Democratic and Popular Republic of Algeria

Ministry of Higher Education and Scientific Research

The University of Adrar

Faculty of Sciences and Technology

Department of Mathematics and Computer Science

Master Thesis in Computer Science

Option: Networks and Intelligent Systems

Title:

Efficient Sensors Network Management: Global and Local Clustering based on Genetic Algorithms and Fuzzy Logic

Presented By:

- HARMA Abdelkarim
- BENAICH Salem

Jury members :

President : Mr. ELMIR Youssef
Advisor : Dr. OMARI Mohammed
Examiner : Mr. CHERAGUI Mohamed Amine

November 2013

Abstract

Energy consumption is one of the major factors to be considered in wireless sensors networks (WSNs). In fact, routing protocols in WSNs are different than those applied in regular ad hoc networks since energy consumption is rarely considered in the latter. Sensors' clustering is a class of protocols where sensors are gathered into classes and only a cluster head is responsible for transmission to the base station, thus, reducing energy consumption in regular sensors. The LEACH protocol is a clustering protocol that selects cluster head sensors based on their residual energy. LEACH-GA is a modified version of LEACH where the base station determines sensors' clustering based on a genetic algorithm method. In this thesis, we introduce a protocol (GAGLCP) that exchanges roles between regular nodes and cluster heads in a round robin manner following the token ring methodology. The equi-distribution of cluster head burden over all sensors in the same cluster reduces the need of expensive periodic reclustering. In addition, an initial division of the WSN into domains of limited number of sensors is performed in order to reduce the GA search time. The domain memberships of edge sensors are handled through fuzzy logic based on the residual energy. We conducted several simulations of the GAGLCP and the LEACH protocols to test the residual energy and the number of life nods over rounds of clustering. The simulation results showed a better performance of GAGLCP over LEACH in terms of extending the lifetime of a WSN network.

Keywords: Global clustering, local clustering, wireless sensors, genetic algorithms, fuzzy logic sets, routing.

Dedicates

BENAICH Salem

I would like to dedicate this thesis to my parents,
For their endless love, support and encouragement

HARMA Abdelkarim

This thesis is dedicated to my darling parents and all my family, and any one give me the courage to go on

Acknowledgements

It would not have been possible to write this thesis without the help and support of the kind people around as, to only some of whom it is possible to give particular mention here.

This thesis would not have been possible without the help, support and patience of ours principal supervisor, Prof. OMARI Mohammed, not to mention his advice and unsurpassed knowledge of problems. The good advice, support and friendship of ours friend Prof. DEMRI Mohammed, has been invaluable on both an academic and a personal level, for which we are extremely grateful.

Table of Contents

List of Figures

List of Tables

INTRODUCTION

A wireless sensor network consists of spatially distributed autonomous sensor nodes to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants [1][2]. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance. They are now used in many industrial and civilian application areas, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control. Data collected in WSNs are transmitted among all nodes through routing, and then converged into the sink node [26]. Therefore, the routing algorithm determines whether the information can be transmitted completely or not, and greatly affect the performance of the network.

Sensor nodes are usually powered by lightweight batteries, and replacing or recharging these batteries is often not feasible [2][26]. Therefore, in many cases, the lifetime of a sensor network is over as soon as the battery power in critical node(s) is depleted [27]. Therefore, many protocols were proposed to reduce the energy consumption throughout the whole network and, as a result, extend the network lifetime. Some of these protocols depend on reduction of communication overhead over long paths through using multi-hopping from the source node to the base station. Other protocols suggested improvements through routing algorithms [26] [27] [28]. One of the most important algorithms that were suggested in the field of WSNs is the LEACH algorithm [1], which introduced an innovative implementation of clustering, aggregation, and dynamic scheduling. Despite these capabilities of LEACH, it still has a weakness directed toward the selection of the optimal number of cluster heads.

The primary objective of this thesis is to develop a new clustering mechanism based on the LEACH strategy, yet enhancing the lifetime of sensors by shifting the clustering load from the base station to the cluster itself. Therefore, we aim to determine the optimal cluster heads and reducing the energy consumption. Genetic algorithms are a branch of stochastic yet guided optimization methods that has been introduced recently in the domain of efficient clustering [26]. Fuzzy logic was also used to find the best clustering that can survive more rounds on.

The thesis is organized as follows. In Chapter 1 we present a general overview of wireless sensor networks (WSNs) and their application field. In Chapter 2, a background of genetic algorithms and fuzzy logic is presented followed by their application in the WSNs field in Chapter 3 especially the series of LEACH methods. Our mechanism of enhancing the lifetime of a WSN based on genetic algorithms, fuzzy logic and token ring approach is introduced in Chapter 4 with details of the proposed method phases. In Chapter 5 simulation results are presented along with some interpretation and analysis. Chapter 6 is the conclusion.

CHAPTER I

Wireless Sensor Networks:

Definitions and Basics

1.1. Introduction

Wireless Sensor Networks (WSNs) are considered to be one of the potential emerging computing technologies, edging closer towards widespread feasibility [5]. Cheap and smart sensors networked through wireless communication with the Internet hold remarkable prospects for controlling and monitoring environment, homes, health care, military, and other strategic applications.

The Base Station (BS) selects a number of sensor nodes to act as Cluster Heads (CHs). Each non-cluster head node is associated to a CH thereby forming the clusters. After that, each node senses the environment and measures physical phenomenon of interest (e.g., temperature, pressure, smoke, humidity). Each node then aggregates and transmits its measurements and information to its associated CH. The CH compresses this data in a single signal and transmits it to the BS (sink).

1.2. Sensor Network Applications

The original motivation behind the research into WSNs was military application. Examples of military sensor networks include large-scale acoustic ocean surveillance systems for the detection of submarines, self-organized and randomly deployed WSNs for battlefield surveillance and attaching micro sensors to weapons for stockpile surveillance [4]. As the costs for sensor nodes and communication networks have been reduced, many other potential applications including those for civilian purposes have emerged. The following are a few examples.

1.2.1. Environmental monitoring

Environmental monitoring [4] can be used for animal tracking, forest surveillance, flood detection, and weather forecasting. It is a natural candidate for applying WSNs [5], because the variables to be monitored, e.g. temperature, are usually distributed over a large region They collect data from sensor nodes installed within the ice and the sub-glacial sediment without the use of wires which could disturb the environment.

Their WSN deployment is used to provide spatially dense measures to the Swiss authorities in charge of risk management, and the resulting model will assist in the prevention of avalanches and accidental deaths.

1.2.2. Health monitoring

WSNs can be embedded into a hospital building to track and monitor patients and all medical resources [4]. Special kinds of sensors which can measure blood pressure, body temperature and electrocardiograph (ECG) can even be knitted into clothes to provide remote nursing for the elderly.

1.2.3. Traffic control

Sensor networks have been used for vehicle traffic monitoring and control for some time [4]. At many crossroads, there are either overhead or buried sensors to detect vehicles and to control the traffic lights. Furthermore, video cameras are also frequently used to monitor road segments with heavy traffic. However, the traditional communication networks used to connect these sensors are costly, and thus traffic monitoring is usually only available at a few critical points in a city [5]. WSNs will completely change the landscape of traffic monitoring and control by installing cheap sensor nodes in the car, at the parking lots, along the roadside, etc. Street line incorporation is a company which uses sensor network technology to help drivers find unoccupied parking places and avoid traffic jams.

The solutions provided by Street line can significantly improve the city traffic management and reduce the emission of carbon dioxide.

1.2.4. Industrial sensing

As plant infrastructure ages, equipment failures cause more and more unplanned downtime [7]. The ARC Advisory Group estimates that 5% of production in North America is lost to unplanned downtime. Because sensor nodes can be deeply embedded into machines and there is no infrastructure, WSNs make it economically feasible to monitor the “health” of machines and to ensure safe operation. Aging pipelines and tanks have become a major problem in the oil and gas industry. Monitoring corrosion using manual processes is extremely costly, time consuming, and unreliable. A network of wireless corrosion sensors can be economically deployed to reliably identify issues before they become catastrophic failures.

1.2.5. Infrastructure security

WSNs can be used for infrastructure security and counterterrorism applications. Critical buildings and facilities such as power plants, airports, and military bases have to be protected from potential invasions. Networks of video, acoustic, and other sensors can be deployed around these facilities [5]. An initiative in Shanghai Pudong International Airport has involved the installation of a WSN-aided intrusion prevention system on its periphery to deter any unexpected intrusions.

1.3. Sensors Types and Characteristics

There are different types of wireless smart sensors currently in use [6]. A more representative example is the sensor nodes of the smart dust project developed at UC Berkeley [3]. The sensor node is supported by Tiny OS, UC Berkeley's open source operating system for sensor networks. Each sensor node has limited resources comprising of an 8 bit, 4 MHz CPU with 8K of Instruction flash memory, 512 bytes of RAM and 512 bytes of EEPROM with 10 Kbps communication and 433 MHz radio.

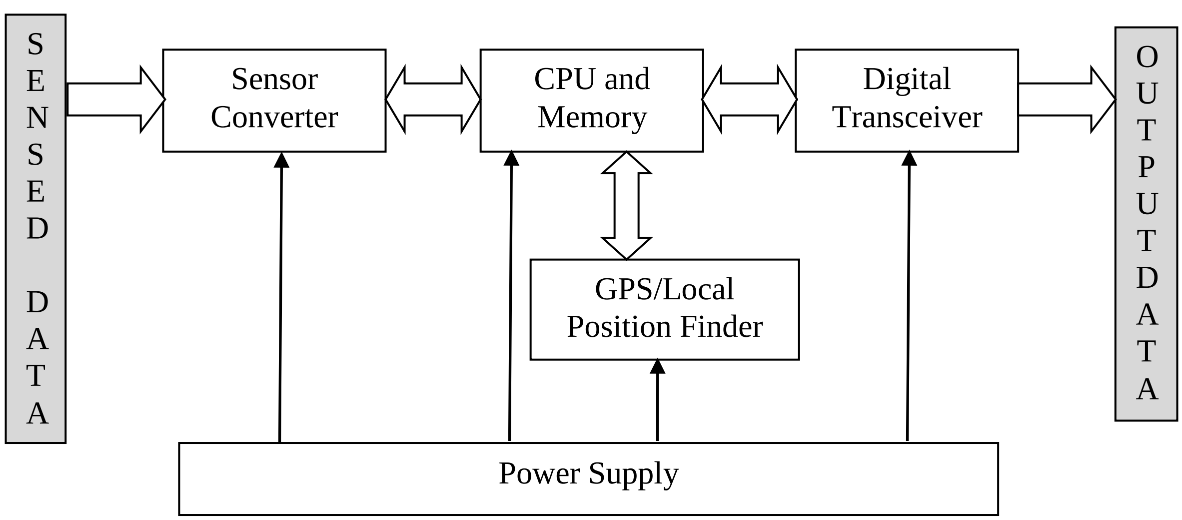

Figure 1.1: Sensor hardware platform.

Different characteristics of sensor nodes (Figure 1.1) include size, battery consumption, power level, lifetime of operation, movement characteristics (indicating whether the nodes are stationary or mobile), position characteristics (indicating whether the nodes are embedded into the system or independent of its surroundings), failure characteristics (indicating if the sensor has failed, or is degrading slowly) [6].

1.4. Sensor Network Architecture

Typical sensor network architecture consists of a sensor field, which is the physical environment where the sensor nodes or devices are deployed (see Figure 1.2). Sensor nodes can possibly be deployed in extremely large numbers, on the order of thousands of sensor nodes in the field. Consequently, the cost of these nodes should be low. A low-cost device can thus be expected to have fairly limited computational and communication capabilities, considering the fact that sensing capabilities are also to be included in the device. Moreover, in many applications, sensor nodes are deployed in hostile areas or physically inaccessible regions where it is not easy to have human intervention to maintain sensor nodes. Such sensor nodes have to operate on limited battery power and the batteries cannot be replaced easily. In such cases, sensor nodes have to be designed so that power-consuming operations such as the central processing unit or the radio used for communications are shut down when they are not being used. Of course, for specific applications (e.g. physical intrusion detection using cameras), sensor nodes may have more advanced capabilities. Thus, sensor devices may range from millimeter-sized devices fabricated on custom silicon to more general purpose cell-phone-sized devices with advanced capabilities.

Figure 1.2 shows a schematic of simple sensor network architecture. Sensor nodes with limited capabilities deployed in the sensor field communicate to a powerful BS that links them to the Internet and a central manager for processing the sensed data. Communications to the BS have to go through several sensor nodes first, because all sensor nodes will not be typically able to communicate directly with the BS. This may be due to limited communication range, distance from the BS, intermittent sensor activity, and so on.

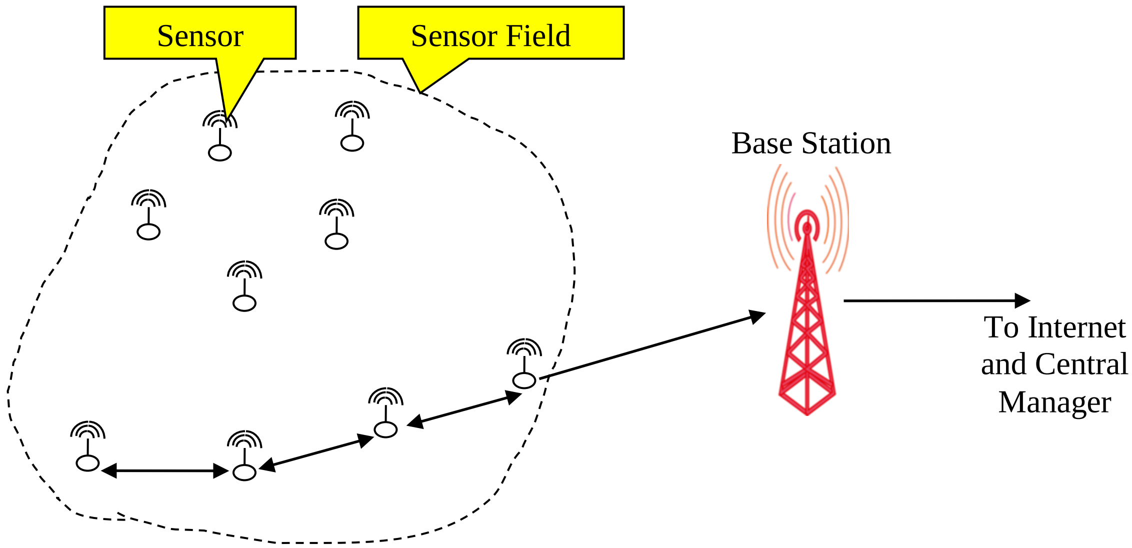

Figure 1.2: Typical sensor network architecture.

