed by the transceiver,
- E_{sleep} is the energy consumed by the transceiver during sleep operation,
- E_{idle} is the energy consumed by the transceiver while in the idle state,
- E_{Tx} is the energy consumed by the transceiver to send a data message, and
- E_{Tr} is the energy consumed by the transceiver in receiving a data message

Figure 2.4 shows typical values of current consumed for a Tmote Sky wireless sensor that is based on IEEE 802.15.4 WSN mote [42], [43]. It can be seen that the energy consumed to transmit and receive is the most; energy consumed in idle and sleep states and sensing unit can be neglected. Then the energy consumption model can be approximated to:

$$E_{Wsens} \approx E_{aggreg} + E_{Tx} + E_{Tr} \tag{2.3}$$

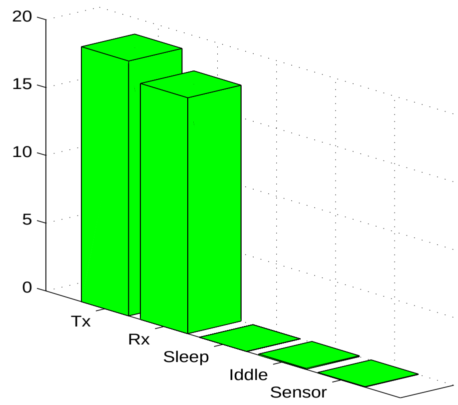


Figure 2.4: Current Draw of Tmote Sky wireless sensor (mA)

2.5.1 Radio Model

We adopted the radio model described in [44]. Figure 2.5 shows the block diagram of the radio model. The energy consumed in aggregating data depends on the message size. Then the energy for aggregating an m-bit message is modeled in Equation 2.4. This RF model uses either the free space model, or the multipath model according to the distance between the transmitter and the receiver. When the distance between the transmitter and the receiver lies within a threshold, d_0, the free space model is used. But when the distance exceeds the threshold, then the multipath model is used. The energy consumed in transmitting an m-bit message is given in Equation 2.5. The energy consumed in receiving an m-bit message is given in Equation 2.6. Typical values of constant coefficients are given in Table 2.3.

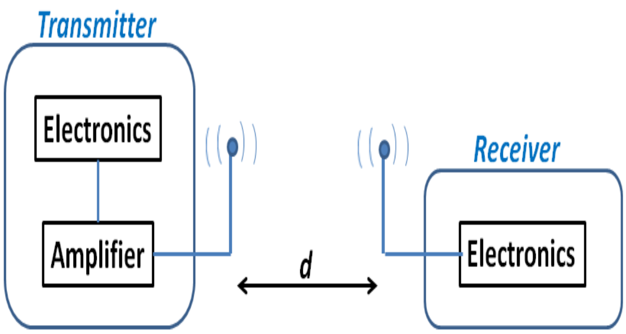


Figure 2.5: Block diagram of Transceiver Radio Model

$$E_{aggreg} = m * E_{DA} \tag{2.4}$$

$$E_{Tx} = \begin{cases} mE_e + \epsilon_{fs} d^2 & \text{for } d < d_0 \\ mE_e + \epsilon_{mp} d^4 & \text{for } d \geq d_0 \end{cases} \tag{2.5}$$

$$E_{Tr} = mE_e \tag{2.6}$$

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{2.7}$$

where:

- m is the number bits in a data message,
- E_{DA} is the energy consumed in aggregating one bit,
- E_e is the energy consumed in the electronic circuit of the transceiver to transmit or receive the signal,
- ϵ_{fs} is the energy consumed by the amplifier to transmit using the free space model,
- ϵ_{mp} is the energy consumed by the amplifier to transmit using the multipath model,
- d is the distance between the transmitter and receiver.

Table 2.3: Radio model parameter values

Parameter	Value
E_{DA}	$5nJ/bit$
E_e	$50nJ/bit$
ϵ_{fs}	$10pJ/m^2$
ϵ_{mp}	$0.0013pJ/m^4$
d_0	87 m

2.6 Traditional Clustering Protocols

Several traditional clustering protocols were implemented. In this section, three famous clustering techniques are mentioned: Low Energy Adaptive Clustering Hierarchy (LEACH), Hybrid Energy-Efficient Distributed clustering (HEED), and Energy Efficient Unequal Clustering (EEUC).

2.6.1 LEACH Protocol

LEACH Protocol is an early proposed single hop clustering protocol in WSN [45]. It is now one of the most famous conventional clustering protocols that is now used as a benchmark for testing clustering algorithms. LEACH implements clustering by the rotation of CHs during transmission rounds [46]. LEACH repeats a two-phase round: setup phase and steady-state phase. In the setup phase, each node elects itself to be a CH with a probability $P_i(t)$ as follows:

$$P_i(t) = \begin{cases} \frac{k}{N-k*(r.mod\frac{N}{k})} & i \in G_r \\ 0 & otherwise \end{cases} \tag{2.8}$$

- r: round number
- k: expected number of clusters
- N: number of nodes
- G_r: nodes that haven't been CHs in the last $r.mod\frac{N}{k}$ rounds

When the node's probability is less than LEACH's random number, the node becomes a CH. The CH then advertises itself to the network, and non CH nodes join their nearest CH. In the steady-state phase, every CH gathers the data from its member nodes and sends the aggregated data message to the base station. The election probability prevents the node from being a CH once more unless all the nodes have been chosen as CHs. The process of election repeats again when all the nodes performed as CHs. Figure 2.6 shows the operation of LEACH protocol.

LEACH claims to balance energy consumption of the sensor nodes. However, the rotation of CHs can cause more energy loss [47]. Moreover, clustering does not ensure even best distribution of clusters [48]. It also assumes that the CH consumes the same energy as the member node. It is expensive and not applicable to be deployed in large geographic region. Another problem with LEACH is that it is highly stochastic; its response is not robust because the algorithm mainly depends on randomly generated numbers.

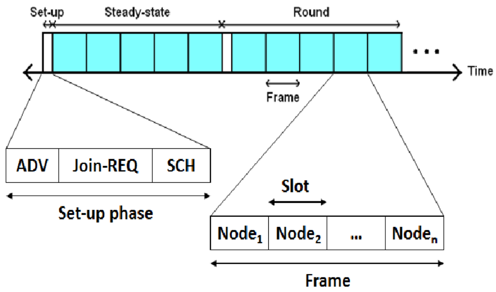


Figure 2.6: LEACH phases

The authors in [49] modified LEACH's threshold equation $P_i(t)$ to include the energy of the nodes. The proposed algorithm, named LEACH-E, runs in three phases. The first phase selects the CHs, then the second phase forms the clusters, and the last phase is responsible for collecting data and sending it to the sink. The problem with LEACH-E, as the authors state, is its non-uniformity in CH distribution and ignoring nodes location in CH selection.

Similarly, the authors in [50] modified the threshold equation $P_i(t)$ to include two factors. One factor aims to choose the CH among dense node distribution. The other factor focuses on the distance between nodes and CHs. However, the improvement in the results shown is not significant.

The authors in [51] proposed a new version of LEACH, named LEACH-G. LEACH-G is based on a formula derived for the optimal number of clusters. The base station selects candidate CHs based on ”maximum return time” of the received messages. Then each CH selects its members using ”minimum return time”. After the clusters are formed, final CHs are chosen for each cluster based on the residual energy. The results showed slight improvement in the lifetime, making the algorithm non-significant.

2.6.2 HEED Protocol

HEED protocol [52] is an extension of LEACH protocol. It depends on residual energy in the election of CH. Each node sets its probability of becoming a CH according to the following equation:

$$CH_{prob} = C_{prob} \times \frac{E_{resid.}}{E_{max}} \tag{2.9}$$

where:

- C_{prob} is the initial percentage of CHs. The initial value is set to 5%.
- $E_{resid.}$ is the node's residual energy.
- E_{max} is the node's maximum energy.

If CH_{prob} is more than HEED's random number, the node becomes a CH. This protocol minimizes the communication overhead with less costly algorithms. It extends the network lifetime and forms compact cluster with better distributed CH. The problem with HEED is that it consumes high energy for local communication and also for communication between CH and base station.

2.6.3 EEUC Protocol

EEUC protocol adopts variable cluster size architecture and multihop routing [53]. It adopts the rotation of CHs while considering the residual energy. Based on the fact that CHs placed near the base station will contribute in routing communication more than far CHs, theses CHs will consume more energy than far CHs. In order to balance the energy consumption among the nodes, the algorithm distributed smallest clusters to be the nearest to the base station. In each round, every node generates a random number. Compared with the algorithm's threshold, the algorithm decides which nodes become possible candidates. Then CHs are chosen from the possible candidates according to their residual energy levels. CHs near the BS will contain lower members than far CHs. The problem with EEUC algorithm is that it is not practical for real world because it assumes circular distribution of nodes.

2.7 Clustering based K-Means

K-means was applied to perform WSN clustering in [54, 55, 56]. In [57], the authors proposed using K-means central clustering algorithm to cluster an indoor WSN. The proposed algorithm first run K-means to calculate the best number of clusters based on the following formula:

$$K_{opt} = \sqrt{\frac{N}{2\pi}}\sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}}\frac{M}{d^2_{toBS}} \tag{2.10}$$

where N is the number of nodes in the network, M is the side of the squared-geographical area. ϵ_{fs} and ϵ_{mp} are the amplifier energy values based on the RF energy dissipation model. The K-means clustering was used to select the CH and cluster members. However, the proposed algorithm was neither simulated nor tested.

Sasikumar and Khara [58] developed two K-means clustering models: centralized and distributed. The centralized K-means model was run on the base station. The distributed model is processed in each node. The information needed was gathered from messages broadcasted by every node. Simulation results showed that the distributed processing time was considerably less. However, there was no difference in the energy consumed. Moreover, the distributed algorithm imposed communication overhead because broadcasting consumes the node's energy.

2.8 Clustering WSN using Soft Computing

Clustering is an NP hard problem that is ineffectively solved by traditional techniques. The dynamic nature of the WSN makes the problem more complex due to repetitive change in the clusters and CHs which can't be modeled by the traditional mathematical methods. Traditional clustering algorithms suffer from non-uniformity in clusters and CH distribution [59]. They are expensive algorithms. The following subsections discuss previously implemented clustering algorithms based on SC techniques: GA and PSO. These techniques are explained in the next chapters.

2.8.1 Clustering based GA

In [60] the authors have proposed a GA clustering algorithm for optimizing the number of CHs. The chromosome is a simple 9-bit binary representation where the bit value of 1 represents a CH, and a 0 represents an ordinary node, as shown in Table 2.4. The fitness function is defined as:

$$Fitness = w * (D - distance_i) + (1 - w) * (N - H_i) \tag{2.11}$$

where D is the total distance from all nodes to sink, $distance_i$ is the total distances from regular nodes to their cluster head, N is the total number of nodes, H_i is the number of cluster heads, and w is preset weight. The results showed that the cluster layout depends on the location of the base station. More cluster heads are elected when the base station is around the center of the network.

Table 2.4: 9-bit Binary Chromosome Representation

1	0	1	1	0	0	1	0	1

The authors in [48] used the same model as [60], but with different mutation factor and sink location. They showed that better fitness value is reached when CH saturated to 25 percent of the total nodes. However, the choice of CH was not based on its residual energy. This could lead to network disconnection because if the optimal CHs have at least one CH with low energy, it will fade quickly and disconnect part of the network.

Mehr improved the previous work of [60], and [48] by adding the residual energy in the fitness function calculation [61]. The Fitness function used is shown in Equation 2.12

$$Fitness = RE + SE + (w * (D - distance_i)) + ((1 - w) * (N - H_i)) \tag{2.12}$$

where RE is the total cluster heads' energy, and SE total energy needed to send data from cluster heads to sink. The results were compared with LEACH and showed proper distribution of clusters and improvement in the network lifetime.

In [40], the authors used GA to optimize the clustering problem based on minimizing the energy consumption. In their model, the radio transmission technology is used in their calculations. The fitness calculation depended on the distance between nodes, CHs and sink. The GA's outcome was the optimal Cluster Heads. The base station then identifies the cluster members and the transmission schedule. The results have been applied on a simulator in [62]. Each CH is assumed to send directly to the sink. Multiple hop communication between CHs was not considered. Although their algorithm performed better than LEACH, the improvement was not significant. This is because of the complexity of the fitness function. Many parameters have been taken into consideration and each one is assigned a weight that is updated at each generation.

In [1], the authors proposed a GA approach to minimize the communication distance. Moreover, a two-dimensional chromosome representation is used. The chromosome mapped the actual node layout of the deployment area. The gene's value is either zero indicating non-existing node, or one for sensor node or two for CH. The algorithm used the result of the LEACH as an initial condition to GA algorithm. The fitness function used is as follows:

$$Fitness = \sum_{i} \sum_{j} d^2_{CH(ij)} + \sum_{i} d^2_{SN(i)} \tag{2.13}$$

where i is the number of CHs and j is the member number in cluster i.

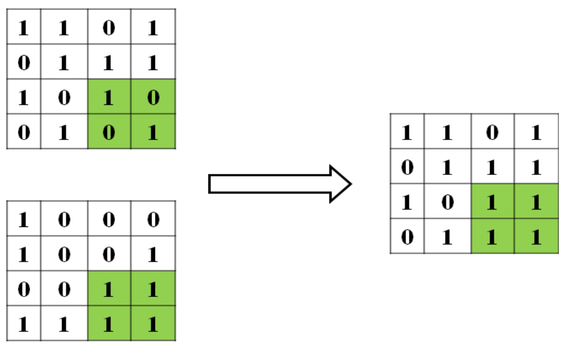


Figure 2.7: 2D crossover [1].

The chromosome is divided into sectors, and crossover is performed by exchanging sectors between parents to ensure that the genes move with their neighbors, as shown in Figure 2.7. The results proved better performance than LEACH.

The authors in [63] applied GA to perform WSN clustering. The algorithm optimizes the number of clusters, CHs and members according to the following fitness:

$$Fitness = \frac{100}{E} + \frac{TD - RCSD}{TD} - \frac{10(N - TCH)}{N} \tag{2.14}$$

where E is the total energy consumed during one transmission, TD is the total distance from every node to the base station, $RCSD$ represents the total clustered communication distance, N is the number of nodes, and TCH is the number of clusters. Simulation results showed improvement over LEACH

In [64] the authors proposed dividing the network into clusters. But instead of sending the measured data from the node to CH, they used mobile agent with each cluster that migrated through its nodes. The mobile agent collects the data and sends it to the base station. The authors divided the chromosome into two arrays: group array and sequence array (see Table 2.5). The group array contains the number of member in each cluster. The sequence array identifies the nodes that belong to each cluster. Crossover only exchanges the nodes in the same group. The mutation changes the number of nodes in two groups to ensure consistency. They used GA to calculate the optimum number of mobile agents (i.e. clusters) and the cluster layout. The authors' objective criterion was based on the network latency. The energy condition was not included. Analysis of simulation results showed that the sensor nodes traversed by the mobile agent will result in energy depletion.

Table 2.5: Group array and sequence array Chromosome Representation

Sequence Array	7	4	1	6	8	5	3	2
Group Array	3	1	2	2	0	0	0	0

In [65] the authors implemented GA to optimize CH selection. The algorithm applies two phases: the first phase divides the network into optimal number of clusters, then the second phase applies GA to select the best CH for each cluster based on the node's energy and the location of the CH with respect to other nodes in the cluster. Simulation results showed improvement over LEACH.

The authors in [66] implemented GA hierarchical clustering. It optimizes the number of clusters and the CHs using the same fitness of Equation 2.11. The algorithm sets the population size and number of generations to be the same as the network size. Once the optimum network topology is selected, a multihop protocol is set up. Each CH sends the message to either a nearest CH, or to the base station if closer than the nearest CH. Results were compared with [60] rather than LEACH, and showed better results.

2.8.2 Clustering based PSO

In [67], the authors' objective was to group the network into equal sized clusters using PSO. The clustering algorithm consists of two main phases. The first phase divides the network into equal clusters using a recursive PSO algorithm. This phase mainly searches for the best line that divides the area into two equal regions. Then recursive PSO division is made where each region is divided into equal regions, and so on till the required number of clusters in obtained. The second phase applies PSO to select the best CH in every cluster based on minimizing the communication distance. Compared with the K-means clustering algorithm, the application successfully formed equal clusters (which is not guaranteed by K-means). However, K-means produced lower communication distance since the CHs are in the middle of the clusters. The application is computationally expensive since PSO is run recursively. Also, no comparison with any protocol has been addressed.

Guru et. al. applied PSO to obtain the optimum cluster layout using a fitness function based on distance calculations, as shown in Equation 2.15 [68]. Residual energy calculations were not included.

$$F = \sum_{j=1}^{k} \sum_{i=1}^{n_j} (d_{ij}^2 + \frac{D_j^2}{n_j}) \tag{2.15}$$

where d_{ij} is the distance between node i and its cluster head j, D_j is the distance from cluster head j to the base station, and n_j is the number of nodes in the cluster j. The authors applied the PSO algorithm while varying inertia weight, or the acceleration constant. Analysis of the results are discussed in details in [69].

The authors in [70] used a centralized PSO algorithm to minimize inter-cluster distance and maximize the average energy level. The radio model is used in calculations. The nodes were assumed to have from $2J - 5J$ initial energy. The authors preset the number of clusters to 5% of the total nodes. Results showed improvement over LEACH.

In [59], the authors proposed using improved PSO algorithm to solve the uneven clustering. They chose the number of clusters to be 5 percent of the total nodes, and each cluster has the same number of nodes. Their fitness was based on the communication distance. The particle structure used contains the ID of the cluster head followed by the IDs of its members, as shown in Table 2.6.

Table 2.6: PSO Particle Structure

Cluster 1				Cluster 2				Cluster 3			
12	8	7	1	4	10	6	11	3	5	9	2
CH	members			CH	members			CH	members		

Moreover, the PSO dynamic inertia weight was modified to include the particles' diversity. The CHs resulted from the PSO algorithm is then checked for their energy level. If their energy level falls below a threshold, then they are replaced by the nearest node whose energy is more than the threshold. Compared with LEACH and improved LEACH, the proposed PSO algorithm showed better results. However, the overall nodes' remaining energy and lifetime is not considered.

In [71], the authors adopted a distributed algorithm that selects two CHs for each cluster: a master CH (MCH) and a vice CH (VCH). The MCH collects, aggregates and sends the aggregated data message to the VCH. The role of VCH is to only send the message to the base station. Similar to LEACH protocol, the algorithm consists of setup phase and steady state phase. In the setup phase, PSO selects the best MCHs and VCHs based on a fitness function that includes both the residual energy and the cluster communication distance, as shown in Equation 2.16. The results showed improvement over the LEACH protocol. However, the algorithm is computationally expensive, since the PSO algorithm runs periodically in the setup phase. Also, the value of the constant factor ϵ in the fitness function can affect the optimum CHs selected; this makes the choice of the factor itself an optimization problem.

$$F = \epsilon \times \frac{CHenergy}{ClusterEnergy} + (1 - \epsilon) \times \frac{ClusterMemebersCount}{ClusterDistance} \tag{2.16}$$

The authors in [72] modified the LEACH protocol to apply PSO in the setup phase to select the optimal CHs. The PSO algorithm adopted the square of the communication distance as the fitness function. The dataset are randomly deployed and the radio model is adopted. The simulation results outperformed LEACH protocol. However, applying PSO algorithm in the setup phase rather than applying it only when CH topology changes forces complexity in WSN operation.

In [47] the authors proposed using an embedded PSO-Cuckoo search algorithm to minimize the communication distance and energy consumed. The fitness was based on minimizing both the communication distance and the energy consumed. The model used for the energy calculation is the radio model. The authors assumed uniformly distributed nodes, a predefined number of clusters and predefined nodes that have more energy than others. The fitness function adopted is as follows:

$$F = c * f_1 + (1 - c) * f_2 \tag{2.17}$$

$$f_1 = \frac{\sum E(n)}{\sum E(CH)} \tag{2.18}$$

$$f_2 = \sum d(n, CH) \tag{2.19}$$

f_1 is the ratio between the total nodes' energies to the total CHs energies. f_2 is the total distances from each node to its CH. The proposed algorithm is aimed to speedup convergence. Instead of abandoning discovered nests, the proposed algorithm used PSO on those nests to move them to an acceptable and best solution. These new solutions replaced the abandoned ones in the cuckoo search individually. The results showed faster convergence when compared with GA, PSO, and CS seldom. The lifetime considerably increased compared to LEACH. However, this model considered only the global distance between the nodes and its cluster head. The communication distance between the cluster head and the sink is not considered. Also, the new proposed algorithm is more complex because PSO runs inside each CS iteration to get the best PSO result.

PSO is used in [73] to cluster the WSN based on minimizing the communication distance and energy consumption, as shown in the following equations:

$$F = \phi_1 f_1 + \phi_2 f_2 + (1 - \phi_1 - \phi_2) f_3 \tag{2.20}$$

$$f_1 = \sum \frac{D(n,p)}{Ncount_p} \tag{2.21}$$

$$f_2 = \sum \frac{E(p)}{E(Cluster)} \tag{2.22}$$

$$f_3 = \frac{1}{H(p)} \tag{2.23}$$

$D(n,p)$ is the Euclidean distance between the node n and the CH p,and $Ncount_k$ is the number of members belonging to the CH p. $E(p), E(Cluster)$ are the energy of the selected CH and the total energy of the cluster respectively. $H(p)$ is the head count associated with the CH p. The algorithm was compared with the famous LEACH protocol. The results showed improvement over LEACH.

The authors in [74] proposed non uniform clustering mechanism using PSO. The square geographic area is divided into four quarters. Each quarter is then divided into square layers where each layer is a cluster, as shown in Figure 2.8. The base station is located in the center of the region. The network adopts multihop routing; every layer's CH sends aggregated data to its neighbor layer's CH. PSO searches for the layers best CH based on minimization of an energy fitness proposed. However, there is no guarantee on the reliability of this mechanism since it is not evaluated with a benchmark.

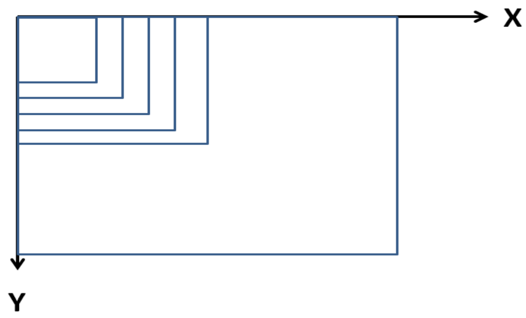


Figure 2.8: 2D crossover.
[74]

2.9 Clustering based Multi-Objective Optimization

The authors in [75] proposed a centralized protocol having two phases per round: setup and steady state. The PSO works in rounds, specifically in the steady state phase to select the optimal number of clusters and the optimal CHs based on two objective functions. The first objective function is the energy consumption , while the second fitness is the intracluster distance. PSO algorithm adopts binary particle values. The particle position is either 0 or 1. The velocity, after calculation, is changed to a binary value using a special sigmoid function compared to a random threshold value. Although the results showed improvement over LEACH protocol, the algorithm adds complexity to the protocol since PSO is run during every round.

[16] proposed a multi-objective genetic algorithms (MOGA) approach for achieving two objectives: cluster membership and routes-to-sink. They aimed to maximize the coverage while minimizing the battery usage. Each sensor node is selected to be one of four options: inactive node, cluster-head, inter-cluster router (ICR), sensor node. Each cluster is managed by a cluster-head, and cluster-members are represented by inactive/active nodes and ICRs. Cluster-head performs data-fusion from various nodes while ICR routes cluster data to the sink. The MOGA fitness functions are the Total Node Fitness (TNF) and the Total Route fitness (TRF). The total node fitness is a weighted sum Cluster-Head Fitness (CHF), Node Communication Fitness (NCF), Battery Status Fitness (BF),and Router Load Fitness (RLF). The total route fitness contains the battery fitness, the node communication fitness and the inter cluster routing data. The results showed convergence to optimal fitness. However, this was done at the expense of convergence time.

The authors in [76] proposed an adaptive PSO (APSO) for selecting the CHs. They introduced modification to the PSO algorithm to calculate the cognitive and social factors ϕ_1 and ϕ_2 respectively. The new values of these factors depend on the local and best positions achieved p_{id}, and p_{gd} described in Section 3.2. Moreover, the algorithm replaces 25% of the particles that remain stationary, for a fixed number of iterations, with new initialized particles. The algorithm uses two objective functions that represent the energy and communication distance respectively. The results outperformed LEACH in 60 % of the whole network lifetime, while LEACH outperformed the proposed algorithm in the remaining rounds.

Zahmatkesh and Yaghamaee proposed a multi-objective GA solution to clustering problem [77]. They depended on uniformly distributed heterogeneous nodes. They also assumed that the nodes communicate with each other. The authors used the same model in [16], but in a simpler form. The first fitness function used is aimed decide the optimal number of clusters to avoid cluster overhead. The same first fitness function is used without routing calculations. The second fitness function decides the cluster layout. However, the transmission cost function proposed assumed small distance between the cluster head and the base station.

2.10 Literature Review Summary

Although traditional approaches focused on improving LEACH protocol, non-significant results were reported. The problem with K-means algorithm is that it does not guarantee reaching to an optimum solution. It is most likely to get stuck in local optima. Techniques such as GA and PSO were presented to handle the clustering problem of WSN in number of ways. Some research work focused on developing a suitable fitness function that minimize energy loss or communication distance. While other research, focused on refining the presentation of GA or PSO techniques to obtain better clustering outcomes. Some authors proposed hybridizing techniques to minimize the communication distance and energy consumed. Still, developing a low-computational and high performance clustering algorithm is a challenge. Deciding the number of clusters to be produced is an important issue. If low clusters are to be produced, then cluster overloading occurs. If the number of clusters is high, then more nodes will connect to the sink. In both cases, the CH nodes will deplete faster which will result in sub-network disconnection. Finding the relation between the acceptable number of the clusters and the node density is still and open issue.

Chapter Three

Soft Computing

Clustering is an NP hard problem that can't be solved efficiently using traditional methods. Soft Computing paradigms are suitable to adapt for WSN dynamic nature. SC has proved to solve successfully NP hard problems. SC was introduced by Lotfi Zadeh in 1992. SC is a family of methods that imitate human intelligence, biological evolution and social behavior of species. SC mainly constituted fuzzy logic, neural networks and probabilistic reasoning. Recent trends of SC included evolutionary algorithms and swarm intelligence. This chapter explains the SC techniques used in this thesis. They are: Genetic Algorithm, and Particle Swarm Optimization.

3.1 What is GA?

GA is an adaptive search algorithm which was presented by J. Holland [78], and extensively studied by Goldberg [79, 80], De Jong [81, 82, 83], and others. GA successfully handled many areas of applications and was able to solve a wide variety of difficult numerical optimization problems. GA have been applied successfully in many optimization problems as pattern recognition [84], [85], and [86]. It has been applied for robotics path planning [87], [88] and [89]. An interesting field that used GA is software engineering problems as in [90], [91], [92], and [93].

GA is much less likely to get trapped in local minima on multi-modal search spaces. GA found to be quite insensitive to the presence of noise [94]. GA is inspired from Charles Darwin's theory of evolution: 'the survival of the fittest" where only the best individuals are able to survive in harsh biological conditions. Thus as generations run, better individuals are obtained by reproduction genetic operations as crossover and mutation of chromosomes.

3.1.1 Encoding the Problem

The main challenge in solving the problems using GA is encoding the problem into a set of chromosomes; each representing a solution to the problem. A possible solution is represented by an encoded version of parameters. The solution is a concatenated string of genes of the same value type: binary digits, numbers, characters,...etc. A single solution is named 'chromosome' or 'individual'. A population is a set of chromosomes that represent possible solutions of the encoded problem. An example of a chromosome that is a string of binary digits is shown in Table 3.1

Table 3.1: Example of a GA chromosome

1	0	1	1	0	0	1	0

3.1.2 Fitness Function

In order to evaluate the quality of the proposed solution, i.e. chromosome, a fitness function is used. The fitness function generates a number that indicates if the corresponding solution is better or worse. The fitness formula depends on the encoded problem.

