**Optimization of CH selection with energy efficient clustering and routing for lifetime maximization of CRWSN**

*Submitted in partial fulfillment of the requirements for the degree of*

**Bachelor of Technology**

in

**Programme**

*By*

**Abhishek Iyer**

**19BEC0806**

**Under the guidance of Dr. Karthikeyan**

**SENSE VIT, Vellore.**



**DECLARATION**

I hereby declare that the thesis entitled “Optimization of CH selection with energy efficient clustering and routing for lifetime maximization of CRWSN" submitted by me, for the award of the degree of *Bachelor of Technology in Programme* to VIT is a record of bonafide work carried out by me under the supervision of Dr.Karthikeyan.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

|  |  |
| --- | --- |
| Place: Vellore | Abhishek Iyer |
| Date: 11-04-23 | **Signature of the Candidate** |

**CERTIFICATE**

This is to certify that the thesis entitled “Optimization of CH selection with energy efficient clustering and routing for lifetime maximization of CRWSN” submitted by **Abhishek Iyer & 19BEC0806**, **SENSE**, VIT, for the award of the degree of *Bachelor of Technology in Programme*, is a record of bonafide work carried out by him under my supervision during the period, 01. 12. 2022 to 30.04.2023, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

|  |  |
| --- | --- |
| Place: Vellore | Professor Name:  Dr.Karthikeyan A |
| Date: 16th April 2023 |  |
|  |  |

|  |  |
| --- | --- |
|  |  |

ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have gotten this experience all along the completion of my project. I respect and thank Dr.Karthikeyan A, for providing me the opportunity to work under him and for providing the guidance required to complete the project. I owe a deep gratitude to him for taking out time from his busy schedule.

I would also like to express my gratitude to the chancellor, Dr. G Viswanathan, the dean, Dr.Sivanantham S, the HoD, Dr.Karthikeyan B for providing me with an environment to work in and for their inspiration during the tenure of the project.

In a jubilant mood I express ingeniously my whole-hearted thanks to all the teaching staff and members working as limbs of the university for their not-self centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. Additionally, I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At the last but not the least, I express my gratitude and appreciation to all those who have helped me directly or indirectly towards the successful completion of this project.

**Student Name: Abhishek Iyer**

Executive Summary

Wireless sensor networks are small devices that are interlinked with each other that need to send information in the form of data to the base station. The WSN node usually comprise of a sensing node, transceivers, a processor and a battery. The lifetime of a battery is limited and the lifetime of the WSN is dependent on the lifetime of the batteries as it would not be possible to replace each battery once its dead. Therefore, to maximize the lifetime of the network, we need to efficiently utilize the energy. The energy can be maximized based on how efficiently a node sends a packet to the sink. This thesis proposes a hybrid routing algorithm to increase the lifetime of the network.   
In addition to an efficient routing algorithm this paper introduces the concept of cognitive radio wireless sensor networks as the rapid proliferation of low-cost wireless applications in unlicensed spectrum bands has resulted in spectrum scarcity among those bands. The performance of these networks would decrease over time as the popularity of these networks increase. To solve this issue, this paper proposes the sharing of the under-utilized licensed spectrum when the primary user (PUs) is not active. Cognitive radio is a new paradigm in wireless communication that allows sensor nodes as the unlicensed users or Secondary Users (SUs) to detect and use the under-utilized licensed spectrum temporarily. The primary users would not be aware of the existence of the secondary users, the SU opportunistically uses the licensed spectrum on a temporary basis thereby increasing the performance of the network. A reinforcement learning algorithm called Q learning is used to simulate the dynamic channel allocation. The algorithm results in an efficient increase in lifetime of the nodes as well as future-proofs the wireless sensor network by incorporating cognitive radio.

List of figures

|  |  |  |  |
| --- | --- | --- | --- |
| **Figure No.** |  | **Title** | **Page No.** |

1.3.2.1 Architecture of CRWSN 14

2.1.7.1 WPO-EECRP protocol

2.2.1 Cognition Cycle 23

3.1 Initial clustering and CH selection 26

3.2 Flowchart for GSO 27

3.3 Flowchart for FFOA 28

3.4 Flowchart for FGF 29

3.5 Block diagram for Dynamic channel Allocation 30

7.1.1 Analysis of alive nodes Vs Number of rounds (100 nodes) 43

7.1.2 Analysis of alive nodes Vs Number of rounds (200 nodes) 44

7.2.1 Analysis of throughput Vs Number of rounds (100 nodes) 45

7.2.2 Analysis of alive nodes Vs Number of rounds (200 nodes) 45

7.3.1 Distance analysis for GSO and FFOA 48

7.3.2 Distance analysis for FGF 48

7.4.1 Number of interfering channels Vs Time (ε=0.8) 49

7.4.2 Number of interfering channels Vs Time (ε=0.4) 49

List of tables

**Table No. Title Page No.**

* 1. Comparison between WSN and CRWSN 12

2.1.1.1 Advantages and Disadvantages of GSO 17

2.1.2.1 Advantages and Disadvantages of FFOA 20

2.1.3.1 Advantages and Disadvantages of ABC 23

2.4.1.1 Advantages and Disadvantages of ALO 26

7.1.1 Alive nodes analysis 44

7.2.1 Throughput analysis 46

List of abbreviations

WSN Wireless sensor network

CRWSN Cognitive radio wireless sensor network

CH Cluster Head

PU Primary User

SU Secondary User

GSO Glowworm Optimization Algorithm

FFOA Fruit Fly Optimization Algorithm

ALO Ant Lion Optimization Algorithm

ABC Artificial Bee Colony Algorithm

FGF Fitness based Glowworm swarm with Fruitfly Algorithm

**Contents**

|  |
| --- |
| **Acknowledgments……………………………..……………………………………iii**  **Executive Summary……………………………….………………….…………….iv**  **Contents…………………………………………..………………………………….v**  **List of Figures……………………………….………………………….…………..vi**  **List of table……………………………….…………………………………..……vii**  **List of abbreviations………………………………………………………..….…viii**  **Symbols and Notations………………………………………………………….…ix** |
| [**1. INTRODUCTION 12**](#_Toc132538942)  [**1.1 Objective 12**](#_Toc132538943)  [**1.2 Motivation 12**](#_Toc132538944)  [**1.3 Background 13**](#_Toc132538945)  [**1.3.1 Comparison between WSN and CRWSN 13**](#_Toc132538946)  [**1.3.2 Terms related to WSN/CRWSN 14**](#_Toc132538947)  [**1.3.3 CRWSN Architecture 14**](#_Toc132538948)  [**2. LITERATURE SURVEY 16**](#_Toc132538949)  [**2.1 Energy optimization algorithms 16**](#_Toc132538950)  [**2.1.1 Glowworm optimization algorithm 16**](#_Toc132538951)  [**2.1.2 Fruit fly optimization algorithm 18**](#_Toc132538952)  [**2.1.3 Artificial Bee Colony Algorithm 21**](#_Toc132538953)  [**2.1.4 Ant lion optimization Algorithm 24**](#_Toc132538954)  [**2.1.5 A modified Glowworm swarm optimization 27**](#_Toc132538955)  [**2.1.6 Particle swarm optimization-based clustering to prevent residual nodes in a wireless sensor network 28**](#_Toc132538956)  **2.1.7** **WPO-EECRP: Energy-Efficient Clustering Routing Protocol Based on Weighting and Parameter Optimization in WSN………………………………………………………………………………………………….30**  [**2.2 Cognitive radio wireless sensor networks 35**](#_Toc132538957)  [**2.2.1 Cognitive Radio-based Wireless Sensor Networks 35**](#_Toc132538958)  [**2.2.2 Q-Learning for Cognitive Radios 38**](#_Toc132538959)  [**3. PROJECT DESCRIPTION AND GOALS 41**](#_Toc132538960)  [**4. TECHNICAL SPECIFICATIONS 49**](#_Toc132538961)  [**5. DESIGN APPROACH AND DETAILS 50**](#_Toc132538962)  [**6. SCHEDULE, TASKS AND MILESTONES 60**](#_Toc132538963)  [**7. PROJECT DEMONSTRATION 61**](#_Toc132538964)  [**7.1 Number of alive nodes Vs Number of rounds 61**](#_Toc132538965)  [**7.2 Throughput of the Network 63**](#_Toc132538966)  [**7.3 Deviation in CH Distance 65**](#_Toc132538967)  [**7.4 Interference Vector Analysis 66**](#_Toc132538968)  [**8. RESULTS AND DISCUSSION 69**](#_Toc132538969)  [**9. REFERENCES 70**](#_Toc132538970) |

1. INTRODUCTION

A wireless sensor network (WSN) is a collection of multiple sensor nodes that are spread throughout an area which are used to capture surrounding data such as temperature, humidity, air flow and so on. These sensor nodes that are interconnected to each other wirelessly then send this data to the sink node which then sends the data to the cloud through a gateway. The energy required to operate the sensor nodes in a WSN is supplied via a non-replaceable battery. The energy consumption from the battery is greater during activities such as data sensing, data computation and data communication between the respective nodes. Re-energizing or replacement of the batteries are considered to be impractical and, in some applications, impossible. Hence it is crucial to design an energy efficient WSN for it to be considered reliable.