The simple architecture shown in Figure 1.2 is expanded upon by Hill et al [4], where four classes of sensor networking devices are described. At the lowest level of the hierarchy, the actual sensing device could be very specialized, with a tiny form factor and very limited capabilities. Such devices may not even be capable of receiving information and may simply transmit information when they sense an event or perform other application- related activity. We will call these devices sub motes in this chapter for ease of reference.

A WSN is a network consisting of numerous sensor nodes with sensing, wireless communications and computing capabilities. These sensor nodes are scattered in an unattended environment (i.e. sensing field) to sense the physical world. The sensed data can be collected by a few sink nodes which have accesses to infrastructured networks like the Internet. Finally, an end user can remotely fetch the sensed data by accessing infrastructured networks. Figure 1.3 shows the operation sketch map of WSNs.

In Figure 1.3, two kinds of network topologies are shown. The sensor nodes in a flat network also act as routers and transfer data to a sink (base station). Whereas, in a clustered network, regular nodes send data to a central node (cluster head), which is responsible for transferring data to the base station.

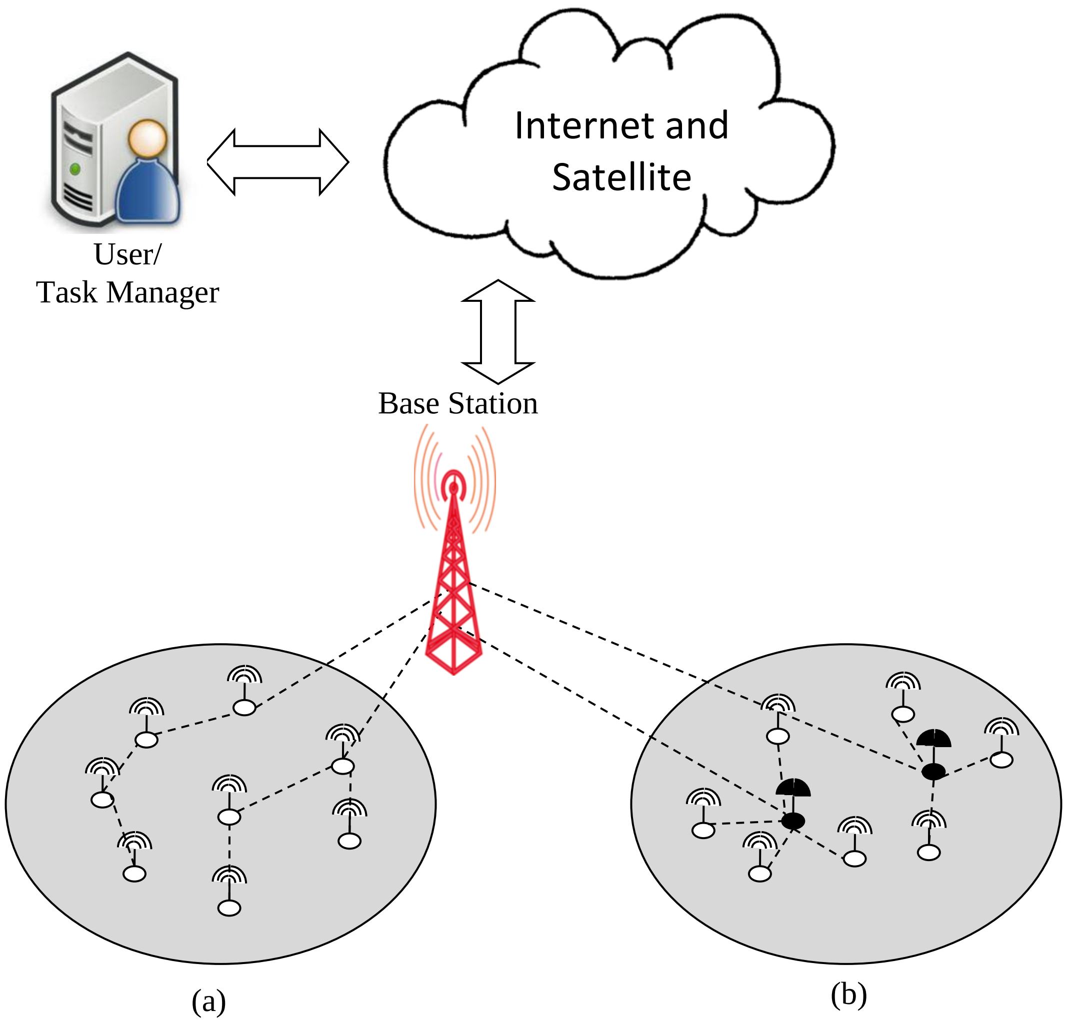

Figure 1.3: The Operation of WSNs: (a) Flat sensor network (b) Clustered sensor network.

1.5. WSN Topologies

As explained earlier, WSNs consist of many inexpensive, portable wireless nodes, with limited power, memory and computational capabilities. The energy supply of the sensor nodes is one of the main constraints in the design of this type of network [4]. Since it is infeasible to replace batteries once WSNs are deployed, an important design issue in WSNs is to lessen the energy consumption with the use of energy conserving hardware, operating systems, and communication protocols.

WSNs have one or more centralized control units called the base station. The base station serves as a gateway for each sensor node to send data to another network (Figure 1.2). Thus it can be an interface to interact with the network, to extract and transfer information to the sensor nodes. Unlike sensor nodes, base stations are many times more powerful and have an AC power supply, high communication bandwidth, and larger processing power and storage facilities [2].

The energy consumption in a WSN can be reduced by allowing only some nodes to communicate with the base station. These nodes called cluster-heads collect the data sent by each node in that cluster, compressing it and then transmitting the aggregated data to the base station [1]. The model is suitable considering the amount of redundancy found in WSNs; direct transmissions the base station will consume large amount of transmit power from each node.

For a WSN we make the following assumptions:

- The base station is located far from the sensor nodes and is immobile.
- All nodes in the network are homogeneous and energy constrained.
- Symmetric propagation channel.
- Base station performs the cluster-head election.

- Nodes have location information that they send to the base station with respective energy levels during set up phase.
- Nodes have little or no mobility.

1.6. Problem of Cluster-Head Selection

Clustering helps the nodes to minimize the overall energy dissipation in the network by allowing only some nodes to take part in the transmission to the base station. Moreover it also helps to reuse the bandwidth and thus utilizes better resource allocation and improved power control. LEACH (Low Energy Adaptive Clustering Hierarchy) is a popular current approach for cluster-head selection and has formed the basis for many other approaches [2] [5] [7]. Algorithms like LEACH use only the local information in the nodes to select cluster-heads stochastically. But this method of selecting cluster-heads using only local information has its own limitations. Since each node probabilistically decides whether or not to become the cluster-head, there might be cases when two cluster-heads are selected in close vicinity of each other. Moreover, the node selected can be located near the edges of the network, in which the other nodes will expend more energy to transmit data to that cluster-head.

In fact considering only one factor, like energy, is not suitable to elect the cluster-head properly. This is because other conditions like centrality of the nodes with respect to the entire cluster, also gives a measure of the energy dissipation during transmission for all nodes. The more central the node is to a cluster, the more is the energy efficiency for other nodes to transmit through that selected node. The concentration of the nodes in a given region also affects in some way for proper cluster-head election. It is more reasonable to select a cluster-head in a region, where the node concentration is high.

1.7. Advantages of Base Station Control Algorithm

A central control algorithm in the base station will produce better decision, since the base station has the global knowledge about the network. Moreover, base stations are many times more powerful than the sensor nodes, having sufficient memory, power and storage. In this approach energy is spent to transmit the location information of all the nodes to the base station (possibly using a GPS receiver). Considering WSNs are meant to be deployed over a geographical area with the main purpose of sensing and gathering information, we assume that nodes have minimal mobility, thus sending the location information during the initial setup phase is sufficient.

1.8. Disadvantages of Local Information Processing

Several disadvantages are there for selecting the cluster-head using only the local information in the nodes as in the case of LEACH:

- Since each node probabilistically decides whether or not to become the cluster head, there might be cases when two cluster-heads are selected in close vicinity of each other increasing the overall energy depleted in the network.
- The number of cluster-head nodes generated is not fixed so in some rounds it may be more or less than the preferred value.
- The node selected can be located near the edges of the network, wherein the other nodes will expend more energy to transmit data to that cluster-head.
- Each node has to calculate the threshold and generate the random numbers in each round, consuming CPU cycles.

CHAPTER II

Overview on Genetic Algorithms and Fuzzy Logic

2.1. Genetic Algorithms: Introduction

Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure, and apply recombination operators to these structures in such a way as to preserve critical information [26].

Genetic algorithms are often viewed as function optimizers, although the range of problems to which genetic algorithms have been applied is quite broad.

An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to 'reproduce' than those chromosomes which are poorer solutions. The 'goodness' of a solution is typically defined with respect to the current population.

The genetic algorithm (GA) is an optimization and search technique based on the principles of genetics and natural selection. A GA allows a population composed of many individuals to evolve under specified selection rules to state that maximizes the “fitness” (i.e., minimizes the cost function).

2.1.1. Evolution and genetic algorithms

Evolutionary computing was introduced in the 1960s by I.RECHENBERG in the work “Evolution strategies” [22]. This idea was then developed by other researches. Genetic Algorithms (GAs) was invented by John Holland and developed this idea in his book “Adaptation in natural and artificial systems” in the year 1975. Holland proposed GA as a heuristic method based on “Survival of the fittest”. GA was discovered as a useful tool for search and optimization problems.

Recombination or sexual reproduction is a key operator for natural evolution [22].Technically, it takes two genotypes and it produces a new genotype by mixing the gene found in the originals. In biology, the most common form of recombination is crossover, two chromosomes are cut at one point and the halves are spliced to create new chromosomes. The effect of recombination is very important because it allows characteristics from two different parents to be assorted. If the father and the mother possess different good qualities, we would expect that all the good qualities will be passed into the child. Thus the offspring, just by combining all the good features from its parents, may surpass its ancestors. Many people believe that this mixing of genetic material via sexual reproduction is one of the most powerful features of Genetic Algorithms. As a quick parenthesis about sexual reproduction, Genetic Algorithms representation usually does not differentiate male and female individuals(without any perversity). As in many livings species (e.g., snails) any individual can be either a male or a female. In fact, for almost all recombination operators, mother and father are interchangeable.

The magnificent phenomenon called the evolution of species can also give some insight into information processing methods and optimization in particular. According to Darwinism, inherited variation is characterized by the following properties:

- Variation must be copying because selection does not create directly anything, but presupposes a large population to work on.
- Variation must be small-scaled in practice. Species do not appear suddenly.
- Variation is undirected. This is also known as the blind watchmaker paradigm.

While the natural sciences approach to evolution has for over a century been to analyze and study different aspects of evolution to find the underlying principles, the engineering sciences are happy to apply evolutionary principles, that have been heavily tested over billions of years, to attack the most complex technical problems, including protein folding.

2.1.2. GA operators

The operation of GAs begins with a population of a random string representing design or decision variables [26]. The population is then operated by three main operators; reproduction, crossover and mutation to create a new population of points.

The operators are described in the following steps.

2.1.2.1. Reproduction

Reproduction (or selection) is an operator that makes more copies of better strings in a new population [22]. Reproduction is usually the first operator applied on a population. Reproduction selects good strings in a population and forms a mating pool. To sustain the generation of a new population, the reproduction of the individuals in the current population is necessary. For better individuals, these should be from the fittest individuals of the previous population.

There exist a number of reproduction operators in GA literature, but the essential idea in all of them is that the above average strings are picked from the current population and their multiple copies are inserted in the mating pool in a probabilistic manner.

2.1.2.2. Crossover

A crossover operator is used to recombine two strings to get a better string[22]. In crossover operation, recombination process creates different individuals in the successive generations by combining material from two individuals of the previous generation. In reproduction, good strings in a population are probabilistic-ally assigned a larger number of copies and a mating pool is formed. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool.

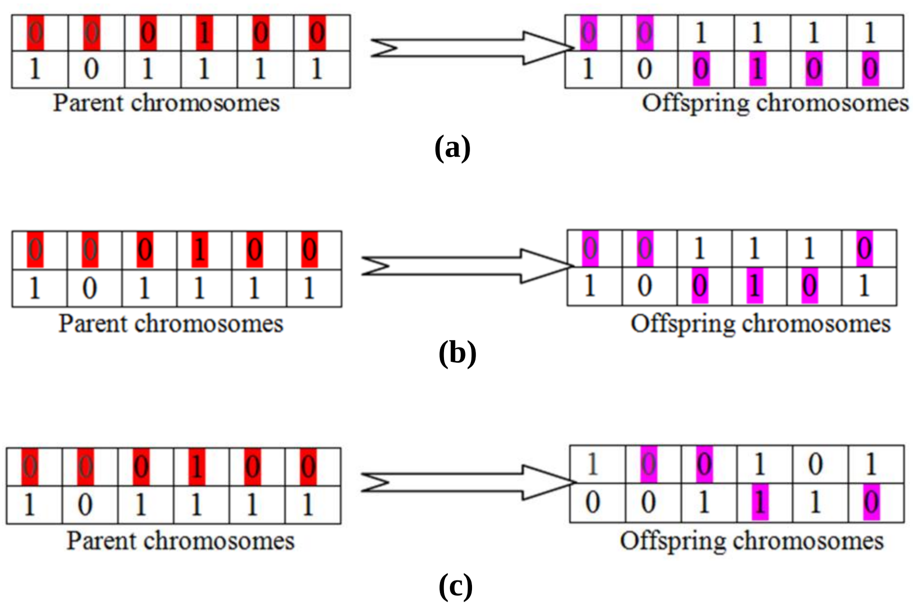

Figure 2.1: Crossover methods: (a) One-point (b) Two point (c) Uniform.