3.1.3 Selection

This operation selects the parents that will be used to create the next population (individuals). Examples of the population selection methods are: Random Selection, Roulette Wheel Selection, and Tournament Selection [79]. In Roulette Wheel Selection each chromosome is denoted by the ratio of its fitness to the sum of all chromosomes' fitness. Thus the individuals with better fitness are more likely to be chosen. The Tournament Selection randomly selects k individuals, ($k \geq 2$), and then the one with best fitness is chosen.

3.1.4 Crossover

Crossover forms new chromosomes from the selected parents by exchanging genes from the selected parent pairs. Figure 3.1 shows an example of a 1-point crossover. The crossover point is chosen randomly and the genes after this point are exchanged between parents. Figure 3.2 shows a 2-point crossover. In Uniform crossover, each gene of one pair is separately exchanged by a randomly chosen gene from the other pair.

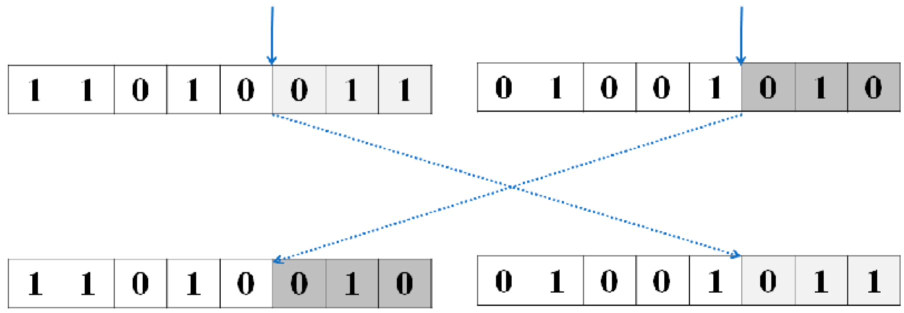


Figure 3.1: Example of One-point crossover

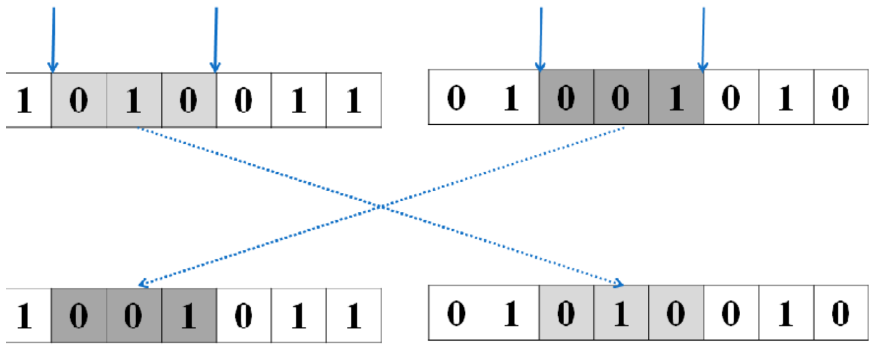


Figure 3.2: Example of Two-point crossover

3.1.5 Mutation

Mutation is important since it prevents being trapped in local minima. This is done by making a minor change in the chromosome. Mutation changes one or more genetic element in the produced offspring. Table 3.2 shows a simple 1-point mutation. In binary chromosomes representation, the digit is inverted. In numeric chromosomes representation, non-uniform mutation is an example. It changes one of the parameters of the parent based on a non-uniform probability distribution.

Table 3.2: Example of Mutation

Before Mutation:	1	**1**	0	0	1	0	1	1
After Mutation:	1	**0**	0	0	1	0	1	1

3.1.6 How does GA work

Algorithm 1 summarizes the basic steps of GA [80, 95]. GA starts by first encoding the problem into a set of chromosomes. First, an initial population is performed randomly. Then for each generation a group of parents are selected, based on their fitness value, to be parents for the next generated population. Then crossover and mutation are applied on them, to produce a new set. This process is repeated until no best solution is discovered. GA algorithm ensures that, in most cases, the fittest individuals are chosen to participate in the next generation, and eventually the best solution is obtained [96].

Algorithm 1: Basic steps of GA

begin GA
  $g = 0$ generation counter
  Initialize population
  Compute fitness for population $F(P(g))$
  Repeat
    $g = g + 1$
    Select $P(g)$ **from** $P(g-1)$
    Crossover $P(g)$
    Mutate $P(g)$
    Compute fitness for population $F(P(g))$
  Until (Terminating condition is reached)
end GA

3.2 What is PSO?

Particle Swarm Optimization was developed in 1995 [97] by James Kennedy and Russell Eberhart. PSO is a robust stochastic nonlinear- optimization technique based on movement and intelligence of swarms. It is inspired from social behavior of flock of birds or fish, where a group of birds randomly search for food in an area by following the nearest bird to the food. They interact with each other to identify which bird is the nearest. Then, the birds explore the area around that nearest one to locate another nearest place to the food. And so on, all the birds eventually reach the source of the food.

The PSO algorithm has been applied successfully in many different application areas such as training neural networks [98], [99], [100], [101], and [102]. It has been applied in various electric power system optimization problems [103], [104], [105]. It has also been used in designing electronic circuits [106], [107].

PSO consists of number of individuals, called particles. The group of particles constitutes the swarm. Each particle represents a possible solution. The particles move in the search space looking for the optimal solution. Every particle is characterized by three properties:

1. the position of the particle in the search space
2. the best position it has individually reached
3. the velocity of the particle

Moreover, each particle exchanges information with its neighborhood particles to memorize the best position reached by the swarm. Using these properties, each particle updates its position. Similar to GA, a fitness function is used to evaluate the quality of the solution denoted by the particle's current position. The fitness value obtained helps in directing the particles to the right direction towards the optimal solution.

3.2.1 PSO Basic Equations

In implementation of PSO, each particle is represented by its position in the search space, x. Then the position of the particle i at time t, $x_i(t)$, is represented as:

$$x_i(t) = x_{i1}(t), x_{i2}(t), ...x_{id}(t) \tag{3.1}$$

where $x_{id}(t)$ is the position of particle i in dimension d.

Each particle tries to modify its position using information as: its current position, its current velocity $v_i(t)$, the distance between the current position and best solution individually found, and the distance between the current position and the best solution found in its neighborhood [108]. The basic PSO equations are given as follows:

$$x_i(t+1) = x_i(t) + v_i(t+1).\delta t \tag{3.2}$$

Standard implementations of PSO use $\delta t = 1$, then the PSO equations used are given in Equations 3.3 and 3.4 as follows:

$$v_{id}(t+1) = v_{id}(t) + \phi_1 * r_1 * (p_{id} - x_{id}) + \phi_2 * r_2 * (p_{gd} - x_{id}) \tag{3.3}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{3.4}$$

where:

v_{id} represents the velocity of particle i in dimension d,

x_{id} represents the position of particle i in dimension d,

ϕ_1, ϕ_2 are positive constants,

r_1, r_2 are random numbers

p_{id} represents the best position reached so far by the particle, and

p_{gd} represents the global best position reached by the neighborhood.

As shown from the Equation 3.3, the velocity update equation is influenced by two components: $(p_{id} - x_{id})$ represents the cognitive component, and $(p_{gd} - x_{id})$ represent the social component. Figure 3.3 shows the influence of the PSO equation components on the movement of the particle.

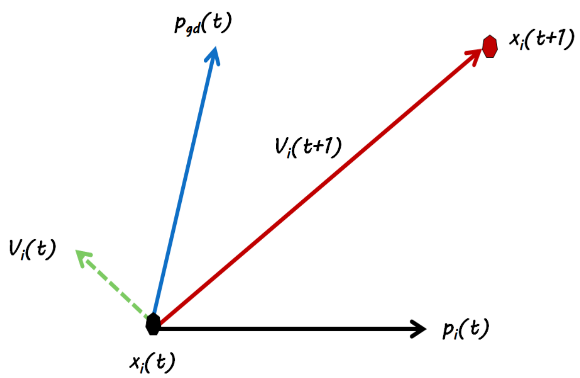


Figure 3.3: A 2D representation of PSO mechanics

3.2.2 PSO Neighborhood Topology

The choice of subgroups of particles communicating with each other is called 'neighborhood topology'. Two famous topologies are shown in Figure 3.4. They are:

- **Star topology**: In this topology, all particles in the swarm communicate with each other. In this case there is only one neighborhood and only one p_{gd}, called 'gbest', is obtained in each generation.

- **Ring topology** : Each particle communicates only with only two neighborhoods. Every neighborhood contains only three particles and each particle is a member in three neighborhoods. In this case there is no global best position for all particles; each particle has its own p_{gd}, called 'lbest' value, p_{ld}.

Figure 3.4: (a) PSO Ring topology (b) PSO Star topology

The Star topology has the advantage of fast convergence. However, this fast convergence is misleading in some cases, where premature convergence is reached. The Ring topology is characterized by having slower and less premature convergence and performs better on multimodal problems. Different topologies are discussed in [109, 110]

3.2.3 Maintaining PSO Particles within Search Space

The velocity of PSO is prune to reach infinity which causes the particles to reach a state of instability and go beyond the search space. Therefore, the velocity of the particle should not exceed a maximum value, v_{max}. The performance can suffer if maximum velocity is inappropriately set. If it is too high, the particles can fly past optimal solutions, and if it is too low, they can get stuck in local minimal.

Also the distance of the particles should not go beyond the search space limits. Therefore, the position of the particle is subjected to the following limiting equation:

$$x_{id}(t+1) = \begin{cases} x_{dmin} & \text{if } x_{id}(t+1) < x_{dmin} \\ x_{dmax} & \text{if } x_{id}(t+1) > x_{dmax} \end{cases} \tag{3.5}$$

where x_{dmin} and x_{dmax} are the lower and upper limits in dimension d.

3.2.4 Inertia Weight PSO Model

A newly implemented model has been developed and named the Inertia Weight PSO model [108]. The inertia weight equation added a slight modification to the standard PSO model presented in Equation 3.3. The modification was introduced by [111]. An inertia weight w is added that controls the velocity of the particle. The PSO Equation is described as:

$$v_{id}(t+1) = w_i(t).v_{id}(t) + \phi_1 * r_1 * (p_{id} - x_{id}) + \phi_2 * r_2 * (p_{gd} - x_{id}) \tag{3.6}$$

Experiments have been made to explore the effect of the inertia weight value on the performance of PSO particles. Setting inertia value to less than 1 can cause particle poor exploration due to jumping in the search space. When values of inertia weight exceed 1, the particles rarely change its direction and may lead to velocity explosion. Experiments lead to adjusting the inertia to be linearly decreasing from value $w = 0.9$ to $w = 0.4$ during the iterations. This helps the particles perform exploration at earlier stages and focus the search in later stages.

3.2.5 Clerc PSO Model

Another model is explored that added a constriction factor to the standard PSO model [112]. The Clerc PSO equation is denoted as follows:

$$v_{id}(t+1) = \psi[w_i(t).v_{id}(t) + \phi_1 * r_1 * (p_{id} - x_{id}) + \phi_2 * r_2 * (p_{gd} - x_{id})] \tag{3.7}$$

where constriction factor ψ is defined as:

$$\psi = \begin{cases} \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} & \text{for } \phi > 4, \phi = \phi_1 + \phi_2 \\ 1 & \text{otherwise} \end{cases} \tag{3.8}$$

Setting inappropriate coefficient values can cause particle explosion or local optima entrapment. Extensive experimentation has been performed to identify the best coefficient values. Empirical results showed that setting $\phi = 4.1$, $\phi_1 = \phi_2 = 2.05$ and $\psi = 0.72984$ ensure particle convergence.

3.2.6 Trelea PSO Model

There are two Trelea PSO models: Trelea-1 and Trelea-2 [113]. They use the same standard PSO model, but with specified values used for the constant factors. Trelea-1 is denoted by:

$$v_{id}(t+1) = 0.6 * v_{id}(t) + 1.7 * r_1 * (p_{id} - x_{id}) + 1.7 * r_2 * (p_{gd} - x_{id}) \tag{3.9}$$

Trelea-2 is denoted by:

$$v_{id}(t+1) = 0.7296 * v_{id}(t) + 1.494 * r_1 * (p_{id} - x_{id}) + 1.494 * r_2 * (p_{gd} - x_{id}) \tag{3.10}$$

3.2.7 PSO- Time Varying Acceleration Coefficients Model

The authors in [68] provided the PSO- Time Varying Acceleration Coefficients Model (PSO-TVAC) by modifying the Inertia Weight PSO model. The authors used Equation 3.6 and modified the values of the ϕ_1 and ϕ_2 to change by iteration. ϕ_1 decreases from 2.5 till 0.5, while ϕ_2 increases from 0.5 till 2.5. Thus, at earlier stages, the algorithm gives more weight to cognitive component in order to explore the search space thoroughly. Then at later stages the social component focuses on the promising region to get the best result. The values of ϕ_1 and ϕ_2 change according to the following equations:

$$\phi_1 = 2.5 - \frac{2 \times iterationNumber}{MaxIterations} \tag{3.11}$$

$$\phi_2 = 0.5 + \frac{2 \times iterationNumber}{MaxIterations} \tag{3.12}$$

3.2.8 How PSO Works

The PSO algorithm is straightforward (see Algorithm 2). First, initialize particles with random position and velocity vectors. For each particle: evaluate the fitness and if it is better than the best individual fitness then update it. After that, update the best global fitness. Then obtain the new velocity and position for each particle. This procedure is repeated for a number of iterations or until convergence is beyond a certain limit.

Algorithm 2: Basic steps describing the PSO algorithm

begin PSO
  Randomly initialize the position and velocity of the particles: $X_i(0) and V_i(0)$
  while (While terminating condition is not reached) do
    for for i = 1 to number of particles
      Evaluate the fitness:= $f(X_i)$
      Update p_i and g_i
      Update velocity of the particle V_i
      Update position of the particle X_i
    Next for
  end while
end PSO

3.3 PSO versus GA

PSO and GA are very similar [108]. Both are population based stochastic optimization that starts with a group of randomly generated populations. They have fitness values to evaluate their population, and update the population and search for the optimum with random techniques. Also, both include random components to overcome local minima. However, they differ in the type of operators used to reach the optimal solution. PSO differs from GA in that there is no selection, crossover and mutation. PSO uses the velocity component as the main operator. PSO particles do not die; the same particle changes its position in the search space. The information sharing mechanism in PSO is significantly different; PSO applies neighborhood topology while GA applies crossover. Finally, PSO proved computational efficiency and simplicity [114], [115].

Chapter Four

Research Overview

The main objective of our proposed WSN clustering approaches is to increase the network lifetime by grouping the sensor nodes into a number of clusters. To cluster n nodes WSN, an exhaustive algorithm has to go through $2^n - 1$ solutions to find the optimal clustering layout. Solving such problem is known to be NP hard. We propose three approaches to achieve efficient and effective clustering for WSN, on assumption that the network communication is established. This chapter describes in details our proposed approaches.

4.1 Solution 1: Hybrid K-Means PSO Clustering Approach 'KPSO'

We proposed a Hybrid K-means PSO Clustering Approach 'KPSO' to solve the energy consumption problem based on clustering. Figure 4.1 shows the phases of our proposed approach.

- Phase 1: The first phase applies K-means to partition the network into k clusters.
- Phase 2: Next, the PSO searches for the best CH within each cluster obtained by K-means.
- Phase 3: Finally, the last stage evaluates the obtained cluster layout.

Our proposed hybrid approach decreases the computational effort; the search space will be reduced to about $\frac{n}{k}$ solutions for each cluster.

4.1.1 Phase 1: Clustering based K-Means

K-means is a simple unsupervised clustering algorithm. It was successfully used to solve a variety of clustering problems. It is heuristic in nature, and proved to be effective in determining clusters of spherical shapes [116, 58]. It classifies a given data set into a predefined

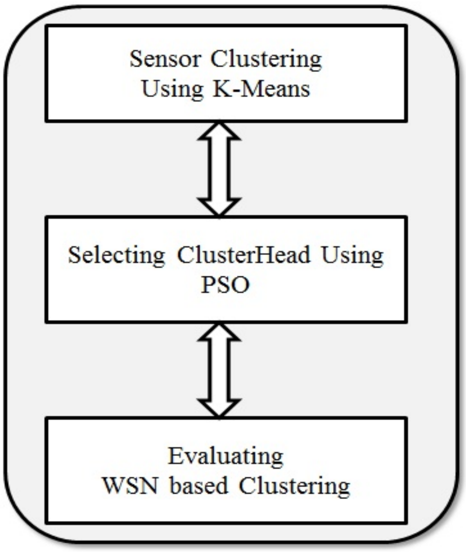


Figure 4.1: K-means PSO Hybrid clustering

number of clusters, k. Its main idea is to define k centroids, one for each cluster. It was initially proposed by Stuart Lloyd in 1982 [117]. It partitions a set of data points into predefined number of clusters [118], [119]. It mainly maximizes the inter-cluster distance while minimizing the intra-cluster distance. The algorithm for obtaining the centroids aims at minimizing an objective cost function which is a squared error function [120]. The objective function can be simply defined as:

$$D = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^j - \mu_j \right\|^2 \tag{4.1}$$

where $\left\| x_i^j - \mu_j \right\|^2$ is the Euclidean distance measure between a data point x_i^j and the cluster center μ_j , D is an indicator of the distance of the n data points from their respective cluster centers. Different matrices, other than Euclidean distance, that can be used as objective function for K-means is stated in [121, 122].

The selection of the number of clusters in our approach is not arbitrary. This phase runs a module that decides the number of clusters to be produced based on the distribution of the generated nodes. The module computes the total computed clustered distance for different number of clusters. Then the number of clusters producing the minimum computed distance is selected. Algorithm 3 shows the steps for selecting the best number of clusters.

In our proposed approach, the base station will select k nodes as CHs, then each node joins its nearest CH. A new CH is chosen as the middle of the cluster. These steps are

Algorithm 3: Algorithm for selecting the best number of clusters

begin
  n = number of nodes
  For i=5 to 30
    $k = \frac{i \times n}{100}$
    Form k clusters using K-means
    Calculate the computed clustered distance
  next i
  Find k for minimum computed distance
end

repeated until no new CH is selected. The distance between two nodes s_1, s_2 is computed based on the following Euclidean distance calculation:

$$D(s_1, s_2) = \sqrt{(x_{s1} - x_{s2})^2 + (y_{s1} - y_{s2})^2} \tag{4.2}$$

Where x and y are the node's x-coordinate and y-coordinate, respectively. This phase will divide the network to disjoint clusters. The base station will save information about each cluster's sensor node ID, location and energy level. The steps of WSN K-means clustering is summarized as follows:

1. Arbitrarily choose k nodes to be CHs.
2. Join each node to the closest CH.
3. Calculate the new cluster center by calculating the mean distance between each CH and all sensors in its cluster.
4. If at least one new CH is changed then go to step 2, else stop the process.

4.1.2 Phase 2: Cluster Head Selection using PSO

PSO is then applied to select the optimal cluster head from each cluster obtained by the K-means phase. For instance, if the K-means partitioned z nodes to 3 clusters with l, m, and n

nodes, respectively, then the PSO model will select three cluster heads; one from the l nodes, the second from the m nodes and the third from the n nodes. PSO will search the space of all possible CHs and provide a convergence to the best solution.

Encoding the Problem

The size of the PSO particle is fixed. It consists of k entries, where k is the number of WSN clusters, as shown below. It represents an array containing the indices of each node, for every cluster, elected as CH.

CH_1	CH_2	......	CH_k

Fitness Function

The objective of the PSO phase is to maximize the fitness function in order to achieve the maximum lifetime of the cluster head. Maximizing the WSN lifetime is done by maximizing the nodes residual energy and minimizing the dissipated energy. According to the RF model, the dissipated energy is proportional to the square of the distance between the transmitter and receiver, d. But when d exceed the threshold (d_0), the dissipated energy is proportional to d^4. Thus, we proposed a novel fitness function given in Equation 4.3. It represents an indicator to the number of transmissions a CH can perform during its lifetime.

$$F = \begin{cases} \sum_{i=1}^{k} \dfrac{E_i}{d_i^2} & \text{for } d_i < d_0 \\ \\ \sum_{i=1}^{k} \dfrac{E_i}{d_i^4} & \text{for } d_i \geq d_0 \end{cases} \tag{4.3}$$

where

- k represents the number of clusters,
- E_i represents the CH's actual energy,
- d_i is the Euclidean distance between the cluster head and the base station, and
- $d_o = 87m$ [46]

4.1.3 Phase 3: Model Evaluation

This phase simulates the energy consumed by every node in the network during each transmission. In a transmission cycle, each node sends a data message to its CH , then the CH aggregates the received data into one message and sends it to the base station. The dead nodes are identified along with the alive nodes after each transmission. This phase also counts for dead and remaining alive nodes. Each node's consumed energy is simulated based on the RF model provided in Section 2.5. The dissipated energy (DE) by a CH (DE_{CH_i}) is given in Equation 4.4. It calculate the energy consumed in receiving the messages from the member nodes, aggregating data plus transmitting an m-bit message to short and long distances, respectively. Equation 4.5 shows the dissipated energy by a non-CH (DE_{nonCH_i}) node. The member node only sends the data to the CH.

$$DE_{CH_i} = \begin{cases} n_i m E_e + n_i m E_{DA} + \epsilon_{fs} d_{toBS}^2 & \text{for } d_i < d_0 \\ n_i m E_e + n_i m E_{DA} + \epsilon_{mp} d_{toBS}^4 & \text{for } d_i \geq d_0 \end{cases} \tag{4.4}$$

$$DE_{nonCH_i} = \begin{cases} m E_e + \epsilon_{fs} d_{toCH}^2 & \text{for } d_i < d_0 \\ m E_e + \epsilon_{mp} d_{toCH}^4 & \text{for } d_i \geq d_0 \end{cases} \tag{4.5}$$

where:

- n_i is the number of nodes belonging to CH_i,
- m is the size of the message in bits,
- d_{toBS} is the Euclidean distance between the CH and the base station,
- d_{toCH} is the Euclidean distance between the node and its CH,
- $E_e = 50nJ/bit$, $E_{DA} = 5nJ/bit$, $\epsilon_{fs} = 10pJ/m^2$, and $\epsilon_{mp} = 0.0013pJ/m^4$.

4.2 Solution 2: PSO Variable Clustering Approach 'PSO-VC'

PSO is implemented to evolve complete clustering for WSN. The approach is aimed to obtain the optimal number of clusters, optimum CHs and optimum clusters layout; i.e. the members in each cluster. Figure 4.2 shows the phases of our proposed approach.

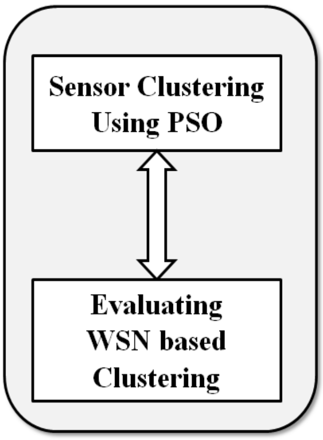


Figure 4.2: PSO variable clustering block-diagram

4.2.1 Phase 1: Clustering using PSO

In this phase, PSO is applied to select the optimum WSN cluster layout. PSO outputs the following optimum parameters:

- Number of clusters,
- Cluster Heads, and
- Members in each cluster.

Encoding the Problem

In this approach the PSO particle structure consists of the number of clusters, followed by the CH index for each cluster. The PSO particle representation will be as follows:

k	CH_1	CH_2	......	CH_k

where k is the number of clusters, CH_i is the index of the CH of cluster i. The particle structure is of variable size depending on the number of clusters obtained by the PSO particle during each iteration. For example, if PSO particle choose 5 clusters ($k = 5$) then the particle size will be 6 entries. PSO particle structure should include unique CH entries. If a CH index is repeated in multiple entries, then repeated indices are removed and the number of clusters entry is updated to include only the number of unique entries of the PSO particle structure.

Fitness Function

PSO-VC aimed to maximize the number of transmissions which a CH can perform before the node depletes its energy. The number of transmissions can be defined as:

$$\text{Number of Transmissions} = \frac{\text{Residual Energy}}{\text{Dissipated Energy}} \tag{4.6}$$

The dissipated energy depends on the distance between the node and its CH, and the distance between the CH and the sink. However, another attribute that has to be considered is the number of member nodes in a cluster. Excluding this attribute from the fitness leads PSO to select nodes close to the sink to be CHs, and thus uniform cluster distribution will not be achieved. Moreover, adding this attribute helps in obtaining balanced cluster density. The authors in [68] proved that the dissipated energy based on the distance and number of members can be represented as:

$$\text{Energy Loss} = \sum_{i=1}^{k} \Big(\sum_{j=1}^{q} d^2_{CH(i,j)} + \frac{d^2_{SN(i)}}{n_i} \Big) \tag{4.7}$$

where:

- k represents the number of CHs,
- q represents the number of members,
- d_{CH} is the distance from a node to its CH,
- d_{SN} is defined as the distance between node to the base station,
- E_i represents the CH's residual energy,
- n_i represents the number of members in cluster i, and
- d_0 is 87 meters.

Thus we proposed a fitness function as shown in Equations 4.8 and 4.9.

$$F = \sum_{i=1}^{k} \frac{E_i}{\sum_{j=1}^{q} d^p_{CH(i,j)} + \frac{d^p_{SN(i)}}{n_i}} \tag{4.8}$$

$$p = \begin{cases} 2 & \text{for } d < d_0 \\ \\ 4 & \text{for } d \geq d_0 \end{cases} \tag{4.9}$$

4.2.2 Phase 2: Model Evaluation

This phase simulates the energy consumed by every node in the network, as explained in Section 4.1.3.

4.3 Solution 3: Hybrid K-means PSO Clustering Approach 'KPSO-PSO'

We propose a Hybrid K-means PSO Clustering approach to solve for clustering WSN in order to increase the network lifetime. It is an extension to KPSO, the first proposed approach. Figure 4.3 shows the phases of our proposed approach. This approach outputs the best CHs, the best CH's antenna pattern, and the best members. It has the following phases:

1. Phase 1: apply the KPSO approach to partition the network into k clusters and select the best CH for each cluster.

2. Phase 2: obtain the optimal antenna pattern radius, and the optimal member distribution of CHs using PSO.

3. Phase 3: evaluate the resulting cluster layout.

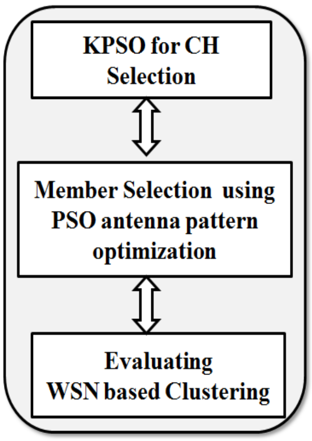


Figure 4.3: KPSO-PSO block-diagram

4.3.1 Phase 1: KPSO

This phase runs the first approach "KPSO" provided in 4.1. This phase will output the optimal CHs for the selected number of clusters. The KPSO phase consists of two sub-phases:

- *First: K-means for partitioning the nodes into clusters.*
- *Second: PSO for selecting the best CH from each cluster*

4.3.2 Phase 2: Antenna Pattern Optimization and Member Selection

In this phase, PSO is applied to assign the optimal members to its corresponding CH obtained by KPSO phase by optimizing the antenna pattern shape radius. The idea behind this approach is that a symmetric round shape is virtually assigned for each CH. The approach particularly works on two shapes: circle and ellipse. For each CH selected, the approach assigns a value, r, that is the radius of the virtual shape whose center is the CH. Sensors that fall within the shape will be chosen as the members for this cluster. The approach specifies the following rules:

1. A CH is not permitted to be a member inside another cluster. The maximum allowable shape radius of every CH does not exceed the Euclidean distance to the nearest CH neighbor.
2. A sensor that falls in more than one shape will be assigned to the nearest CH.
3. A sensor that does not fall in any shape will be assigned to the nearest CH.