In recent years, there has been a rapid proliferation of low-cost wireless sensor application in the unlicensed spectrum. This limited unlicensed radio spectrum is being shared by many wireless applications including Bluetooth, WiFi, WiMAX, and Zigbee. In addition to these wireless applications, microwave ovens and cordless phones operate in this spectrum. As the demand of these wireless applications increases the performance of the traditional WSNs will inevitably begin to degrade. Sharing of the under-utilized licensed spectrum amongst unlicensed spectrum is a promising solution. Cognitive radio (CR) in wireless sensor networks allows the sensor nodes called the secondary users (SUs) to use the under-utilized licensed spectrum (white spaces) opportunistically without interfering the primary user (PUs).

1. Objective

The main objective of this thesis is to increase the lifetime of a network by efficiently selecting a cluster head based on a fitness criteria using a hybrid CH selection algorithm and to incorporate the dynamic channel allocation in the WSN as the popularity of the devices that use the unlicensed spectrum is increasing by the day, to avoid the degradation of the WSNs, whitespaces from licensed spectrum must be used opportunistically.

1. Motivation

This paper will focus on the following challenges and requirements to design a reliable WSN.

**Increased lifetime:** Lifetime of a network is defined as the time taken by the first node in the network to fail. Wireless sensor nodes are usually either deployed in un-inhabitable or remote locations across a wide area. It is not practical to re-charge each sensor node once its battery drains out. Hence lifetime of the network is considered to be one of the most important metrics to analyze the efficiency and reliability of a network.

**Mitigating the effects of Shadowing and Fading:** The errors caused while spectrum sensing due to obstructions between the receiver and transmitter or due to the limitations in range can be catastrophic. The main objective of cognitive radio is the utilization of the licensed spectrum without the interference of the primary user. The effects of shadowing and fading can cause the sensors to brand the spectrum as a white space when it is technically in use. CRWSN applications should implement algorithm to mitigate these errors.

**Improved performance of channel selection:** The cognitive radio model should be able to choose the channels dynamically. When a channel is preoccupied by the PU, the SU must be switched to a backup channel.

**Compliance with timing requirements:** The channel occupation time for a node can range from a couple of milliseconds to hours. It is vital that the model has an efficient switching mechanism which complies with the timing requirements. For instance, a SU node must detect the PU’ signal within the Channel Detection Time (CDT) for a signal greater than the Incumbent Detection Threshold (IDT). During notification and recovery, a node must cease all transmissions within the Channel Move Time (CMT).

**Adaptability:** WSNs should be able to adapt to different scenarios that can take place in the real world. Sensor nodes might be added or removed. Nodes can cease transmission or nodes can even die due to energy constraints. The network should be able to remain stable and efficiently utilize its limited energy through optimized routing protocols to maximize its lifetime.

* 1. Background
     1. Comparison between WSN and CRWSN

Table 1.1 Comparison between WSN and CRWSN

|  |  |  |
| --- | --- | --- |
| **Parameters** | **WSN** | **CRWSN** |
| Medium used | Wireless – ISM Band | Wireless – Licensed band |
| Channel used | Single channel | Multi-channel |
| Channel requirements | Possible to create multi-channel requirement | Required to create multi-channel requirement |
| Memory | Restricted | Have huge capacity |
| Hardware availability | Readily available | Not readily available |
| Communication range | Short | Short but requires to be adaptable |
| Computation capabilities | Moderate computation | High computation |
| Routing topology | Broadcast from/to sink is relatively easier | Broadcast from/to sink is a complex process |
| Suitable | ISM band when it is not overcrowded | ISM band when it is overcrowded |
| Hardware constraints | Low processing capabilities, low memory capacity | Intelligent, cognition capabilities, moderate processing capabilities, moderate memory capacity. |
| Standards | Zigbee, IEEE 802.15.4, ISA100, IEEE1415 | IEEE 802.22, UHF/VHF |

* + 1. Terms related to WSN/CRWSN

**Sensor node:** A sensor node is an individual processing unit that is responsible for sensing, processing and transmitting data.

**Cluster head:** A node responsible for aggregating data and transmitting it to the base station.

**Sink/Base station:** The data that is sent by all the cluster heads are routed to the base station. It is also called a gateway as it acts as a bridge between the sensor nodes and the external environment.

**Cognitive radio:** The concept of the Cognitive radio technology aims at opportunistically making use of the licensed spectrum band by an unlicensed user with minimum allowable interferenceto the licensed user

**Primary user:** The primary user or the licensed user are the exclusive devices that use the licensed spectrum.

**Secondary user:** The secondary users or the unlicensed user are the devices opportunistically use the licensed spectrum without causing any interference to the primary user.

**Cognition cycle:** Cognition cycle enables the CR network to achieve context-awareness and intelligence through six main states.

* + 1. CRWSN Architecture

The architecture of a Cognitive radio wireless sensor network consists of:

1. RF Unit
2. Cognitive radio platform
3. Sensing unit
4. Processing unit
5. Power unit

**RF unit:** The RF unit is responsible to detect the various spectrums in use and relay the information to the cognitive radio platform.

**Cognitive radio platform:** The cognitive radio platform consists of the cognitive engine which enables the CR node to dynamically adapt their communication parameters with the help of the CR policy.

**Sensing unit:** The sensing unit consists of a couple of sensors and an analog to digital converter which then sends the converted digital signals to the processing unit.

**Processing unit:** The processing unit consists of a microcontroller which controls the application of the sensors present.

**Power unit:** The power supply is responsible for powering up the sensor nodes. The power supply is limited and hence it is vital that the sensor nodes use their power efficiently to extend their lifetime.



1. LITERATURE SURVEY

The current research in the field is mostly limited to only WSN and the different energy optimization algorithms of traditional WSNs. WSNs usually operate within the overcrowded unlicensed band of radio frequency spectrum and according to the studies sponsored by the Federal communications commission (FCC), the current static spectrum allocation has led to an overall low spectrum utilization where up to 70% of the allocated spectrum remains unused. These unused spectra are also called white spaces. Therefore, a balanced approach between energy utilization and the opportunistically utilization of the white spaces to avoid the performance degradation has been considered.

* 1. Energy optimization algorithms
     1. **Glowworm optimization algorithm**

Glowworm optimization algorithm is a swarm intelligence algorithm modeled after the behavior of glowworms. It prescribes a decentralized decision-making solution which are useful for wireless sensor networks. Each glow-worm or agent in GSO is assumed to carry a luminous pigment called luciferin, whose quantity encodes the fitness of its location in the objective space. This allows the agent to glow at an intensity approximately proportional to the function value being optimized. The glow-worms that have a higher luciferin intensity attracts the glow-worms with a lesser luciferin value. The algorithm incorporates an adaptive neighbourhood range by which the effect of distant glow-worms is discounted when a glow-worm has sufficient number of neighbours.

It follows three major mechanisms

**Fitness broadcast:** Glowworms carry a luminescent pigment called luciferin, whose quantity encodes the fitness of their locations in the objective space. This allows them to glow at an intensity that is proportional to the function value being optimized. It is assumed that the luciferin level of a glowworm as sensed by its neighbor does not reduce due to distance.

**Positive taxis:** Each glowworm is attracted by, and moves toward, a single neighbor whose glow is brighter than that of itself; when surrounded by multiple such neighbors, it uses a probabilistic mechanism to select one of them.

**Adaptive neighborhood:** Each glowworm uses an adaptive neighborhood to identify neighbors A glowworm i considers another glowworm j as its neighbor if j is within the neighborhood range of i and the luciferin level of j is higher than that of i.

The implementation of GSO follows 4 major stages

1. **Initialization:** The glowworms are distributed into the search space randomly having identical luciferin energies and similar decision domain GM0.
2. **Luciferin-Update**: The luciferin intensity of the glowworm is related to the location of the glowworm. The glowworm’s position gets varied as the iterations increases and the value of the luciferin gets updated after each iteration. During the luciferin-update phase, each glow-worm adds, to its previous luciferin level, a luciferin quantity proportional to the fitness of its current location in the objective function space. Also, a fraction of the luciferin value is subtracted to simulate the decay in luciferin with time. The luciferin update rule is given by:

…. (2.1.1.1)

Where LUg(t) is the luciferin intensity of g­­th glowworm at time t

J(Xg(t)) is the objective function of gth glowworm at time t

η refers to the luciferin improvement constant.

ν refers to the luciferin decay constant.

1. **Movement phase** During the movement phase, each glow-worm decides, using a probabilistic mechanism, to move toward a neighbour that has a luciferin value higher than its own. All the glowworms choose their neighbors and follow it with that distinctive possibility. The glowworms are attracted to neighbors that glow brighter. The glowworm needs to satisfy two particular criterions, the first, the glowworm needs to be in the decision domain and the second, the luciferin value needs to be greater than that of its neighbors. Further g glowworms move towards w neighbors which has a probability PTgj(t).

(t) = …. (2.1.1.2)

Once the movement takes place, the position of all the glowworms is updated by the following equation

= + size \* () …. (2.1.1.3)

1. **Neighborhood range update:** The neighborhood range update by after the implementation of te movement phase. Each glow-worm tries to find its neighbors. A glowworm is a neighbor of a glow-worm i only if the distance between glow-worms are lesser than the decision range of that glowworm and additionally, if the glowworm has a higher luciferin value compared to glowworm i. If one glow-worm has multiple neighbours, it chooses one at random with probability proportional to the luciferin level of this neighbour. Finally, a glow-worm moves one step in direction of the chosen neighbour.