In one site crossover, a crossover site is selected randomly (shown as vertical lines). The portion rights of the selected site of these two strings are exchanged to form a new pair of strings. The new strings are thus a combination of the old strings.

Two site crossovers is a variation of the one site crossover, except that two crossover sites are chosen and the bits between the sites are exchanged as shown in Figure 2.1.

2.1.2.3. Mutation

Mutation adds new information in a random way to the genetic search process and ultimately helps to avoid getting trapped at local optima [22]. It is an operator that introduces diversity in the population whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operators. Mutation may cause the chromosomes of individuals to be different from those of their parent individuals.

The need for mutation is to create a point in the neighborhood of the current point, thereby achieving a local search around the current solution. The mutation is also used to maintain diversity in the population. For example, the following population having four eight bit strings may be considered: 01101011, 00111101, 00010110, and 01111100. It can be noticed that all four strings have a 0 in the left most bit position. If the true optimum solution requires 1 in that position, then neither reproduction nor crossover operator described above will be able to create 1 in that position. The inclusion of mutation introduces probability pm of turning 0 into 1.

2.1.2.4. Replacement

Once the new offspring solutions are created using crossover and mutation, we need to introduce them into the parental population[27]. There are many ways we can approach this. Bear in mind that the parent chromosomes have already been selected according to their fitness, so we are hoping that the children (which include parents which did not undergo crossover) are among the fittest in the population and so we would hope that the population will gradually, on average, increase its fitness. Some of the most common replacement techniques are outlined below.

- **Delete-all:** This technique deletes all the members of the current population and replaces them with the same number of chromosomes that have just been created. This is probably the most common technique and will be the technique of choice for most people due to its relative ease of implementation. It is also parameter-free, which is not the case for some other methods.

- **Steady-state:** This technique deletes n old members and replaces them with n new members. The number to delete and replace, n, at any one time is a parameter to this deletion technique. Another consideration for this technique is deciding which members to delete from the current population. Do we delete the worst individuals, pick them at random or delete the chromosomes that you used as parents? Again, this is a parameter to this technique.

- **Steady-state-no-duplicates:** This is the same as the steady-state technique but the algorithm checks that no duplicate chromosomes are added to the population. This adds to the computational overhead but can mean that more of the search space is explored.

2.1.3. A simple genetic algorithm

Given a clearly defined problem to be solved and a bit string representation for candidate solutions, a simple GA works as follows:

1. Start with a randomly generated population of n l–bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness f(x) of each chromosome x in the population.
3. Repeat the following steps until n offspring have been created:
   a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is

done "with replacement," meaning that the same chromosome can be selected more than once to become a parent.

b. With probability p (the "crossover probability" or "crossover rate"), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring.

   If no crossover takes place, form two offspring that are exact copies of their respective parents.

   (Note that here the crossover rate is defined to be the probability that two parents will crossover in a single point. There are also "multi−point crossover" versions of the GA in which the crossover rate for a pair of parents is the number of points at which a crossover takes place.)

c. Mutate the two offspring at each locus with probability p (the mutation probability or mutation rate), and place the resulting chromosomes in the new population.

   If n is odd, one new population member can be discarded at random.

4. Replace the current population with the new population.
5. Go to step 2.

Each iteration of this process is called a generation. A GA is typically iterated for anywhere from 50 to 500 or more generations. The entire set of generations is called a run. At the end of a run there are often one or more highly fit chromosomes in the population. Since randomness plays a large role in each run, two runs with different random−number seeds will generally produce different detailed behaviors. GA researcher’s often report statistics (such as the best fitness found in a run and the generation at which the individual with that best fitness was discovered) averaged over many different runs of the GA on the same problem [27].

The simple procedure just described is the basis for most applications of GAs. There are a number of details to fill in, such as the size of the population and the probabilities of crossover and mutation, and the success of the algorithm often depends greatly on these details.

The Genetic algorithm process is discussed through the GA cycle in Figure 2.2:

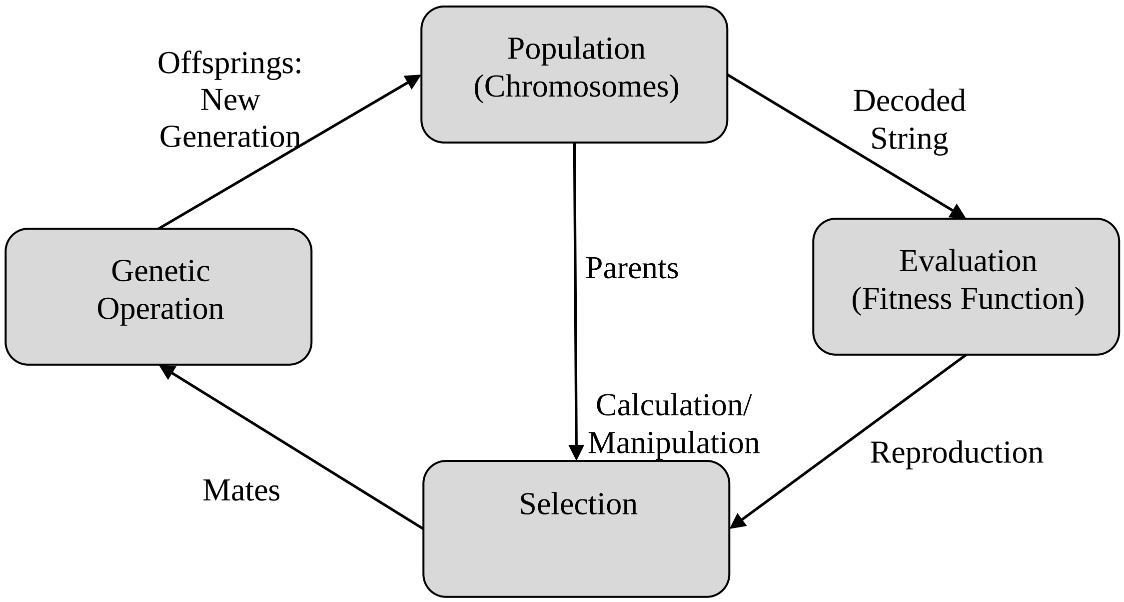

Figure 2.2: Genetic algorithm cycle.

Reproduction is the process by which the genetic material in two or more parentis combined to obtain one or more offspring. In fitness evaluation step, the individual’s quality is assessed. Mutation is performed to one individual to produce a new version of it where some of the original genetic material has been randomly changed. Selection process helps to decide which individuals are to be used for production and mutation in order to produce new search points.

The flowchart showing the process of GA is as shown in Figure 2.3.

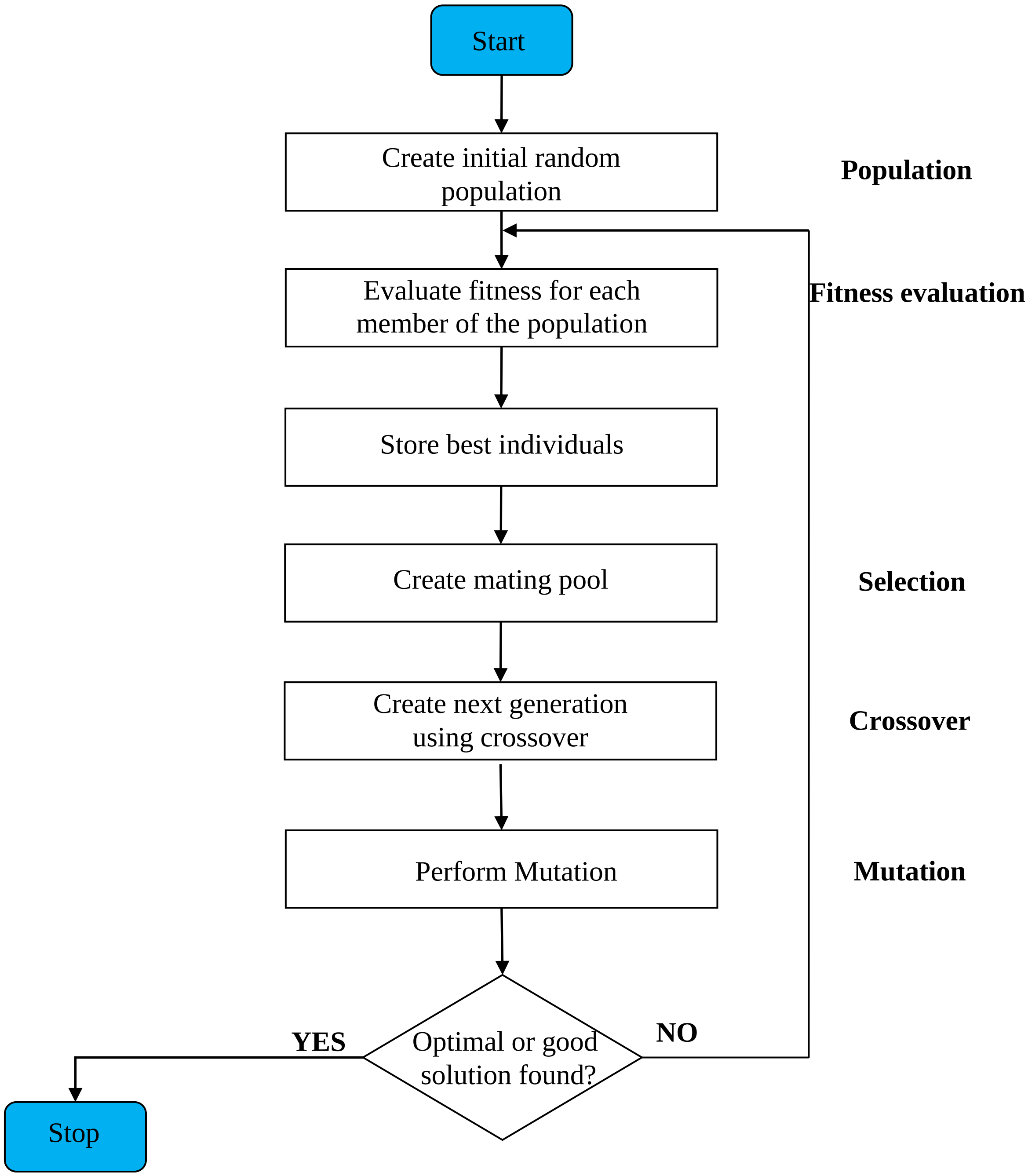

Figure 2.3: Flowchart of genetic algorithm.

Based on the above, the important criteria for GA approach can be formulated as given below:

- **Completeness:** Any solution should have its encoding
- **Non redundancy:** Codes and solutions should correspond one to one
- **Soundness:** Any code (produced by genetic operators) should have its corresponding solution
- **Characteristic perseverance:** Offspring should inherit useful characteristics from parents.
- In short, the basic four steps used in simple Genetic Algorithm to solve a problem are:
    - The representation of the problem.
    - The fitness calculation.
    - Various variables and parameters involved in controlling the algorithm.
    - The representation of result and the way of terminating the algorithm.

2.1.4. Applications of genetic algorithms

Genetic algorithms have been used for difficult problems (such as NP-hard problems), for machine learning and also for evolving simple programs[22]. They have been also used for some art, for evolving pictures and music. A few applications of GA are as follows:

- Nonlinear dynamical systems–predicting, data analysis
- Robot trajectory planning
- Evolving LISP programs (genetic programming)
- Strategy planning

- Finding shape of protein molecules
- TSP and sequence scheduling
- Functions for creating images
- Control–gas pipeline, pole balancing, missile evasion, pursuit
- Design–semiconductor layout, aircraft design, keyboard configuration, communication networks
- Scheduling–manufacturing, facility scheduling, resource allocation
- Machine Learning–Designing neural networks, both architecture and weights, improving classification algorithms, classifier systems
- Signal Processing–filter design
- Combinatorial Optimization–set covering, traveling salesman (TSP), Sequence scheduling, routing, bin packing, graph coloring and partitioning

2.2. Fuzzy Logic

2.2.1. Introduction

Fuzzy logic is based on the idea that all things admit of degrees [28]. It attempts to model our sense of words, our decision making and our common sense.

In 1965 Lotfi Zadeh [28], extended the work on possibility theory into a formal system of mathematical logic with the application of natural language terms to create 'Fuzzy Logic'.

Unlike Boolean logic having two values, fuzzy logic is multi-valued and uses continuum of logical values or degrees of membership between 0 and 1.

2.2.2. Crisp and fuzzy sets

The basic idea of the fuzzy set theory is that an element belongs to a fuzzy set with a certain degree of membership [23]. This degree is usually taken as a real number in the interval [0, 1]. For a given fuzzy set, the x-axis represents the universe of discourse – the range of all possible values applicable to a chosen variable and the y-axis represents the membership value of the fuzzy set.

The characteristic function of a crisp set A, can be defined as $f_A(x)$, where X is the universe of discourse with its elements denoted as x.

$$f_A(x) = \begin{cases} 1, if\ x \in A \\ 0, if x \notin A \end{cases}$$

$$f_A(x): X \rightarrow \{0,1\}$$

Hence, for any element x of universe X, characteristic function $f_A(x)$ is equal to 1 if x is an element of set A, and is equal to 0 if x is not an element of A.

In fuzzy logic, fuzzy set A of universe X is defined by the membership function [8] of set A, $(x)_A$m . For any element x of universe X, membership function $(x)_A$m equals the degree to which x belongs to set A. The degree of membership value ranges between 0 and 1 (see Figure 2.4).

$$m{:}\,X[0,1],$$

$$where{:}\begin{cases} (x)_A m = 1 \quad if\ x\ is\ totally\ in\ A \\ (x)_A m = 0 \quad if\ x\ is\ not\ in\ A \\ 0 < (x)_A m < 1 \quad if\ x\ is\ partially\ in\ A \end{cases}$$

Sigmoid, Gaussian and other linear functions can be used to represent the fuzzy sets. Implementing non-linear functions increases the computational complexity for the algorithm.