Encoding the Problem

Our objective is to find the best antenna pattern which provides the best region for clustering. The gain of the antenna, G, is given in Equation 4.10 [123].

$$G = \frac{S}{A} \tag{4.10}$$

where S is the Area of isotropic sphere, and A is area of the antenna pattern.

Approximation of an elliptical antenna pattern is shown in Figure 4.4. Assuming the antenna pattern is uniform, the gain of the antenna G in this case is computed as a function of the parameters a and b. Given that: D is the distance from antenna location to the main antenna pattern, a and b are the ellipse radii. Thus, the gain G is computed as:

$$A = \pi ab \tag{4.11}$$

$$S = 4\pi D^2 \tag{4.12}$$

Thus:

$$\begin{aligned} G &= \frac{4\pi D^2}{\pi ab} \\ &= \frac{4D^2}{ab} \end{aligned} \tag{4.13}$$

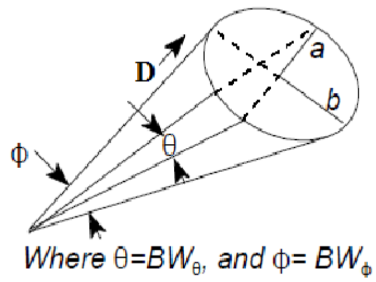


Figure 4.4: Approximating the antenna pattern as an elliptical area

As shown, the developed gain equation is a function of the elliptic dimension a and b. Thus, finding the optimal values of these parameters can help in developing a better pattern which leads to energy saving. In case of circular antenna pattern, where $a = b = r$, the gain is a function of the circle radius r as shown in the following equation:

$$G = \frac{4D^2}{r^2} \tag{4.14}$$

It is assumed that the size of PSO particle is fixed. Thus, we adopted the following representation for PSO. In case of applying the circle pattern with k clusters, the representation will be encoded as follows:

r_1	r_2	......	r_k

where r_1 represents the radius of the circle whose center is CH_1. In case of applying the ellipse pattern, horizontal and vertical radius parameters, a and b, need to be optimized. The PSO representation will be as follows:

a_1	a_2	......	a_k	b_1	b_2	......	b_k

Fitness Function

The goal of the fitness function is to reduce the energy consumed by the network and prolong the network lifetime. The fitness function has also to consider many conditions that could appear during the evolutionary process. That is why we consider a penalty term which helps minimizing the cluster overlapping, as shown in Equation 4.15. We adopted the fitness function introduced in [68]. Thus, we need to minimize the communication distance and provide the optimal radius for each CH.

$$F = E_{dd} + E_{n0} * (0.7n_0 + 0.3n_{\geq 2}) \tag{4.15}$$

$$E_{dd} = \sum_{i=1}^{k} \sum_{j=1}^{n_i} (d^2_{CH(i,j)} + \frac{d^2_{SN(i)}}{n_i}) \tag{4.16}$$

where

- E_{dd} represents the communication distance,
- k is the number of clusters,
- n_i is the number of nodes belonging to CH_i,
- $d_{CH(i,j)}$ is the Euclidean distance between the sensor and its CH,
- $d_{SN(i)}$ is the Euclidean distance between the cluster head and the base station,
- n_0 is the number of nodes that do not belong to any CH,
- $n_{\geq 2}$ is the number of nodes that belong to more than one CH, and
- E_{n0} is sum of square of Euclidean distance of all n_0 nodes to their nearest CH.

4.3.3 Phase 3: Model Evaluation

In this phase, the WSN cluster layout obtained from phase 2 is evaluated as explained in Section 4.1.3.

4.4 Comparing the Proposed Solutions with other Clustering Approaches

In order to evaluate the effectiveness of our proposed solutions, it has to be compared with previous well known benchmark clustering mechanism. LEACH protocol (Section 2.6.1) is a common benchmark clustering protocol used to evaluate previously proposed clustering approaches. Moreover, in order to ensure the effectiveness of using the PSO technique, we have to compare it with another soft computing technique. GA is used to compare with PSO. Thus, the same three proposed approaches are re-implemented using GA instead of PSO. The GA chromosomes used are identical to the PSO particle structures. Also, the same fitness functions are used. The typical three GA approaches implemented are:

- *KGA: Hybrid K-means GA Approach*
- *GA-VC: GA Variable Clustering*
- *KGA-GA: Hybrid K-means GA Clustering Approach*

Chapter Five

Hybrid K-Means and PSO Clustering Approach

In this chapter, our proposed hybrid K-means PSO approach, named KPSO, is evaluated and compared with two clustering approaches. KPSO consists of three phases. The first phase partitions the network into predefined number of clusters using K-means. In the second phase, PSO selects the optimum CH for each cluster. Evaluation of the obtained cluster layout is performed in the last phase. PSO selects the CHs based on our novel fitness function that aims to maximize the number of transmissions a CH performs. Evaluation of the network layout is based on the RF communication model.

KPSO approach was used over number of experiments with various layouts in order to test it efficiently. It is compared with K-means and the LEACH protocol. Also, KPSO is compared with KGA approach, where the same approach is implemented with GA rather than PSO. We explore the effect of varying: the number of nodes, the energy of the nodes, and the base station location. Also, we explore the effect of our developed hybrid model on the WSN lifetime on two conditions: 1) When the first node dies, 2) When 10% of nodes dies.

5.1 Experimental Setup

5.1.1 Assumptions

For proper operation of our proposed approaches, some assumptions were made. We assumed a fixed number of static nodes that are randomly deployed in a two dimensional geographical area. The nodes are assumed to have an initial energy level. The base station has no constraints on its energy and computing resources. It has an updated record of each node's location and energy level. It is assumed that the communication in the network is established. There are no communication problems between the nodes and the base station, and also between the nodes and their CHs. Within each cluster, each pair of sensor nodes is guaranteed to be within the effective transmission range. So each two nodes in the cluster can communicate with each other directly. The energy dissipated due to environmental disturbances (e.g. signal fading, packet loss) is ignored. These assumptions are also adopted for the remaining approaches.

5.1.2 Experimental WSN Data

Table 5.1 lists the values of the WSN parameters adopted in our experiments. We adopted 100 nodes to be randomly generated in a geographic area of $100 \times 100m^2$. Since in the field heterogeneous nodes are used, we adopted nodes that have different energy levels. We chose the base station to be the corner of the geographic area. The dataset is generated by a 'Random Node Generator' implemented in WSN-CAT toolbox (Chapter 8). Figure 5.1 shows the dataset generated by the Random Node Generator. The nodes marked with 'red star' are those having high energy.

Table 5.1: WSN simulation parameters

Parameter	Value
Field Size	$100 \times 100m^2$
Number of Sensor nodes	100
Energy of Sensor nodes	80% have 2J ; 20% have 5J
Base Station location	(0,0)
Size of message	4000 bits

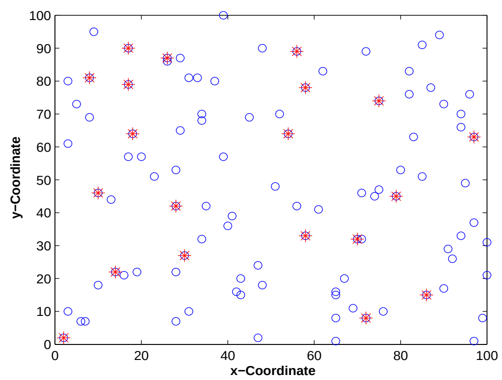


Figure 5.1: 2D Layout of 100 Sensor nodes

5.2 Selecting the Number of Clusters

A pretest was made to discover the number of clusters producing the minimum communication distance. Figure 5.2 shows the total communication distance versus the number of clusters for the data shown in the above table (Table 5.1). The graph showed 9 clusters as the best number to be used.

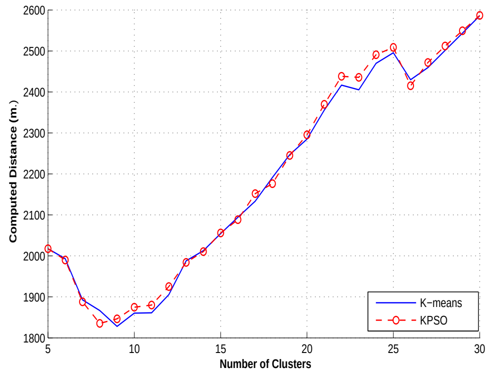


Figure 5.2: Communication distance vs. number of clusters

5.3 Selection of best PSO Equation

A pretest was made to choose the most suitable PSO equation. Figure 5.3 shows the fitness values for the five PSO models. Although both PSO-Inertia and PSO-TVAC models converge to the best fitness value, TVAC model converges faster. Therefore, the PSO-TVAC model is adopted. Table 5.2 lists the parameters of the PSO model used. Figure 5.4 shows the fitness conversion curves with various numbers of particles. In our case, 20, 40, 60 and 80 PSO particles were used, respectively, and the best result is chosen.

Table 5.2: PSO simulation parameters

Parameter	Value
PSO model	TVAC
Inertia weight	[0.1:0.9]
ϕ_1	[0.5:2.5]
ϕ_2	[0.5:2.5]
No. of particles	{20,40,60,80}

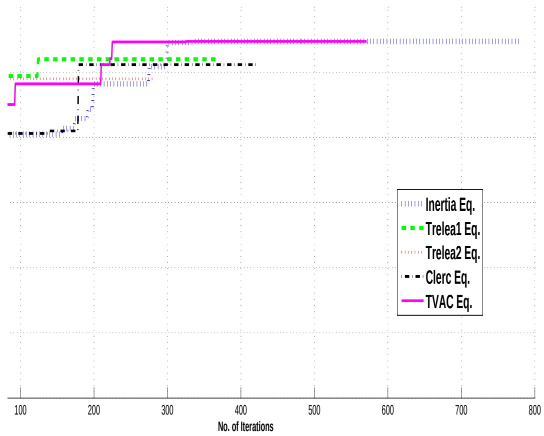


Figure 5.3: Fitness conversion of various PSO models

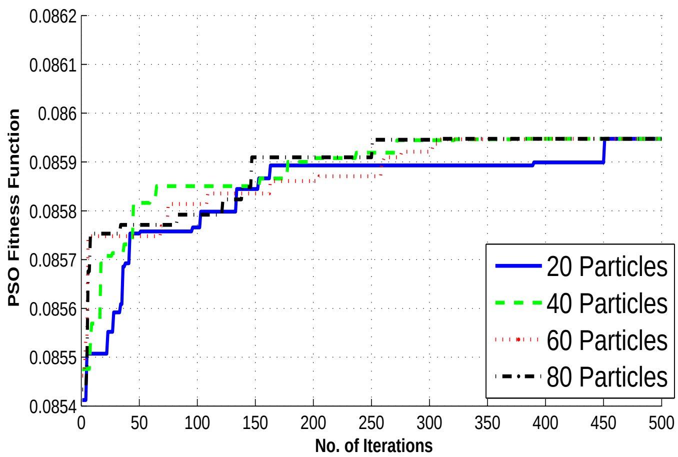


Figure 5.4: PSO fitness function conversion

5.4 Comparison between the Proposed Fitness Function and other Fitness Functions

In order to evaluate our proposed fitness, we explored previously used fitness functions. The proposed fitness is shown in Equation 5.1 (Section 4.1.2). Equations 5.2 and 5.3 were explored and compared with our proposed fitness function, F. F_1 represents the distance fitness; i.e. sum of the distances between member node to CH plus the distance from CH to Sink. F_2 represents the distance square fitness; sum of the square of the distances between member node to CH plus the square of the distance from CH to Sink. Figure 5.5 shows the simulation results of our proposed fitness as well as F_1 and F_2. The graph shows that our proposed fitness produces more number of transmissions than F_1 and F_2 given a number of nodes are alive.

$$F = \begin{cases} \sum_{i=1}^{k} \frac{E_i}{d_i^2} & \text{for } d_i < d_0 \\ \\ \sum_{i=1}^{k} \frac{E_i}{d_i^4} & \text{for } d_i \geq d_0 \end{cases} \tag{5.1}$$

$$F_1 = \sum_i \sum_j d_{CH(i,j)} + \sum_i d_{SN(ij)} \tag{5.2}$$

$$F_2 = \sum_i \sum_j d^2_{CH(i,j)} + \sum_i d^2_{SN(ij)} \tag{5.3}$$

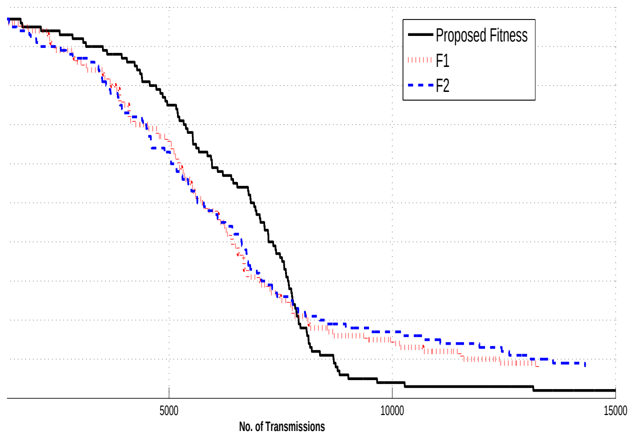


Figure 5.5: Total number of alive nodes vs. number of transmissions

5.5 Developed Cluster Layout

The developed cluster layout of K-means and the layout after applying our KPSO approach are presented in Figure 5.6. Figure 5.7 shows the clusters layout formed by the LEACH protocol. The nodes marked with 'red star' are those having high energy. The LEACH distribution shows higher number of randomly chosen clusters. The cluster head selected from the K-means is the center of the cluster. However, the cluster head selected from our KPSO approach is not necessarily the center of the cluster. Our KPSO approach selects the cluster head that can survive for maximum number of transmissions.

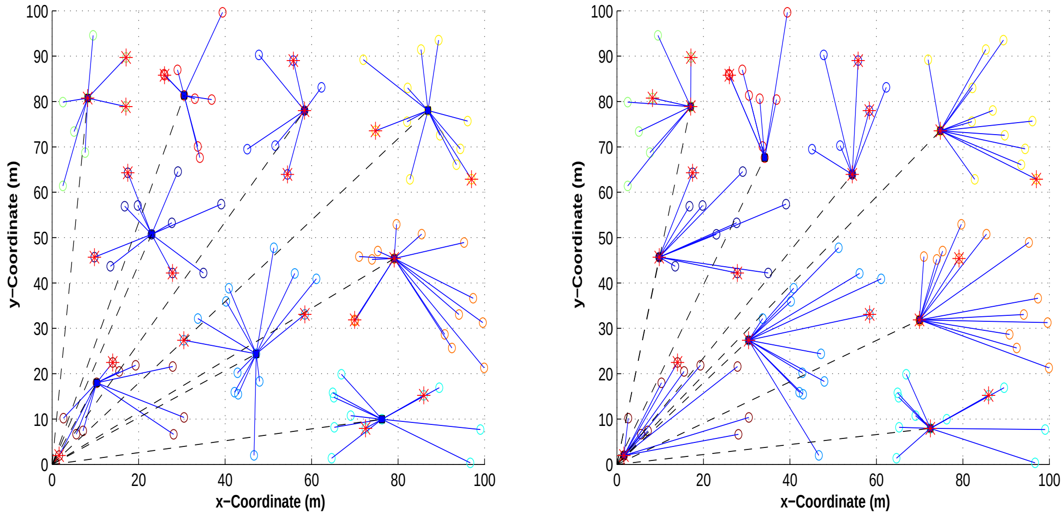


Figure 5.6: (a) K-means clusters (b) KPSO clusters

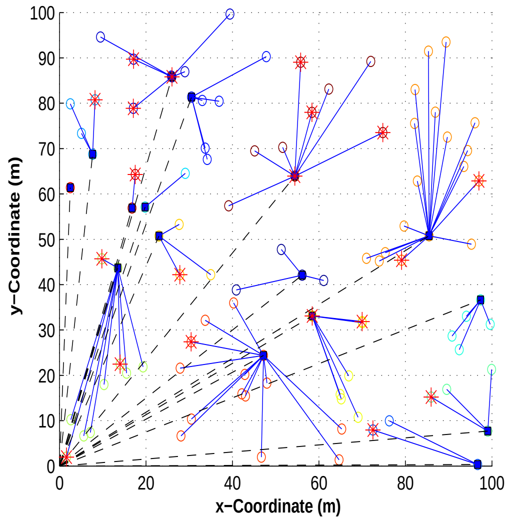


Figure 5.7: LEACH clusters

5.6 Comments on Energy Computation

In this section, we discuss some observations on our developed results. It is essential to increase the network life time by managing the network remaining energy and the total number of alive nodes during the simulation. The total remaining energy based on the three approaches studied in this research (i.e. K-means, LEACH protocol and our KPSO approach) are presented in Figure 5.8(a). Figure 5.8(b) shows the number of alive nodes versus the number of transmissions. We concluded the following facts:

- LEACH protocol consumes the highest energy between the three approaches.
- K-means consumes less energy than our KPSO approach because its CH is the center of the cluster leading to minimum communication distance.
- The CH as presented in our proposed KPSO approach is not always the center of the cluster. Thus, the communication distance is greater or equal to that of the K-means communication distance, in some cases.
- Our proposed approach shows a higher number of alive nodes than the other cases of K-means and LEACH. To show how we made the comparison, we considered a threshold of 30% of the total WSN nodes as a guide for comparison. Above this level, our KPSO approach performs better than the K-means, and LEACH protocol.
- KPSO approach has a longer life time compared to both LEACH protocol and the K-means. The reason is our adopted fitness has the advantages of considering both the node's actual energy and the communication distance at the same time.

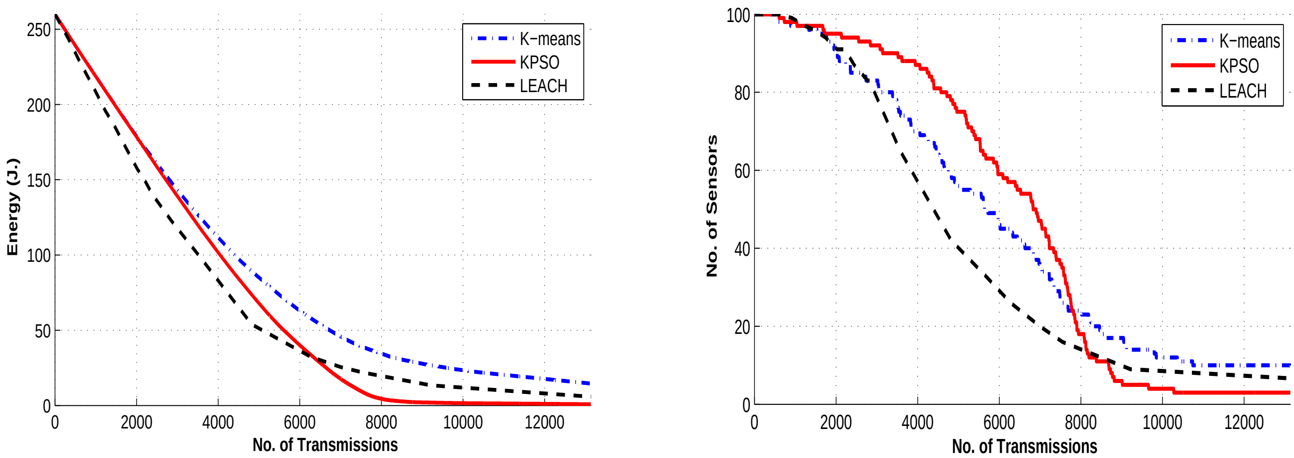


Figure 5.8: (a)Total remaining energy vs. number of transmissions (b)Total number of alive sensor nodes vs. number of transmissions

Table 5.3 shows the simulated number of transmissions performed by the network given a fixed number of nodes are alive and the percentage improvement of KPSO over LEACH and K-means. The percentage improvement over LEACH is calculated as follows:

$$Improvement = \frac{KPSO_{Transmissions} - LEACH_{Transmissions}}{LEACH_{Transmissions}} * 100\% \quad (5.4)$$

Table 5.3: Simulated number of transmissions

Number of Alive Sensor nodes	Simulated transmissions			KPSO Improvement over	
	LEACH	K-means	KPSO	LEACH	K-means
100	425	555	604	42.20%	8.83%
90	2582	2251	3518	36.23%	56.29%
80	3422	3223	4708	37.57%	46.08%
70	4006	4606	5350	33.54%	16.15%
60	4620	5264	5955	28.90%	13.13%
50	5235	5720	6897	31.73%	20.58%
40	5901	6623	7193	21.90%	8.61%
30	6660	7675	7550	13.37%	-1.63%
20	7888	8920	8084	2.48%	-9.37%
10	10071	10726	8903	-11.59%	-17.00%
Average KPSO Percentage Improvement over				23.63%	14.17%

5.7 Computing WSN Lifetime

WSN lifetime is defined as the time at which the first sensor dies [13]. It was also defined [14] as the time at which a fraction of nodes die. In this section, we explore the effect of our developed hybrid model on the WSN lifetime on two conditions:

1. When the first node dies,

2. When 10% of nodes dies.

Figure 5.9 shows the simulated number of transmissions for the cases mentioned above. KPSO improved the WSN lifetime than LEACH protocol and K-means in both cases.

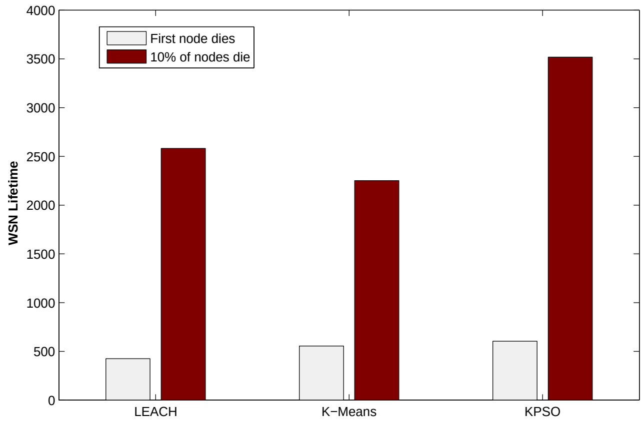
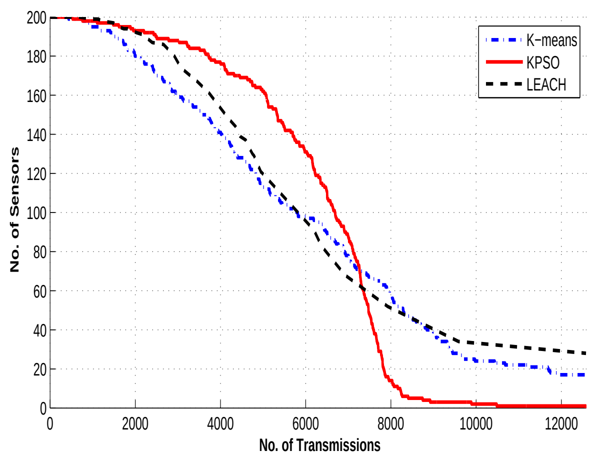


Figure 5.9: WSN lifetime

5.8 Measuring the Effect of Variation of the Number of Sensor Nodes

Network layouts with different number of sensors are examined to evaluate our KPSO approach. Two network layouts explored in our study: 200 nodes WSN and 400 nodes WSN. The base station location was arbitrary fixed at point (0,0). The two networks are assumed to be in the same geographic area of $100m \times 100m$. We assumed that 80% of the nodes are having 2J energy, while the rest of nodes are having 5J energy. Figure 5.10 shows the number of alive nodes for the two network layouts having total number of nodes equals 200, and 400, respectively.



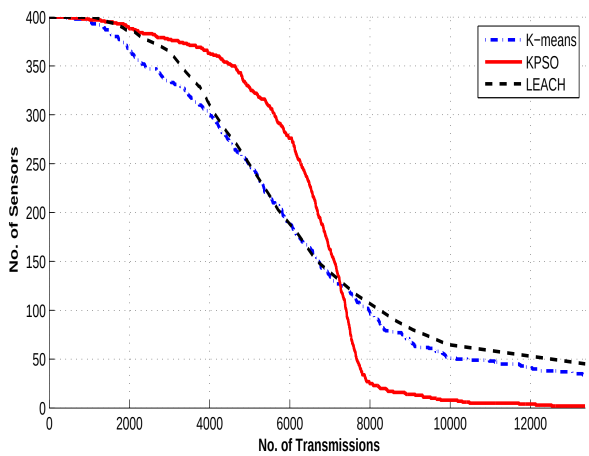


Figure 5.10: Total number of alive nodes for (a) 200 nodes-WSN (b) 400 nodes-WSN

The results show that our KPSO approach performs more transmissions than LEACH protocol when the number of alive nodes in the network is more than 30% of the total nodes in the network. When the number of alive nodes fall below 30% of the total number of nodes in the network, KPSO nodes deplete faster than LEACH and K-means. However, the data gathered with only 30% of the of the total nodes does not give detailed information on the region. Table 5.4 lists the average improvement in the KPSO performance compared to the LEACH approach. The performance improvement of KPSO over K-means is almost not affected by the number of sensors.

Table 5.4: Average performance improvement

No. of Sensor nodes	Average improvement in the No. of transmissions (%)	
	Improvement over K-means	Improvement over LEACH
100	14.17 %	23.63 %
200	20.25 %	9.98 %
400	19.58 %	9.07 %

5.9 Measuring the Effect of Variation of Energy of Sensor Nodes

Varying the node's energy is always essential for WSN performance and it affects the network life time. That is why, we decided to explore the effect of having sensors with various energy distributions on the three approaches. We changed the energy of the randomly generated nodes. Three situations were considered. They are:

- Case 1: all sensor nodes have the same energy of 2J.
- Case 2: 80% of the sensor nodes has 2J and 20% of sensor nodes have 5J.
- Case 3: 50% of the sensor nodes has 2J and the other 50% have 5J.

Figure 5.11 shows the alive nodes graph for cases 1 and 3 respectively, while case 2 is already discussed in Figure 5.8. Studying Case 1 and Case 3, our proposed approach performs slightly better than the other two approaches. In Case 2, our KPSO approach outperforms both the K-means, and LEACH protocol. This is more likely to be the case in practice. There is no guarantee that all the sensors shall have the same energy distribution in the field. Figure 5.12 shows the WSN lifetime for the layouts of cases 1 and 3. The lists of the average improvement of KPSO compared to both the K-means, and LEACH protocol is presented in Table 5.5.