If the luciferin density of the neighborhood is not up to the required level, we increase the neighborhood, else the neighborhood gets minimized.

…. (2.1.1.4)

This process is then repeated for a set number of rounds.

Table 2.1.1.1 Advantages and Disadvantages of GSO

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| GSO can deal with highly non- linear, multi-modal optimization problems naturally and efficiently | The dynamic change of decision domains in the method of glowworms moving, the algorithm slows convention speed and has poor local search ability delayed in the iteration |
| The speed of convergence of GSO is very high in probability of finding the global optimized answer | GSO is prone to premature convergence to suboptimal solutions if the parameters are not set correctly. |
| GSO can be adapted to different optimization problems by adjusting its parameters and tuning its behavior. | GSO is poor in high dimensional problems. |
| GSO can be used to optimize large-scale problems with a large number of variables and constraints. | GSO requires careful tuning of its parameters to achieve optimal performance. |

1. **Fruit fly optimization algorithm**

FFOA, or Fruit Fly Optimization Algorithm, is a swarm optimization technique inspired by the foraging behavior of fruit flies. The algorithm uses the vision and smell function of the fruit fly to optimize a given objective function. FFOA is based on the concept of clustering, which involves dividing a large population into smaller groups or clusters to simplify problem solving. FFOA aims to minimize the distance between cluster heads (CHs) and cluster nodes, which are the data points within each cluster.

FFOA is implemented in four major phases. The first phase involves measuring the distance between neighboring CHs and taking the reciprocal of the value obtained. The FF algorithm is then used to select the minimum distance between CHs and cluster nodes. This algorithm sums the values along a particular distance and applies the two main functions of fruit flies. The vision function selects the minimum distance from all possible distances, while the smell function identifies the direction with the maximum odor concentration. By applying these functions, FFOA is able to optimize the objective function by adjusting the position of CHs and cluster nodes to minimize the distance between them.

**1.Initialization phase:**

The FFOA algorithm begins with an initialization phase where the fruit flies are randomly dispersed in the search space along the X and Y axes. This randomization helps in exploring a wide range of the search space, increasing the chances of finding the global optimum solution. The random vector rv is also introduced during this phase to add further randomness to the initialization process.

The random vector rv adds an element of unpredictability to the initialization phase, preventing the algorithm from getting stuck in local optima. This ensures that the algorithm explores a wider range of the search space, increasing the likelihood of finding a better solution. By combining the random dispersal of fruit flies and the addition of the random vector rv, the initialization phase sets the stage for the rest of the algorithm to operate effectively.

This random vector is sampled from a uniform distribution.

= + rv …. (2.1.2.1)

= + rv …. (2.1.2.2)

2. **Path construction phase**:

The path construction phase in FFOA is the process of constructing a path from the fruit fly's current position to the food source, which represents the optimal solution of the optimization problem. This phase consists of two steps:

Vision function: In this step, the fruit fly evaluates the distance between its current position and all other positions in the search space. It then selects the position that is closest to the food source, based on its vision function. The vision function of a fruit fly represents its ability to detect and navigate towards the food source using its vision.

Smell function: Once the fruit fly has selected the position closest to the food source based on its vision, it applies its smell function to determine the direction in which it should move to reach the food source. The smell function represents the fruit fly's ability to detect and follow the chemical trails left by other fruit flies that have already found the food source.

By combining these two functions, the fruit fly constructs a path towards the food source. This path may not be the optimal path, but it is a path that leads the fruit fly closer to the food source. The fruit fly repeats this process several times, constructing different paths towards the food source, and eventually converges to the optimal solution.

We determine the distance value and the smell concentration value in this phase using the equations given below.

= …. (2.1.2.3)

= …. (2.1.2.4)

Where Distancei indicates the distance between the ith fruit fly to the food location and SMiC indicates the smell concentration value.

3. **Fitness evaluation:**

The fitness evaluation phase in FFOA is where the fitness value of each fruit fly is evaluated based on the objective function. In other words, the objective function is applied to the current position of each fruit fly to determine how well it solves the optimization problem at hand.

The fitness function is problem-dependent and is usually provided by the user. It is a measure of the quality of a solution based on the problem at hand. The fitness value of a solution is used to determine its probability of being selected for the next generation. Higher fitness values mean higher chances of being selected for the next generation, while lower fitness values mean lower chances of being selected.

After evaluating the fitness of each fruit fly, the algorithm proceeds to the next phase, which is the update phase. In this phase, the best solution found so far are used to update the position of the fruit flies, leading to the discovery of better solutions.

We now use the smell concentration values as an indicator to achieve the best fitness value.

= *function* () …. (2.1.2.5)

= max () …. (2.1.2.6)

Where smelli indicates the value of smell concentration of distinctive fruit fly and smellbest and indexbest represent the highest element and its respective indices.

**4.Movement phase:**

The movement phase is the third phase of the FFOA algorithm. In this phase, the fruit flies adjust their positions based on their current positions, the positions of their neighbors, and the information they have gathered about the environment in the previous phases. The main goal of this phase is to improve the fitness of the fruit flies by guiding them towards the promising regions of the search space.

To adjust their positions, each fruit fly considers the following three factors:

Individual attraction: Each fruit fly is attracted towards the direction of the best position it has visited so far.

Social attraction: Each fruit fly is attracted towards the average position of its neighboring fruit flies.

Global attraction: Each fruit fly is attracted towards the global best position discovered by any fruit fly in the swarm.

These three factors are combined using weighted coefficients to determine the new position of each fruit fly. The weights are determined using a random process that ensures exploration of the search space.

After adjusting their positions, the fruit flies repeat the path construction and fitness evaluation phases to update their fitness values and explore the search space further. This process continues until a stopping criterion is met, such as a maximum number of iterations or attainment of a satisfactory fitness level.

Table 2.1.2.1 Advantages and Disadvantages of FFOA

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Efficient in solving complex optimization problems. | May get stuck in local optima |
| Flexible and adaptable algorithm for different types of problems. | Can be slow in finding the global optimal solution. |
| Low computational cost and simplicity. | Can suffer from premature convergence. |
| Provides high-quality solutions. | Performance may depend on the tuning of parameters |
| Convergence is fast in the early stages of optimization. | Can require a large number of iterations to reach convergence. |
| Robust to noise and does not require gradient information. | The algorithm may not scale well for very large-scale problems. |
| Can handle multi-modal problems. | The convergence speed may decrease for problems with a large number of dimensions. |

1. **Artificial Bee Colony Algorithm**

Artificial Bee Colony (ABC) is one of the most recently defined algorithms motivated by the intelligent behavior of honey bees. It is as simple as Particle Swarm Optimization (PSO) and Differential Evolution (DE) algorithms, and uses only common control parameters such as colony size and maximum cycle number.

ABC as an optimization tool, provides a population-based search procedure in which individuals called foods positions are modified by the artificial bees with time and the bee’s aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. In ABC system, artificial bees fly around in a multidimensional search space and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one. Thus, ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process.

In ABC, a population-based algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees is equal to the number of solutions in the population.

At the first step, a randomly distributed initial population (food source positions) is generated. After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise, she keeps the position of the one in her memory.

After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts.

ABC possesses three kinds of agents: employed bees, onlooker bees and scout bees. These bees are represented in the form of partitioned adjacency matrix of the molecule. The algorithm goes through three phases: the employed phase, the onlooker phase and the scout phase.

**Employed Phase:**

In the employed phase, a set of feasible solutions are created by changing one group at a time.

Employed bees search for new food sources (υm→) having more nectar within the neighbourhood of the food source (xm→) in their memory. They find a neighbour food source and then evaluate its profitability (fitness). For example, they can determine a neighbour food source υm→ using the formula given by equation

= + ( - ) …. (2.1.3.1)

where xk→ is a randomly selected food source, i is a randomly chosen parameter index and ϕmi is a random number within the range [−a,a] . After producing the new food source υm→ , its fitness is calculated and a greedy selection is applied between υm→ and xm→ .

The fitness value of the solution, fitm(xm→) , might be calculated for minimization problems using the following formula

…. (2.1.3.2)

where fm(xm→) is the objective function value of solution xm→ .

**Onlooker Phase:**

In the onlooker phase, these solutions are evaluated by onlooker bees and ranked from best to worst depending on their property values.

Unemployed bees consist of two groups of bees: onlooker bees and scouts. Employed bees share their food source information with onlooker bees waiting in the hive and then onlooker bees probabilistically choose their food sources depending on this information. In ABC, an onlooker bee chooses a food source depending on the probability values calculated using the fitness values provided by employed bees. For this purpose, a fitness-based selection technique can be used, such as the roulette wheel selection method

The probability value pm with which xm→ is chosen by an onlooker bee can be calculated by using the expression given in equation

= …. (2.1.3.3)

After a food source xm→ for an onlooker bee is probabilistically chosen, a neighbourhood source υm→ is determined by using equation (6), and its fitness value is computed. As in the employed bees phase, a greedy selection is applied between υm→ and xm→ . Hence, more onlookers are recruited to richer sources and positive feedback behaviour appears.