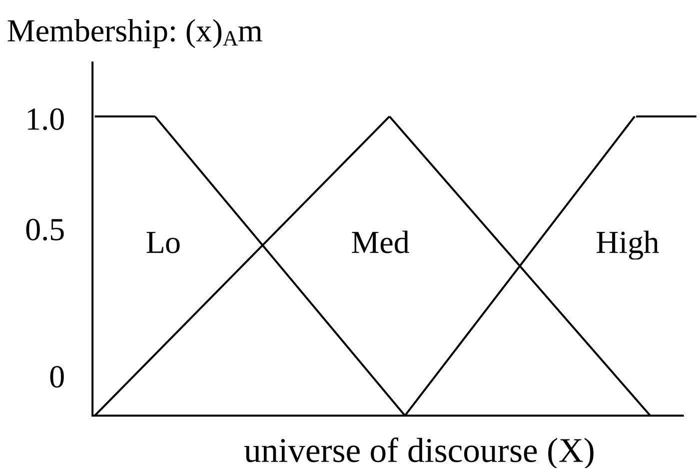

Figure 2.4: A basic fuzzy set.

2.2.3. Operations on fuzzy sets

Complement, containment, intersection and union are the four major operations on fuzzy sets. Figure 2.5 shows the different operations done on fuzzy sets schematically [19].

Complement: The complement of the fuzzy set is the opposite of the given set. Fuzzy complement for fuzzy set A, on universe of discourse X is given by equation,

$$(x)_{-A}m = 1 - (x)_A m, \qquad where\ x \in X$$

Containment: Containment or subset is a set which can contain other sets or which is apart (partial or full) of another set. It is important to note fuzzy subset elements have lower or equal membership values than the set it is part of.

Intersection: An intersection between two sets contains the common elements of the two sets. In fuzzy sets, an element can partly belong to both sets with different memberships. Thus we choose the minimum of the two membership values to find the fuzzy intersection. The fuzzy intersection of two fuzzy sets A and B on universe of discourse X is given by,

$$(x)_{A\cap B}m = min[(x)_A m,\ (x)_B m] = (x)_A m\ \cap (x)_B m, \qquad where\ x \in X$$

Union: The union is the opposite of the intersection. As a consequence, the union is the largest membership value of the element in either set.

The fuzzy union of two fuzzy sets A and B on universe of discourse X is given by,

$$(x)_{A\cup B}m = max[(x)_A m,\ (x)_B m] = (x)_A m\ \cup\ (x)_B m, \qquad where\ x \in X$$

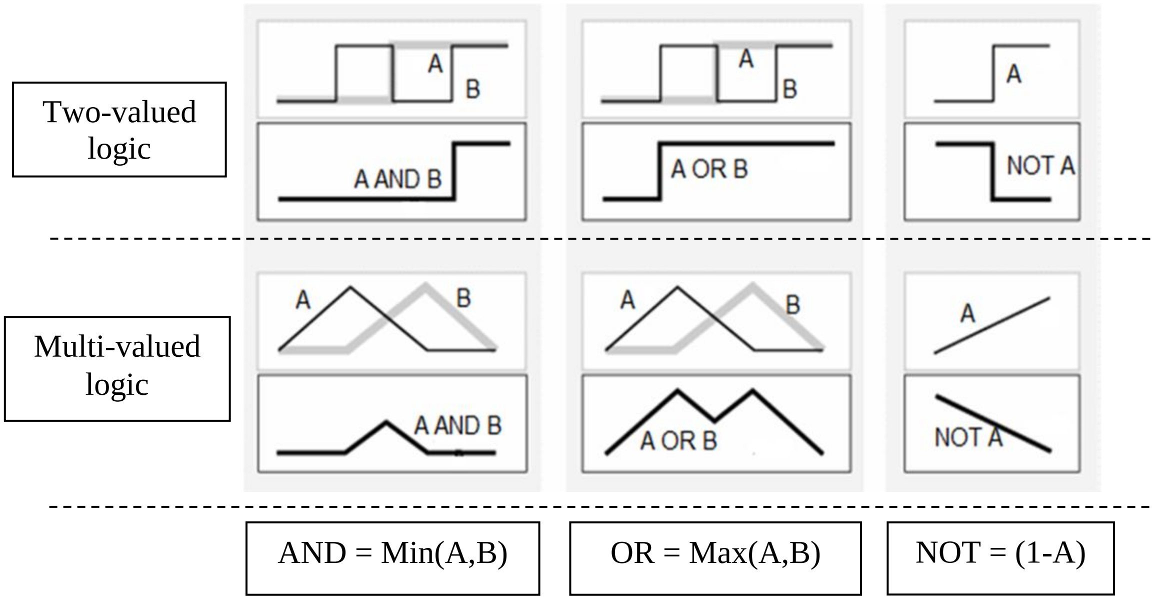

Figure 2.5: Fuzzy set operations.

2.2.4. Fuzzy rules and fuzzy sets

A conditional statement based on two linguistic values A and B on universe of discourse X and Y with linguistic variables x and y of the form IF x is A THEN y is B, defines a fuzzy rule [23].

In fuzzy expert systems, linguistic variables are used in fuzzy rules which relates to fuzzy sets. For example:

- IF energy is high
- THEN cluster-head chance is high
- The range of possible values of a linguistic variable represents the universe of discourse of that variable. To site an example, the universe of discourse of the linguistic variable energy may have the scaled range between 0 and 100, and can include fuzzy sets as low, medium, and high. All rules fire to some extent, depending upon the degree of membership to which the antecedent relates with the consequent. A simple example maybe,
- IF the energy is low and

- IF the concentration is low and
- IF the centrality is far
- THEN the node’s cluster-head election chance is very small.

2.2.5. Fuzzy inference

Up to this point we have provided the details of the fuzzy sets, fuzzy rules and the working of different operators like union and intersection. In this section, a commonly used fuzzy inference technique called the Mamdani method is reviewed. It requires us to find the centroid of a two-dimensional shape by integrating across a continuously varying function [23]. From the computational point of view this method is not so efficient. Another inference technique, called Sugeno fuzzy inference proposes the use of a single spike, called singleton, as a membership function of the rule consequent [8]. But since the popular MAMDANI technique allows us to describe the expert knowledge in a more intuitive manner, it has been used very frequently [23].

To explain how the complete steps work, we take a basic example:

Here we consider two basic fuzzy rules:

IF energy is medium THEN cluster-head chance is medium

and IF energy is high THEN cluster-head chance is high.

These are the steps that the system would undertake to carry out the fuzzy inference:

1. Input of crisp value and fuzzification,

2. Rule evaluation,

3. Aggregation of the rule outputs,

4. Defuzzification.

Step 1: Input of crisp value and fuzzification

This is rather straight forward. The input is a crisp value, energy level with a single point of reference. Based on the crisp number, we determine the input values from the fuzzy sets. It is seen that the crisp input energy value of 60 intersects the input fuzzy sets with a membership value of 0.3 and 0.5 from the first and second rules respectively.

Step 2: Rule evaluation

Based on the fuzzification values of 0.3 and 0.5 obtained earlier, we deduce these values to the output rules to determine the new fuzzy output set. For a given fuzzy rule having multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number that represents the result of the antecedent evaluation. This number (the truth value) is then applied to the consequent membership function.

Step 3: Aggregation of the rule outputs

For the crisp value of energy (= 60) both the defined rules apply. Aggregation is the process of unification of the outputs of all rules. Based on two fuzzy sets, a new fuzzy output set is created. Aggregation is an OR logic because we are looking at the aggregation of both rules, thus we use the fuzzy logical operator OR to create the new aggregate fuzzy set.

Step 4: Defuzzification

This is the last step where we determine the final fuzzy system crisp output. We produce a single crisp number from the output fuzzy set. The Mamdani technique is used to calculate the inference value. Centroid defuzzification method [8] finds a point representing the centre of gravity of the fuzzy set. The center of gravity is a weighted average calculation that takes each subdivided area's center and averages according to the weight that that area contributes to the whole. Therefore after defuzzification it gives a value of 55 as the cluster-head election chance.

2.2.6. Fuzzy systems in wireless networks

There are varied applications of intelligent techniques in wireless networks [20].The increase in the growth of wireless application demands for the wireless network to have the capability to trace the locations of mobile users. Location updating scheme using fuzzy logic controls that adaptively adjusts size of the location area for each user.

Different approaches in improving the reliability and accuracy of measurement information from the sensor networks have been described. It offers a way of integrating sensor measurement results with association information, available or a priori, derived at aggregating nodes by using some optimization algorithm. They have considered both neuro-fuzzy and probabilistic models for sensor results and association information. The models carry out classification of the information sources, available in sensor systems.

CHAPTER II

Related Work: Wireless Sensor Networks Clustering Protocols

3.1. Introduction

'Power aware' routing protocol [16] in wireless networks relies on utilizing routes that have high energy nodes in its path but are longer to reach the base station than the route sthat have shorter paths and low energy nodes respectively. This is done to minimize the overall energy of the network.

Another early approach involves selecting the MTE (minimum transmission energy) routing [15], where the nodes are chosen for routing such that the total transmit energy is minimized. We assume d^2 power loss for the total transmitting energy at distance d [$E_{Tx}(d)$]. Thus for three nodes A, B:

$$E_{Tx}(d = d_{AB}) + E_{Tx}(d = d_{BC}) < E_{Tx}(d = d_{AC})$$

$$or$$

$$d_{AB}^2 + d_{BC}^2 < d_{AC}^2$$

The drawback of this approach is that, although the transmitting distance is being taken into consideration, the energy present at the nodes are not, it might not generate the lowest energy routes.

3.2. Tree-Based Approach

In the tree-based approach, instead of sensor nodes sending the data to the cluster-heads directly, each node sends it to its parent. The base station selects some of the sensor nodes to be its children. The election criterion is based on factors like concentration of nodes in a given area and its closeness to immediate neighbors. Other nodes in the network associate with the child nodes, as selected by the base station, on the basis of the received signal strength.

Thus in this approach the number of long distance transmissions is reduced by having the nodes to send their data to their parent and in turn to the base station. This hierarchical approach with the base station as the root node (as shown in Figure 3.1) generates a spanning tree for the network. Only the immediate children of the base station are required to make the high energy transmission, after collecting and compressing the data received from its offspring.

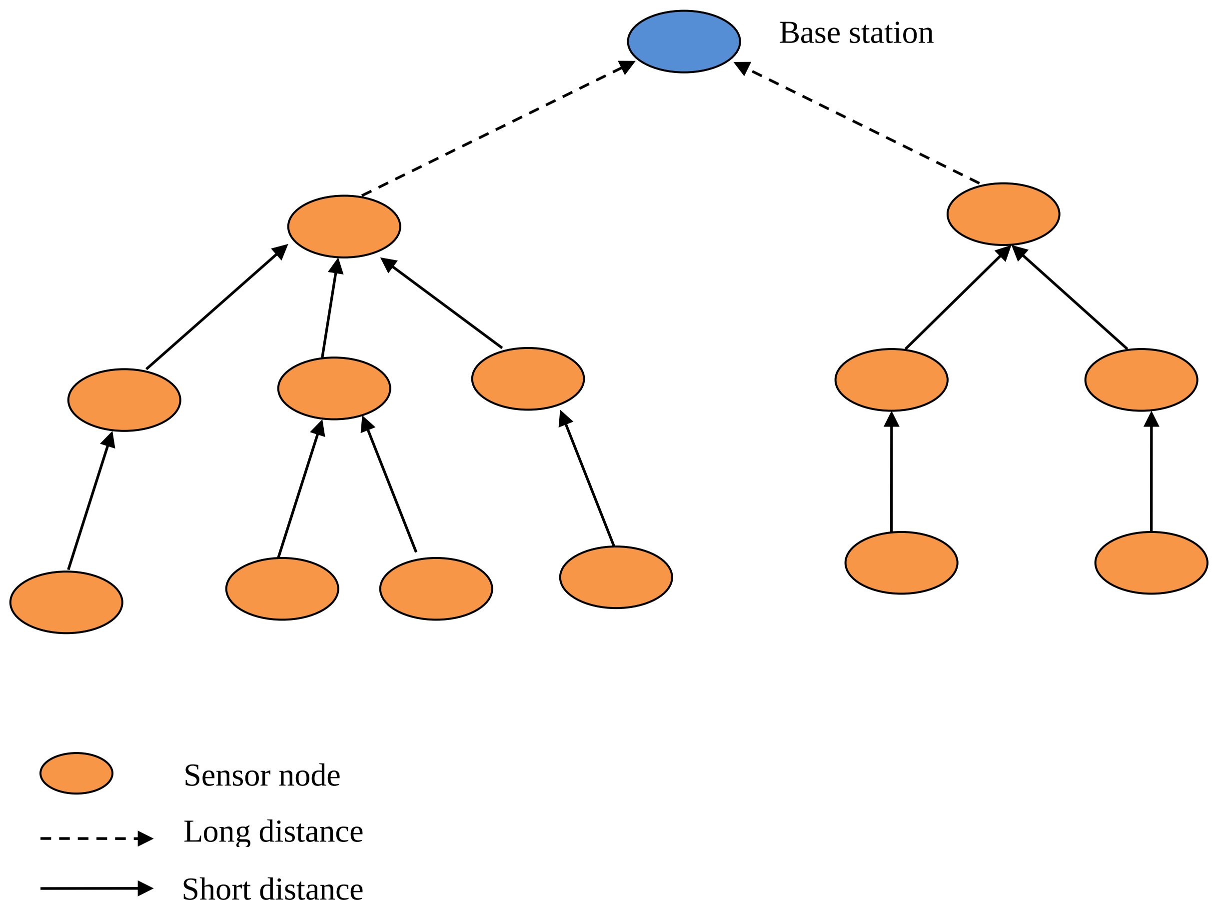

Figure 3.1: Tree based WSN.