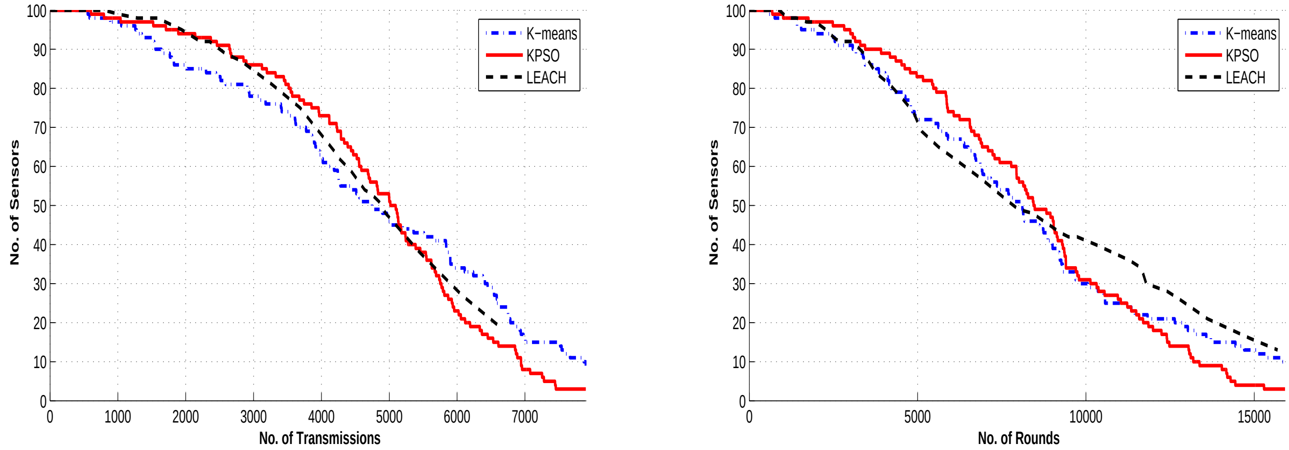


Figure 5.11: Total number of alive sensors with energies of (a) 2J each (b) 50% with 2J and 50% with 5J.

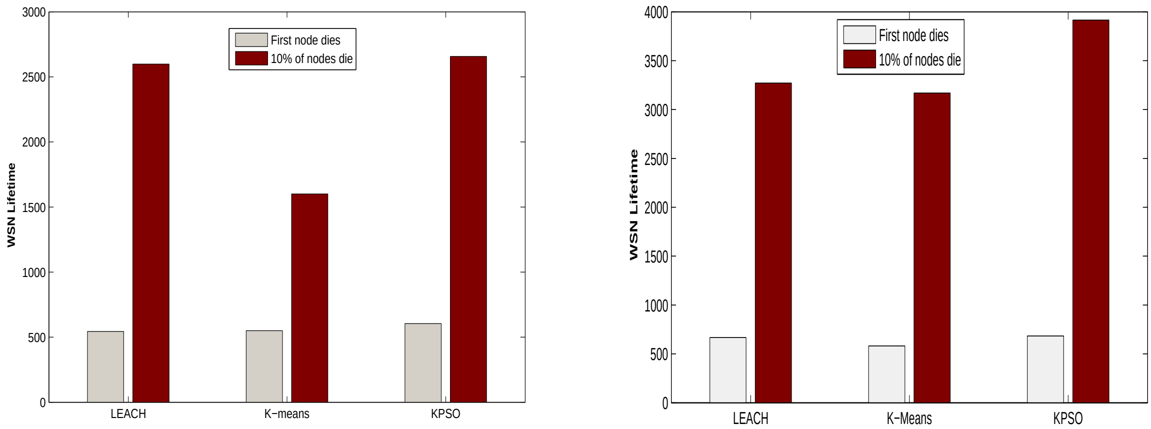


Figure 5.12: WSN lifetime (a) all nodes have 2J (b) 50% of nodes have 2J and 50% 5J

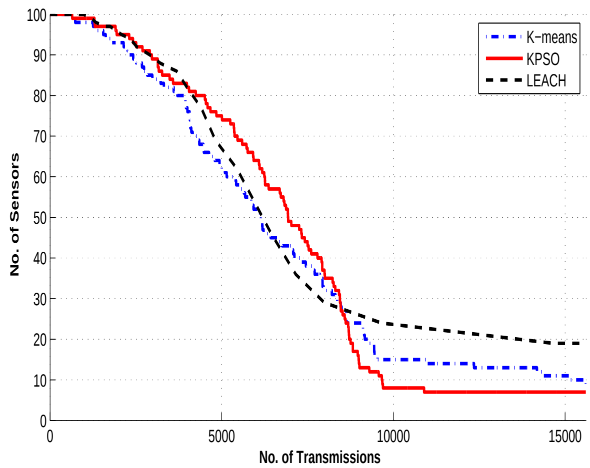
5.10 Measuring the Effect of Variation of the Base Station Location

Of course, the location of the base station is an essential element which can affect the WSN operation lifetime. To study its effect, we have arbitrary chosen two different locations for a base station simulation. We chose one location to be within the field, and the other far from the field. The locations are the point (50,50) and the point (50,175) in the work environment.

Table 5.5: Average performance improvement

Energy of Sensor nodes	Average improvement in the No. of transmissions (%)	
	Improvement over K-means	Improvement over LEACH
2J each	8.0251 %	5.26 %
80% have 2J and 20% have 5J.	17.1305 %	23.63 %
50% have 2J and 50% have 5J.	9.0486 %	6.68 %

The performance is simulated, as shown in Figure 5.13. The KPSO approach performs more transmissions than the other two approaches when the number of alive nodes is greater than 40% of the total network nodes. The results show that our KPSO approach performs better than K-means algorithm and LEACH protocol. Our KPSO approach was able to select the CHs that performs more number of transmissions based on the fitness criteria; not based on the random process as in LEACH protocol. Figure 5.14 shows that KPSO prolongs the lifetime of the network. Table 5.6 lists the average improvement in the KPSO performance compared with both the K-means algorithm, and LEACH protocol.



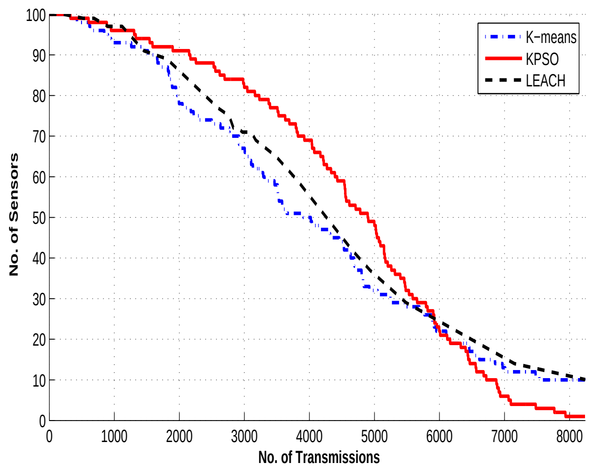


Figure 5.13: Total number of alive sensor nodes when the base station located at point (a) (50,50) (b) (50,175) in the environment

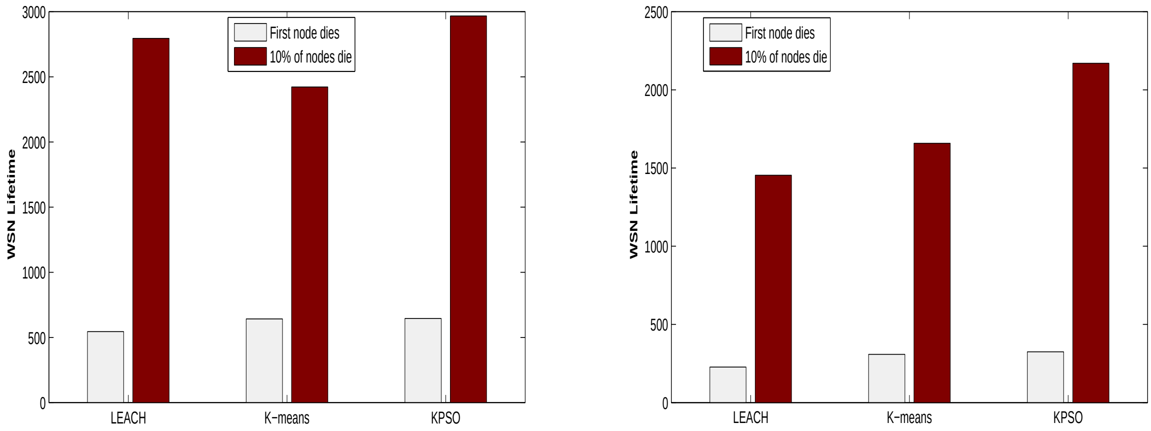


Figure 5.14: WSN lifetime for 100 nodes-WSN with base station location at(a) (50,50) (b) (50,175)

Table 5.6: Average performance improvement

Base Station Location	Average improvement in the No. of transmissions (%)	
	Improvement over K-means	Improvement over LEACH
(0,0)	14.17%	23.63%
(50,50)	5.93%	4.63%
(50,175)	19.04%	28.67%

5.11 KPSO Results versus KGA Results

In this section, we evaluate using PSO in our proposed approach. The hybrid K-means GA approach ”KGA” results are recorded and compared with KPSO. The results of applying our hybrid approaches for WSN Layout stated in Table 5.1 are discussed. The following subsections explore the effect of varying: 1) the number of nodes 2) the node’s energy and 3) the base station location.

Figure 5.15 shows the GA and PSO fitness conversion. The graph shows that PSO converges to higher (i.e. better) fitness than GA, with higher number of generations. In Figure 5.16, we show the developed cluster layout for the by the KGA approach. The cluster head selected by KGA approach is, like KPSO, not necessarily the center of the cluster. KGA adopted the same proposed fitness that selects the cluster head that can survive for maximum number of transmissions.

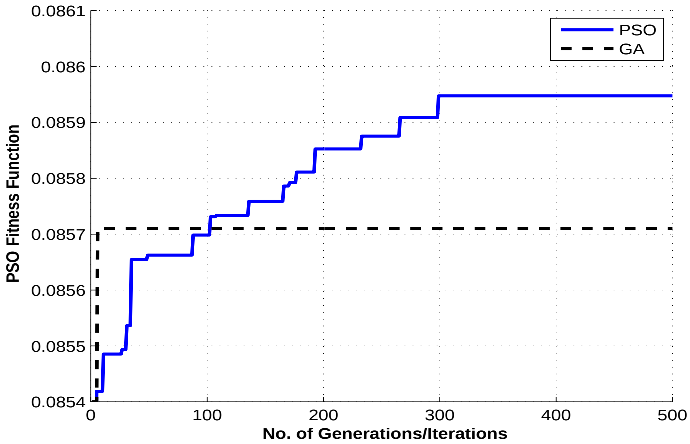


Figure 5.15: PSO and GA fitness conversion graphs

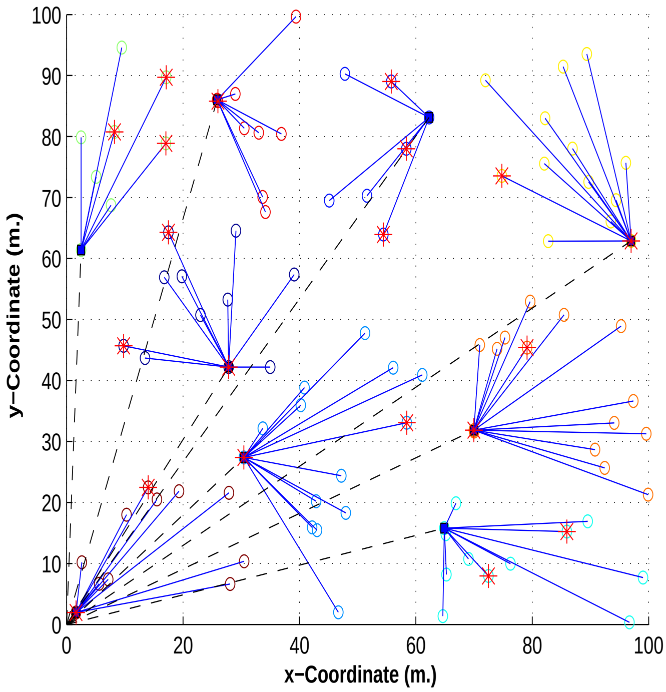


Figure 5.16: KGA clusters Layout

The total remaining energy based on the two approaches studied in this chapter (i.e. KPSO and KGA approaches) are presented in Figure 5.17. Figure 5.18 shows the number of alive nodes versus the number of transmissions. Both KPSO and KGA approaches preserve the same total energy during the first half of the network lifetime. In the second half, KPSO consumes more energy than KGA. However, KPSO performs more transmissions than KGA.

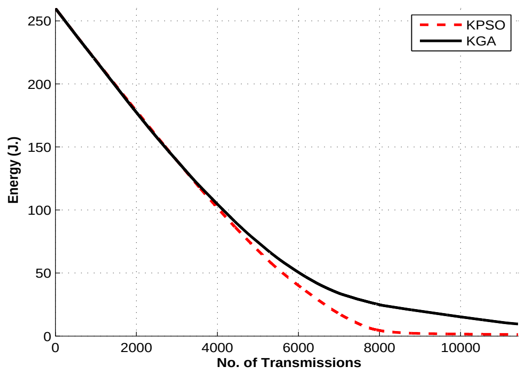


Figure 5.17: Total remaining energy vs. number of transmissions

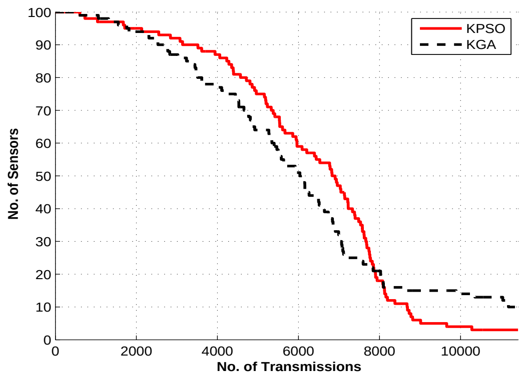


Figure 5.18: Total number of alive sensor nodes vs. number of transmissions

Figure 5.19 shows the WSN lifetime when the first node dies and when 10% of the nodes die. The simulated results show that KPSO performs more transmissions than KGA. For example, if we consider the network lifetime as when 10% of the sensors die: the number of KPSO transmissions is more than KGA transmissions by about of 47%; thus the WSN lifetime using KPSO is more than WSN lifetime using KGA. The calculated the average percentage improvement of KPSO over KGA was 8.46%.

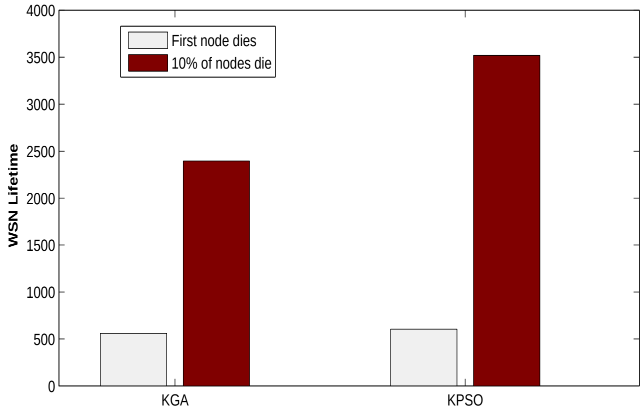


Figure 5.19: WSN lifetime

5.11.1 Measuring the Effect of Variation of the Number of Nodes

The two network layouts examined in Section 5.8 are used to evaluate our KGA and compare with KPSO approaches. The two networks use the same simulation parameters of Table 5.1, except for the number of sensors. Figure 5.20 shows the number of alive nodes for the two network layouts having total number of nodes equals 200, and 400, respectively.

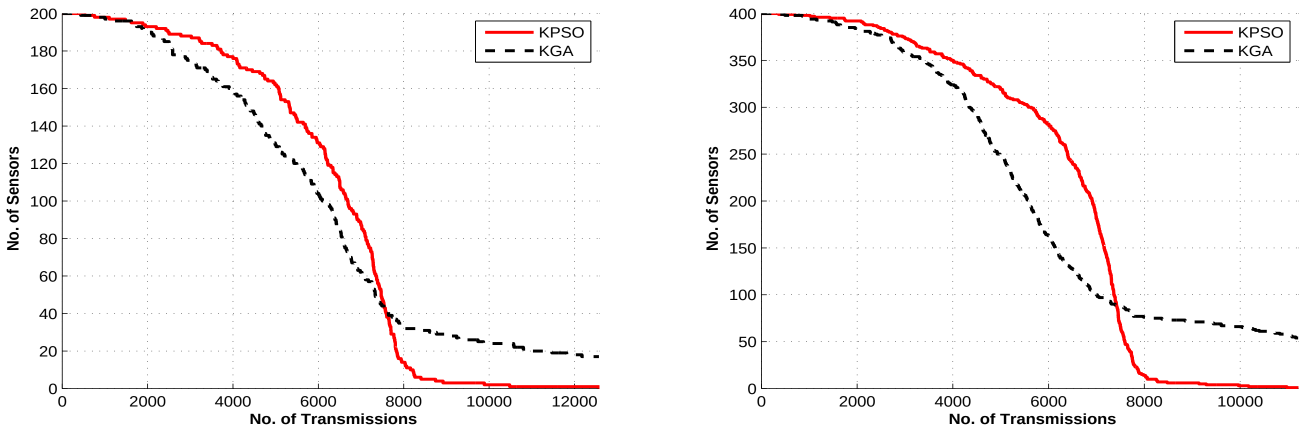


Figure 5.20: Total number of alive nodes for (a) 200 nodes-WSN (b) 400 nodes-WSN

Table 5.7 lists the average improvement in the KPSO performance compared to the KGA approach. The results show that KPSO approach achieved more transmissions than KGA when the number of alive nodes in the network is more than 20% of the total nodes in the network. When more than 80% of the total nodes depletes their energy, the KGA approach performs more transmissions than KPSO. However, the data gathered from the network in this case does not give full information about the region of interest. Figure 5.21 show that using PSO instead of GA resulted in more WSN lifetime.

Table 5.7: Average performance improvement

No. of Sensor nodes	Improvement over KGA
100	8.46 %
200	11.6282 %
400	11.3039 %

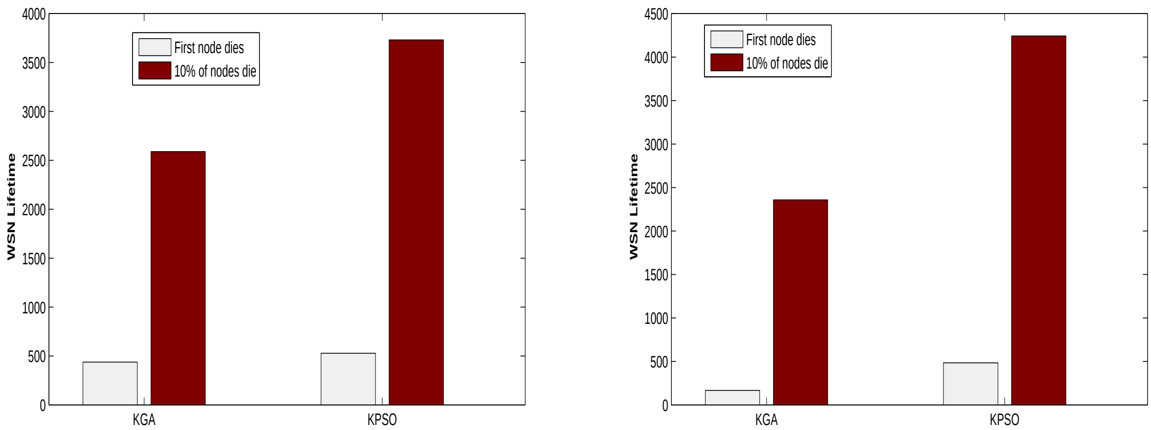


Figure 5.21: WSN lifetime for (a) 200 node-WSN (b) 400 node-WSN

5.11.2 Measuring the Effect of Variation of the Nodes Energy

In this subsection, we explore the effect of having nodes with various energy distributions on KGA approach and compare it with the results of KPSO approach. Specifically, we will explore the three cases that were explored with KPSO in Section 5.9. Figure 5.22 shows the performance graph for cases 1 and 3, case 2 is already shown in Figure 5.18. The lists of the average improvement of KPSO over KGA are presented in Table 5.8. When the nodes have the same energy, both KPSO and KGA approach perform nearly similar. However, with varying energy, which is the practical situation, KPSO provide better performance. Figure 5.23 shows the lifetime of the WSN for cases 1 and 3. Thus using KPSO resulted in more lifetime than using KGA.

Table 5.8: Performance Improvement of KPSO over KGA

Energy of Sensor Nodes	Improvement over KGA
2J each	2.8232%
80% have 2J and 20% have 5J.	8.46%
50% have 2J and 50% have 5J.	10.367%

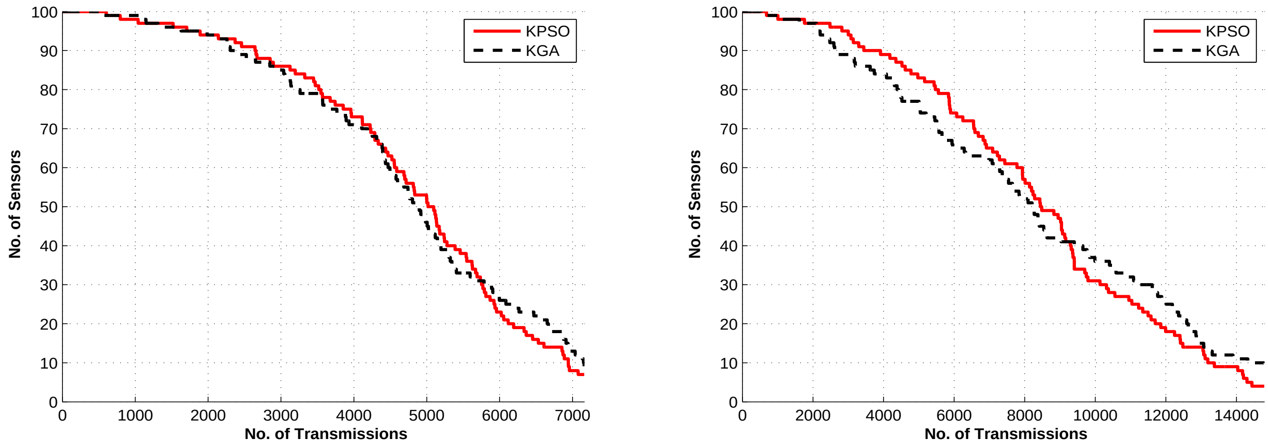


Figure 5.22: Total number of alive sensors with energies of (a) 2J each. (b) 50% with 2J and 50% with 5J.

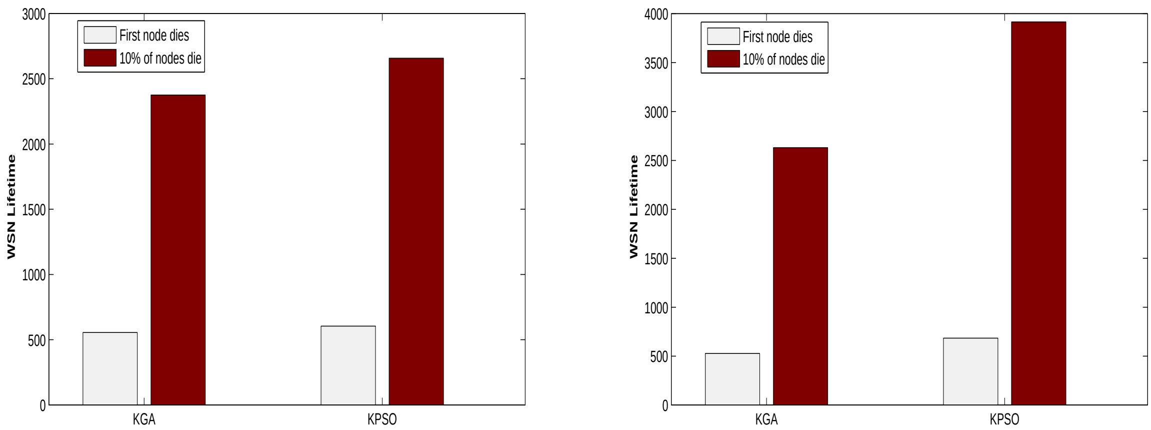


Figure 5.23: WSN lifetime (a) all nodes have 2J (b) 50% of nodes have 2J and 50% have 5J

5.11.3 Measuring the Effect of Variation of the Base Station Location

KGA approach results are recorded when the base station is at (50,50) and the point (50,175) in the work environment. The results are compared with that of KPSO approach. Figure 5.24 shows the simulated number of alive nodes for both base station locations. Figure 5.25 shows the WSN lifetime for both approaches. KPSO lifetime is slightly better than KGA when the first node dies. When 10% of the nodes die, KPSO lifetime increased by 6% at base station (50,50) and 31.8% when the base station is located at (50,175).The graphs show that the KPSO approach performs better than KGA especially when the base station is located away from the geographical area. Table 5.9 lists the average improvement in the KPSO performance compared with KGA approach.

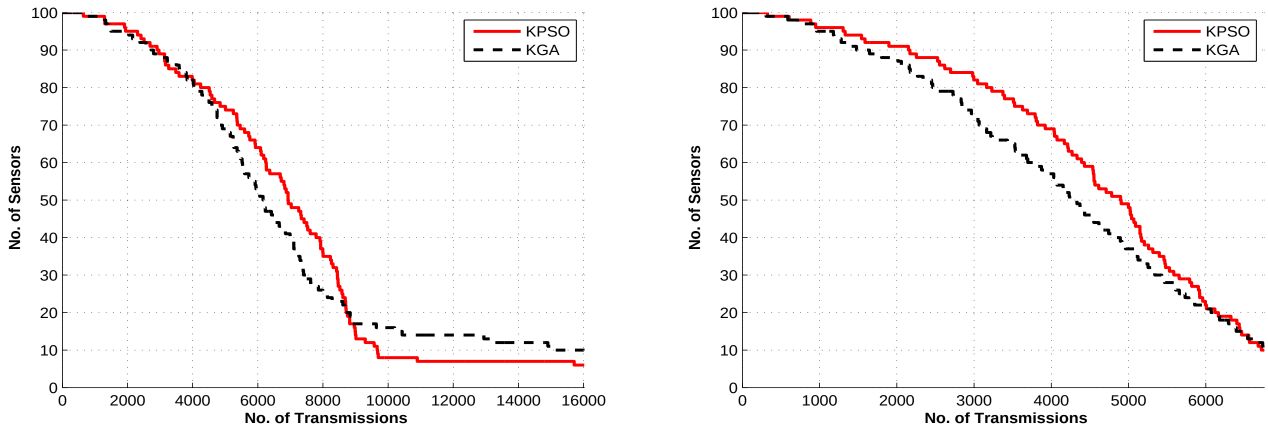


Figure 5.24: Total number of alive sensor nodes when the base station located at point (a) (50,50) (b) (50,175) in the environment

Table 5.9: Average performance improvement

Base Station Location	Improvement over KGA
(0,0)	8.46%
(50,50)	3.59%
(50,175)	13.42%

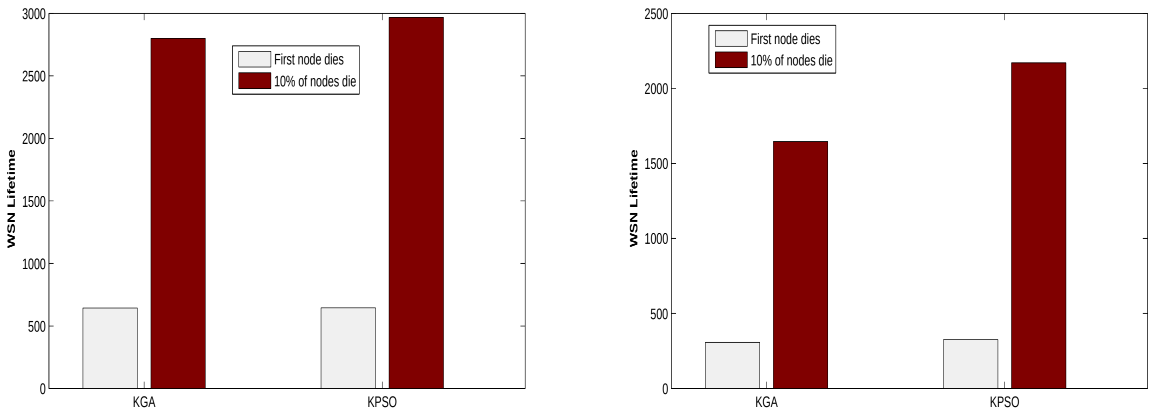


Figure 5.25: WSN lifetime (a) all nodes have 2J (b) 50% of nodes have 2J and 50% have 5J

Chapter Six

Clustering Based Variable Structure PSO

PSO Variable Clustering approach, named PSO-VC, develops a clustering approach using PSO. PSO-VC outputs the optimum number of clusters, optimal CHs and optimal members for each cluster. The PSO particle structure contains the number of clusters followed by CHs ID for each cluster. Therefore, PSO particle size is not fixed. Its size is always varying from PSO iteration to another depending on the number of clusters selected by PSO particles. A novel fitness function is adopted that aims to maximize the number of transmissions a CH performs during its lifetime.