**Scout Phase:**

Finally, in the scout phase, if a solution is not improved within a certain number of iterations, it is abandoned and replaced with a new solution.

The unemployed bees who choose their food sources randomly are called scouts. Employed bees whose solutions cannot be improved through a predetermined number of trials, specified by the user of the ABC algorithm and called “limit” or “abandonment criteria” herein, become scouts and their solutions are abandoned. Then, the converted scouts start to search for new solutions, randomly. For instance, if solution xm→ has been abandoned, the new solution discovered by the scout who was the employed bee of xm. Hence those sources which are initially poor or have been made poor by exploitation are abandoned and negative feedback behavior arises to balance the positive feedback.

Table 2.1.3.1 Advantages and Disadvantages of ABC

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| The structure of the algorithm is favorable for parallel processing, thus saving time | Slow down when used in sequential processing. |
| Ability to explore local solutions | Slow to obtain accurate solutions. |
| Global optimizer, with effective search process even under high complexity, and with low risk of premature convergence | The possibility of losing relevant information on the behavior of the function to be optimized. |
| Broad applicability, even in complex functions, or with continuous, discrete or mixed variables | High number of objective function evaluations. |

1. Ant lion optimization Algorithm

The Ant Lion Optimizer (ALO), also known as Antlion Optimizer, is a meta-heuristic optimization algorithm inspired by the interaction between ants and antlions in nature. The algorithm uses a mathematical model to simulate the behavior of ants exploring a bounded search space while avoiding being trapped by antlions. The ALO is designed to solve optimization problems by implementing a set of rules that govern the random walk of the ants, the construction of traps, the entrapment of prey, and the rebuilding of traps.

The ALO algorithm utilizes a fitness function to evaluate the performance of the ants as they move through the search space. The ants move through the world using a stochastic policy, which involves a random walk that allows them to explore the search space while avoiding being trapped by antlions. The goal of the ALO is to find the optimal solution to an optimization problem by converging the population of ants to a set of good solutions. By simulating the behavior of ants and antlions in nature, the ALO provides an easy-to-use and effective optimization method that can be applied to a wide range of optimization problems.

The random walk of an ant is computed using the following equation:

In this case is the cumulative sum function. We also use a random function which we show below:

…. (2.1.3.3)

In the Ant Lion Optimizer, the antlions are not just passive entities waiting for ants to come to them. On the contrary, they play an active role in the optimization process. As solutions to the problem, the antlions dig pits, which serve as traps for the ants. When an ant falls into a pit, it becomes trapped and cannot escape. This affects the random walk of the ant, as it is forced to stay within the bounds of the pit. Over time, as more ants become trapped in a pit, the corresponding antlion becomes fitter, reflecting an improvement in the quality of the solution it represents.

To model this behaviour, we represent the antlions with a matrix. Each element of the matrix corresponds to a pit that the antlion has dug. The size and shape of each pit is determined by the antlion's position in the matrix, which is updated over time as the antlion captures more ants. The ants, on the other hand, are search agents that move through the space using a stochastic policy. Their goal is to find the best possible solution to the optimization problem, while avoiding the traps set by the antlions. The fitness of an ant is evaluated using a fitness function, which takes into account the distance it has travelled between antlion traps. By balancing exploration and exploitation, the Ant Lion Optimizer is able to effectively search the solution space and converge to a high-quality solution.

The random walk of ants, modelled to include the effect of antlion traps is shown below:

= + …. (2.1.3.4)

In the random walk equation above, we are normalizing the ant’s position. Here we represent the minimum value of the random walk along dimension as and the maximum value along dimension as. This is a normal min-max scaling. We extend this scaling to include the effect of antlion traps.

= + …. (2.1.3.5)

= + …. (2.1.3.6)

These normalization terms are used to keep the random walk within the bounds of a chosen antlion trap, by decreasing the bounds overtime we can achieve the effect of trapping an ant. This impacts the dynamics of the walk itself. If an ant enters a trap, the antlion attempts to knock it towards the center of the pit.

An ant is knocked towards the center of a trap by reducing the hypersphere an ant can explore around the trap. This is achieved through applying a ratio to the lower and upper bounds of the ant’s orbit. We will decay the size over time.

= …. (2.1.3.7)

= …. (2.1.3.8)

is a ratio represented as. In this equation is a constant for a given time interval and increased in a stepwise fashion. In this way allows us to control the amount of exploitation that takes place.

**Method**

The Ant Lion Optimizer (ALO) is a powerful optimization algorithm that combines elements of both ant colony optimization and particle swarm optimization. ALO is a population-based algorithm that starts with a random population of antlions (solutions) and uses ants to explore the solution space through random walks. The random walks are bounded within a specific region, and their movement is influenced by the proximity to antlions.

One of the key features of ALO is the way in which it gradually reduces the hypersphere an ant can explore around an antlion over time. This is done by setting a lower bound and an upper bound, which allows the algorithm to focus on increasingly narrow areas of the search space as it progresses.

When an ant is caught and becomes fitter than its corresponding antlion, the antlion updates its position in order to optimize its chances of catching more ants. This process helps to ensure that the best antlions converge to an optimal solution over time.

ALO also incorporates an elitism strategy that directs the search towards good solutions. An ant's random walk takes place around two antlions, with the first antlion chosen in a roulette wheel fashion proportional to its fitness, and the second being the elite antlion, which is the solution with the highest fitness discovered so far. The two proposed walks of the ant are then averaged together.

One of the key advantages of ALO is its ability to explore a high degree of the search space while avoiding local optima. The random walks of ants and the roulette wheel selection of antlions help to avoid stagnation in local optima, while the shrinking of antlion trap boundaries and the decreasing intensity of ant movements over time encourage convergence towards an optimal solution.

ALO is an easy-to-use, gradient-free, black-box optimization method with only a few tuneable parameters. It can be applied to a wide range of optimization problems and is part of a family of optimization methods, such as Moth Flame Optimization and Gray Wolf Optimization, which can be powerful additions to the optimization toolbox.

Table 2.4.1.1 Advantages and Disadvantages of ALO

|  |  |
| --- | --- |
| Advantages | Disadvantages |
| Easy to implement and requires minimal parameter tuning. | Not as well-established as other optimization algorithms like Genetic Algorithm or Particle Swarm Optimization. |
| Can handle both continuous and discrete optimization problems. | Convergence to global optima is not always guaranteed, and can be sensitive to initial parameter values. |
| Does not require gradient information, making it suitable for black-box optimization problems. | May not be the most efficient algorithm for high-dimensional optimization problems. |
| Has a good balance between exploration and exploitation, and can avoid getting stuck in local optima. | Can be computationally expensive for large-scale optimization problems. |
| Has shown promising results in a variety of applications, including engineering design, machine learning, and image processing. | May not be suitable for optimization problems with highly complex and nonlinear objective functions. |

1. **A modified Glowworm swarm optimization**

This modified GSO algorithm is used for the dynamic cluster formation and cluster head selection for an IoT based application. It uses the conventional GSO algorithm as the base parameters and changes few of the basic steps from the GSO algorithm. The system model consists of a set of sensors with sensor ranges and decision ranges. The initial voting index(v0) is set to 0 and the initial energy level is set to e0 and an initial luciferin level of l0. The initial cluster heads and their respective normal nodes are chosen. All of these normal nodes have their unique Id.

The neighborhood discovery is done as a request-response model. As a response of Hello message for neighbor discovery, nodes would send their unique Id. The sender node is able to compute the distance depending upon the response time taken by the receiver node. There is no need to know the position of all the nodes. After deployment of the nodes, the sink node gathers the initial properties of each node. After that, it initiates the cluster head selection algorithm for the first time. Hereafter, whenever energy of any cluster head becomes less than a threshold value, it sends a request to re-invoke the algorithm.

This algorithm mainly consists of three stages.

1. Luciferin update phase
2. Cluster formation phase
3. Neighborhood range update phase

Luciferin update phase:

Initially, each sensor has its own luciferin intensity. The luciferin intensity of each sensor node in the modified GSO algorithm is updated depending upon its luciferin intensity l(t), remaining residual energy e(t), decision range |Ht| and the voting index v(t).

The luciferin value is highly dependent on the connectivity index as that reduces the number of required clusters and a higher voting index is proportional to its luciferin index as it will help the selection of cluster heads to be geographically distributed and this would further improve the coverage of the network.

The luciferin update value for sensor si is given by

…. (2.1.3.1) Where w, x, y, z denotes the weighing factor. [11]

Cluster formation phase:

In the modified GSO algorithm the movement phase is replaced with the cluster formation phase. This phase is further divided into 2 sub-phases namely (1) Voting phase and (2) the cluster formation phase.

In the voting phase, an initial message is sent to all the single hop neighbors. Each node that receives the message, sends a reply to its sender along with a unique Id. During the cluster head selection phase, a voting mechanism is executed to select the cluster head with respect to every sensor node on behalf of every node, a vote is registered for the node that has the highest luciferin value among its neighbors. In case two nodes that the same luciferin value, the vote is registered to the one with the lower index. In the cluster head selection phase, all the non-cluster head nodes become a member of the cluster ci which has the maximum voting index among nodes.