3.3. Cluster-Based Approach

In the cluster-based approach, only some of the nodes in the network are allowed to transmit and receive information from the base station, which is located at a large distance from the sensor nodes [1]. The key issue here is that, this allows sensor nodes to sense and transmit the information to the cluster-heads directly, instead of routing it through its immediate neighbors. Also, since communication energy is proportional to the square of the distance, having all nodes to transmit its sensed data individually to the base station, exhausts the energy of each node drastically and hence the life time operation of the network gets significantly reduced. As a consequence it does not serve the purpose with which WSNs are designed for, namely network should be operational for a long period of time.

Figure 3.2 shows a typical cluster-based WSN. Cluster-heads collect the data sent by each node in that cluster, compressing it and then transmitting the aggregated data to the base station for further processing.

Comparative analyses of the performance of the two approaches have been made in [11]. A multi-hop tree based approach is considered inefficient for routing in WSN considering the global distribution of nodes as shown in [24].

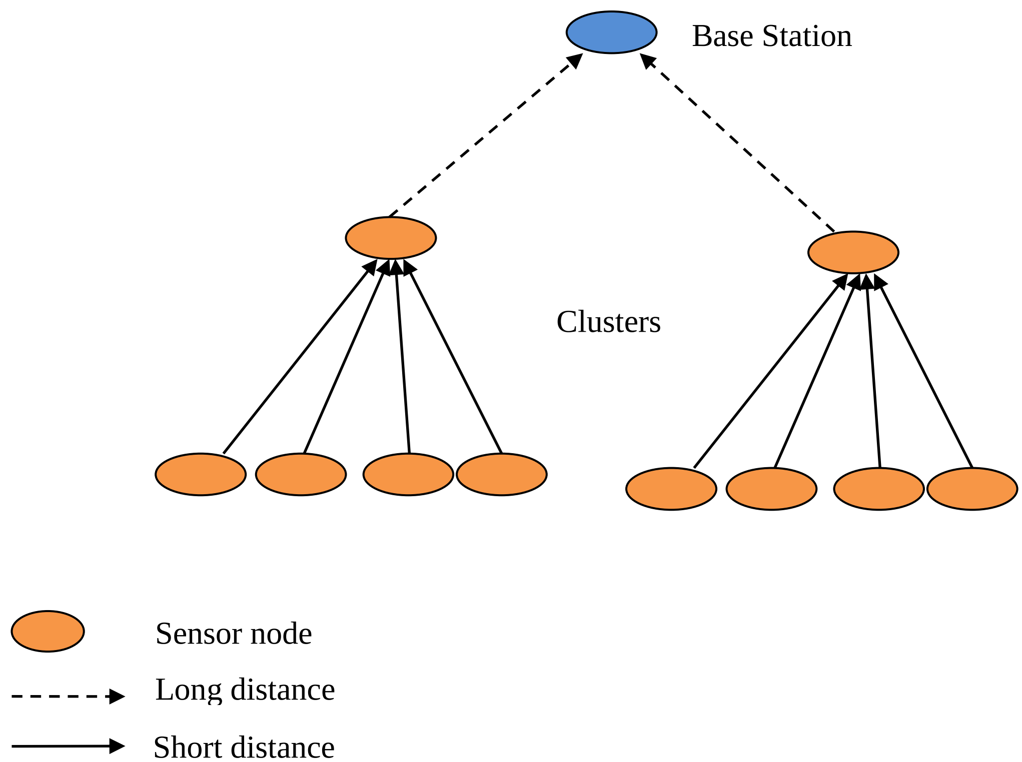

Figure 3.2: Cluster based WSN

3.4. LEACH Protocol

LEACH (Low Energy Adaptive Clustering Hierarchy) is a popular and significant communication protocol that helps the nodes to minimize the overall energy dissipation in the network using clustering [1]. It is the first significant protocol for the minimization of the overall energy in this type of network.

LEACH organizes nodes into clusters with one or multiple nodes from each cluster acting as a cluster-head. The aim of the protocol is to randomize the cluster-head election in each round so that the energy among the sensor nodes becomes evenly distributed. In this approach, the base station is assumed to be fixed and all the nodes are assumed to be energy constrained in nature.

The motivation for this approach came from the MIT's AMPS project - the fact that communication energy between sensor nodes and base station is expensive and thus it is infeasible for the sensor nodes to sense and gather data and send them to the base stations individually in a single hop, which is a high power operation. Moreover, if cluster-head selection is static, the nodes selected as cluster-heads will quickly drain out its limited power and die quickly. Apart from selecting the cluster-head, priorities were given to data aggregation and data fusion methodologies. Data fusion combines different data measurements and then reduces the uncorrelated noise to provide a more accurate signal. The randomized rotation of nodes that is necessary to be cluster-heads, for even distribution of energy consumption over all nodes in the network is the main characteristic of this algorithm.

LEACH operation is broken into rounds, having a set-up phase and a steady-state phase. In the beginning of the set-up phase, each node probabilistically decides whether or not to be a cluster head.

To become a cluster-head, each node n chooses a random number between 0 and 1. If the number is less than the threshold T(n), the node becomes the cluster-head for the current round. The threshold is set at:

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

where, P is the cluster-head probability, r the number of the current round and G the set of nodes that have not been cluster-heads in the last 1/P rounds. Thus, after 1/P -1 rounds T*(n)* =1 for all nodes that have not been a cluster-head.

After electing itself as a cluster-head, the node broadcasts an advertisement message announcing its intention to the rest of the nodes, using CSMA-MAC (Carrier Sense Multiple Access- Medium Access Control) protocol [25]; all the cluster-heads broadcast using the same transmit energy. Each non cluster-head node thus receives advertisements from the cluster-heads and selects the cluster to join based on the largest received signal strength of the advertisement. This implies that minimum amount of transmission energy is needed for communication with the selected cluster-head. Nodes inform the cluster head of the cluster they intend to join, using the CSMA-MAC protocol. Each cluster-head then assigns a TDMA (Time Division Multiple Access) schedule for the nodes in the cluster, for sending sensed data. The TDMA slots are being calculated based on the number of nodes present in the cluster and are then broadcasted back to the cluster nodes.

In the steady-state phase, each cluster-head waits to receive data from all nodes in its cluster and then sends the aggregated result back to the base station.

Simulation proved that LEACH reduces the communication energy as much as 8xtimes as compared to direct transmission.

3.5. LEACH-C

Motivated by the previous approach, Heinzelman et al. proposed a centralized clustering algorithm having the steady state phase operation similar to LEACH, called LEACH-C [2]. The protocol offers a way out to the shortcoming in the earlier protocol. LEACH offers no assurance about the placement of the cluster-head nodes in the cluster. Thus other nodes in that cluster may expend more energy transmitting through the selected node located far from the cluster centroid.

Moreover there is no guarantee about the number of cluster-head nodes selected per round. Since nodes become cluster-head when the random number it generates is less than the threshold, it might happen at times that more than the desired percentage of nodes becomes cluster-head.

Consequently LEACH-C uses a centralized algorithm and provides another approach to form clusters, as well as selecting the cluster-heads using the simulated annealing technique. Centralized control algorithm approach produces better cluster-heads by dispersing the cluster-heads throughout the network. During the setup phase, the nodes end the location co-ordinate information and the energy level to the base station. The base station computes the average node energy in each round. Thus, in addition to determining good clusters, this approach eliminates the nodes to become the cluster-heads whose energy falls below the average energy of the network. It then forms the clusters based on the remaining nodes as cluster-heads, using the simulated annealing technique [9].

A relevant contribution to this work comes from the fact that the algorithm tries to minimize the amount of energy required by non cluster-head nodes to transmit their data thorough the cluster-head, by minimizing the total sum of the squared distances between all the non cluster-head nodes and the possible cluster-head. After the clusters and the cluster-heads have been calculated at the base station, the base station broadcasts cluster head ID for each node. If a match occurs between the node ID and the cluster-head ID, the node is a cluster-head. Otherwise the nodes find out the TDMA time slots for transmitting the data to the cluster-head. The steady state phase is identical to the steady state phase of LEACH [9].

3.6. Deterministic Model of LEACH

Handy et al. [9] proposed an algorithm for reducing the power consumption of the wireless sensor network. The stochastic method of selecting cluster-heads in the LEACH approach was extended by adding a deterministic component. They propose to make it energy efficient by multiplying the factor

$$\frac{E_{n\_current}}{E_{n\_max}}$$

Where, $E_{n\_current}$ is the current energy and E_{n_max} the initial energy of the node. Hence the new threshold is:

$$T(n) = \begin{cases} \dfrac{P}{1 - P \times \left(r \bmod \frac{1}{P}\right)} \dfrac{E_{n\_current}}{E_{n\_max}} & if\ n \in G \\ 0 & otherwise \end{cases}$$

Now since the ratio is less than one, from the second round onwards, the total number of cluster-heads generated in each round reduces, by multiplication of a factor less than one. This can significantly impact the average energy dissipated per round as demonstrated in [2].

3.7. Other Approaches

In each node calculates its distance from the area centroid in the specified cluster [25]. Performance issues for biased distribution of nodes are one of the drawbacks of this algorithm. The single nodes that are strategically placed closed to the area centroid will be more often selected as cluster-heads. Consequently it leads to overall high energy consumption in the network for other nodes to transmit data through the selected node located relatively far from the cluster.

Recently, Voigt et al. [17] propose a new approach to utilize solar power in WSNs and extended LEACH to become solar-aware. They propose a handover scheme that allows changes of cluster-heads during the steady state phase of the LEACH protocol, proving that making LEACH solar aware increases the lifetime of the network substantially.

For solar-aware distributed LEACH, they modify the LEACH equation to incorporate two different constraints. Firstly, they suggest that the solar powered nodes must become cluster-heads with a higher probability than a non solar powered node. Secondly, a node that has been solar powered while it is a cluster-head should have the provision to become a cluster-head even during next 1/P rounds.

CHAPTER IV:

Our Approach: Genetic Algorithm based on Global and Local Clustering Protocol (GAGLCP)

4.1. Basic Idea

In order to achieve our goal of enhancing the WSNs clustering, we need to perform more control of the clustering process, i.e., detecting cluster heads and their members. In fact, such operation is based on two main arguments. The first one is the distances between cluster heads and their members as well as the distance between cluster heads and the base station. The second argument is related the residual energy of the nodes.

With the first argument, we need to apply a global clustering that should select the area of the highest density (large zone) as shown in Figure 4.1. The selection is related to the sum of distances between clusters heads and the members where they should as minimal as possible. In order to guarantee a best performance of the second argument (energy of the entire network), at the end of each round we must chose the best clustering that guarantees the following:

1- The minimum energy consumed to transmit the aggregated data to cluster heads and base station.

2- Maintain the maximum number of live nodes.

3- Try to keep the network working in large area in real time.

4.2. Global Clustering

It resembles the clustering of LEACH and LEACH-GA (LEACH based on genetic algorithms). Yet, we extended the LEACH protocol by applying fuzzy logic to divide the network into domains. A domain is set by a threshold that defines the maximum number of sensor nodes inside each domain. Once the domains are defined based on sensors locations and their current energy, the base station runs a genetic algorithm with chromosomes that correspond to domains instead of the entire network. Hence, the generated chromosomes will be of short size compared to LEACH-GA. Moreover, when the network increases in size, more domains are defined, but the size of the chromosome in each domain will maintain the same maximum size.

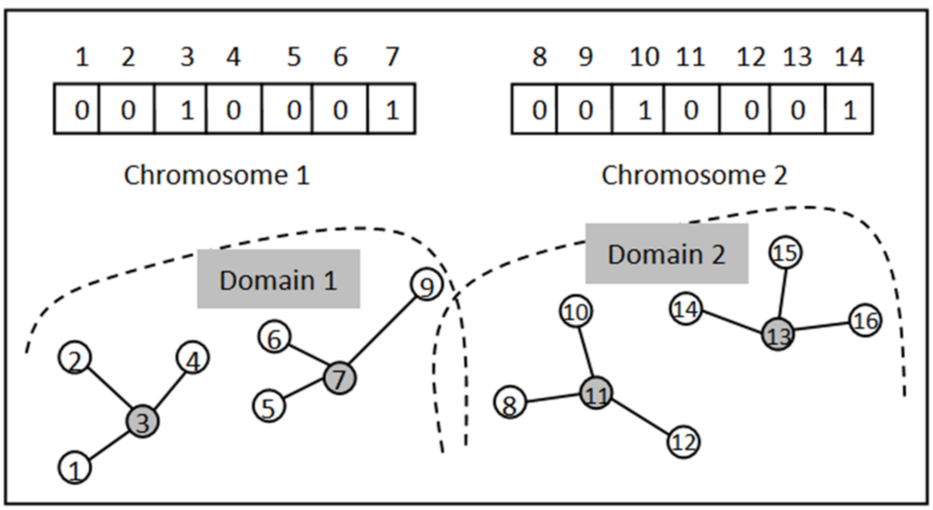

Figure 4.1: Dividing the network into domains to enhance genetic algorithms clustering.

In order to enhance the time search, the network is divided into k domains: D_1, D_2, … D_k. Sensors belonging to different domains have low chance to be in the same cluster. To define the domains we use the fuzzy logic concept.

4.3. Fuzzy Logic Phase

In our approach we try to keep the number of nodes in each domain near to 30 elements and also the number of nodes in each cluster is 10 elements to better control the energy level and have low complexity of computing.

4.3.1. How to define the domains

To define new domain, select a group of nodes with high energy and density and determining the nodes included in each domains using fuzzy logic.

The number of domains was estimated using the following formula:

$$N_{Domains} = \frac{N_{nodes} \times 3}{100}$$

A domain is defined as follows: if two sensors s_1 and s_2 belong to two separate domains D_1 and D_2, then the chance to have both of them in one cluster is very low.

To calculate the chance we need the set the rules of defuzzification .