In this chapter, our second proposed approach, PSO-VC, is evaluated and compared with traditional LEACH protocol. Moreover, we re-implemented this approach but using GA instead of PSO, named GA-VC (GA Variable Clustering). Finally, PSO variable Clustering is compared with our first approach KPSO.

6.1 Selection of best PSO Equation

A pretest was made to choose the most suitable PSO equation for our simulation. Figure 6.1 shows the fitness values after applying the PSO models. The graph shows that $PSO - Inertia$ equation converges fastest (with convergence value of 0.8990). The PSO-Trelea1 equation converges best (with convergence value of 0.8997), but at the expense of convergence time. Figure 6.2 shows that the PSO-Inertia and PSO-Trelea-1 equations produce results almost with the same performance. Therefore, the PSO-Inertia equation is adopted since it produces fast convergence and almost best performance results. Table 6.1 shows the values of PSO parameters adopted in our experiments.

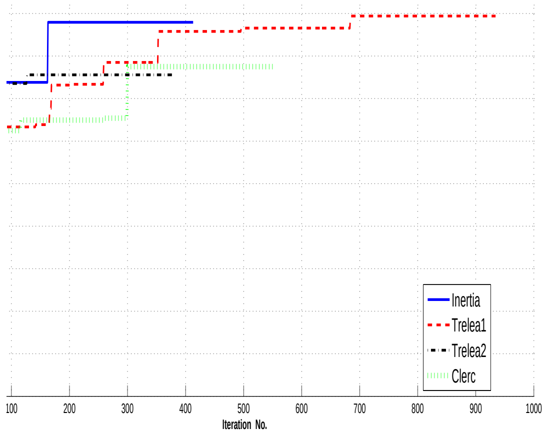


Figure 6.1: PSO Conversion with different PSO models

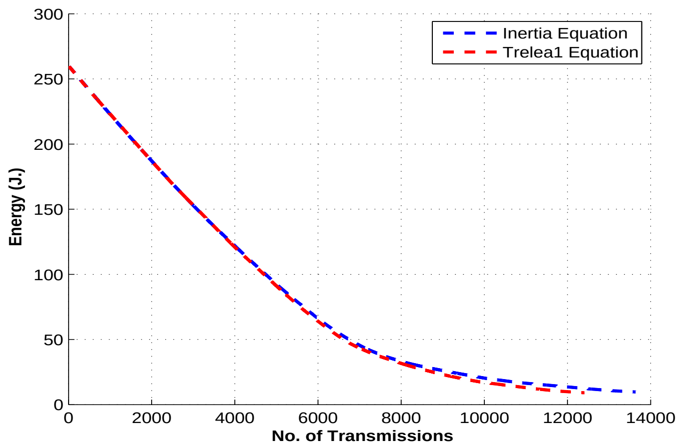


Figure 6.2: Total Remaining Energy using different PSO Equations

Table 6.1: PSO simulation parameters

Parameter	Value
Inertia weight	from 0.9 to 0.4
No. of particles	20, 40, 60, and 80

6.2 Comparison between the Proposed Fitness Function and other Fitness Functions

In this study we explored number of fitness functions to improve our results. In Equations 6.1 and 6.2, we introduce two other fitness function which shall be explored in the results section. F_1 represents the sum of the square of the distances between member node to CH plus the distance from CH to base station [1]. F_2 represents the sum of the square of the distances between member node to CH plus the distance between CH to sink divided by the number its members [68]. Our Novel Fitness adopted is shown in Equations 6.3 and 6.4 (Section 4.2.1).

$$F_1 = \sum_{i}^{k} \sum_{j}^{q} d^2_{CH(i,j)} + \sum_{i}^{l} d^2_{SN(i)} \tag{6.1}$$

$$F_2 = \sum_{i=1}^{k} (\sum_{j=1}^{q} d^2_{CH(i,j)} + \frac{d^2_{SN(i)}}{n_i}) \tag{6.2}$$

$$F = \sum_{i=1}^{k} \frac{E_i}{\sum_{j=1}^{q} d^p_{CH(i,j)} + \frac{d^p_{SN(i)}}{n_i}} \tag{6.3}$$

$$p = \begin{cases} 2 & \text{for } d < d_0 \\ \\ 4 & \text{for } d \geq d_0 \end{cases} \tag{6.4}$$

Figure 6.3(a) shows the total energy for the fitness functions over various transmissions, and Figure 6.3(b) shows the total alive nodes over the number of transmissions. The graphs show that our fitness is superior over F_1 and F_2. The reason is our proposed fitness considered 3 attributes: 1) node's residual energy, 2) dissipated energy, and 3) number of member nodes within each cluster. F_1 included only the communication distance, and ignored both the CH energy and number of member per cluster. F_2 included the communication distance and the number of nodes within each cluster, and excluded the CH energy from its calculation. Excluding the CH's energy from fitness leads PSO to select a CH with low residual energy and thus the CH depletes fast leading to poor WSN lifetime. The dissipated energy is an important attribute that affects the clustering organization. Finally, the third attribute helps in obtaining balanced cluster density and uniform cluster distribution.

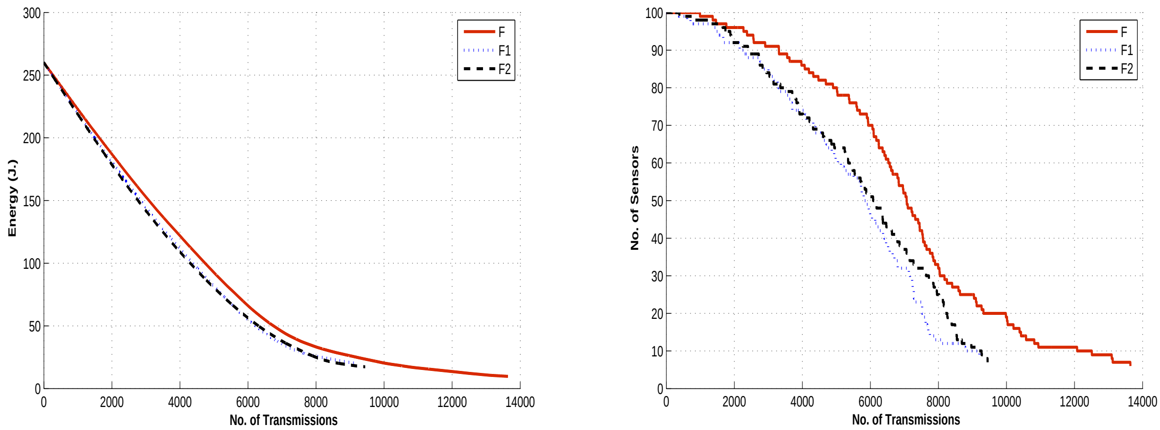


Figure 6.3: Comparison of different fitness functions (a) Total remaining energy (b) Total number of alive nodes: for 100-nodes with variable energy

6.2.1 Developed Cluster Layout

In this section we will explore and discuss the simulation results of the randomly generated WSN as specified in Table 5.1. The proposed PSO approach runs several times with different number of particles (20, 40, 60 and 80) and the best results were chosen. Figure 6.4 shows a sample of fitness value for different number of particles. Figure 6.5 shows the cluster layout after applying our proposed approach. The layout resulted in more clusters than KPSO clusters. Some nodes that are close to the base station send to it directly; they are considered as one cluster.

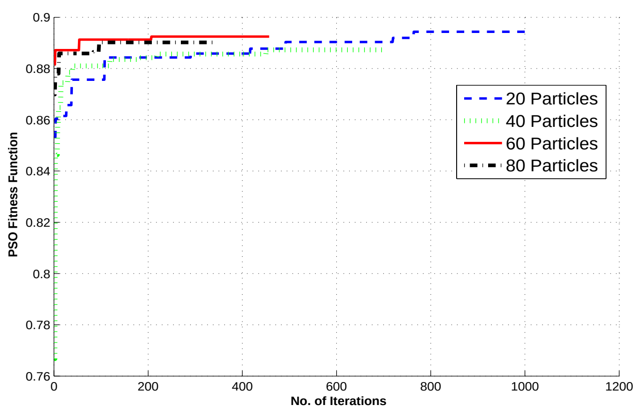


Figure 6.4: PSO fitness conversion with various numbers of particles

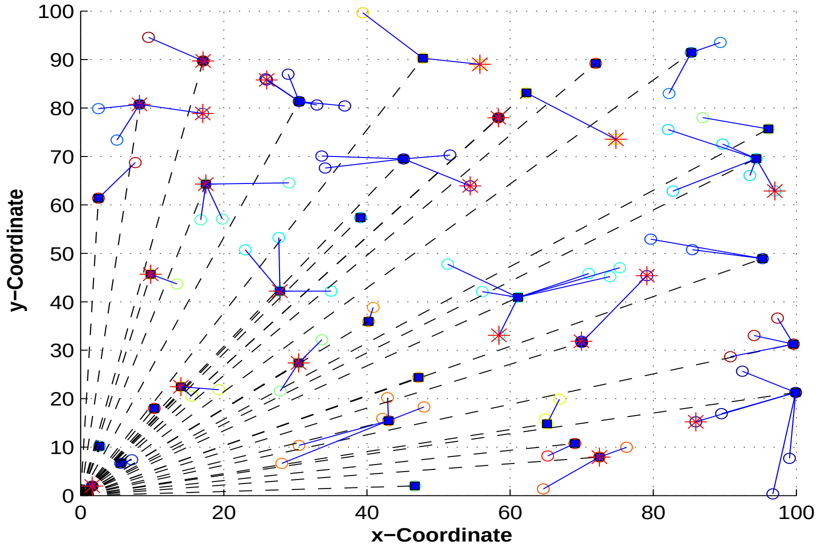


Figure 6.5: Developed cluster layout based our PSO approach

Figures 6.6 (a) and (b) show the total number of alive nodes, and the total remaining energy vs. the number of transmissions. PSO-VC approach proved preserving higher energy than LEACH protocol. PSO-VC also preserved more energy than GA-VC approach when more than 30% of the total nodes are alive. Below this threshold the nodes of PSO-VC loose energy faster than GA-VC. PSO-VC alive nodes outperformed LEACH protocol during the whole operation of the network. Above the threshold of 30% of the total alive nodes, the nodes of PSO-VC performed more number of transmissions than GA-VC. Figure 6.7 shows the WSN lifetime for the two cases studied in KPSO. PSO-VC improved WSN lifetime than LEACH and GA-VC. PSO-VC doubled the LEACH-lifetime when the first node dies. When 10% of nodes die, PSO-VC increased the WSN lifetime by about 37%. PSO-VC slightly increased the lifetime over GA-VC by about 4%-10%.

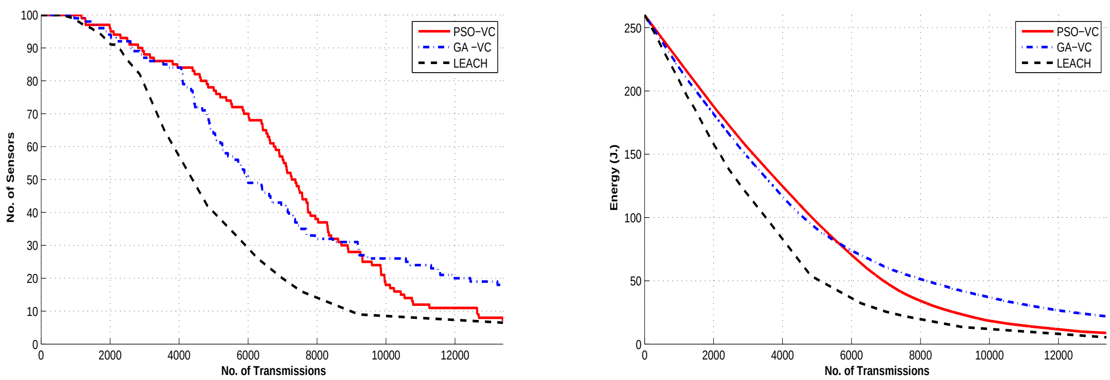


Figure 6.6: (a) Total number of alive nodes (b) Total remaining energy: for 100-nodes with variable energy

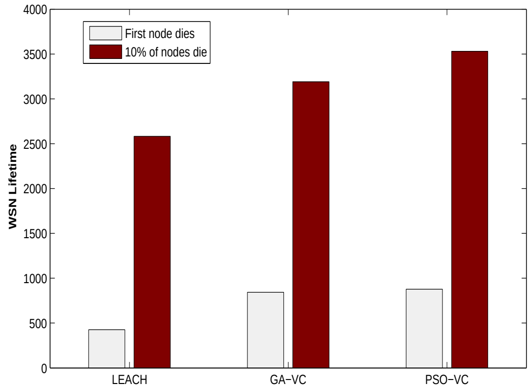


Figure 6.7: WSN Lifetime for 100 Nodes with variable energy

6.2.2 Measuring the Effect of Variation of the Number of Sensor Nodes

We examined the PSO-VC approach on the network layouts of sizes 50, and 200 randomly generated nodes with variable energy. The base station is fixed at the corner of the geographic area; (0,0). Figure 6.8 showed that PSO-VC performed more number of simulated transmissions given a fixed percent of nodes are alive. Thus, PSO-VC increased the network lifetime. Table 6.2 shows the average improvement of PSO-VC over LEACH and GA-VC.

Table 6.2: Average performance improvement

WSN data			PSO-VC Improvement over	
No. of nodes	Energy	Base Station	LEACH	GA-VC
50	80% 2J ; 20% 5J	(0,0)	37.62 %	8.65 %
200	80% 2J ; 20% 5J	(0,0)	31.64 %	9.33 %

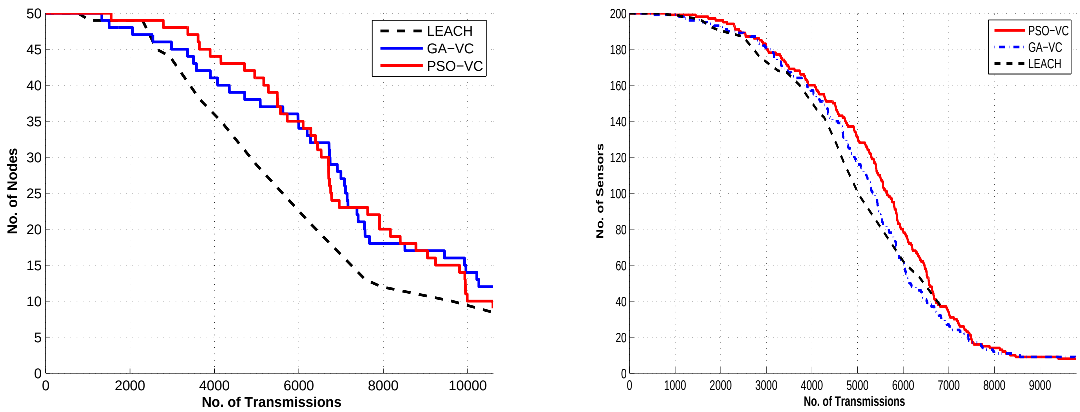


Figure 6.8: Total number of alive nodes vs. number of transmissions for (a) 50 nodes-WSN (b) 200 nodes-WSN

6.2.3 Measuring the Effect of Variation of Energy of Sensor Nodes

To explore the effect of nodes' energy on the performance of our PSO-VC approach, we considered the cases mentioned in Section 5.9. Case 2 has already been reported in Figures 6.6 and 6.7. Figure 6.9(a) shows the total number of alive nodes vs. the simulated transmissions when equalizing the initial energy of the 100 node WSN to 2J each, while Figure 6.9(b) shows the number of alive nodes when half nodes with 2J and the second half with 5J. PSO-VC outperformed LEACH in both cases when the alive nodes are above a threshold of 20%, and below this threshold the LEACH nodes survived for more transmissions. PSO-VC alive nodes are more than GA-VC nodes until a threshold of 30%. Figure 6.10 reports the WSN lifetime for the two network layouts. PSO-VC improved WSN lifetime than LEACH by 77-82%, while slightly improved over GA-VC. When 10% of the nodes die, PSO-VC improvement over LEACH exceeded 25% for layout with same energy and 40% for layout with varying energy. PSO-VC resulted in prolonging lifetime when compared with GA-VC.

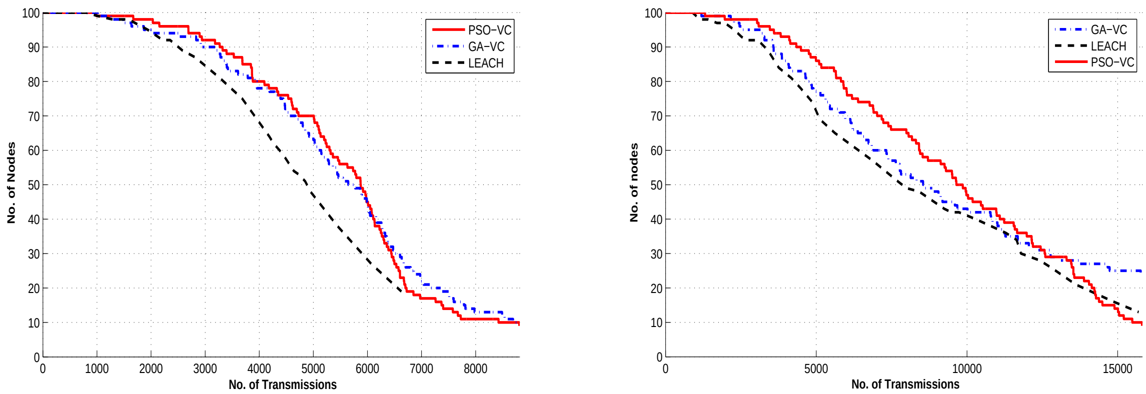


Figure 6.9: Total number of alive nodes vs. number of transmissions for 100 node WSN with energy (a) 2J each (b) 50% 2J and 50%5J

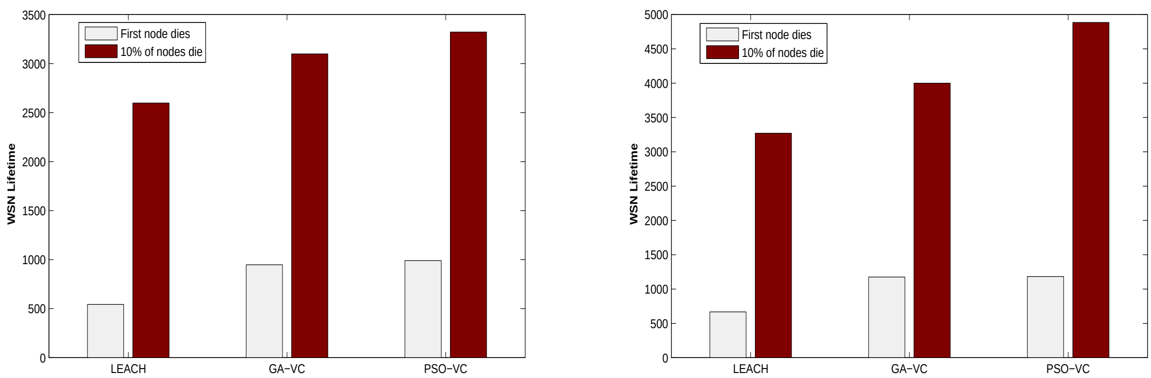


Figure 6.10: WSN Lifetime for 100 Nodes with (a) 2J each (b) 50% 2J and 50% 5J

6.2.4 Measuring the Effect of Variation of the Base Station Location

In this section, the effect of changing the base station location of the WSN lifetime is studied on two locations: (50,50) and (50,175). Figure 6.11(a) shows the total number of alive nodes vs. the simulated number of transmissions for base station at (50,50), while Figure 6.11(b) shows the results for (50,175). The graphs show that PSO-VC results in more simulated number of transmissions than LEACH protocol, and thereby better WSN lifetime. PSO-VC also provides better performance than GA-VC especially when base station is not within the nodes region. Figure 6.12 shows the WSN lifetime. PSO-VC considerably increased the WSN lifetime than LEACH and GA-VC.

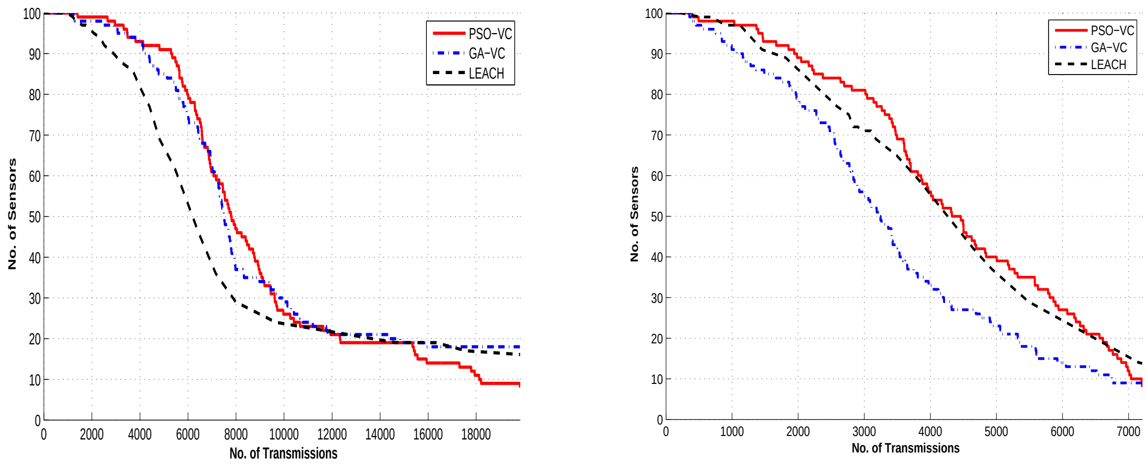


Figure 6.11: Total number of alive nodes vs. number of transmissions for 100 node WSN with base station located at (a) (50,50) (b) (50,175)

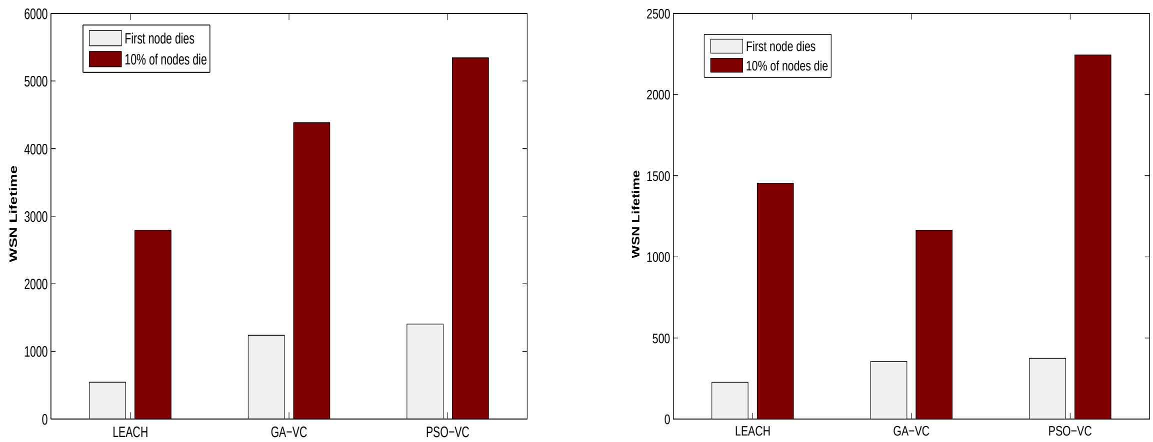


Figure 6.12: WSN Lifetime for 100 Nodes at base station of (a) (50,50) (b) (50,175)

6.3 PSO-VC versus KPSO

In this section we compare between the results of the first proposed approach (KPSO), and this second proposed approach (PSO-VC). We explain the results w.r.t energy and lifetime. Figure 6.13(a) shows the total remaining energy of the 100-node WSN having 80% 2J and 20% 5J. The base station is located at the corner (0,0). Figure 6.13(b) shows the number of alive nodes versus the simulated number of transmissions.

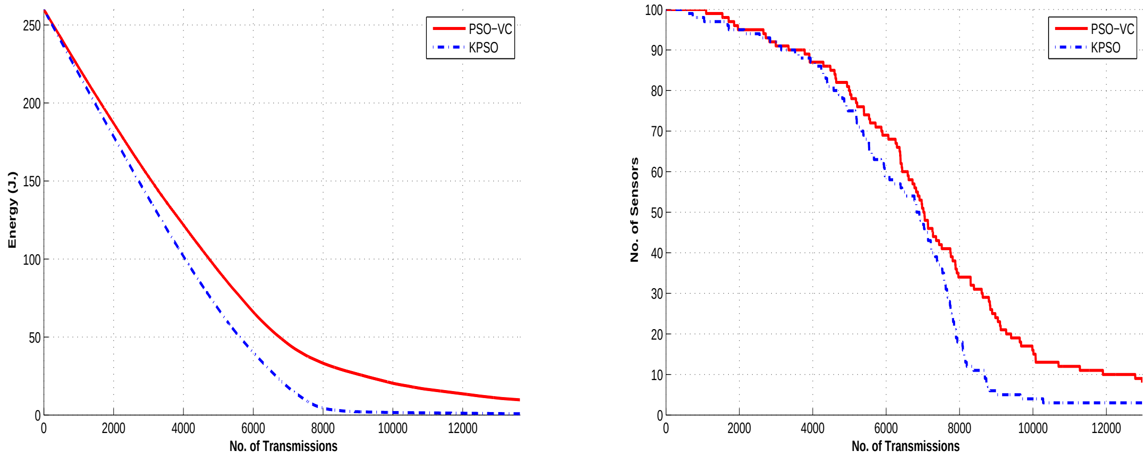


Figure 6.13: (a) Total remaining energy, and (b) Total number of alive nodes vs. number of transmissions for 100 node WSN with varying energy

PSO-VC, has the flexibility to search for optimal number of clusters and CHs for each cluster based on the CH's residual energy and estimated consumed energy. This resulted in more clusters with considerable less member nodes than KPSO. The graphs show that PSO-VC saved more energy than KPSO approach. PSO-VC resulted in more number of alive nodes given a fixed time. This means that the nodes survived for more number of transmissions and thus WSN lifetime is increased.