Thus, in the voting phase, each node votes the node with the highest luciferin value in its neighbourhood, and voting index of each node is calculated. In the cluster formation phase, all the voted nodes are informed that they have been selected as cluster heads and also the set of nodes which have voted it become the member nodes of its respective cluster. This voting phase also prevents the overlapping of the clusters

Neighbourhood range update phase: The neighbourhood range update phase is identical to the conventional GSO algorithm and is given by the following formula

The above procedures are repeated until all the clusters have the desired number of neighbors or the clusters have become stable i.e., the cluster head has not been changes for 3 iterations.

1. Particle swarm **optimization-based** clustering to prevent residual nodes in a wireless sensor network

Clustering is a vital aspect of a wireless sensor network system and the Clustering method of data aggregation and transmission results in better lifetime as it eliminates the data redundancies. The clusters are formed and a cluster head is elected in each cluster, the sensing nodes send the data to the cluster head which sends the information to the base station. During the cluster formation phase, it is not sure if all the nodes become members of any cluster and few nodes are bound to be left out. These are generally called as individual nodes. These nodes utilize high energies to transmit data to the sink. The nodes may be grouped into a cluster during the next iteration but other nodes that were once part of the cluster for the previous iteration might find themselves as individual nodes and would be required to either transmit data to the sink directly or else such nodes need to send many control messages to find the next best hop for constructing the optimal routing path. This form of clustering pays a heavy toll on the network lifetime.

Another major factor that affects the network lifetime is that of the routing path construction. The constructed path should be highly reliable. In the proposed system, the concept of Particle Swarm optimization (PSO) is used to form the clusters while eliminating all the individual nodes in the network and Gravitational Search Algorithm (GSA) is used to find the next best hop. Parameters like position of the node, velocity and force between the cluster heads are considered for selecting next best hop. The clusters with maximum force are selected to forward the aggregated data from cluster head until it reaches the Base Station

Particle Swarm Optimization can be referred to as a random optimization technique based on population which was developed taking inspiration of the social behavior of fish schooling or birds flocking. They demonstrate solutions of complex non-linear optimization by imitating the flocking of the birds. PSO generally optimizes an issue based on candidate population in the search-space in conformity with the mathematical formulae over velocity and position of the particle. In each iteration, velocity of each particle is updated using the current velocity of the particle and the previous local best and global best position. The new velocity and the new positions can therefore be estimated using these values. The fitness of each particle revolves in the search space so far and an appropriate solution would be demonstrated by the neighbouring positions. The swarms will then meet at the optimal solution.

In other words, the particles are first initialized randomly and each and every particle has specific fitness value which is assessed by the aspect of fitness function. They possess velocities which manoeuvres the movement of particles and by the means of pursuing the current optimum particles, the particles could fly around the search space and update every generation that is better than that of the previous one.

Prediction of number of clusters in the network:

The (X,Y) be the sensing region and let r be the coverage of any sensor node. The network is divided into smaller portions called cluster with a radius of r. We then take (x,y) as the coordinates of a cluster in the sensing region.

The total number of clusters formed can be calculated using the formula given below:

The formula give above is used for calculating clusters formed for the lower bound regions and the Eq. <insert eq number> is used for calculating upper bound.

When x=y=t, X=Y and t=r=√2.

Clustering using PSO:

Once the nodes are deployed in the sensing region, the base station broadcasts a message called info\_collection\_request message to all the sensor nodes in the network to collect the node’s information. The sensing nodes then send the information to the base station which contains

1. The position of the sensor nodes
2. The velocity of the nodes V=(v1,v2) where v1 is the average velocity of the sensor node and v2 is the current velocity
3. Energy of the sensor node.

These values are maintained and updated in the base station and the base station supervises the performance of clustering.

Cluster formation takes place by considering the fitness value of each particle. The node which is accessible to maximum number of sensor nodes is elected as cluster particle, a cluster particle is also referred to as the cluster head. The fitness value is calculated for choosing a cluster particle depends on the following three factors namely

1. Energy of the particle.
2. Energy of the particles within the radio range from a particular particle
3. Distance of those particles within the radio range from a particular particle

The fitness values of each particle can be calculated using the following equation

Where α1 and α2 lie in the range of [0,1] and α3=1- α1- α2.

Where N is a set of number of nodes reachable from particle p and CN is the number of nodes reachable from particle p.

The below equation shows the updated velocities of each and every particle given by,

Where,

ω1 is the weight of the node velocity

ω2 and ω3 are the weight of node location

νt-1 is the velocity of previous position

pt-1 and pt are the previous position of the node and current position of the node respectively.

Location update of the particle can be done with the knowledge of the particle’s previous position and the updated velocity.

The fitness function of each particle is updated at each iteration and the node with the maximum fitness value is taken as the reference and constructing the cluster by making the nodes in its radio range as its cluster members. Additionally a cluster assistant (CA) is calculated which has the second highest fitness in that particular cluster and acts as a supporting node for the cluster head. The CA can act as a CH if the CH dies.

After the clustering is completed, the GSA algorithm is used for constructing an optimal path for transmitting the sensed data to the base station. The source node checks for the next best hop to transmit data to the base station. A route\_request message is sent to the neighbouring nodes. The message consists of valuable data such as position, velocity and energy of the node. Neighbour nodes forward the same request to its available neighbours by replacing the received position, velocity and energy value by its own value. The same process is repeated until it reaches the Base Station.

The gravitational search algorithm uses principles of Law of Gravity by Newton [21]. Newton’s Law of Gravity states that, ―Every particle attracts every other particle with a force F which is directly proportional to the product of masses and inversely proportional to the square of distance between them.

Where F is the force of attraction

G is the gravitational constant ((G = 6.8x10-11 m3 kg-1 s-2)

M1 and M2 are mass of the particle

R is the distance between the two particles.

The force of attraction is more for the nearer node than the farthest node. Higher the force of attraction implies that better transmission efficiency and reliability. In addition to that, the node with minimal distance is considered to choose the next hop. The distance between the two cluster heads from a particular cluster head varies. Route construction takes place by considering the next best hop with minimum distance and high force if attraction

**2.1.7 WPO-EECRP:**

Wireless sensor networks (WSNs) are becoming increasingly popular due to their ability to gather information from the environment and transmit it to a base station for analysis. However, WSNs are often deployed in remote and hard-to-reach locations, which makes replacing or recharging their batteries a challenging and expensive task. Therefore, energy efficiency is a critical factor to consider when designing WSNs.

One way to achieve energy efficiency in WSNs is by clustering the sensor nodes. Clustering involves grouping the sensor nodes into clusters and selecting a cluster head to communicate with the base station on behalf of the other nodes in the cluster. This reduces the overall energy consumption in the network as the nodes can communicate with the base station through the cluster head, thus reducing the number of transmissions required.

There have been several clustering protocols proposed in the literature, but the energy efficiency of these protocols varies depending on the clustering factors considered. Therefore, we propose a new energy-efficient clustering routing protocol called WPO-EECRP. This protocol considers multiple clustering factors related to energy consumption to select the cluster head. These factors include the residual energy of the nodes, their distance from the base station, the number of neighbors, and the number of neighbors of their neighbors.

To determine the optimal cluster head, we weigh these factors and transform the clustering problem into an optimization problem with two parameters: the neighbor communication range R and the weight coefficient W. We then configure the optimal parameters Ropt and Wopt to divide the network into clusters, which operate until data communication is completed.

WSN has been widely used in various fields, including military defense security, industrial and agricultural environmental monitoring, biological medical services, remote surveillance of dangerous zones, and many others, and its potential practical value has brought great convenience to people. However, each sensor node has limited power supply, which is used in data processing, such as data aggregation, multicast, and broadcast. As a result, energy supplementing is costly, and the network's survival time faces a challenge. Therefore, to prolong the network's lifetime, energy-efficient protocols and architectures are required.

Among the energy-saving protocols in WSN, clustering is an effective way to reduce energy consumption and extend the network lifetime. The clustering protocol divides the network into several clusters, with each cluster consisting of a cluster head (CH) and multiple cluster members (CMs). The CH is responsible for collecting and processing the data acquired by the CMs and sending the fused data to the base station (BS) or sink node. Research on clustering protocols has continued, mainly focusing on the CH election mechanism because CH election directly affects the energy consumption of networks.

One of the earliest classical clustering protocols for sensor networks is Low-Energy Adaptive Clustering Hierarchy (LEACH). It is a distributed, dynamic clustering protocol in which CHs are randomly generated through a probability formula in each round. The CH will announce to all nodes, and nodes select CH according to the received signal strength. LEACH-C is the centralized control version of LEACH. This protocol considers the residual energy as a constraint to ensure that the node whose remaining energy is greater than the average energy of the network can act as CH.

Weighted clustering algorithm (WCA) is a specially reactive clustering algorithm where CH election is based on a score function called "combined weight." This score function is a weighted linear combination of the degree, the mobility level, the transmission power, and the residual energy of the sensor node. The sensor node having the least weight value acts as CH. Energy-Efficient Hierarchical Clustering (EEHC) is another distributed, proactive, dynamic clustering algorithm where each sensor node becomes a candidate CH with probability p. Each node announces itself to its k-hops neighbor nodes, and once a node receives the announcement, the node chooses the closest candidate CH as its CH. Sensors that are not CHs and don't join any cluster become "Forced CHs." Eventually, this algorithm transforms the question of clustering to minimize energy consumption into the optimization of parameters p and k.