4.3.2. Rules used in fuzzy logic

In this phase, we use three parameters: energy level of the CH, distance between the CH and the BS and the distance from non-CHs to the CH. We choose these parameters because of their importance for extending the network lifetime as we will see in the next chapter. In order to study how much they are effecting the lifetime of the network, and to make these parameters more flexible, we divided each 'linguistic variable' that we used to into three levels (Table 4.1): (low, medium, high) for energy level of the CH and (close, medium, far) for the distance to the BS and the distance between the CHs and the node.

TABLE 4.1: DIFFERENT LINGUISTIC VALUES USED IN FUZZY LOGIC OPERATIONS

Energy	Distance to CH	Distance to BS
High	Close	Close
Medium	Medium	Medium
Low	Far	Far

4.4. Genetic Algorithm Phase

4.4.1. The flowchart

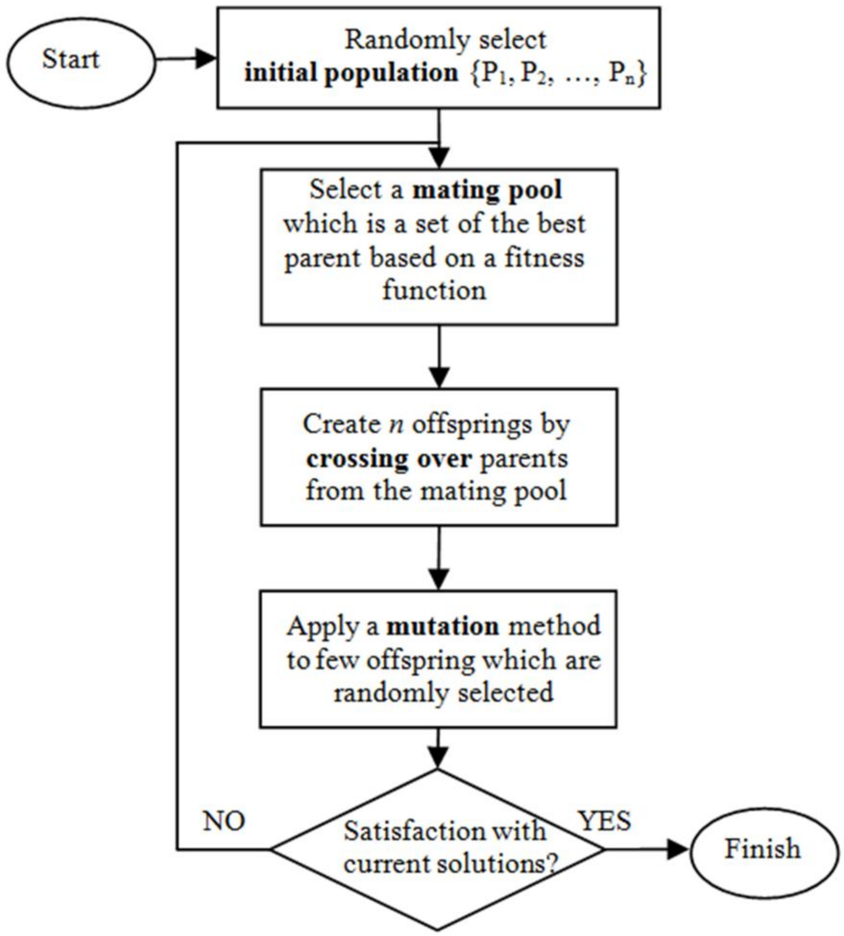

Figure 4.2: Flowchart of genetic algorithms phase.

When the network is divided into domains using fuzzy logic, each domain generate a chromosome that give an initial search space equals to factorial number of nodes in each domain (Figure 4.2). Thus, we reduced the search space in the network to one third less than the network size, which will affect greatly the clustering results in terms of time and accuracy.

Sensors belonging to different domains have low chance to be in the same cluster. Each sensor in the chromosome represents a gene that is labeled with an ID and has a value (0 for the regular sensor and 1 for the cluster head) as shown in Figure 4.3.

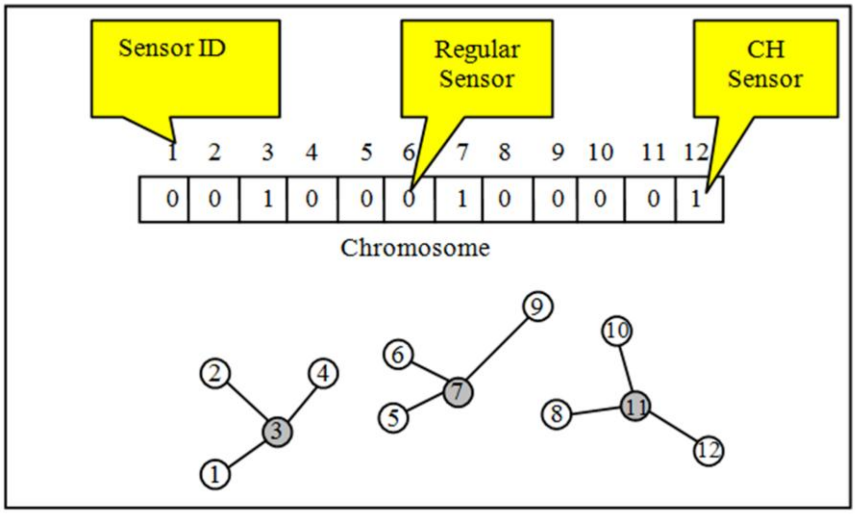

Figure 4.3: Enhanced LEACH clustering based on genetic algorithms.

GA generates a random set of chromosomes and starts the GA cycles to find optimal clustering. At the end of each round, an evaluation is performed on each chromosome using a fitness function as shown in the next section.

4.4.2. The fitness function

$$\text{Fitness}(C_k) = \sum_{j \in CHS} E(j) + \frac{1}{\sum_{i \in RSS} \text{StoH}(i)} + \frac{1}{\sum_{j \in RSS} \text{HtoB}(j)}$$

Where:

Ck: Chromosome k.

E(j): Current energy of a cluster head.

StoH(i): Energy to transmit one packet from a regular sensor i to the associate cluster head.

HtoB(j): Energy to transmit one aggregated-data packet to from a cluster head j the base station.

RSS: Regular sensors set corresponding to chromosome C_k.

CHS: Cluster heads set corresponding to chromosome C_k.

Our goal is to maximize the function $F(C_k)$ in order to obtain Low energy consumption and Best clustering. Next is a flowchart that combines the three phases of our mechanism: Fuzzy logic, genetic algorithms and operational phases (Figure 4.4).

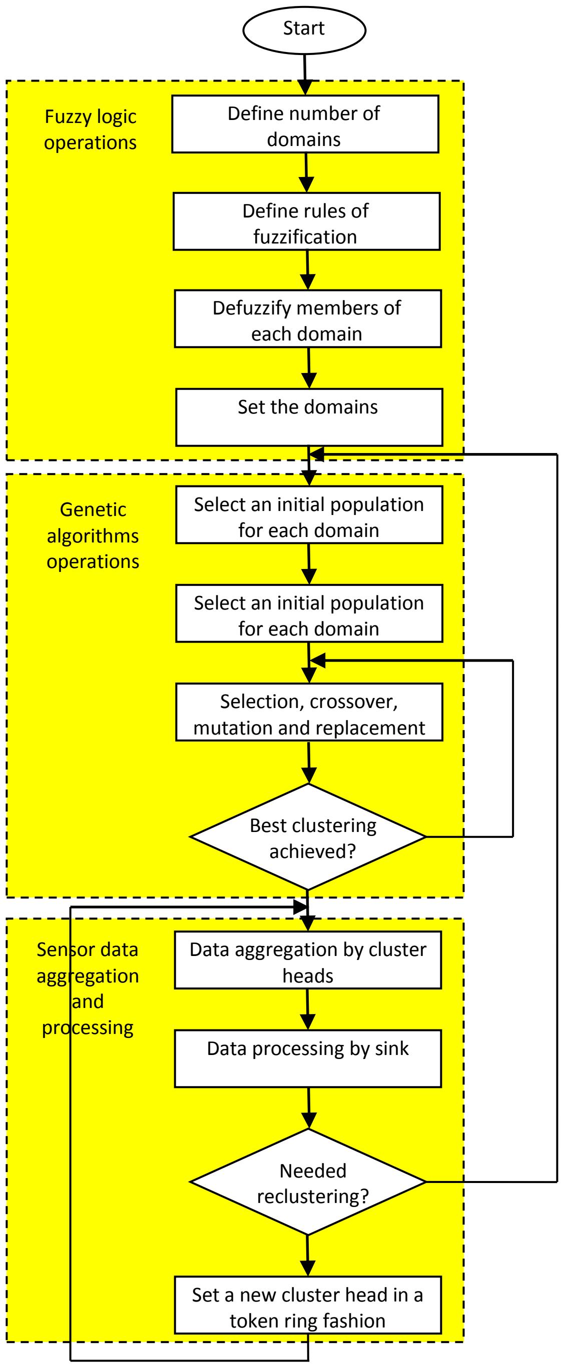

Figure 4.4: Flowchart of global clustering using fuzzy logic and local clustering using genetic algorithms.

4.5. Local Clustering: Token Ring Topology

The base station selects a cluster topology after running a GA algorithm. The base station should decide sort of a ring topology inside each cluster where each sensor knows its successor. Therefore, when the clusters are formed, the initial cluster head must hold a token in order to aggregate data and send it to the base station (Figure 4.5). Thereafter, it passes the token to the next node to assign it as a new cluster head.

When a node gets isolated after all cluster nodes are of low energy (the token cannot be passed to other node), it transmits its current info to the base station for new topology and clustering assignment. On the other hand, when a node is about to be down, it sends an alerting packet to its predecessor in order to inform it to forward the token in the future to the current next sensor in line.

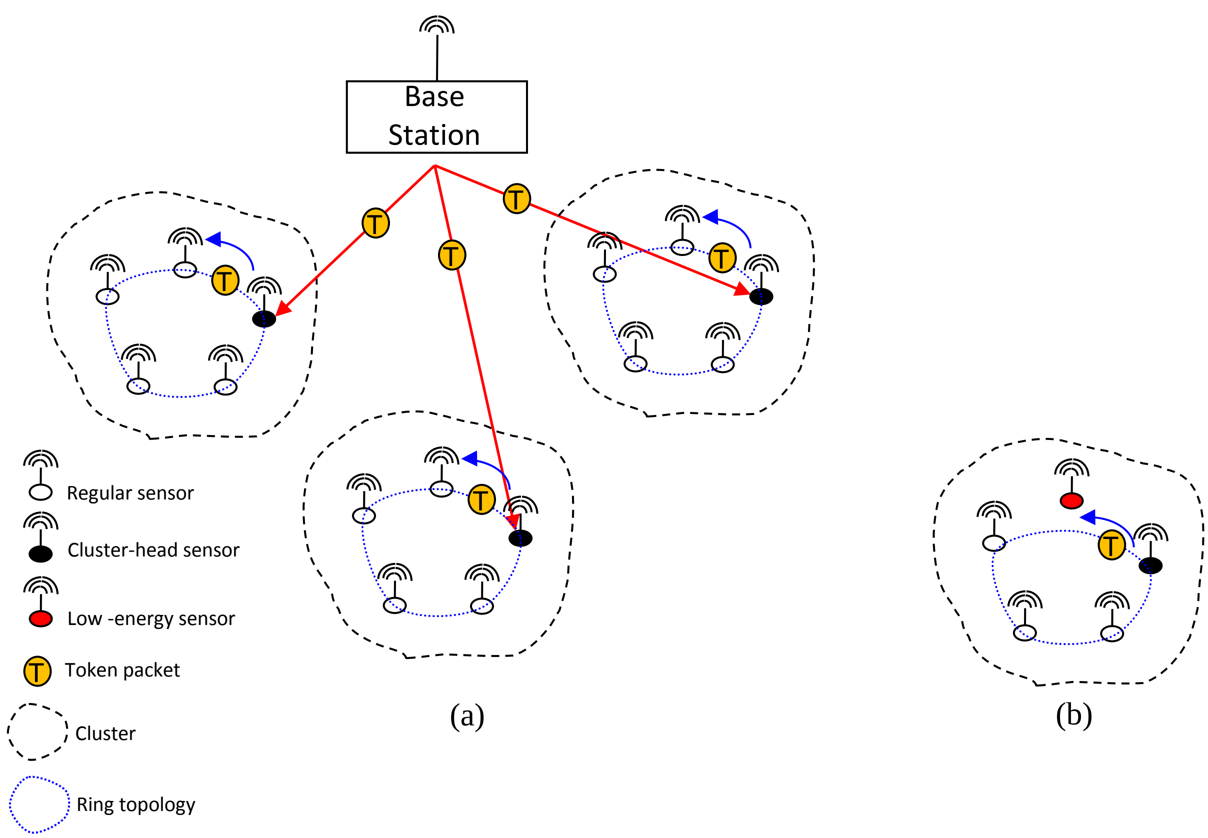

Figure 4.5: Local cluster head switching (a) normal token forwarding (b) escaping low battery nodes.

CHAPTER IV:

Simulations, Results and Analysis

5.1. Introduction

In this chapter, we analyze the performance of our algorithm via conducted simulations in MATLAB. We also compare our clustering algorithm with the LEACH algorithm. Since energy conservation is the primary objective of our work, performance metrics such as network lifetime, energy consumed per round, and the residual energy level of sensor nodes are of particular interest.