Figure 6.14 shows the WSN lifetime values of KPSO and PSO-VC when the first node dies, and Figure 6.15 shows the WSN lifetime values when 10% of nodes die. The graphs proved that PSO-VC resulted in more WSN lifetime. WSN lifetime is reported for the following WSN setup:

- **100:** 100 nodes, 80% 2J and 20% 5J, BS (0,0)
- **2J:** 100 nodes, 2J , BS (0,0)
- **50% 5J:** 100 nodes, 50% 2J and 50% 5J, BS (0,0)
- **(50,50):** 100 nodes, 80% 2J and 20% 5J, BS (50,50)

- **(50,175):** 100 nodes, 80% 2J and 20% 5J, BS (50,175)
- **200:** 200 nodes, 80% 2J and 20% 5J, BS (0,0)
- **200-2J:** 200 nodes, 2J, BS (0,0)

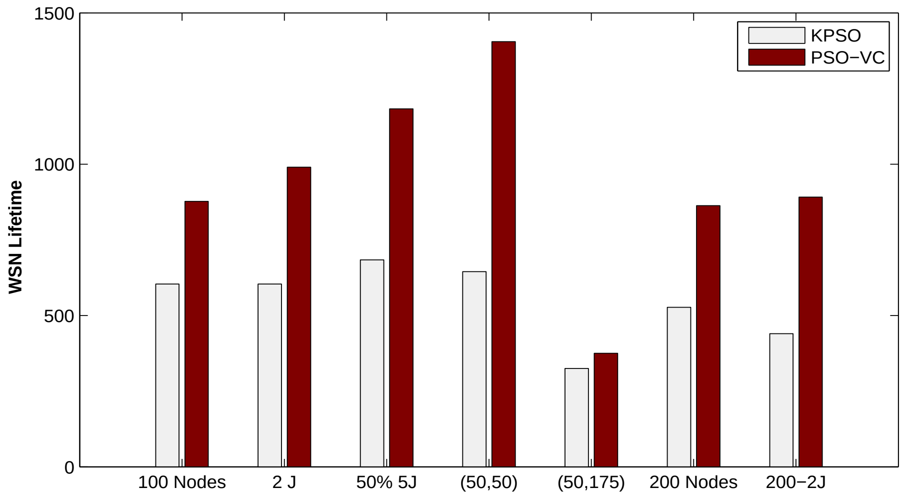


Figure 6.14: Comparison between KPSO and PSO-VC WSN Lifetime when the first node dies

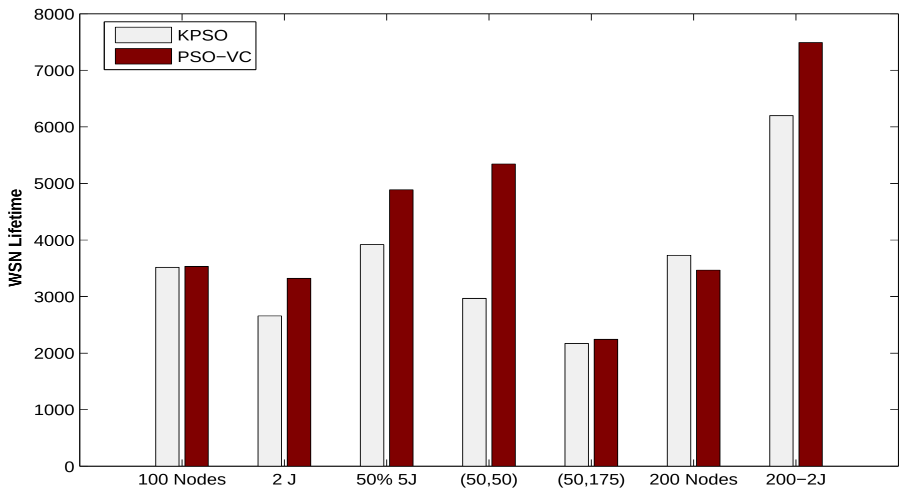


Figure 6.15: Comparison between KPSO and PSO-VC WSN Lifetime when 10% of the nodes die

Chapter Seven

Clustering Based Extended Hybrid K-Means and PSO

The main objective of this thesis is to prolong the WSN lifetime using clustering techniques, which is shown in the previously implemented approaches. Another approach to save the consumed energy, and thereby prolong the network lifetime, is to control the power of the nodes' antenna. We proposed KPSO-PSO approach that optimizes antenna pattern radius and network clustering. KPSO-PSO is a hybrid K-means PSO approach that is an extension to the first approach (KPSO), where another PSO phase is added. This approach first selects the best CH using KPSO. Then the role of the extended PSO phase is to optimize the antenna pattern radius and cluster members.

In this chapter, the results of applying our hybrid approach KPSO-PSO are described and evaluated. KPSO-PSO is compared with LEACH protocol, KPSO and PSO-VC approaches. Also the same approach is re-implemented using GA instead of PSO, named KGA-GA. The results are compared with KPSO-PSO results. KPSO-PSO approach is evaluated according to the following three perspectives:

1. KPSO-PSO vs. KGA-GA,
2. KPSO-PSO vs. KPSO, and PSO-VC
3. Circle-shaped criteria vs. Ellipse-shaped criteria
4. Variations in WSN setup

7.1 Selection of best PSO Equation

In order to select the PSO model for the extended PSO phase, PSO models were explored. Figure 7.1 shows the conversion of PSO models. Trelea-1 shows the best convergence. Table 7.1 lists the parameters of the PSO parameters adopted for the extended phase of KPSO-PSO.

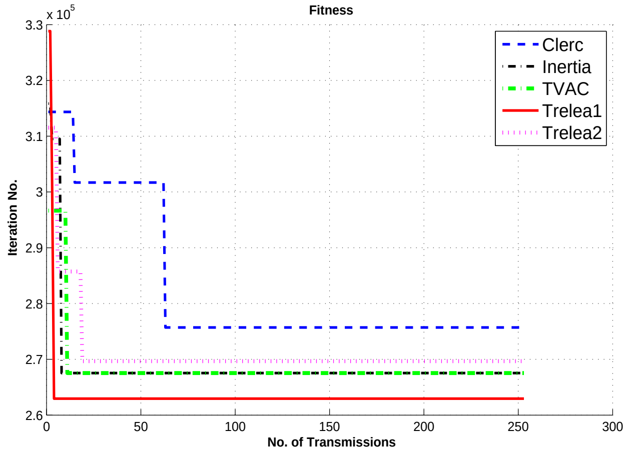


Figure 7.1: PSO conversion of KPSO-PSO phase 2

Table 7.1: PSO simulation parameters

Parameter	Value
PSO model	Trelea-1
Inertia weight	0.6
a	1.7
b	1.7
No. of particles	{20,40,60,80}

7.2 Developed Cluster Layout

In this section, we report the results of applying KPSO-PSO on the WSN layout dataset specified in Table 5.1. Two cases are adopted in exploring the proposed approach. They are:

- **Case 1:** KPSO-PSO using circle pattern
- **Case 2:** KPSO-PSO using ellipse pattern

Figure 7.2 shows the developed cluster layout for the KPSO approach. The cluster layout formed by our proposed KPSO-PSO using circle shape criteria is shown in Figure 7.3, and Figure 7.4 shows the cluster layout formed by the KPSO-PSO using ellipse shape criteria.

In Figures 7.2 and 7.3(a), the node labeled '**node a**' is belonging to two clusters. Thus our proposed KPSO-PSO caused a change in the cluster layout distribution. Figures 7.3(a) and 7.4(a) show that the ellipse criteria layout differs from the circle criteria layout; the node labeled '**node b**' is a sample of the difference.

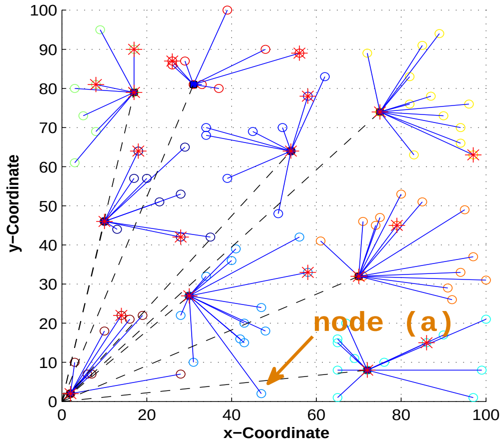


Figure 7.2: KPSO WSN Clustering Layout

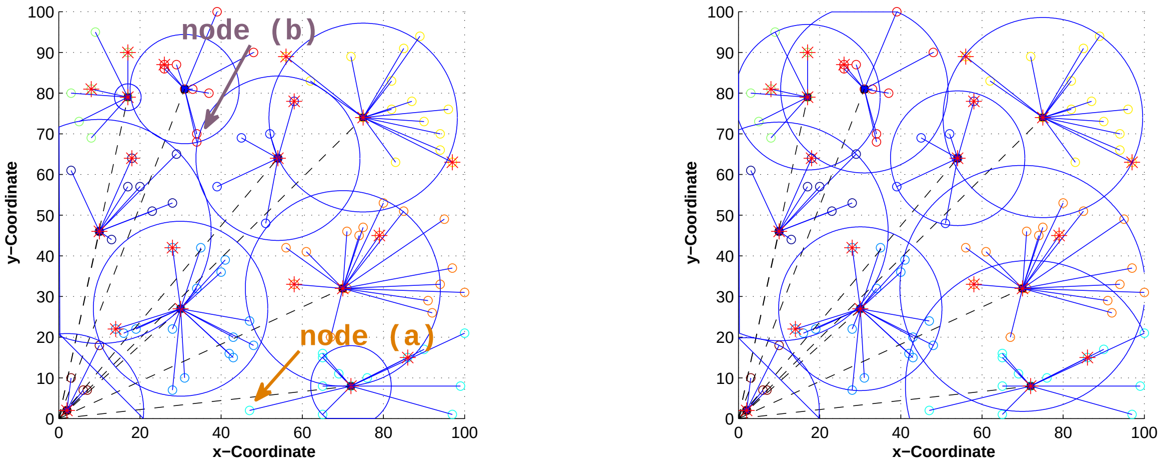


Figure 7.3: KPSO-PSO Clustering Layout for (a) Circle shape (b) with corrected radius

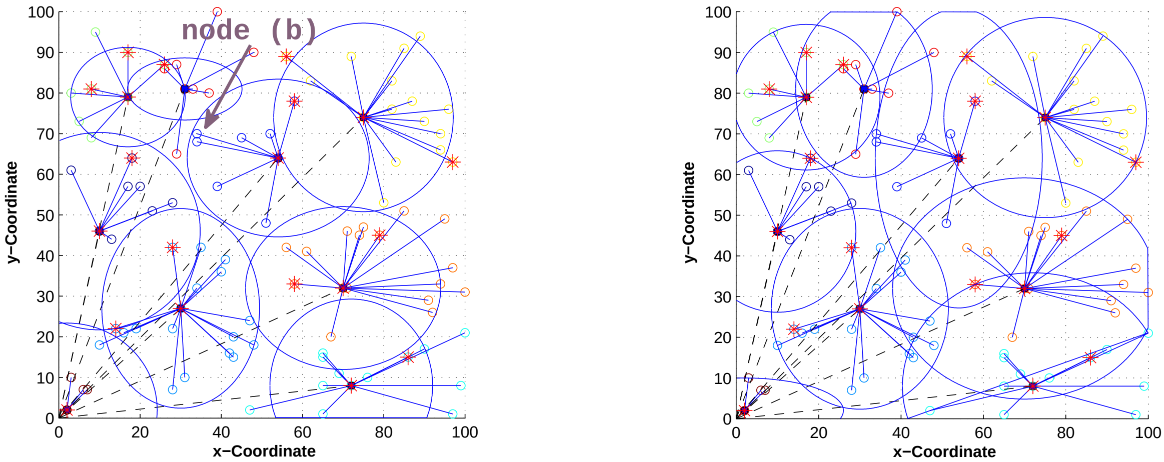


Figure 7.4: KPSO-PSO Clustering Layout for (a) Ellipse shape (b) with corrected radii

Inspite of selecting the optimal antenna pattern radius, some of the nodes are shown outside the antenna pattern range. The KPSO-PSO approach assigns nodes that are not within the antenna pattern range to connect to the nearest cluster. In this case, the KPSO-PSO approach corrects the optimum antenna pattern radius to include those nodes. The corrected antenna pattern radius is the Euclidean distance between the CH and the farthest member node. Figures 7.3(b) and 7.4(b) show the corrected circle and elliptical antenna pattern radii.

Figure 7.5(a) shows the total remaining energy for KPSO-PSO circle and ellipse pattern, and LEACH protocol. Figure 7.5(b) shows the total number of alive nodes versus simulated number of transmissions. Both pattern criteria outperformed LEACH protocol; they preserved the network's energy and more nodes survived during the network operation.

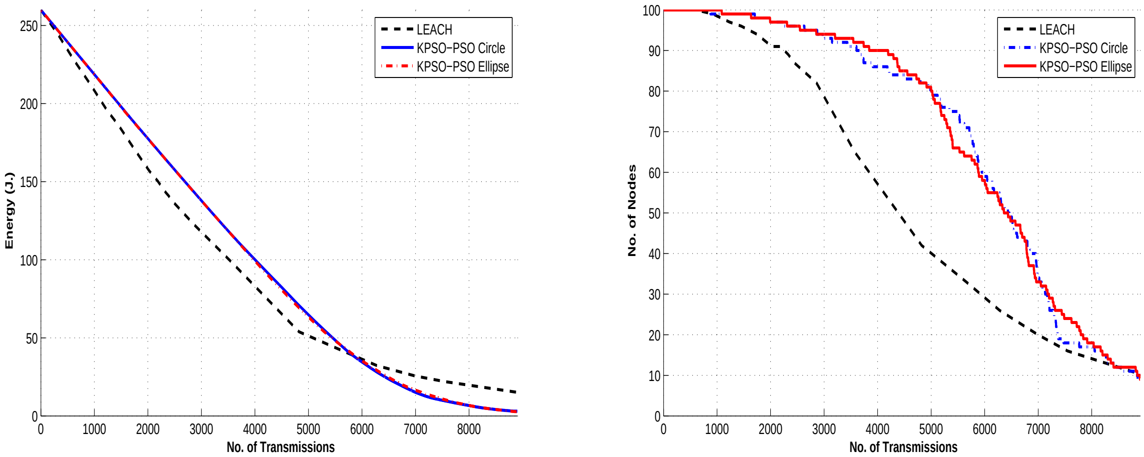


Figure 7.5: (a) Total remaining energy (b) Total number of alive nodes vs. number of transmissions for 100 nodes WSN with varying energy

7.3 WSN Lifetime Results

Figure 7.6 shows the WSN lifetime for the two patterns and LEACH protocol. Comparing with LEACH, KPSO-PSO ellipse criteria showed the best WSN lifetime improvement over LEACH. The number of alive sensors was doubled when the first node dies. When 10% of nodes die the KPSO-PSO WSN lifetime improved about 45-62%.

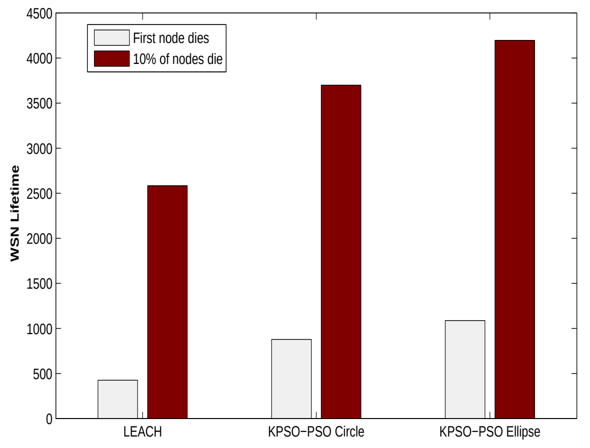


Figure 7.6: WSN Lifetime for 100 nodes-WSN with varying energy

7.4 KPSO-PSO vs. KGA-GA

Figure 7.7 shows the GA and PSO fitness conversion. PSO converges to a better fitness value than GA but with larger number of iterations. Figures 7.8, 7.9 and 7.10 show the developed cluster layouts for KGA approach, KGA-GA using circle shape criteria, and KGA-GA using ellipse shape criteria respectively. It can be shown that the three result in different clustering layouts; nodes labeled (a) and (b) are examples of the difference.

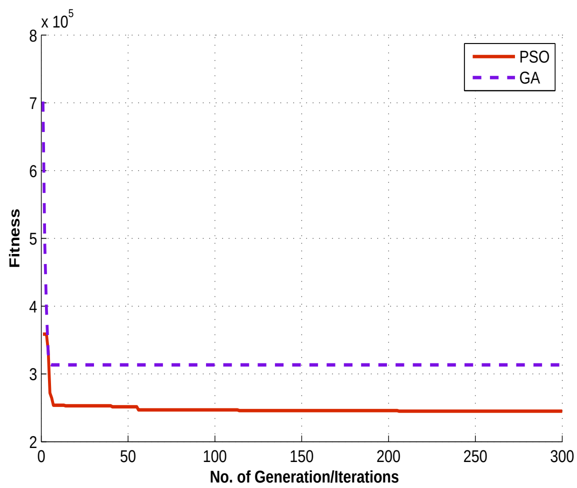


Figure 7.7: PSO and GA fitness conversion graphs for Circle Shape Criteria

We recorded the simulated number of transmissions along with the number of alive nodes ranged from 100 nodes to 10 nodes. Table 7.2 shows the simulated results for our KGA-GA and KPSO-PSO approaches. The simulated results showed that our proposed KPSO-PSO produced more transmissions than LEACH protocol and KGA-GA for both shape criteria: circle and ellipse by a given number of alive sensors. We also calculated the percentage

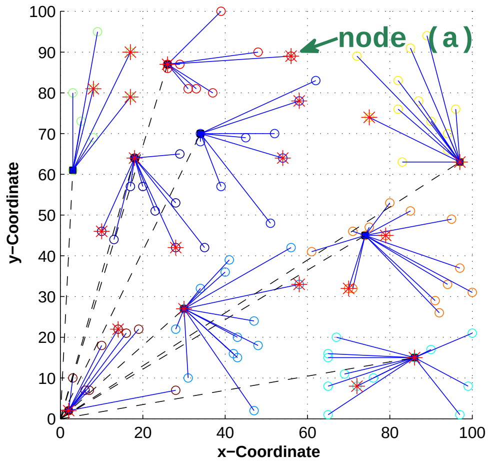


Figure 7.8: KGA WSN Clustering Layout

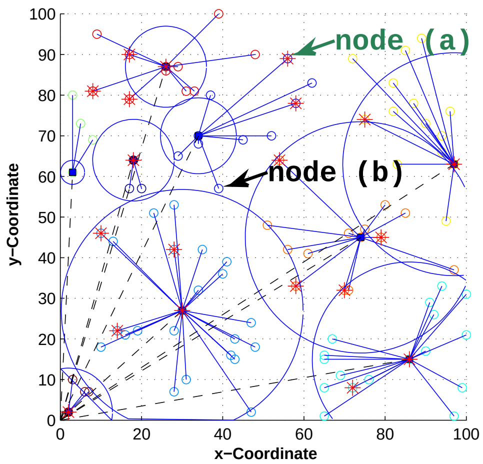


Figure 7.9: KGA-GA Clustering Layout for Circle shape

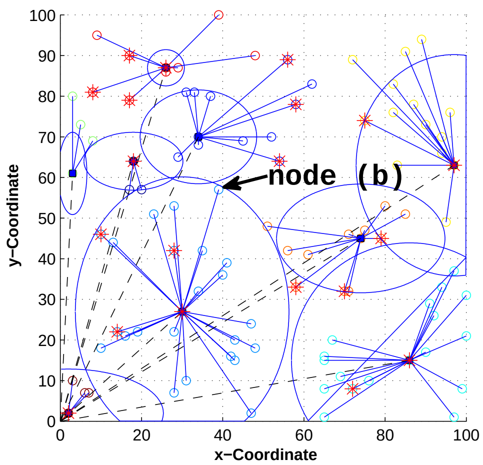


Figure 7.10: KGA-GA Clustering Layout for Ellipse shape

Table 7.2: Simulated number of transmissions

Number of Alive Sensor nodes	Simulated number of transmissions				
	LEACH	KGA-GA		KPSO-PSO	
		Circle	Ellipse	Circle	Ellipse
100	610	633	736	877	1087
90	2183	2803	2748	3699	4196
80	2858	4035	3807	4843	5022
70	2858	5009	4899	5614	5364
60	3572	5539	5711	6296	5892
50	3572	6019	6978	6600	6436
40	4822	6749	7238	6856	6796
30	5655	7206	7679	7447	7202
20	6995	8440	8452	7813	7841
10	7549	10590	11179	8491	8950

of improvement of KPSO-PSO over KGA-GA. The average improvement of KPSO-PSO over KGA-GA is 10.27% using the circle-shaped criteria, while the ellipse-shaped criteria resulted in 11.82% improvement of KPSO-PSO over KGA-GA.

7.5 KPSO-PSO vs. KPSO and PSO-VC

In this section, we analyze the results of the work compared to our developed KPSO and PSO-VC approaches. Figure 7.11 shows the WSN lifetime of the three proposed approaches on the test case network layout. KPSO-PSO prolonged the WSN lifetime than KPSO and PSO-VC. Thus controlling the antenna pattern resulted in more enhancement in the WSN lifetime by readjusting the members to be within the optimum antenna range.

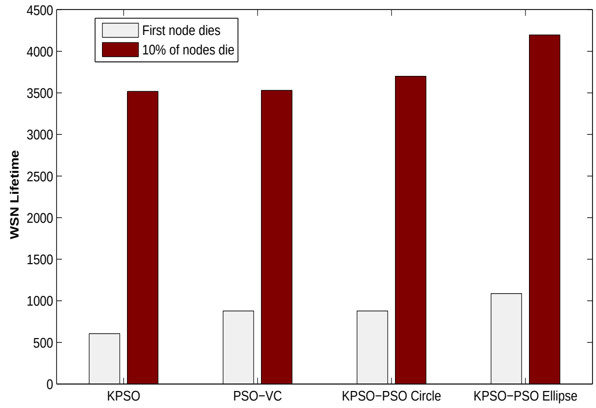


Figure 7.11: WSN Lifetime for 100 nodes-WSN with varying energy

7.6 Circle-shaped criteria vs. Ellipse-shaped criteria

The proposed approaches produced the best evolved radius of the virtual shapes adopted in this research. Table 7.3 shows the results of the best radii according to Equations 4.13 and 4.14. In case of ellipse pattern, the developed virtual ellipse of some CHs found to be similar to the network coverage in the case of a circle; i.e. the output radii of the ellipse are equal. The calculated average performance improvement of the ellipse-pattern over circle-pattern is 2.17% using KGA-GA approach, while KPSO-PSO resulted in 2.93% improvement.

Table 7.3: Antenna Pattern Circle and Ellipse Radii

CH No.	KGA-GA			KPSO-PSO		
	Circle	Ellipse		Circle	Ellipse	
	r	a	b	r	a	b
CH 1	10.05	12.19	7.13	27.02	22.33	24.09
CH 2	9.34	14.35	11.55	18.42	22.64	19.21
CH 3	29.78	26.32	33.15	21.75	22.96	24.30
CH 4	23.94	29.06	28.92	22.44	23.57	16.08
CH 5	2.96	3.53	10.13	14.09	14.09	14.09
CH 6	27.43	24.13	27.24	21.52	22.04	23.21
CH 7	28.46	20.74	13.39	24.03	24.03	20.54
CH 8	9.98	4.54	4.46	14.09	11.37	6.56
CH 9	10.82	23.36	10.85	23.25	6.42	24.71

7.7 Variations in WSN setup

This section explored the effect of varying WSN setup on the results of KPSO-PSO. WSN lifetime is reported for the following WSN setups:

- **50:** 50 nodes, 80% 2J and 20% 5J, BS (0,0)
- **100:** 100 nodes, 80% 2J and 20% 5J, BS (0,0)
- **200:** 200 nodes, 80% 2J and 20% 5J, BS (0,0)
- **(50,50):** 100 nodes, 80% 2J and 20% 5J, BS (50,50)
- **(50,175):** 100 nodes, 80% 2J and 20% 5J, BS (50,175)
- **50% 2J:** 100 nodes, 50% 2J and 50% 5J, BS (0,0)

Figure 7.12 shows the WSN lifetime when the first nodes dies, and Figure 7.13 shows the WSN lifetime when 10% of the nodes die. The graphs report the WSN lifetime for both shape criteria adopted; circle and ellipse. The graphs proved that KPSO-PSO improved WSN lifetime compared to LEACH protocol and our KPSO approach.

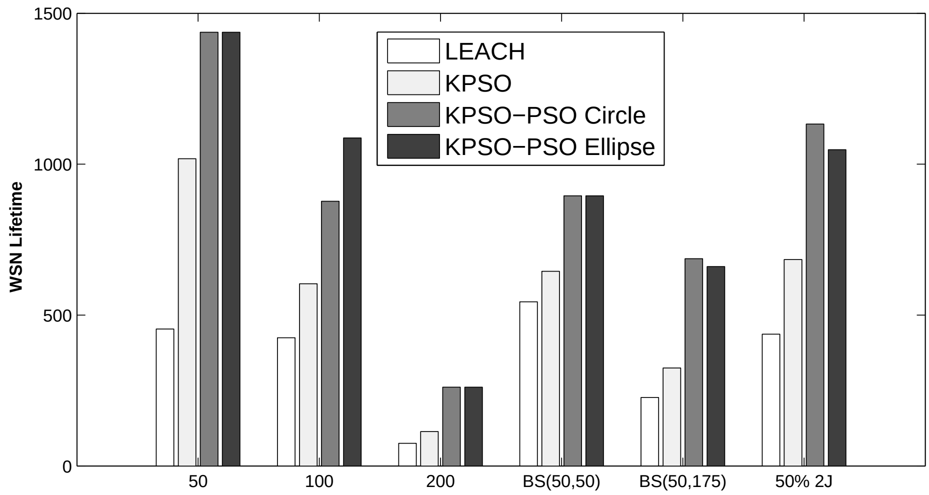


Figure 7.12: KPSO-PSO lifetime for various network layouts when the first node dies

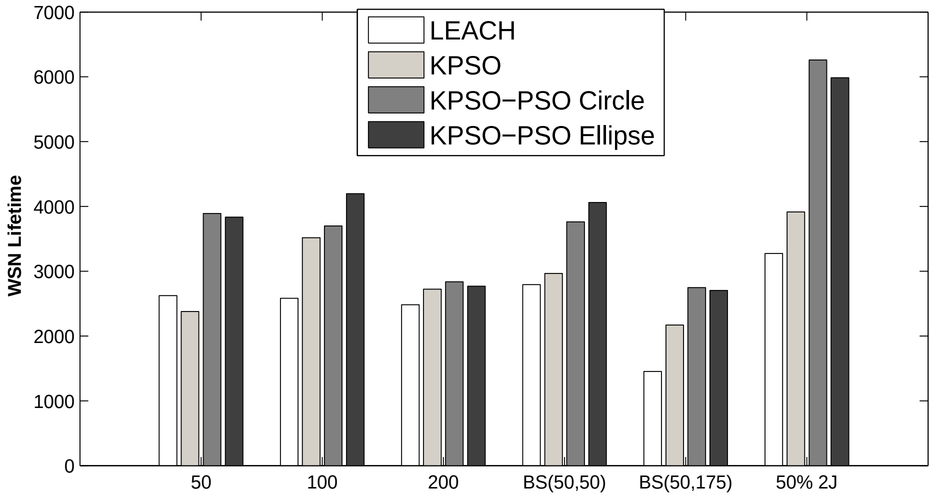


Figure 7.13: KPSO-PSO lifetime for various network layouts when 10% of the nodes die

Chapter Eight

WSN Computer Aided Design Toolbox

We implemented a WSN Clustering Aided Toolbox, named WSN-CAT, to develop and simulate our proposed approaches. The toolbox was designed such that it is simple and easy to be used with novice user. Our Matlab WSN-CAT toolbox is composed of the four modules. They are: WSN Data module, Clustering Data module, Clustering module and Simulation module. The PSO software tool is inspired from [124]. The chapter describes the modules of WSN-CAT, and how it works.

8.1 WSN-CAT Graphical User Interface

Figure 8.1 shows the graphical user interface (GUI) of the developed toolbox implemented by MATLAB program. The upper part of the GUI represents the WSN data that the user has to input. The input parameters of the WSN-CAT toolbox are:

- Length of the geographic area,
- Width of the geographic area,
- Base Station Location: X, and Y coordinates,
- Number of sensor nodes
- WSN nodes generation criteria,
- Nodes energy option,
- Clustering approach to be used,
- Fitness function to be used,
- Number of clusters required.