To achieve the goal of energy conservation, a new energy-efficient clustering routing protocol called WPO-EECRP has been proposed by combining the revelations of previous clustering protocols. This protocol considers multiple clustering factors related to energy consumption to select the CH, such as residual energy, distance from the node to the base station, neighbors, and the number of neighbors through weighting. Finally, it transforms the question of efficient clustering into the optimization of two parameters: neighbor communication range R and weight coefficient W of clustering factors. Therefore, the network is divided into clusters under the configuration of optimal parameters Ropt and Wopt and operates until it completes data

Algorithm:

The WPO-EECRP protocol is a fully distributed algorithm used to elect cluster heads in a wireless sensor network. The algorithm allows each node in the network to participate in the election process by exchanging information with its neighboring nodes and updating its score until it becomes a cluster head.

1. At the beginning of network operation, all nodes are ordinary nodes. Each node broadcasts a packet called NI\_ADV to its R-range neighbor nodes containing its ID. Upon receiving a NI\_ADV from its neighbor, each node creates a neighbor information table to store the ID and distance of the neighbor node.
2. Using the information gathered from the neighbor information table, each node calculates its clustering factors Q1, Q2, Q3, and Q4, and then calculates the score function using Equation (28). The score function determines the probability of a node becoming a cluster head.
3. To avoid contention during the cluster head announcement, each node backs off for a period of time Tbackoff before announcing itself as a cluster head. The value of Tbackoff is determined by multiplying the reverse of the node's score by a constant factor n, which ensures that Tbackoff is reasonable. The node with the largest score backs off for the shortest time and becomes the cluster head. It broadcasts a packet called CH\_ADV to all the nodes to inform them of its status.
4. Upon receiving CH\_ADV, all nodes cancel their back-off time and do not try to compete for becoming a cluster head. For all neighbor nodes of the cluster head, each node broadcasts a packet called JOIN\_ACK to its neighbors, including the cluster head node, to confirm that it has become a cluster member (CM) and will not participate in the competition as a cluster head in this round.
5. For neighbor nodes of the cluster member node that do not contain nodes that have already become CMs, each node receives the JOIN\_ACK packet and knows that the CM node has become a cluster member. This confirmation ensures that the node is no longer included in its neighbor nodes. The node then updates the number of its neighbor nodes, the distance between its neighbor node and itself, its clustering factors, and finally its score. These nodes then participate in the competition for the next cluster head. The process continues with steps until all nodes join the cluster, and each node becomes either a cluster head or a cluster member.

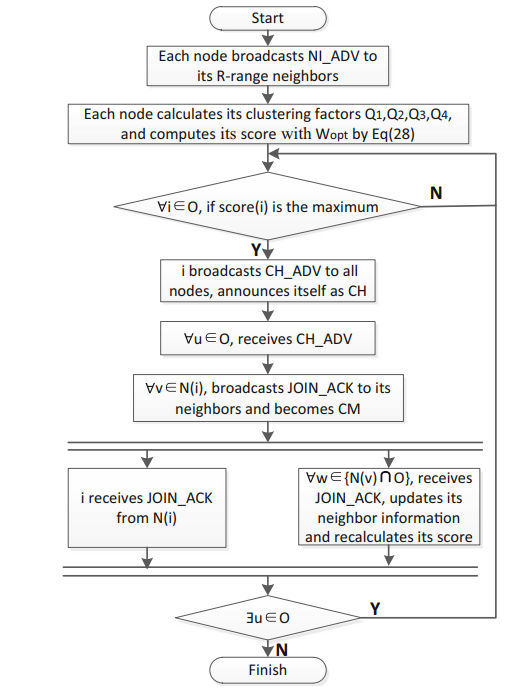


Fig 2.1.7.1 WPO-EECRP protocol

In conclusion, the WPO-EECRP protocol is proposed as a solution for the network clustering problem. This protocol utilizes a comprehensive approach in selecting the cluster head (CH) by considering all impact factors, including the node's distance to the sink, the remaining energy, and the degree of connectivity. The clustering parameter R and the weight parameter W are optimized to control the clustering in each round.

To evaluate the effectiveness of the WPO-EECRP protocol, multiple experiments are conducted. The results demonstrate that the protocol can significantly reduce energy consumption, prolong the network lifetime, and achieve better energy-efficiency performance compared to existing protocols. Moreover, the WPO-EECRP protocol has good scalability, and its clustering control can be further refined by adjusting the precision of the clustering parameters.

However, it is worth noting that the use of the timer in CH election and synchronization is a tradeoff between quality and time delay. While a longer timer can improve the quality of CH selection, it also increases the convergence time of the network. Therefore, it is essential to find a balance between these factors to achieve optimal performance.

In the future, the WPO-EECRP protocol will be applied to large-scale WSNs to investigate its advantages and disadvantages. Additionally, researchers aim to address related problems and make further improvements to enhance the protocol's performance. Overall, the WPO-EECRP protocol is a promising solution for energy-efficient clustering in WSNs.

* 1. Cognitive radio wireless sensor networks

The rapid proliferation of low-cost wireless applications in unlicensed spectrum bands has resulted in spectrum scarcity among those bands. Since most applications in Wireless Sensor Networks (WSNs) utilize the unlicensed spectrum, network-wide performance of WSNs will inevitably degrade as their popularity increases. Unlike traditional wireless sensor networks, cognitive radio based wireless sensor networks opportunistically use the licensed spectrum when the primary user is not active.

* + 1. **Cognitive Radio-based Wireless Sensor Networks**

As a Next Generation Sensor Network: Concept, Problems and Prospects gives a basic understanding of CRWSNs and the differences between the traditional WSN compared to CRWSNs and leveraging on the advantages of the opportunistic spectrum access provided by cognitive radio technology. Wireless sensor networks have the potential of operating at lower licensed spectrum band, for example the TV band with efficient spectrum usage and higher energy efficiency due to range extension.

One of the most important concepts of CR is the Cognition Cycle (CC). The CC enables a host to achieve context-awareness and intelligence so that it is able to be aware of its operating environment in order to sense for the white spaces, and use them in an intelligent and efficient manner. It is comprised of six main states.

**Observe:** In the observe state, a CR host senses its operating environment.

**Orient:** The orient state determines the importance and priority of the sensing outcome, such as Normal, urgent, immediate. The immediate priority leads to an act state, urgent to the decide state, and normal to plan state.

**Plan:** The plan state draws up a longer-term course of actions.

**Decide**: The decide state determines the next action.

**Act**: In the act state, the chosen action is executed.

**Learn:** All of the actions are sent to the learn state.

Normal

Establish priority

**PLAN**

**ORIENT**

**OBSERVE**

**DECIDE**

Urgent

**ACT**

Immediate

Fig 2.2.1 Cognition cycle

Different applications depend on different topologies, [20]

1. Cluster-based topology: Few sensor nodes are elected as a cluster head for a particular group of sensor nodes. The CHs can be assigned different responsibilities such as spectrum sensing, and local bargaining of spectrum. A cluster-based topology is appropriate for effective operation dynamic spectrum management in CRWSN.
2. Hierarchical heterogenous Topology: Introduction of hierarchy into the network, whereby special nodes equipped high power source are capable of a longer transmission range. These nodes may be used as relay nodes. This gives rise to a heterogenous and hierarchical topology consisting of ordinary nodes, high-power relay nodes and the sink. The introduction of the heterogeneity brings about additional challenge in the face of the efficient dynamic spectrum access such as increased communication overhead and the deployment of sensor, special sensor needs be resolved in this topology.
3. Ad Hoc topology: An infrastructure-less topology where nodes communicate directly with the sink in a multi-hop, ad hoc fashion. Spectrum sensing is performed by each node individually or cooperatively in a distributed manner.

Challenges associated with CRWSN

1. Sleep-Wake Strategy:   
   In traditional WSN, the sensor nodes fall into sleep mode to minimize their energy consumption without jeopardizing the network connectivity. During the sleep mode, a node is not aware of any event happening around itself, hence it does not know what has happened in its operating environment upon waking up. Connectivity is vital in a CRWSN environment as it ensures that the CR information such as, operating channel, quiet period and notification on PU detection are transmitted or received whenever necessary.

Currently, there are two kinds of strategies that are used (i) Scheduled Rendezvous (SR), (ii)Asynchronous wakeup (AW). The SR method requires the sleeping nodes to wake up simultaneously periodically. The SR method ensures that all the nodes get the CR information. This method requires a strict synchronization as even a slight clock drift can cause loss in CR information. The AW method does not require synchronization and each node in the network maintains its own wake up schedule. This leads to the nodes waking up more frequently thereby consuming more energy than required.