5.2. The simulation Environments for WSNs

5.2.1 Review of existing simulation tools for WSNs

As fellow we present a set of existing simulation environments for WSNs that we found in the literature:

a. GloMoSim/QualNet

GloMoSim is a scalable simulation environment for wireless and wired network systems, which uses the parallel discrete-event simulation capability provided by Parsec [29], a c based simulation language for sequential and parallel execution of discrete-event simulation models.

b. OPNET Modeler Wireless Suite

OPNET Modeler Wireless Suite [30] is a commercial modeling and simulation tool for various types of wireless networks. It is developed by developed by OPNET Technologies, Inc. and based on the well-known product OPNET Modeler.

c. TOSSIM

TOSSIM (TinyOS mote simulator) [31] is a discrete event simulator for TinyOS sensor networks that is part of the official TinyOS package.

d. OMNeT++

OMNeT++ [32] is an object-oriented discrete network simulation framework. The architecture is rather generic so that various problem domains can be simulated such as protocol modeling, validation of hardware architectures and modeling of wired and wireless communication networks.

e. NS-2

NS (the Network Simulator) [33] is an object-oriented discrete event simulator targeting at networking research. NS-2 is written in C++ and OTcl, an object-oriented version of Tcl.

f. Avrora

Avrora [34] is a set of simulation and analysis tools for programs written for AVR micro-controllers.

g. J-Sim

J-Sim [35] is a component-based compositional simulation environment based on the autonomous component architecture (ACA).

h. ATEMU

ATEMU [36] is one of the first instruction-level software emulators for AVR based systems. Additionally peripheral devices of the MICA2 sensor node platform such as radio is supported.

i. EmStar

EmStar [37] is an environment for WSNs built from Linux-class devices, so called micro servers.

j. SENS

SENS is an application-oriented simulator for WSNs.

5.2.2 The proposed WSN simulation methodology

The environment in which we build our simulation model was MATLAB. The name MATLAB stands for matrix laboratory. MATLAB, developed by MathWorks Inc., is a software package for high performance numerical computation and visualization[38]. The combination of analysis capabilities, flexibility, reliability, and powerful graphics makes MATLAB the premier software package for scientific researchers. MATLAB provides an interactive environment with hundreds of reliable and accurate built-in mathematical functions. These functions provide solutions to a broad range of mathematical problems Simulation Framework of Wireless Sensor Network (WSN) Using MATLAB/SIMULINK Software 269 including matrix algebra, complex arithmetic, linear systems, differential equations, signal processing, optimization, nonlinear systems, and many other types of scientific computations.

The most important feature of MATLAB is its programming capability, which is very easy to learn and to use, and which allows user-developed functions. It also allows access to Fortran algorithms and C codes by means of external interfaces. There are several optional toolboxes written for special applications such as signal processing, control systems design, system identification, statistics, neural networks, fuzzy logic, symbolic computations, and others. MATLAB has been enhanced by the very powerful Simulink program[59].

Simulink is a software package for modeling, simulating, and analyzing dynamical systems. It supports linear and nonlinear systems, modeled in continuous time, sampled time, or a hybrid of the two. Systems can also be multi-rate, i.e., have different parts that are sampled or updated at different rates. For modeling, Simulink provides a graphical user interface (GUI) for building models as block diagrams, using click-and-drag mouse operations. With this interface, you can draw the models just as you would with pencil and paper (or as most textbooks depict them) [38].

5.3. Network Configuration

In our simulation, we used a network operation model as follows. The reference network consists of 150 nodes randomly distributed over an area of100x100 meters. The base station is located at the coordinate 50, 50. In the first phase of the simulation each node has a same energy between 0 and 0.5. The base station computes the concentration for each node by calculating the number of other nodes within the area of 30x25 meters, with that node in the center. Figure 5.1 shows the concentration calculation for the network with a random distribution of nodes.

In this model, the network operation progresses in rounds. Each round in turn consists of a cluster set-up and data transmission phase. In the cluster set-up phase, a set of new CHs is elected from the active nodes and the remaining nodes become cluster members. In the data transmissions phase, each sensor node sends a fixed amount of data to its CH, which is later forwarded to the BS after aggregation.

5.4. Assumptions

The proposed system model uses the assumptions listed below:

1- All the nodes in WSNs are having the same hardware, communication, and computation capabilities.

2- The nodes are deployed randomly in a 2-D plane using uniform distribution.

3- All the nodes have the equal initial energy.

4- The base station position is located in the middle of the WSNs.

5- Nodes consume energy according to the model described in Chapter 2.

6- Nodes are location unaware, they are not equipped with any global positioning system (GPS) device.

5.5. Fuzzy Rule Base

To determine the maximum values for our parameters in our model, we have used the fellow equations:

1- Max energy = initial energy

2- The distance to the base station = $\sqrt{(Dbs_x)^2 + (Dbs_y)^2}$

3- The distance to the cluster head = $\sqrt{(Dch_x)^2 + (Dch_y)^2}$

Where, (Dbs_x ,Dbs_y) are the coordinates of a cluster head in the orthogonal plan where the base station is the origin, and (Dch_x, Dch_y) are the coordinates of a sensor node in the orthogonal plan where the chosen cluster head is the origin.

Since we decided to use fuzzy set in our work, with the three chosen parameters (residual energy, distance to base station, distance to cluster head) each with three possible levels, we end up with 3^3=27 possible values as shown in Table 5.1 below. These values are the basis of the fuzzy IF-THEN rules. These rules fall between two extremely cases as shown next:

Case (1): IF (energy is low)

AND (distance to the BS is far)

AND (distance between the CH and the node is far (no centrality))

THEN (chance is very weak)

Case (2): IF (energy is high)

AND (distance to the BS is close)

AND (distance between the CH and the node is close (centrality))

THEN (chance is very strong)

TABLE 5.1: ALL POSSIBLE CHANCES USED BY FUZZY LOGIC

N°	Energy	Distance to BS	Centrality	Chance to be CH
1	Low	Far	Close	Very Weak
2	Low	Far	Adequate	Weak
3	Low	Far	Far	Little Weak
4	Low	Adequate	Close	Weak
5	Low	Adequate	Adequate	Little Weak
6	Low	Adequate	Far	Little Medium
7	Low	Close	Close	Little Weak
8	Low	Close	Adequate	Little Medium
9	Low	Close	Far	Medium
10	Medium	Far	Close	Little Weak
11	Medium	Far	Adequate	Little Medium
12	Medium	Far	Far	Medium
13	Medium	Adequate	Close	High Medium
14	Medium	Adequate	Adequate	Medium
15	Medium	Adequate	Far	High Medium
16	Medium	Close	Close	Medium
17	Medium	Close	Adequate	High Medium
18	Medium	Close	Far	Little Strong
19	High	Far	Close	Medium
20	High	Far	Adequate	High Medium
21	High	Far	Far	Little Strong
22	High	Adequate	Close	High Medium
23	High	Adequate	Adequate	Little Strong
24	High	Adequate	Far	Strong
25	High	Close	Close	Little Strong
26	High	Close	Adequate	Strong
27	High	Close	Far	Very Strong

All the nodes are compared on the basis of their corresponding chances and the node with the maximum chance is then elected as the cluster-head. If there are multiple nodes having maximum chance, then the node having more energy is selected.

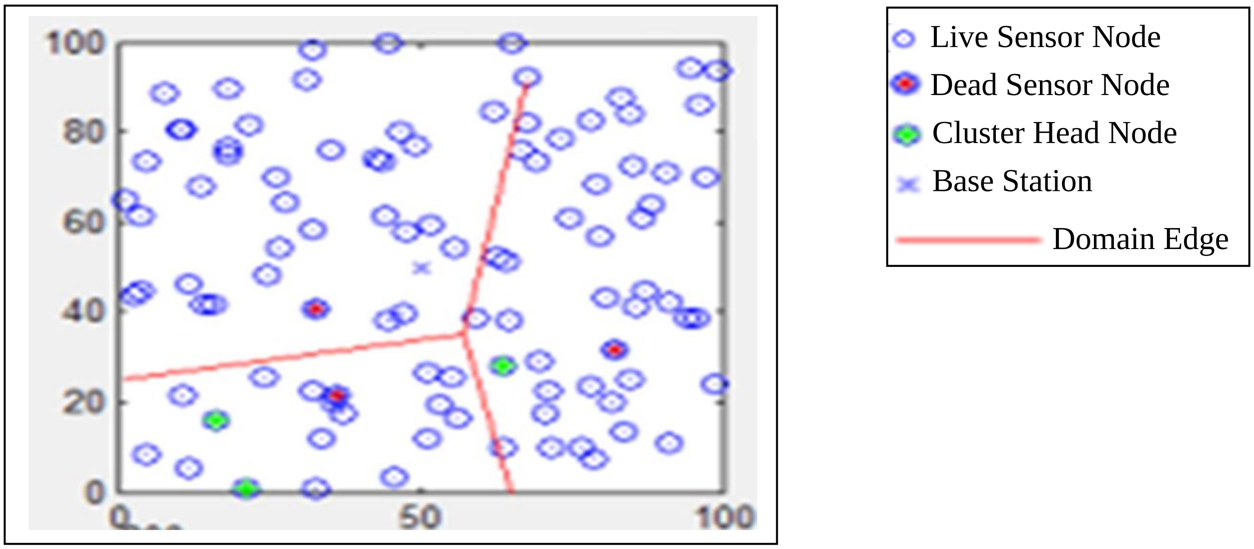

Figure 5.1: Formation of domains using fuzzy logic.

5.6. Results and Analysis

As illustrated in Figure 5.2, the general interface of the simulator is designed as follows: in left the parameters of initial configuration we set the number the live nodes, the number of rounds and the amount of the initial energy. On the top-right section, the deployment of the tow approach LEACH and GAGLCP is shown. In the middle we illustrate a comparison between the two protocols (Figure 5.3), which investigates the live nodes residual energy versus the number of clustering rounds.

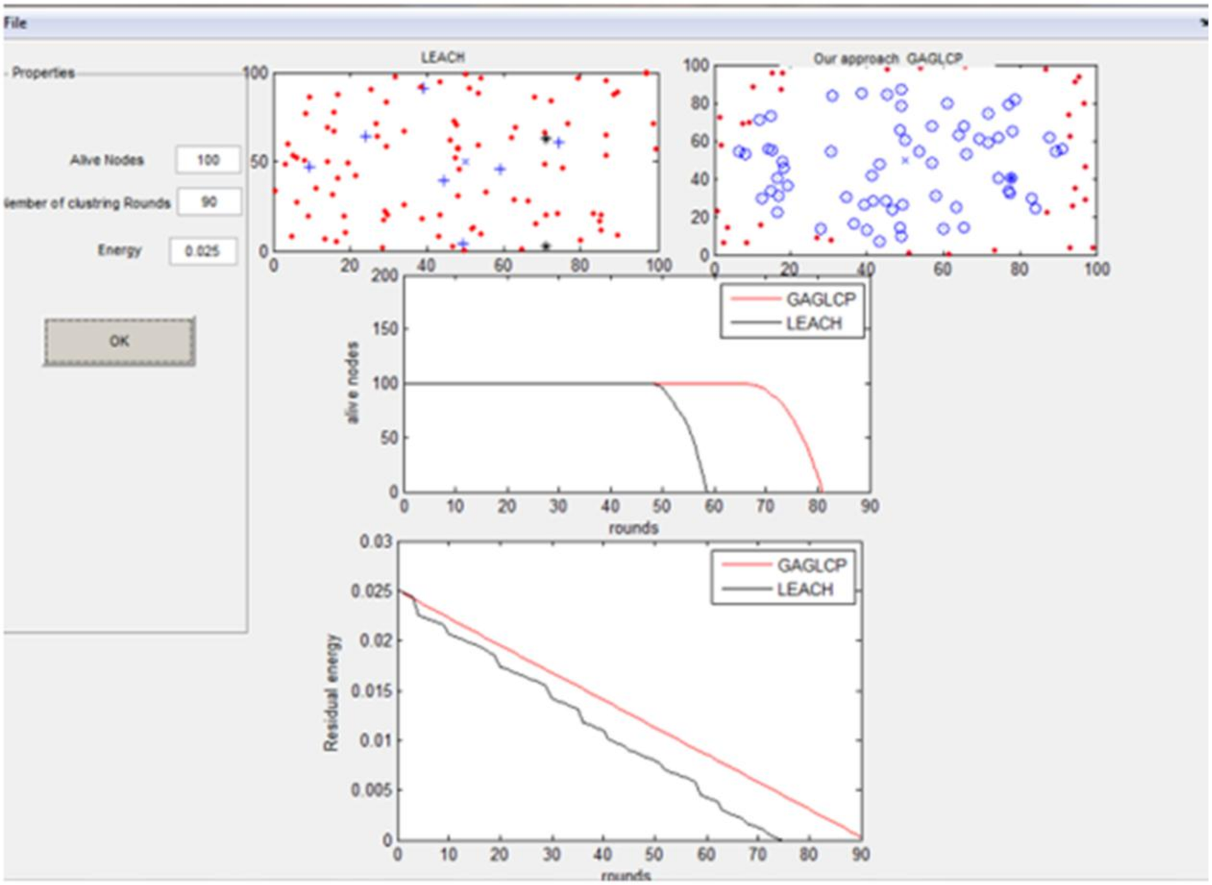

Figure 5.2: MATLAB Simulation interface

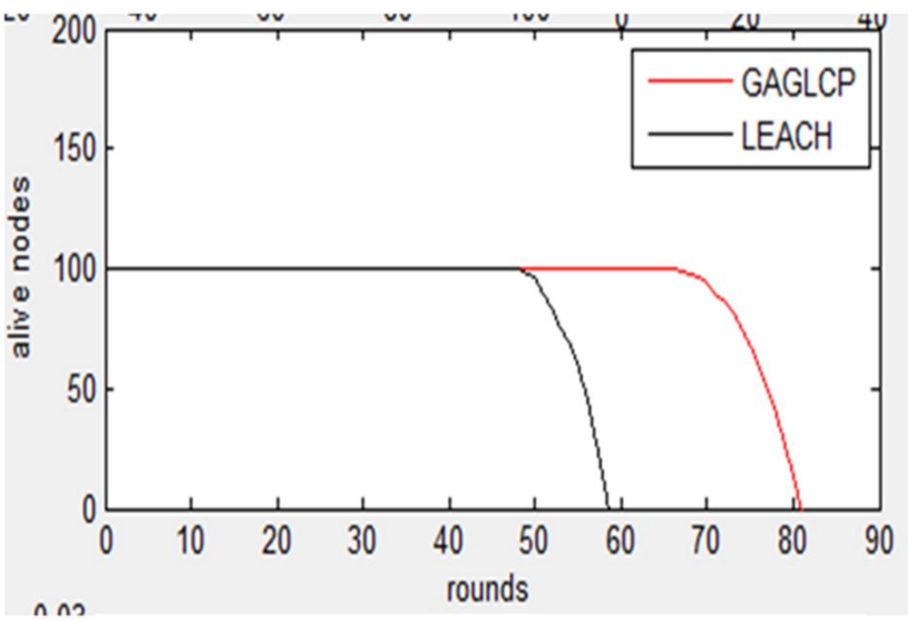

Figure 5.3: Number of live nodes versus the number of rounds for both protocols LEACH and GAGLCP.