The WSN-CAT outputs the obtained clustering and the simulation results, as shown in the lower part of the figure. The graph on the left shows the developed cluster layout resulted from running one of our proposed approaches. The remaining graphs show the simulation resulting from the developed cluster layout. The graphs in the middle and on the right show the total number of alive nodes and the total remaining energy respectively.

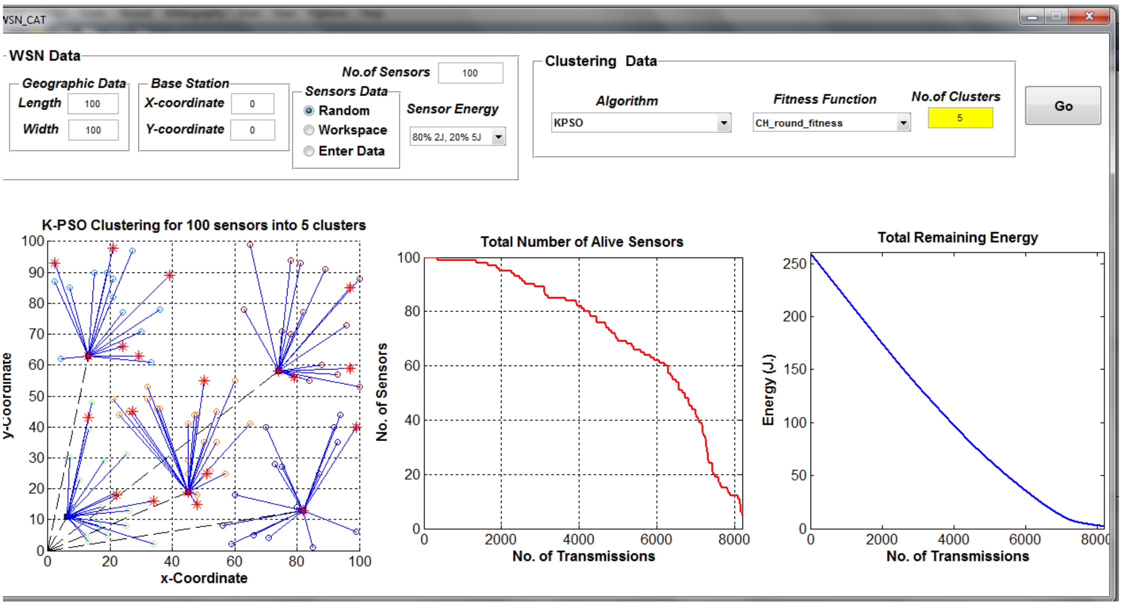


Figure 8.1: WSN-CAT Toolbox GUI

8.2 Components of WSN-CAT

Figure 8.2 shows the block diagram of Our developed MATLAB toolbox. It consists of the following components:

1. WSN Data Module
2. Cluster Data Module
3. Clustering Module
4. Simulation Module

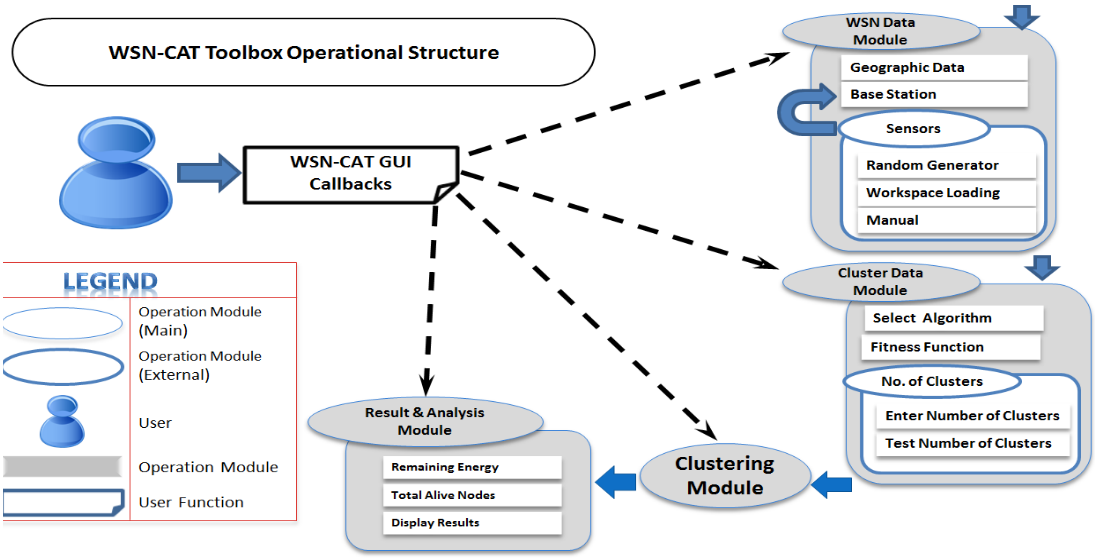


Figure 8.2: WSN-CAT Toolbox Operational Structure

8.2.1 WSN Data Module

The WSN Data module allows the user to enter the simulation parameters: length and width of the geographical area, the base station location and the number of sensor nodes. The user also decides whether the nodes' data are generated randomly using 'Random Node Generator', loaded from a MATLAB workspace, or entered manually. In case the user decides to generate data using our Random Node Generator, the nodes' energy has to be chosen. Figure 8.3 shows the GUI when the user can load WSN data from a MATLAB workspace file. Figure 8.4 shows how to enter WSN data manually. The user enters three types of information: the X-coordinates, the Y-coordinates, and finally the nodes energy.

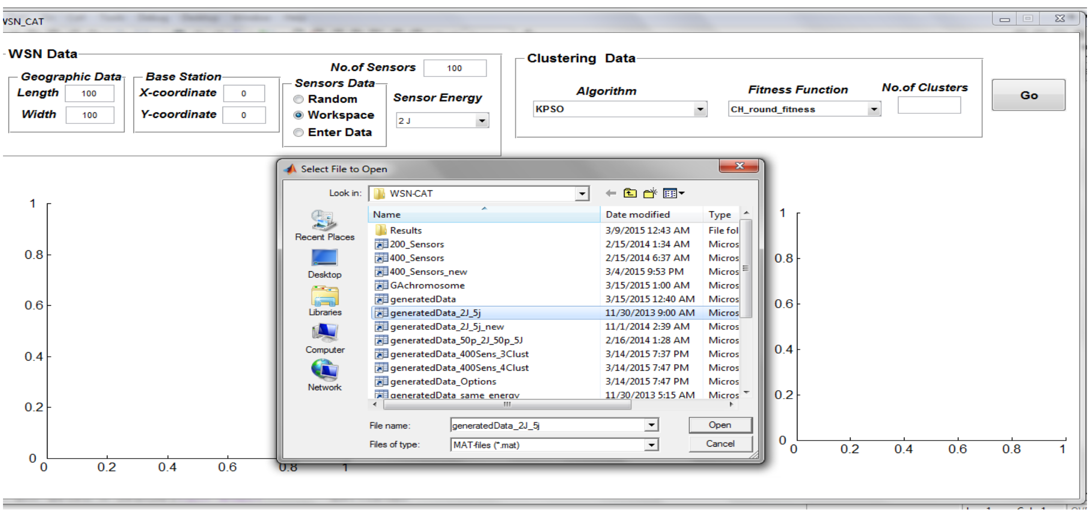


Figure 8.3: WSN-CAT Toolbox: Load from workspace

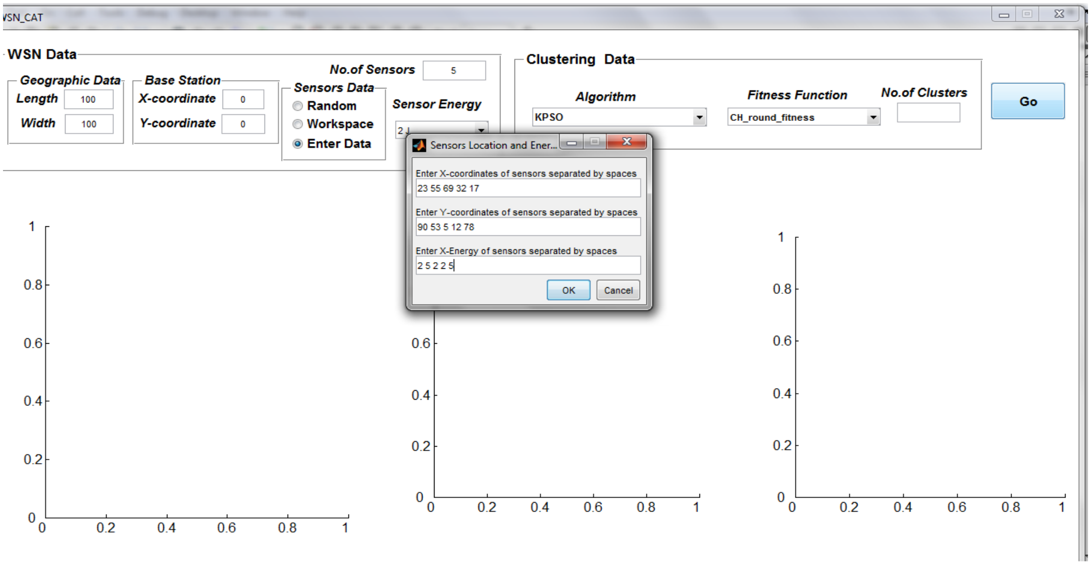


Figure 8.4: WSN-CAT Toolbox: Enter WSN data manually

Random Node Generator

A 'Random Node Generator' is implemented to generate randomly deployed nodes in a 2D rectangular geographic area. The random generator requires input data as : length and width of the simulated geographic area and number of sensor nodes. The user also specifies the criteria for assigning energy to the generated nodes: all nodes have the same or different energy values. Specifically, three criteria are implemented in the random node generator as follows:

1. All nodes are assigned the same energy level. A value of 2 joule is adopted in the toolbox.
2. Nodes are assigned random energy value between two levels, e_{min} and e_{max}. The toolbox adopted default values of $e_{min} = 2J$ and $e_{max} = 5J$.
3. Nodes are assigned one of two energy levels, such that 80% of the nodes are assigned an energy level e_1, and 20% of the nodes are assigned another energy level e_2. The toolbox default values are: $e_1 = 2J$, $e_2 = 5J$.

8.2.2 Cluster Data Module

In this module, the user decides one of our proposed approaches to be used. Figure 8.5 shows how the user chooses the approach. For each approach, at least one fitness function is given. Figure 8.6 shows the possible fitness for the KPSO approach. The toolbox uses the KPSO approach as the default choice, and our novel fitness is the default.

Except for PSO-VC and GA-VC, the user can specify the number of clusters. For KPSO, KGA, KPSO-PSO, and KGA-GA, the user can enter the number of clusters for simulation. If the number of clusters is not specified, the toolbox triggers a 'Test-Clusters' module that tests the nodes layout and calculates the best number of clusters based on K-means.

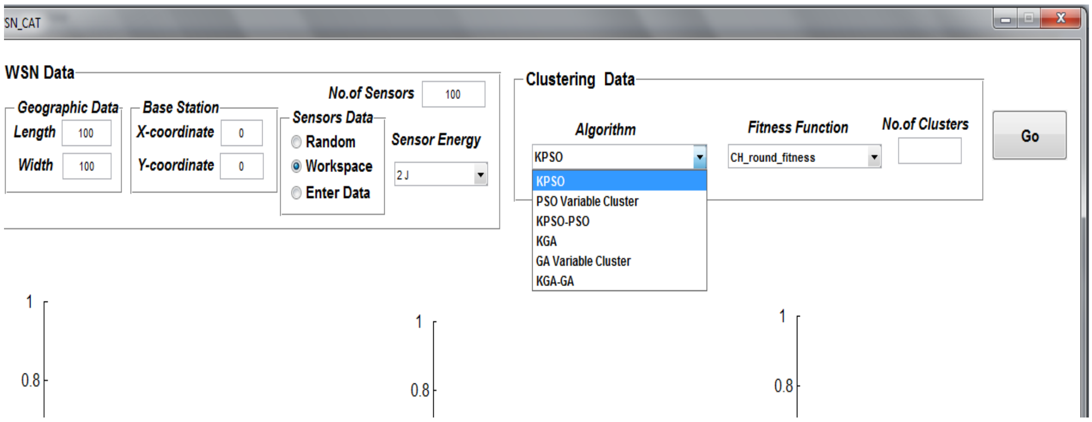


Figure 8.5: WSN-CAT Toolbox: Proposed approaches options

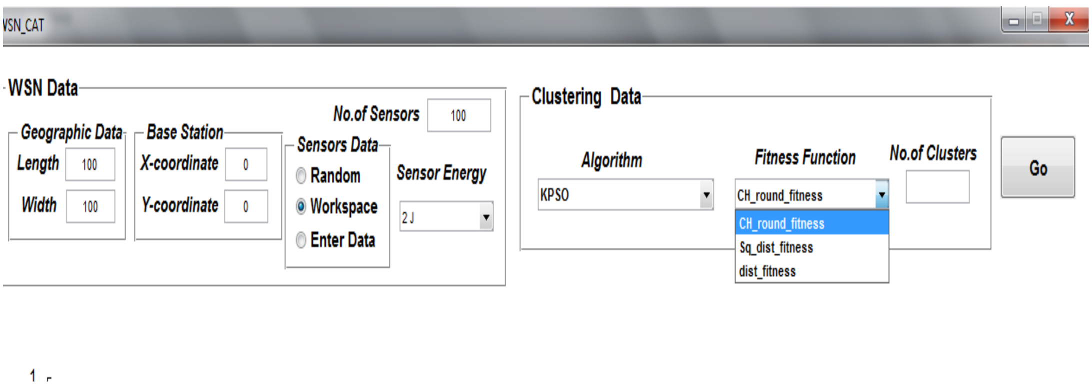


Figure 8.6: WSN-CAT Toolbox: Fitness functions

8.2.3 Clustering Module

This module is triggered only after all the simulation data are entered by the user. It runs one of our proposed approaches according to the user selection (or default approach if the user didn't select any). Figure 8.7 shows the detailed clustering modules for the first three approaches: KPSO, PSO variable clustering, and KPSO-PSO. The clustering modules of the remaining three approaches are the same as those except that GA is used instead of PSO. The output of this module is the cluster layout: CHs and their members.

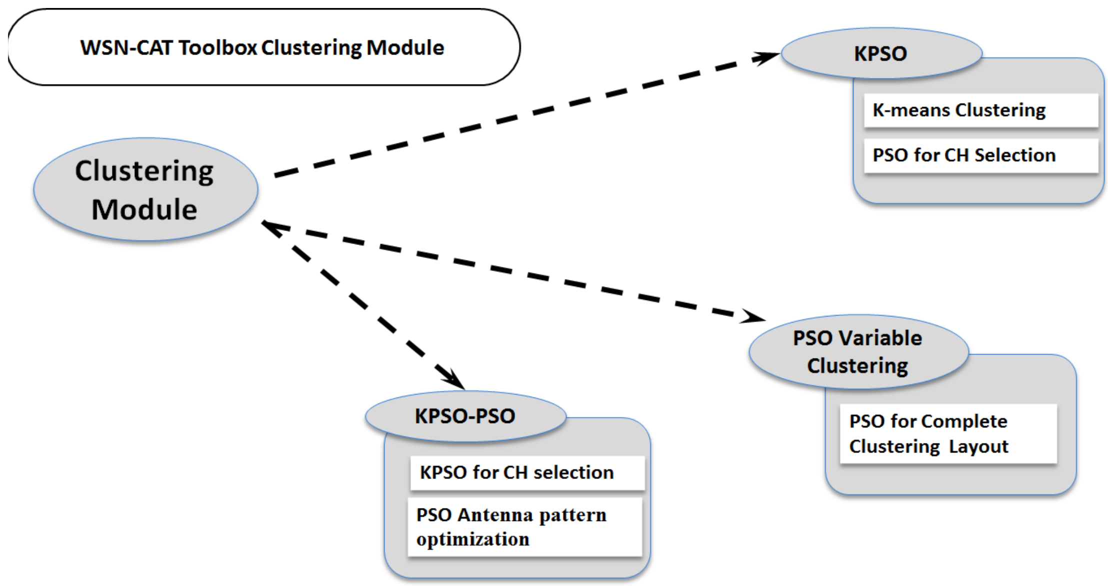


Figure 8.7: Clustering Module of the WSN-CAT Toolbox

8.2.4 Simulation Module

This module is triggered by the clustering module. In this module, we developed our own simulator to evaluate our proposed clustering approach. Our developed simulator reports the WSN data during each transmission. It calculates the total consumed energy for every sensor node, the WSN remaining energy and the total alive nodes based on the radio model described in [44] and modeled in Section 4.1.3. This module displays the total number of alive nodes vs. the number of transmissions, as shown in the middle graph of Figure 8.1. The graph in the right displays the total energy vs. the number of transmissions.

Chapter Nine

Conclusion

9.1 Thesis Contribution

WSN has become an essential component in many real life applications as military, environmental, industrial and many others. Factories, machine commands and control systems are estimated to switch over to rely on wireless sensor nodes. Now, the trend is to replace buildings control and automation systems by WSN. The sensor nodes will become common as light switches and thermostats. However, WSN face some challenges as deployment, security, QoS, and energy. The main constraint faced WSN operation is the limited energy resource during operation in field far away from base station. In most cases, nodes are equipped with batteries that can't be replaced or recharged.

In this thesis, we proposed number of clustering optimization approaches to prolong the WSN lifetime. Clustering is an NP-hard problem that can't be solved using traditional techniques. Two sub-problems have to be solved to cluster WSN. They are: the number of clusters to be produced and the CH for each cluster. The research presented in this thesis investigated WSN clustering using PSO technique to overcome many problems associated with traditional techniques. PSO showed many advantages on handing complex optimization problems by finding solution not have been presented in the past. Three contributions were developed: 1) Hybrid K-means PSO approach, named KPSO, 2) PSO variable clustering, named PSO-VC, and 3) Hybrid K-means PSO approach, named KPSO-PSO. The first contribution, KPSO, designed a three phase approach where each phase of the first two phases solved one clustering sub-problem. In the first phase, K-means investigates for the best number of clusters and then performs clustering. Then the PSO phase selects the best CH according to our novel fitness that maximizes the number of transmissions a CH performs before its energy depletes. In the second contribution, PSO-VC, PSO is tasked to solve the complete clustering sub-problems in one phase. It outputs the best number of clusters, CHs and the cluster members. The cluster layout is performed based on maximizing a novel fitness that concentrates on the network lifetime. Then we decided to perform clustering based on the antenna pattern, leading to our third contribution. The third contribution, KPSO-PSO, is an evolution of the first approach (KPSO) where a new phase is added. The third contribution optimizes the antenna pattern radius and re-arranges the member nodes per cluster to be in the region of the antenna. In order to test the developed approaches, dataset was generated randomly. The RF model was implemented and the network operation was simulated by our own implemented simulator. The approaches were compared with the famous LEACH protocol that is used as benchmark for clustering approaches. In order to test the effectiveness of the proposed approaches, the same solutions were re-implemented with GA instead of PSO, and the results were compared. The following points have been observed:

- The problem with K-means is that it performs clustering based on inter cluster and intra cluster distance. It is independent of the WSN field data environment, as base station location, and assigns the center of the cluster to be CH. Integrating PSO with K-means enabled selecting the CH that proved prolonging the WSN lifetime.
- KPSO showed significant improvement of WSN lifetime compared with traditional LEACH protocol. It also showed promising results when changing: the number of nodes, energy of the nodes, and base station location.
- The second approach (PSO-VC) showed more flexibility in obtaining clusters.
- Although PSO-VC produced considerably more clusters than KPSO approach, the WSN lifetime was improved than KPSO and LEACH. PSO-VC showed promising results to varying network data.
- Optimizing the antenna pattern range re-arranged the cluster members. The approach removed the nodes that were mis-belonged to the clusters produced by K-means and corrected its relation to belong to better CHs.
- KPSO-PSO resulted in enhancement of the WSN lifetime over KPSO and PSO-VC.
- Choosing the base station within the field leads to more number of transmissions due to the decrease of the communication distance between the CH and the base station.
- Setting the nodes with various energy levels, which is more realistic, enabled our approaches to discriminate in its search and converge to the best CH resulting in better enhancement in WSN lifetime.

- Using PSO in our approaches showed better convergence than using GA and better lifetime is reached.

Finally, a WSN clustering aided toolbox, named WSN-CAT, was developed to investigate our proposed approaches. The tool box is user friendly; the user can simply enter the WSN information and clustering data. The toolbox then runs the selected approach to produce the clustering layout. Then our developed simulator is triggered to show the estimated operation of the network.

9.2 Recommended Future Work

We recommend the following ideas that can be used for future:

- The approaches are applied on stationary nodes. We recommend re-implementing the approaches to work as mobile nodes WSN.

- Our solutions were based on a single objective fitness function that maximizes the number of CH transmissions. The approaches can be re-implemented to be multi-objective that adds more criteria, as coverage.

- Some applications require the nodes to be uniformly distributed (e.g. Agriculture), not randomly distributed. We suggest testing the approaches on uniformly distributed data.

- There are different types of directional antenna patterns. The third contribution aimed to optimize the antenna pattern while clustering the WSN. This contribution investigated two shapes: circle and ellipse. New investigations can be made on other antenna pattern shapes.

- Some applications may deploy heterogeneous sensor nodes. Thus, in the same WSN, different antenna patterns may occur. The third contribution could be modified to investigate the use of different antenna patterns within the same network.