1. Effects of quiet period: As the sensor nodes are not allowed to transmit packets during the quiet periods in CRWSNs, the end-to-end delay for a packet from a sensor node to the sink seems to increase compared to traditional wireless sensor networks. However, the end-to-end delay can be reduced by reducing the number of hops between the node and the sink using channels that provide high transmission ranges. The amount of time saved as a result of smaller number of hops must offset the delay caused due to the quiet period.
2. Spectrum sensing algorithms: There is a need to find a balance between energy consumption and sensing. The purpose of the spectrum sensing algorithms are to detect the PU signals within the channel detection time among the active sensor nodes that are communicating within their channel. To reduce energy consumption, a passive sensing approach senses the available channels only when there is data for transmission. In this case real time transmission is not possible. On the other hand, active sensing senses the available channels periodically regardless of its packet arrival. Hence supporting real time transmission. This approach is however not energy efficient.
3. Topology management: The topology of the network changes as various channel frequencies is applied for packet transmission. For the same transmission power, higher frequencies provide a shorter transmission range and thereby reducing the interference caused by the node to the other sensor nodes whereas lower frequencies provide a longer transmission range and reducing the number hops for the data to get to the sink which would invariably increase the interference cause by the node to the network. The reduced number of hops to reach the sink conserves energy at the transmitter sensor nodes. The energy incurred at each hop is given by the equation

Where Eh,i is the energy incurred at each hop and

ni is the number of single-hop neighbours of node i

Ttrans is the transmission duration

Extending the transmission range by choosing a lower frequency increases the value of the number of single hop neighbour’s which in turn increases the energy incurred thereby reducing the network lifetime. Since both short and long transmission range provide distinctive advantages, investigation into optimal methods to ensure network connectivity using different transmission frequencies is necessary.

A possible solution would be to construct a Minimum Spanning Tree (MST) that connects all nodes in an undirected graph with different maximum edge lengths at different channel. This involves the adjustment of both transmission power and operating frequency channel. Nodes can sleep to avoid incurring unnecessary reception energy consumption.

1. Node Deployment: There is the need for proper mathematical analysis for optimum node deployment for various topologies for the purpose of developing efficient and practical node deployment mechanisms.
2. Optimal Network Coverage: As a result of the primary user activity couple with node failure, the spatial location of sensor nodes may vary which would lead to false sensing.
3. Coordinated and Uncoordinated Operation: Operations such as spectrum sensing, spectrum detection, spectrum allocation, spectrum sharing, and spectrum handoff may be performed individually by sensor nodes or cooperatively among sensor nodes causing overhead

An reinforcement learning model for the dynamic channel selection is proposed. The system model has a state of the set of node i's neighbour nodes j and the action that can be taken are the available channels for data transmission and the reward is either a positive value or a negative value depending if the packets that were sent were successfully received or not. The successful transmission of the packet is dependent on many factors including the PUs channel utilization rate, the packet error rate in the channel. The packet transmission is said to be successful when a link-layer acknowledgment is received for the sent packet, else the transmission was unsuccessful.

* + 1. Q-Learning for Cognitive Radios

This shows how reinforcement learning can be used to solve problems such as spectrum sensing and channel allocation in a cognitive radio environment. Q learning is being used because of its low computational demands. The goal of this paper was to focus of the learning of the cognitive nodes rather than the frequency utilization of the secondary users.

The paper uses an off-policy RL algorithm called Q learning to simulate a channel decision making process. The fact that Q learning can converge without having any prior knowledge of the environment makes it an ideal choice for CR. In Q-Learning, an agent goes through a phase of learning before it can converge on an optimal solution for channel allocation. In the learning phase, the node makes decisions on what channels to select pseudo-randomly, the outcome of taking these actions will weigh strongly on what decisions are made later on. Once a node has finished learning, it can then make decisions on what it has learned. The ability for a node to be able to pre-empt whether a channel is going to be in use before accessing it allows it to optimise bandwidth usage for itself and any other nodes that may be accessing the same channel. The algorithm learns to function optimally by receiving a positive or negative feedback from the environment as soon as an action is taken and the current state is changed as show in Fig 2.2.1.

State(st): The set {St} defines the current state of the node. The state is a 2D vector representing the number of channels a particular node is currently transmitting (trt) and the number of channels the node is transmitting on which is either being used by another SU or a PU (tft). Ideally, the value of tft would be 0 no matter what action is being taken.

Action(at): The set {at} defines the available actions that a node can take. Each action changes the environment around the node which leads to either a positive or negative reward for each action.

The Q table keeps an account of all the possible state-action pairs and needs to be updated after each round. When the algorithm finishes training, the algorithm uses this Q table to find the optimal action for a particular state.

The algorithm can be described as a simple value iteration update as shown below:

Q() ← Q() + α()x[ + x Q( + 1, ) – Q()]

where α(st, at) is the learning rate where 0 < α < 1 and which represents to what extent newly acquired information will be taken into account. A learning rate of 1 will mean that only the most recent rewards will be taken into account whereas a learning rate of 0 will mean the agent will learn nothing, and any current reward will be discarded. The discount factor decides how important future rewards are for an agent.

To simulate spectrum sensing, an interference vector is scanned at each iteration. It is populated with 1 if the channel is in use and 0 if the channel is not in use.

There are 3 possible actions that can be taken

1. Do nothing
2. Acquire a channel
3. Drop a channel

Action 1 will neither acquire nor drop a channel, Action 2 will acquire a particular channel from the available channel and Action 3 will drop a current channel that is being used. The channel that is selected for drop or acquire is completely random in this implementation. There is an immediate reward or punishment received for taking an action depending on the action taken.

It is important to ensure that the agent does act in a greedy manner by acquiring as many channels as possible leaving other nodes in the network starved of bandwidth. The reward function used in this implementation ensures that the number of active channels are proportionally greater than the number of interfering channels.

r( = x ) – 1 ……()

This function ensures the channels are evenly distributed between all the nodes of the network.

As the electromagnetic spectrum environment that the agent is working in is very unpredictable, Q learning allows for random exploration before following a particular target policy. The simplest policy that can used is to select the state-action pair with the maximum Q value, or maximum reward given to a particular state-action pair. This would lead to a policy being generated that is extremely greedy and does not allow any exploration to the possible better solution. ε-Greedy is an example of a strategy that overcomes this issue of the algorithm working in a greedy fashion. This policy works in such a way that the best action is chosen 1-ε of the time and the algorithm is allowed to explore for ε of the times. The higher the value of ε, the higher the exploration of the algorithm. A similar strategy called the ε-decreasing strategy which allows the value of ε to eventually degrade whenever the number of explorations done by the algorithm is satisfactory. This algorithm will eventually converge and take the best possible action all the time when the ε value tends to 0.

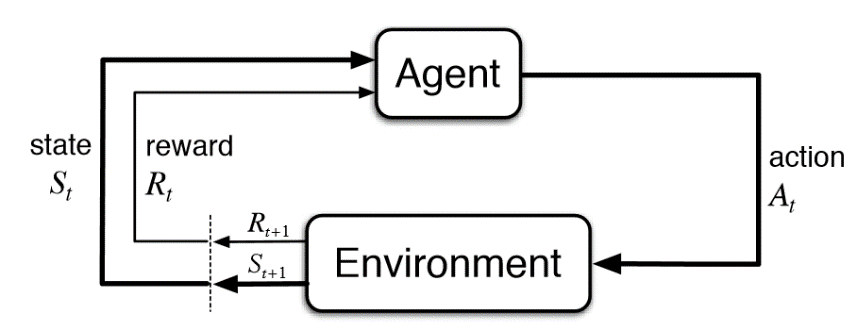


Fig.2.2.1 Block diagram for Q learning

1. PROJECT DESCRIPTION AND GOALS

The proposed algorithm is a hybrid of the glow-worm swarm optimization algorithm and the fruit fly optimization algorithm which aims at solving the poor localization problems, improving the convergence rate in the search space and simultaneously increasing the computational speed of the algorithm. It further enhances the lifetime of the network an minimizes the delay compared to algorithms such as GSO, FFOA, ABC, ALO, etc.

The Wireless sensor networks must first be created and then clustered before using the routing algorithms. We create a network of 50 nodes in an 100x100 area. The nodes are randomly allotted positions in the area and a link is estimated based on the proximity of the nodes i.e., the range of the node. Initially, the CH positions are allotted randomly as the energy of all the nodes are the same. A distance matrix between the CH and the nodes is calculated. A cost function is then used to calculate the cost of selecting these CHs and the CH positions are sorted based on their cost. The best set of CHs are then selected and the initial clustering is then plotted. The triangle is the BS, the rectangles are the CH nodes, the circles are the normal nodes and the blue circles are the Pus.