We observe that the average number of live nodes for LEACH is 100 at round 48, whereas, in the GAGLCP, the average is 100 at round 67 (Table 5.2) . These improvements rely on the nature and number of parameters that have been used for each protocol.

TABLE 5.2: NETWORK LIFETIME IN TERMS OF LIVE NODES

Rounds	**Live nodes in GAGLCP**	**Live nodes in LEACH protocol**	**Difference**
40	100	100	0
50	100	98	2
52	100	82	12
54	100	68	28
55	100	56	44
57	100	25	75
58	100	0	100
69	97	0	97
71	86	0	86
75	68	0	68
78	38	0	38
82	0	0	0

To understand the energy consumption behavior of the sensor nodes, we monitor the residual energy level of the both approaches GAGLCP and LEACH.

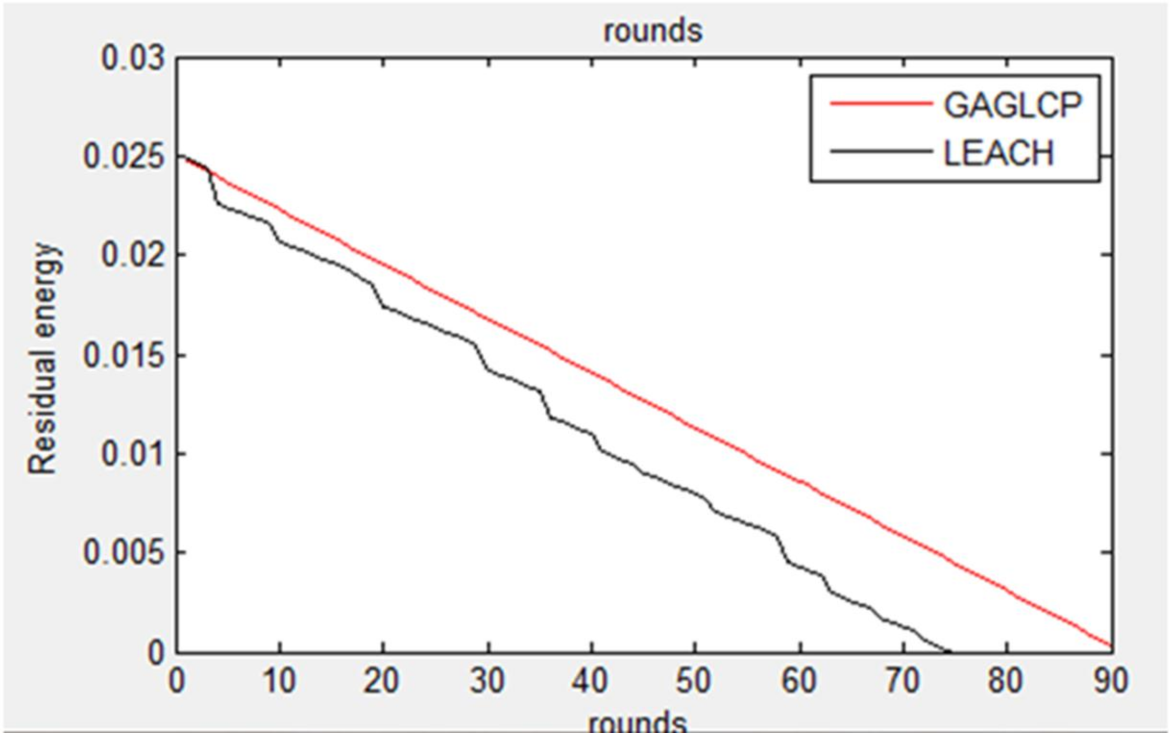

Figure 5.4: Residual energy of the network versus the number of rounds.

Figure 5.4 shows the degradation of the energy levels at both approaches GAGLCP and LEACH. Table 5.3 shows in detail the entire residual energy of the network for both compared protocols.

On LEACH protocol Energy levels break down quickly: at round 10 it records 0.021, on round 60 it records 0.004, and energy was run out at round 74. However, the GAGLCP made high improvement of energy saving: at round 10 it records 0.0235, at round 60 it records 0.0085, and energy ended at round 92, i.e., 18 more rounds than LEACH.

TABLE 5.3: NETWORK ENERGY LEVEL (ACCUMULATED ENERGY *10^3)

Rounds	GAGLCP Total Energy	LEACH protocol Total Energy	Difference
5	24	23.5	0.5
10	23.5	21	1.5
20	19.5	17	2.5
30	17	14	3
40	14.5	11	3.5
50	12	7.5	4.5
60	8.5	4	4.5
70	6.5	1.5	5
80	4.5	0	4.5

5.7. Conclusion

In this chapter, simulation results were presented to demonstrate the performance of the GAGLCP over LEACH and its effects on energy conservation and network lifetime. We used residual energy and live node parameters per round to evaluate the behavior of GAGLCP and LEACH.

Simulation experiments were conducted using many different values of initial energy and number of nodes. Our experiments showed our GAGLCP achieves significant energy savings and enhances network lifetime compared to LEACH protocol. Moreover, GAGLCP showed a better distribution of energy consumption over different sensors in the same cluster.

Conclusion and Future Work

Scalable routing protocols that are based on hierarchical clustering are needed to govern WSNs. In fact, most of the routing protocols used in WSNs were built for ad-hoc networks where energy is not considered as an important factor.

In this thesis, we presented a new approach for cluster formation in WSNs using genetic algorithm and fuzzy logic to enhance the network lifetime. Our approach combines both global and local clustering in order to enhance the energy consumption at the sensor level. So, the sensors network is first divided into many domains based on a membership function that clears the fuzziness between domains. Then, the base station run a genetic algorithm process with each domain separately to find better optimized clustering, i.e., finding cluster heads. Thereafter, the cluster head passes a token to its immediate neighbor in the cluster to exchange roles, and thus saving cluster mean energy. Similar to LEACH, our protocol operates in a distributed manner where decisions are made based on the local information only. That is, each node makes its decision in the cluster formation process using the messages received from CHs in its communication range. No global state maintenance is required.

We have analyzed the performance of our protocol GAGLCP through simulations, and compared its performance with the LEACH protocol. Our GAGLCP shows how the combination between genetic algorithm and fuzzy logic can be used in the cluster formation process to distribute the tasks and energy consumption over all the nodes in a WSN. The results show an outperforming behavior compared to LEACH protocols vis-à-vis the total residual energy and the number of live nodes.

As a future work, some extensions of the GAGLCP approach can be made. For instance, the fitness function can be enhanced to maximize the overall performance to have the best chromosomes, and yet in a lesser search time. The protocol modification can take in consideration different parameters such as centrality or/and density of nodes. Choosing various parameters may further improve the performance of this work.

References

[1] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. of the 33rdAnnual Hawaii International Conference on System Sciences (HICSS),* Maui, HI, Jan. 2000.

[2] W. Heinzelman, A. Chandrakasan and H. Balakrishnan, "An application-specificprotocol architecture for wireless microsensor networks," in *IEEE Transactions on Wireless Communications,* Oct. 2002.

[3] B. Warneke, M. Last, B. Liebowitz and K.S.J. Pister, "Smart Dust: communicating with a cubic-millimeter computer", *IEEE Computer,* Jan. 2001,

[4] E. Cayirci, "Data aggregation and dilution by modulus addressing in wireless sensor networks," *IEEE Communications Letters,* Aug. 2003.

[5] M.J. Handy, M. Haase and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proc. 4th International Workshop on Mobile and Wireless Communications Network,* Sept. 2002.

[6] A. Bhargava, M. Zoltowski, "Sensors and wireless communication for medical care," in *Proc. 14th International Workshop on Database and Expert Systems Applications,* Sept.2003.

[7] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, J. Schiller, "Solar-aware clustering in wireless sensor networks," in *proc. Ninth International Symposium on Computers and communications,* Jun. 2004.

[8] M. Negnevitsky, *Artificial intelligence: A guide to intelligent systems*, Addison-Wesley, Reading, MA, 2001.

[9] M.J. Handy, M. Haase and D. Timmermann, "Low energy adaptive clustering hierarchy with deterministic cluster-head selection," in *Proc. 4th International Workshop on Mobile and Wireless Communications Network,* Sept. 2002.

[10] A. Bhargava, M. Zoltowski, "Sensors and wireless communication for medical care," in *Proc. 14th International Workshop on Database and Expert Systems Applications,* Sept. 2003, pp. 956 - 960.

[11] V. Shankar, A. Natarajan, S.K.S. Gupta, L. Schwiebert, "Energy-efficient protocols for wireless communication in biosensor networks," in *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,* Sept. 2001, pp. D-114 - D-118.

[12] Y. Xu, J. Heidemann and D. Estrin, "Geography-informed energy conservation for adhoc routing," *ACM SIGMOBILE,* Jul. 2001, pp. 70 - 84.

[13] C. Karlof, D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, May. 2003, pp. 113 - 127.

[14] A.D. Wood, J.A. Stankovic, "Denial of service in sensor networks," in *IEEE Computer,* Oct. 2002, pp. 54 - 62.

[15] M. Ettus, "System capacity, latency, and power consumption in multihop- routed SSCDMA wireless networks," in *Proc. Radio and Wireless Conf. (RAWCON),* Colorado Springs, CO, Aug. 1998, pp. 55 - 58.

[16] S. Singh, M. Woo, and C. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proc. 4th Annual ACM/IEEE Int. Conf. Mobile Computing Networking (MobiCom),* Oct. 1998.

[17] T. Voigt, A. Dunkels, J. Alonso, H. Ritter, J. Schiller, “Solar-aware clustering in wireless sensor networks,” in *proc. Ninth International Symposium on Computers and Communications,* Jun. 2004, pp. 238 - 243.

[18] S. Bandyopadhyay, E.J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” *IEEE INFOCOM 2003,* Apr. 2003.

[19] L. Reznik, V. Kreinovich, “Fuzzy and probabilistic models of association information in sensor networks,” in *proc. IEEE International Conference on Fuzzy Systems,* Jul.2004, pp. 185-189.

[20] L.Chou, L. Gen-Chung, “Location management using fuzzy logic control for wireless networks,” in *International Conferences on Info-tech and Info-net,* Oct. 2001.

[21] M.N. Halgamuge, S.M. Guru, A. Jennings, “Energy efficient cluster formation in wireless sensor networks,” in *10th International Conference on Telecommunications,* Mar. 2003, pp. 1571 - 1576.

[22] *http://robotics.eecs.berkeley.edu/~pister/SmartDust/,* Mar. 2005.

[23] A. Safiotti, “Fuzzy logic in autonomous robotics: behavior coordination,” in *proc. Sixth IEEE International Conference on Fuzzy Systems*, Jul. 1997, pp. 573 – 578.

[24] L. Schwiebert, S.K.S. Gupta, J. Weinmann, “Research Challenges in Wireless Networks of Biomedical Sensors,” *ACM SIGMOBILE*, Jul. 2001, Rome, Italy.

[25] Q. Liang, “Cluster head election for mobile ad hoc wireless network,” in *Proc. 14thIEEE International Symposium on Personal, Indoor and Mobile RadioCommunications, (PIMRC),* Sept. 2003, pp. 1623 - 1628.

[26] E. J. Duarte-Melo, M. Liu, Analysis of energy consumption and lifetime of heterogeneous wireless sensor networks, in: Proceedings of the IEEE Global Telecommunications Conference, Taipei, Taiwan, vol. 1, 2002.

[27] K. Akkaya, M. Younis, A survey on routing protocols for wireless sensor networks, IEEE Transactions on Mobile Computing 3 (3) (2005).

[28] F Najafi, M Abbasi, H rostami,“ Prolonging the Network Lifetime by Artificial Immune System”, (IJAEST) International Journal of Advanced Engineering Seiences and Technologies, VOL NO.6. Issue no.2, pp. 261-264. 2011.

[29] S. Technologies, “Qualnet v. 3.9. 5 user’s guide,” 2006.

[30] Jurcık, P., Koubaa, A., “The IEEE 802.15. 4 OPNET Simulation Model: Reference Guide v2. 0,” 2007.

[31] Notani, S., “Performance Simulation of Multihop Routing Algorithms for Ad-Hoc Wireless Sensor Networks Using TOSSIM,” in Advanced Communication Technology,2008. ICACT 2008. 10th International Conference on, vol. 1, 2008.

[32] Varga, A., Hornig, R., “An overview of the OMNeT++ simulation environment,” in Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops table of contents. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering) ICST, Brussels, Belgium, Belgium, 2008.

[33] Downard I., DC, N., “Simulating sensor networks in ns-2,”2004.

[34] Titzer, B., Lee, D. Palsberg, J., “Avrora: Scalable sensor network simulation with precise timing,” in Proceedings of the 4th international symposium on Information processing in sensor networks. IEEE Press Piscataway, NJ, USA, 2005.

[35] Hou, J., Kung, L., “J-Sim: A Simulation and emulation environment for wireless sensornetworks,” IEEE Wireless Communications Magazine, vol. 13, no. 4, pp. 104–119,2006.

[36] Polley, J., Blazakis, D., “Atemu: A fine-grained sensor network simulator,” in Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on, 2004, pp. 145–152.

[37] Girod, L., Ramanathan, N.," A Software Environment for Developing and Deploying Heterogeneous Sensor Actuator Networks," Center for Embedded Network Sensing, p. 101, 2007.

[38] MATLAB Web Site: http://www.mathworks.com/