Appendix 1

WSN-CAT MATLAB Source code

```
function varargout = WSN_CAT(varargin)
% WSN_CAT MATLAB code for WSN_CAT.fig
    gui_Singleton = 1;
    gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @WSN_CAT_OpeningFcn, ...
    'gui_OutputFcn', @WSN_CAT_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
    % End initialization code - DO NOT EDIT
    % — Executes just before WSN_CAT is made visible.
    function WSN_CAT_OpeningFcn(hObject, eventdata, handles, varargin)
handles.output = hObject;
guidata(hObject, handles);
set(handles.Rand_RadBtn,'Value',1);
set(handles.WrkSpace_rdBtn,'Value',0);
set(handles.EntrData_RdBtn,'Value',0);
%============================================
% — Outputs from this function are returned to the command line.
function varargout = WSN_CAT_OutputFcn(hObject, eventdata, handles)
```

```
varargout{1} = handles.output;
%=================================================
% — Executes on selection change in Fitness_pop.
function Fitness_pop_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ....
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%=================================================
% — Executes during object creation, after setting all properties.
function Fitness_pop_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
set(hObject, 'String', {'CH_round_fitness', 'Sq_dist_fitness', 'dist_fitness'});
% Default fitness
Fitness_Selection = 'CH_round_fitness';
%=================================================
% — Executes on selection change in CI_popupmenu.
function CI_popupmenu_Callback(hObject, eventdata, handles)
CI_Selection = get(hObject,'Value');
% control the fitness function options according to the algorithm
switch CI_Selection
case 1 %KPSO
    set(handles.Fitness_pop, 'String', {'CH_round_fitness', ...
    'Sq_dist_fitness', 'dist_fitness'});
case 2 % PSO Variable Cluster
```

```
    set(handles.Fitness_pop, 'String', {'Ech_DistSq_N_fitness',...
    'Dist_Square_fitness','DistSq_N_fitness',});
case 3 % KPSO-PSO
    set(handles.Fitness_pop, 'String', {'CH_round_fitness',...
    'Sq_dist_fitness', 'dist_fitness'});
case 4 % KGA
    set(handles.Fitness_pop, 'String', {'GA_CH_round_fitness'});
case 5 % GA Variable Cluster
    set(handles.Fitness_pop, 'String', {'GA_Alg2_fitness'});
case 6 % KGA-GA
    set(handles.Fitness_pop, 'String', {'GA_CH_round_fitness'});
end
%-----------------------------
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%=============================================
% --- Executes during object creation, after setting all properties.
function CI_popupmenu_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% Default Selection is the KPSO
CI_Selection = 'KPSO';
%=============================================
function Clust_Txt_Callback(hObject, eventdata, handles)
```

```
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
       isempty(BSx) &&  isempty(BSy)
      set(handles.Create_Btn,'Enable','on');
end

%============================================
% — Executes during object creation, after setting all properties.
function Clust_Txt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
      get(0,'defaultUicontrolBackgroundColor'))
      set(hObject,'BackgroundColor','white');
end
%============================================
function SensNo_Txt_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
       isempty(BSx) &&  isempty(BSy)
      set(handles.Create_Btn,'Enable','on');
end
%============================================
% — Executes during object creation, after setting all properties.
function SensNo_Txt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
```

```
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%=============================================
function BSx_Txt_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%=============================================
function BSx_Txt_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%=============================================
function BSy_Txt_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
```

```
%============================================
function BSy_Txt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%============================================
function Len_Txt_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ....
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%============================================
function Len_Txt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%============================================
function Wdth_Txt_Callback(hObject, eventdata, handles)
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
     isempty(BSx) &&  isempty(BSy)
```

```matlab
    set(handles.Create_Btn,'Enable','on');
end
%===========================================
function Wdth_Txt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'), ...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
%===========================================
function Energy_popMenu_Callback(hObject, eventdata, handles)
Energy_Selection_index = 1;
%===========================================
function Energy_popMenu_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),...
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%===========================================
function WrkSpace_rdBtn_Callback(hObject, eventdata, handles)
set(handles.Rand_RadBtn,'Value',0);
set(handles.EntrData_RdBtn,'Value',0);
%----------------------------------------
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if  isempty(Nsensors) && isempty(Len) &&  isempty(Wdth) && ...
     isempty(BSx) &&  isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%===========================================
```

```
function Rand_RadBtn_Callback(hObject, eventdata, handles)
set(handles.WrkSpace_rdBtn,'Value',0);
set(handles.EntrData_RdBtn,'Value',0);
%------------------------------------------------------------
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if isempty(Nsensors) && isempty(Len) && isempty(Wdth) && ...
     isempty(BSx) && isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%=============================================================
function EntrData_RdBtn_Callback(hObject, eventdata, handles)
set(handles.Rand_RadBtn,'Value',0);
set(handles.WrkSpace_rdBtn,'Value',0);
%------------------------------------------------------------
% Check to Enable the Create Button
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); % number of sensors
Wdth = str2num(get(handles.Wdth_Txt,'String')); % number of sensors
BSx = str2num(get(handles.BSx_Txt,'String')); % number of sensors
BSy = str2num(get(handles.BSy_Txt,'String')); % number of sensors
if isempty(Nsensors) && isempty(Len) && isempty(Wdth)...
    && isempty(BSx) && isempty(BSy)
    set(handles.Create_Btn,'Enable','on');
end
%=============================================================
% --- Executes The Clustering Algorithm
%=============================================================
function Create_Btn_Callback(hObject, eventdata, handles)
% get WSN data entered by the user
```

```
%------------------------------------------
Nsensors = str2num(get(handles.SensNo_Txt,'String')); % number of sensors
Len = str2num(get(handles.Len_Txt,'String')); %
Wdth = str2num(get(handles.Wdth_Txt,'String')); %
BSx = str2num(get(handles.BSx_Txt,'String')); %
BSy = str2num(get(handles.BSy_Txt,'String')); %
CI_Selection = get(handles.CI_popupmenu,'Value');
Fitness_Selection = get(handles.Fitness_pop,'Value');
Fitness_Items = get(handles.Fitness_pop,'String');
WSNfitness = char(Fitness_Items(Fitness_Selection));
% Ensure the fitness for GA
    switch CI_Selection
    case 4 %KGA
        WSNfitness = 'GA_CH_round_fitness';
    case 5 %GA alg2
        WSNfitness = 'GA_Alg2_fitness';
    case 6 %KGA-GA
        WSNfitness = 'GA_CH_round_fitness';
    end
    %------------------------------------------
    % generate random data or read from workspace or write data
    %------------------------------------------
    if get(handles.Rand_RadBtn,'Value')==1 % generate random data
        Energy_Selection_index = get(handles.Energy_popMenu,'Value');
        SensorsData = GenerateData (Nsensors, Len, Wdth, Energy_Selection_index);
        save generatedData SensorsData;
    elseif get(handles.WrkSpace_rdBtn,'Value')==1 % load workspace
        [filename,path] = uigetfile('*.mat');
        eval(['load ', filename])
        save generatedData SensorsData;
    elseif get(handles.EntrData_RdBtn,'Value')==1 % write data
        SizeFlag = 0;
        D=inputdlg({'Enter X-coordinates of sensors separated by spaces', ...
        'Enter Y-coordinates of sensors separated by spaces', ...
```

```
    'Enter X-Energy of sensors separated by spaces'},'Sensors Location and Energy');
    x= length(D(1));
    Lx = str2num(D{1});
    Ly = str2num(D{2});
    E = str2num(D{3});
    SensorsData = [Lx',Ly',E'];
    save generatedData SensorsData
    % check that the parameters of written data is consistent with input
    s = size(SensorsData,1);
    if s  = Nsensors
        Nsensors = s;
        set(handles.SensNo_Txt,'String',Nsensors);
        set(handles.SensNo_Txt,'BackgroundColor','y');
    end
    if Len < max(Lx); % sensor outside written range
        Len =floor(max(Lx));
        set(handles.Len_Txt,'String',Len);
        set(handles.Len_Txt,'BackgroundColor','y');
    end
    if Wdth < max(Ly); % sensor outside written range
        Wdth =floor(max(Ly));
        set(handles.Wdth_Txt,'String',Wdth);
        set(handles.Wdth_Txt,'BackgroundColor','y');
    end
end % if
    %------------------------------------------------------------
    % get number of clusters or generate the best number of clusters
    %------------------------------------------------------------
    ClusterNo = str2num(get(handles.Clust_Txt,'String')); % number of Clusters
    if isempty(ClusterNo)
        switch CI_Selection
        case {2,5} % do nothing
        otherwise
            ClusterNo = K_Test_no_Clusters (SensorsData, Len, Wdth,[BSx,BSy]);
```

```
        set(handles.Clust_Txt,'String',ClusterNo);
        set(handles.Clust_Txt,'BackgroundColor','y');
    end % switch
    end % if
        %————————————————————————————————
        % get min/max based on the Selected Algorithm and fitness
        %————————————————————————————————
        switch CI_Selection
        case {1,3} %KPSO or KPSO-PSO
            min_max = 1; % for thr first three choices
                if Fitness_Selection >1
                    min_max = 0;
                end % if
                case 2 % PSO Variable Cluster
                    if Fitness_Selection == 1
                        min_max = 1;
                    else
                        min_max = 0;
                    end % if
                    otherwise
                        min_max = 1;
                    end
                    %———————————————————————————————-
                    % Now Run the Algorithm
                    %————————————————————————————————
                    switch CI_Selection
                    case 1 %KPSO
                        PSOfig = figure;
                        KPSO_Algorithm (SensorsData,Nsensors, Len, Wdth, ...
                        [BSx,BSy],ClusterNo,WSNfitness,min_max);
                        close(PSOfig);
                        Layout_title=sprintf(' K-PSO Clustering for %i sensors into %i
...
                        clusters', Nsensors, ClusterNo);
```

```
case 2 % PSO Variable Cluster
    PSOfig = figure;
    ClusterNo = Var_No_Cluster_PSOalgorithm(SensorsData, ...
    Nsensors,Len,Wdth,[BSx,BSy],WSNfitness,min_max,5,50);
    close(PSOfig);
    Layout_title=sprintf('PSO Variable Clustering for %i ...
    sensors into %i clusters',Nsensors, ClusterNo);
case 3 % KPSO-PSO
    PSOfig = figure;
    KPSO_PSO (SensorsData,Nsensors, Len, Wdth,[BSx,BSy], ...
    ClusterNo,WSNfitness,min_max,2); % Ellipse
    close(PSOfig);
    Layout_title=sprintf(' KPSO-PSO Clustering for %i ...
    sensors into %i clusters',Nsensors, ClusterNo);
case 4 % KGA
    KGA_algorithm (SensorsData,Nsensors, Len, Wdth,...
    [BSx,BSy],ClusterNo,WSNfitness,min_max);
    Layout_title=sprintf(' KGA Clustering for %i ...
    sensors into %i clusters',Nsensors, ClusterNo);
case 5 % GA Variable Cluster
    ClusterNo = WSN_GA_algorithm (SensorsData,Nsensors,...
    Len,Wdth,[BSx,BSy],WSNfitness,min_max,5,50);
    Layout_title=sprintf('GA Variable Clustering for %i ...
    sensors into %i clusters', Nsensors, ClusterNo);
case 6 % KGA-GA
    KGA_GA_2 (SensorsData,Nsensors, Len, Wdth, ...
    [BSx,BSy],ClusterNo,WSNfitness,min_max,2); % Ellipse
    Layout_title=sprintf(' KGA-GA Clustering for %i ...
    sensors into %i clusters', Nsensors, ClusterNo);
end
%——————————————————————————
load WSN_CAT_results
%——————————————————————————-
% Now Plot the Cluster Layout
```

```
%-----------------------------------------------------------
axes(handles.Layout_axes);
%clf
Draw_Clusters( SensorsData,WSN_CAT_membership,Len, Wdth,
...
WSN_CAT_CHx,WSN_CAT_CHy,[BSx,BSy],Layout_title );
if CI_Selection==3 % KPSO-PSO Ellipse
    for ik=1:ClusterNo
        hold on;
        draw_Ellipse(WSN_CAT_CHx(1,ik),...
        WSN_CAT_CHy(1,ik),WSN_CAT_Radius(ik),...
        WSN_CAT_Radius(ik+WSN_CAT_ClusterNo),Len,Wdth);
    end % for
    end % if
        %-----------------------------------------------------------
        % Plot the alive nodes graph vs. Rounds
        %-----------------------------------------------------------
        axes(handles.Alive_axes);
        %clf
        plot(WSN_CAT_alive_nodes,'r-','LineWidth',2);
        grid;
        xlabel('{\bf No. of Transmissions}');
        ylabel('{\bf No. of Sensors}');
        axis([0 WSN_CAT_rounds 0 WSN_CAT_alive_nodes(1)]);
        AliveSen_title=sprintf(' Total Number of Alive Sensors ');
        title(AliveSen_title,'fontweight','bold');
        %-----------------------------------------------------------
        % Plot the Energy graph vs. Rounds
        %-----------------------------------------------------------
        axes(handles.Energy_axis);
        %clf
        grid;
        plot(WSN_CAT_Energy,'b-','LineWidth',2);
        xlabel('{\bf No. of Transmissions}');
```

```
ylabel('{\bf Energy (J.)}');
axis([0 WSN_CAT_rounds 0 WSN_CAT_Energy(1)]);
Energy_title=sprintf(' Total Remaining Energy ');
title(Energy_title,'fontweight','bold');
```

Bibliography

[1] H.-S. Seo, S.-J. Oh, and C.-W. Lee, “Evolutionary genetic algorithm for efficient clustering of wireless sensor networks,” in *Proceedings of the 6th IEEE Consumer Communications and Networking Conference*, pp. 1–5, 2009.

[2] Z. Rezaei and S. Mobininejad, “Energy saving in wireless sensor networks,” *International Journal of Computer Science and Engineering Survey*, vol. 3, no. 1, pp. 23–37, 2012.

[3] marketsandmarkets, “IWSN (Industrial Wireless Sensor Network) market by sensor, technology, application, geography trend, and forecast to 2020,” 2014.

[4] N. Srivastava, “Challenges of next-generation wireless sensor networks and its impact on society,” *Journal of Telecommunications*, vol. 1, no. 1, pp. 128–133, 2010.

[5] A. Pathan, H.-W. Lee, and C. S. Hong, “Security in wireless sensor networks: issues and challenges,” in *Proceedings of the 8th International Conference on Advanced Communication Technology*, vol. 2, pp. 1043–1048, 2006.

[6] V. Kumar, A. Jain, and P. N. Barwal, “Wireless sensor networks: Security issues, challenges and solutions,” *International Journal of Information and Computation Technology*, vol. 4, no. 8, pp. 858–868, 2014.

[7] H. K. Kalita and A. Kar, “Wireless sensor network security analysis,” *International Journal of Next-Generation Networks*, vol. 1, no. 1, pp. 1–10, 2009.

[8] C. Zhu, C. Zheng, L. Shu, and G. Han, “A survey on coverage and connectivity issues in wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619 – 632, 2012.

[9] R. Kulkarni, A. Forster, and G. Venayagamoorthy, “Computational intelligence in wireless sensor networks: A survey,” *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 68 – 96, 2011.

[10] J. Torregoza, I.-Y. Kong, and W.-J. Hwang, “Wireless sensor network renewable energy source life estimation,” in *Proceedings of the 8th International Conference on e-Health Networking, Applications and Services*, pp. 13–18, 2006.

[11] W.-G. Seah, Z. A. Eu, and H. Tan, “Wireless sensor networks powered by ambient energy harvesting (wsn-heap) - survey and challenges,” in *Proceedings of the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, pp. 1–5, 2009.

[12] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.

[13] I. Kang and R. Poovendran, “Maximizing network lifetime of broadcasting over wireless stationary ad hoc networks,” *Mobile Networks and Applications*, vol. 10, no. 6, pp. 879–896, 2005.

[14] N. H. Mak and W.-G. Seah, "How long is the lifetime of a wireless sensor network?," in *Proceedings of the International Conference on Advanced Information Networking and Applications*, pp. 763–770, 2009.

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

[16] R. Khanna, H. Liu, and H.-H. Chen, "Self-organization of sensor networks using genetic algorithms," in *Proceedings of the IEEE International Conference on Communications*, vol. 8, pp. 3377 – 3382, 2006.

[17] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 1415, pp. 2826 – 2841, 2007.

[18] J. R. Patole, "Clustering in wireless sensor network using K-MEANS and MAP REDUCE algorithm," Master's thesis, Department of Computer Engineering and Information Technology, College of Engineering, Pune, 2012.

[19] I. Khemapech, I. Duncan, and A. Miller, "A survey of wireless sensor networks technology," in *Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications*, 2005.

[20] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, pp. 6869 – 6896, 2009.

[21] C.-Y. Chong and S. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247 – 1256, 2003.

[22] F. Martincic and L. Schwiebert, *Introduction to Wireless Sensor Networking*, ch. 1, pp. 1–40. John Wiley and Sons Inc., 2005.

[23] D. C. Steere, A. Baptista, D. McNamee, C. Pu, and J. Walpole, "Research challenges in environmental observation and forecasting systems," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, MobiCom '00, pp. 292–299, 2000.

[24] M. Srivastava, R. Muntz, and M. Potkonjak, "Smart kindergarten: sensor-based wireless networks for smart developmental problem-solving environments," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pp. 132–138, ACM, 2001.

[25] K. Shingha, A. Noor, N. Srivastava, and R. Singh, "Wireless sensor networks in agriculture for potato farming," *International Journal of Engineering Science and Technology*, vol. 2, no. 8, pp. 3955–3963, 2010.

[26] D.-H. Park, B.-J. Kang, K.-R. Cho, C.-S. Shin, S.-E. Cho, J.-W. Park, and W.-M. Yang, "A study on greenhouse automatic control system based on wireless sensor network," *Wireless Personal Communications*, vol. 56, no. 1, pp. 117–130, 2011.

[27] J. P. Lynch and K. J. Loh, "A summary review of wireless sensors and sensor networks for structural health monitoring," *The Shock and Vibration Digest*, vol. 38, no. 2, pp. 91–128, 2006.

[28] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[29] K. Maraiya, K. Kant, and N. Gupta, "Application based study on wireless sensor network," *International Journal of Computer Applications*, vol. 21, no. 8, p. 915, 2011.

[30] V. Katiyar, N. Chand, and N. Chauhan, "Recent advances and future trends in wireless sensor networks," *International Journal of Applied Engineering Research, DINDIGUL*, vol. 1, no. 3, pp. 330 – 342, 2010.

[31] L. Schwiebert, S. K. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pp. 151–165, ACM, 2001.

[32] T. P. Huynh, Y. Tan, and K. J. Tseng, "Energy-aware wireless sensor network with ambient intelligence for smart led lighting system control," in *Proceedings of the 37th Annual Conference on IEEE Industrial Electronics Society*, pp. 2923–2928, 2011.

[33] A. Fernandez-Montes, L. Gonzalez-Abril, J. Ortega, and F. Morente, "A study on saving energy in artificial lighting by making smart use of wireless sensor networks and actuators," *IEEE Network*, vol. 23, no. 6, pp. 16–20, 2009.

[34] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[35] G. Zhao, "Wireless sensor networks for industrial process monitoring and control: A survey," *Network Protocols and Algorithms*, vol. 3, no. 1, pp. 46–63, 2011.

[36] V. Gungor and G. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[37] C. F. Garcia-Hernandez, P. H. Ibarguengoytia-Gonzalez, J. Garcia-Hernandez, and J. A. Perez-Diaz, "Wireless sensor networks and applications: a survey," *International Journal of Computer Science and Network Security*, vol. 17, no. 3, pp. 264–273, 2007.

[38] P. Song, G. Qi, K. Li, and L. Sui, "High performance wireless field measurement system based on wireless sensor network," in *Proceedings of the Workshop on Computational Intelligence and Industrial Application*, vol. 1, pp. 635–639, 2008.

[39] Q. Wang and I. Balasingham, "Wireless sensor networks - an introduction," in *Wireless Sensor Networks: Application - Centric Design* (G. V. Merrett and Y. K. Tan, eds.), InTech, 2010.

[40] S. Hussain, A. Matin, and O. Islam, "Genetic algorithm for energy efficient clusters in wireless sensor networks," in *Proceedings of the fourth International Conference on Information Technology*, pp. 147–154, 2007.

[41] G. Girban and M. Popa, "A glance on wsn lifetime and relevant factors for energy consumption," in *Proceedings of the International Joint Conference on Computational Cybernetics and Technical Informatics*, pp. 523–528, 2010.

[42] Advanticsysy, "Cm5000 tmote sky /main features," 2015.

[43] H. Ba, J. Parvin, L. Soto, I. Demirkol, and W. Heinzelman, "Passive rfid-based wake-up radios for wireless sensor networks," in *Wirelessly Powered Sensor Networks and Computational RFID* (J. R. Smith, ed.), pp. 113–129, Springer New York, 2013.

[44] W. B. Heinzelman, *Application-specific protocol architectures for wireless networks*. PhD thesis, Massachusetts Institute of Technology, 2000.

[45] S. J. Vinay Kumar and S. Tiwari, "Energy efficient clustering algorithms in wireless sensor networks: A survey," *International Journal of Computer Science Issues*, vol. 8, no. 5, pp. 259–268, 2011.

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

[47] M. Karthikeyan and K. Venkatalakshmi, "Energy conscious clustering of wireless sensor network using PSO incorporated cuckoo search," in *Proceedings of the third International Conference on Computing Communication Networking Technologies*, pp. 1–7, 2012.

[48] M. Al-Obaidy, A. Ayesh, and A. F. Sheta, "Optimizing the communication distance of an ad hoc wireless sensor networks by genetic algorithms," *Artificial Intelligence Review*, vol. 29, no. 3, pp. 183–194, 2008.

[49] B. Chen, Y. Zhang, Y. Li, X. Hao, and Y. Fang, "A clustering algorithm of cluster-head optimization for wireless sensor networks based on energy," *Journal of Information and Computational Science*, vol. 8, no. 11, pp. 2129–2136, 2011.

[50] R. Mitra and A. Biswas, "Enhanced clusterhead selection algorithm using leach protocol for wireless sensor networks," *International Journal of Computational Engineering Research*, vol. 2, pp. 766–770, 2012.

[51] H. Chen, C. Zhang, X. Zong, and C. Wang, "LEACH-G: an optimal cluster-heads selection algorithm based on LEACH," *Journal of Software*, vol. 8, no. 10, pp. 2660–2667, 2013.

[52] H. Kour and A. K. Sharma, "Hybrid energy efficient distributed protocol for heterogeneous wireless sensor network," *International Journal of Computer Applications*, vol. 4, no. 5, pp. 37–41, 2010.

[53] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1–8, 2005.

[54] M. Khan, I. Tamim, E. Ahmed, and M. Awal, "Multiple parameter based clustering (MPC): Prospective analysis for effective clustering in wireless sensor network (WSN) using K-Means algorithm," *Wireless Sensor Networks*, vol. 4, no. 1, pp. 18–24, 2012.

[55] G. Y. Park, H. Kim, H. W. Jeong, and H. Y. Youn, "A novel cluster head selection method based on K-Means algorithm for energy efficient wireless sensor network," in *Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops*, pp. 910–915, 2013.

[56] Q. Y. Xie and Y. Cheng, "K-centers min-max clustering algorithm over heterogeneous wireless sensor networks," in *Proceedings of the Wireless Telecommunications Symposium*, pp. 1–6, 2013.

[57] X. H. Li, K. L. Fang, L. Zhang, and J. He, "A clustering algorithm based on k-means for wireless indoor monitoring system," in *Proceedings of the International Conference on Information Technology and Computer Science (ITCS2009)*, vol. 1, pp. 488–492, 2009.

[58] P. Sasikumar and S. Khara, "K-means clustering in wireless sensor networks," in *Proceedings of the Fourth International Conference on Computational Intelligence and Communication Networks*, pp. 140–144, 2012.

[59] J. Hou, X. Fan, W. Wang, J. Jie, and Y. Wang, "Clustering strategy of wireless sensor networks based on improved discrete particle swarm optimization," in *Proccedings of the 2010 Sixth International Conference on Natural Computation (ICNC)*, vol. 7, pp. 3866–3870, 2010.

[60] S. Jin, M. Zhou, and A. S. Wu, "Sensor network optimization using a genetic algorithm," in *Proceedings of the 7th World Multiconference on Systemics,Cybernetics, and Informatics*, 2003.

[61] M. A. Mehr, "Design and implementation a new energy efficient clustering algorithm using genetic algorithm for wireless sensor networks," *World Academy of Science, Engineering and Technology*, vol. 52, pp. 430–433, 2011.

[62] S. Hussain, A. Matin, and O. Islam, "Genetic algorithm for hierarchical wireless sensor networks," *Journal of Networks*, vol. 2, no. 5, pp. 87–97, 2007.

[63] E. Heidari and A. Movaghar, "Intelligent clustering in wireless sensor networks," in *First International Conference on Networks and Communications*, pp. 12–17, 2009.

[64] N. LingaRaj.K, Aradhana.D, "Multiple mobile agents in wireless sensor networks using genetic algorithms," *International Journal of Scientific and Engineering Research*, vol. 3, no. 8, pp. 1–5, 2012.

[65] M. Karimi, H. Naji, and S. Golestani, "Optimizing cluster-head selection in wireless sensor networks using genetic algorithm and harmony search algorithm," in *Proceedings of the 20th Iranian Conference on Electrical Engineering*, pp. 706–710, 2012.

[66] M. Shurman, M. Al-Mistarihi, A. Mohammad, K. Darabkh, and A. Ababnah, "Hierarchical clustering using genetic algorithm in wireless sensor networks," in *Proceedings of the 36th International Convention on Information Communication Technology Electronics Microelectronics*, pp. 479–483, 2013.

[67] J. Tillett, R. Rao, and F. Sahin, "Cluster-head identification in ad hoc sensor networks using particle swarm optimization," in *Proceedings of the IEEE International Conference on Personal Wireless Communications*, pp. 201 – 205, 2002.

[68] S. Guru, S. Halgamuge, and S. Fernando, "Particle swarm optimisers for cluster formation in wireless sensor networks," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 319–324, 2005.

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

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

[71] Z. Ruihua, J. Zhiping, L. Xin, and H. Dongxue, “Double cluster-heads clustering algorithm for wireless sensor networks using pso,” in *Proceedings of the 6th IEEE Conference on Industrial Electronics and Applications*, pp. 763–766, 2011.

[72] K. Bennani and D. El Ghanami, “Particle swarm optimization based clustering in wireless sensor networks: The effectiveness of distance altering,” in *Proceedings of the International Conference on Complex Systems*, pp. 1–4, 2012.

[73] M. Natarajan, R. Arthi, and K. Murugan, “Energy aware optimal cluster head selection in wireless sensor networks,” in *Proceedings of the Fourth International Conference on Computing, Communications and Networking Technologies*, pp. 1–4, 2013.

[74] Z. Jing, T. Le, and Z. Shuaibing, “A novel clustering algorithm based on particle swarm optimization for wireless sensor networks,” in *Proccedings of the 26th Chinese Control and Decision Conference*, pp. 2769–2772, 2014.

[75] N. Abdul Latiff, C. Tsimenidis, B. Sharif, and C. Ladha, “Dynamic clustering using binary multi-objective particle swarm optimization for wireless sensor networks,” in *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–5, 2008.

[76] Z. W. Siew, C. H. Wong, C. S. Chin, A. Kiring, and K. Teo, “Cluster heads distribution of wireless sensor networks via adaptive particle swarm optimization,” in *Proceedings of the fourth International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 78–83, 2012.

[77] A. Zahmatkesh and M. H. Yaghmaee, “A genetic algorithm-based approach for energy- efficient clustering of wireless sensor networks,” *International Journal of Information and Electronics Engineering*, vol. 2, no. 2, pp. 165–169, 2012.

[78] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975.

[79] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1989.

[80] D. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic,Boston, MA, USA, 2002.

[81] K. A. De Jong, *Analysis of Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, University of Michigan, Ann Arbor, MI, 1975.

[82] K. A. De Jong, “Adaptive system design: A genetic approach,” *IEEE Transaction Systems, Man, and. Cybernatics*, vol. 10, no. 3, pp. 556–574, 1980.

[83] K. De Jong, "Learning with genetic algorithms: An overview," *Machine Learning*, vol. 3, no. 3, pp. 121–138, 1988.

[84] S. Auwatanamongkol, "Pattern recognition using genetic algorithm," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 822–828, 2000.

[85] K. Khoo and P. Suganthan, "Structural pattern recognition using genetic algorithms with specialized operators," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 1, pp. 156–165, 2003.

[86] Y.-K. Wang and K.-C. Fan, "Applying genetic algorithms on pattern recognition: an analysis and survey," in *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 2, pp. 740–744 vol.2, 1996.

[87] T. Ueyama, T. Fukuda, and F. Arai, "Coordinate planning using genetic algorithm-structure configuration of cellular robotic system," in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 249–254, 1992.

[88] V. Ayala-Ramirez, A. Perez-Garcia, F. Montecillo-Puente, R. Sanchez-Yanez, and E. Martinez-Labrada, "Path planning using genetic algorithms for mini-robotic tasks," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3746–3750, 2004.

[89] Z. Yongnian, Z. Lifang, and L. Yongping, "An improved genetic algorithm for mobile robotic path planning," in *Proceedings of the 24th Chinese Control and Decision Conference*, pp. 3255–3260, 2012.

[90] R. Garcia, M. de Oliveira, and J. Maldonado, "Genetic algorithms to support software engineering experimentation," in *Proceedings of the International Symposium on Empirical Software Engineering*, pp. 1–10, 2005.

[91] A. F. Sheta, "Estimation of the cocomo model parameters using genetic algorithms for nasa software projects," *Journal of Computer Science*, vol. 2, no. 2, p. 118, 2006.

[92] H. Jiang, C. Chang, D. Zhu, and S. Cheng, "A foundational study on the applicability of genetic algorithm to software engineering problems," in *IEEE Congress on Evolutionary Computation*, pp. 2210–2219, 2007.

[93] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshynanyk, and A. De Lucia, "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms," in *Proceedings of the 35th International Conference on Software Engineering*, pp. 522–531, 2013.

[94] A. Sheta and K. D. Jong, "Parameter estimation of nonlinear systems in noisy environment using genetic algorithms," in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 360–366, 1996.

[95] M. Braik and A. Sheta, *Exploration of Genetic Algorithms and Particle Swarm Optimization in Improving the Quality of Medical Images*, pp. 329–360. Lambert Academic Publishing (LAP), Germany, 2011.

[96] A. Thengade and R. Dondal, "Genetic algorithm - survey paper," *IJCA Proceedings on National Conference on Recent Trends in Computing*, no. 5, pp. 25–29, 2012.

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

[98] B. Al-kazemi and C. Mohan, "Training feedforward neural networks using multiphase particle swarm optimization," in *Proceedings of the 9th International Conference on Neural Information Processing*, vol. 5, pp. 2615–2619, 2002.

[99] M. Braik, A. Sheta, and A. Arieqat, "A comparison between ga and pso in training ann to model the te chemical process reactor," *Proceedings of the AISB symposium on swarm intelligence algorithms and applications*, vol. 11, pp. 24–30, 2008.

[100] I. Vilovic, N. Burum, and D. Milic, "Using particle swarm optimization in training neural network for indoor field strength prediction," in *Proceedings of the ELMAR International Symposium*, pp. 275–278, 2009.

[101] A. Bezborah, "A hardware architecture for training of artificial neural networks using particle swarm optimization," in *Proceedings of the third International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, pp. 67–70, 2012.

[102] I. Aksu and R. Coban, "Training the multifeedback-layer neural network using the particle swarm optimization algorithm," in *Proceedings of the International Conference on Electronics, Computer and Computation*, pp. 172–175, 2013.

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

[104] M. AlRashidi and M. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913–918, 2009.

[105] H. Shen, Y. Zhu, T. Liu, and L. Jin, "Particle swarm optimization in solving vehicle routing problem," in *Proceedings of the Second International Conference on Intelligent Computation Technology and Automation*, vol. 1, pp. 287–291, 2009.

[106] V. Gudise and G. Venayagamoorthy, "Evolving digital circuits using particle swarm," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 468–472, 2003.

[107] G. Yuan-Liang and X. Wen-Bo, "Study on automatic test generation of digital circuits using particle swarm optimization," in *Proceedings of the tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*, pp. 324–328, 2011.

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

[109] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance," in *Proceedings of the 1999 Congress on Evolutionary Computation*, vol. 3, pp. 1931–1938, 1999.

[110] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the 2002 Congress on Evolutionary Computation*, vol. 2, pp. 1671 – 1676, 2002.

[111] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*, pp. 69–73, 1998.

[112] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[113] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, no. 6, pp. 317–325, 2003.

[114] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.

[115] R. Umarani and V. Selvi, "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications*, vol. 5, no. 4, pp. 1–6, 2010.

[116] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[117] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 2006.

[118] M.-T. Chiang, *Intelligent K-means clustering in L2 and L1 versions: Experimentation and Applications*. PhD thesis, School of Computer Science and Information Systems, University of London, 2009.

[119] C. Zhang and S. Xia, "K-means clustering algorithm with improved initial center," in *Proceedings of the second International Workshop on Knowledge Discovery and Data Mining*, pp. 790–792, 2009.

[120] G. A. Wilkin and X. Huang, "K-Means clustering algorithms: Implementation and comparison," in *Proceedings of the second International Multi-Symposiums on Computer and Computational Sciences*, pp. 133–136, 2007.

[121] C. Vercellis, *Business Intelligence: Data Mining and Optimization for Decision Making*. Wiley, 2011.

[122] S. Na, L. Xumin, and G. Yong, "Research on k-means clustering algorithm: An improved k-means clustering algorithm," in *Proceedings of the third International Symposium on Intelligent Information Technology and Security Informatics*, pp. 63 – 67, 2010.

[123] C. A. Balanis, *Antenna Theory: Analysis and Design*. John Wiley & Sons, 3rd ed., 2005.

[124] B. Birge, "Psot - a particle swarm optimization toolbox for use with matlab," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pp. 182–186, 2003.