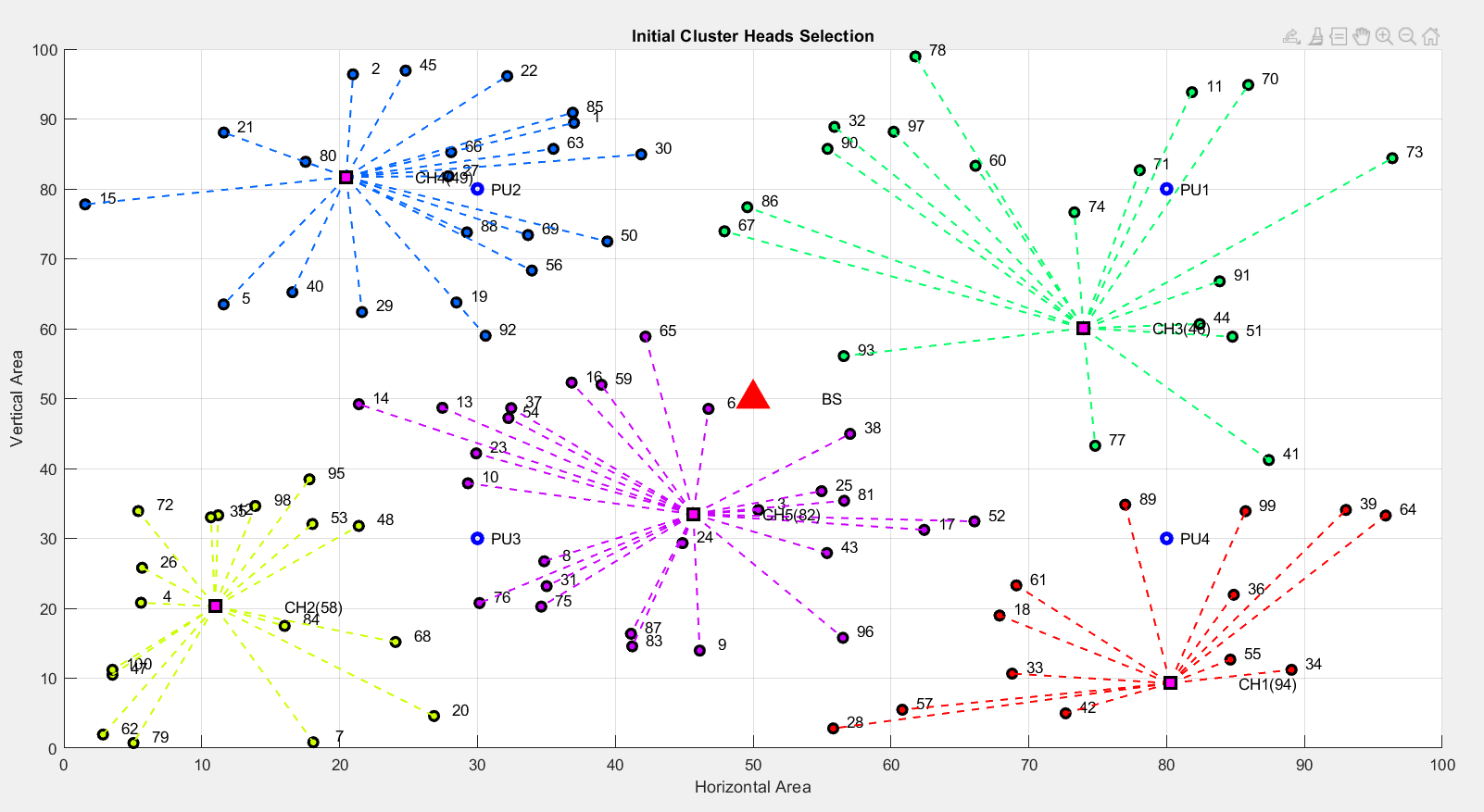


Fig 3.1: Initial Clustering and CH selection

The proposed FGF algorithm incorporates the FFOA algorithm into GSO. The algorithm starts by evaluating the fitness of each node. The fitness values are then sorted and the best five index values are chosen. If the index number is greater than five, FFOA update is done or else the GSO update it done. The flowchart for the developed FGF algorithm is given below.



Fig 3.2 Flowchart for GSO



Fig 3.3: Flowchart for FFOA



Fig:3.4 Flowchart for FGF

Along with the implementation of FGF, Dynamic channel allocation using Q learning is implemented. To simulate the spectrum sensing mechanism, the type, bandwidth, signal interference is calculated along with the channel idle time. These values are used in an objective function to calculate the channel characteristics (theta). The state and action sets are then defined. An interference vector is scanned at each iteration, if the network state changes, or the if PU occupies the channel, an action is taken and the reward for the particular action is updated into the Q table. The reward value for occupying a channel in use, switching a channel, normal communication is -1, -0.1, theta respectively. The epsilon greedy algorithm is used, the value for which is 0.8.i.e., an 80% probability of taking an action randomly and a 20% probability an action is taken depending on the Q table at the initial iteration.



Fig 3.4: Block diagram for Dynamic channel allocation using Q learning

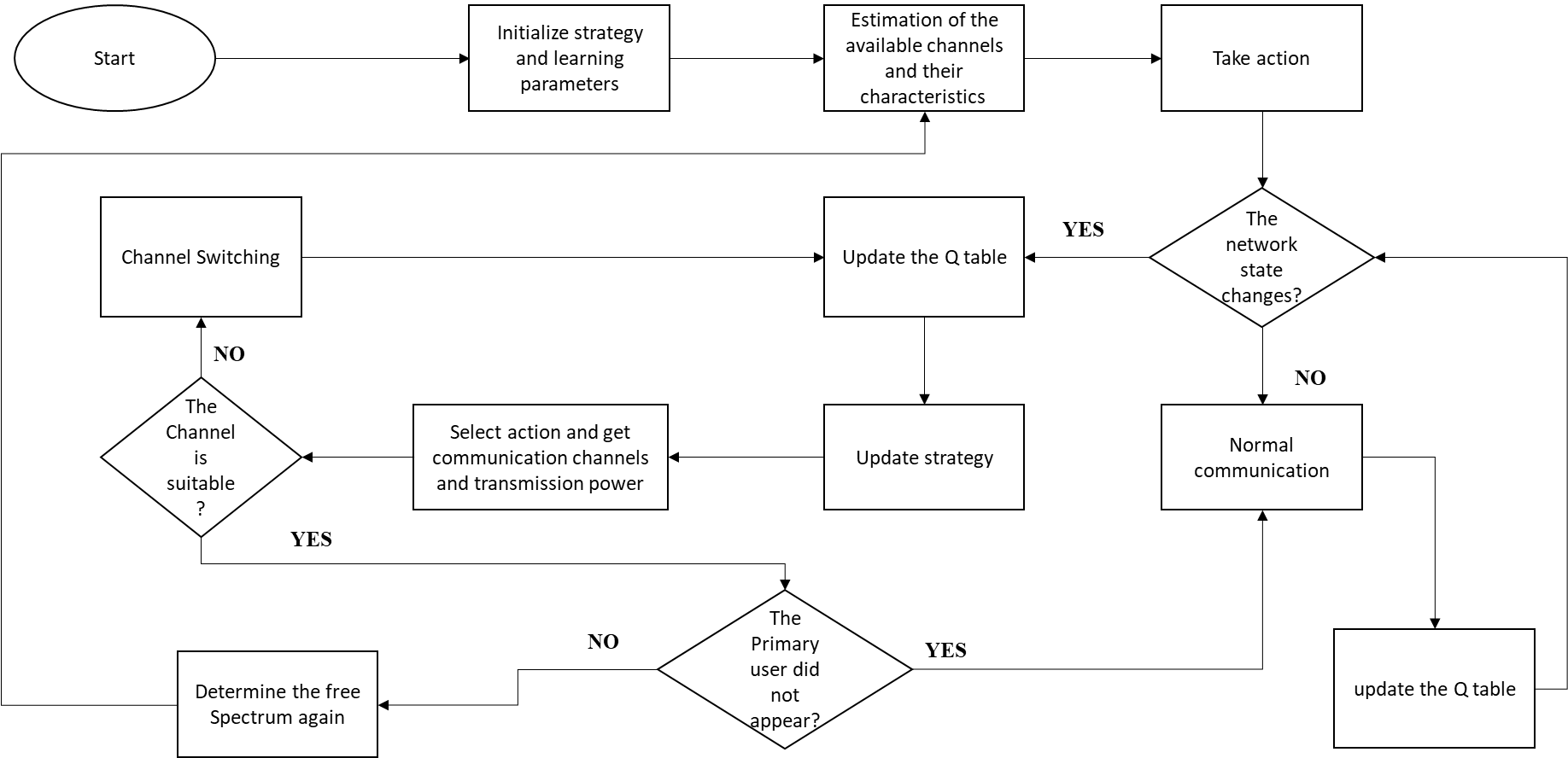


Fig 3.5 Flowchart of CRWSN using Q learning

The main goals of this project are given as follows

1. Increase the lifetime of the network by implementing a hybrid algorithm that utilizes the advantages of both the GSO algorithm as well as FFOA algorithm.
2. Opportunistically utilize the under-utilized whitespaces in the licensed spectrum as well as aid the nodes in accessing multiple channels to solve the problem of collision during packet transmission in a densely deployed sensor network.
3. TECHNICAL SPECIFICATIONS

The proposed algorithm is analysed in two different scenarios where 100 nodes are randomly deployed in a 100x100 network field and 200 nodes are deployed in a 200x200 network field. The simulation is run for a period of 1500 rounds. The base station is located at the centre of the field. There are 4 PU’s available in the field. The table below is used to represent the radio parameters used in the proposed algorithm simulation. The algorithm is compared with GSO [4][5], FFOA [2][3], ABC [6] and ALO [7].

Table 4.1 Energy initializations for the network

|  |  |
| --- | --- |
| PRAMETER | VALUE |
| Region Dimensions | 100 x 100  200 x 200 |
| Number of nodes | 100, 200, |
| Initial Energy (𝐸0) | 0.5 J |
| Energy consumed by radio electronics in transmit mode(𝐸𝑇𝑥) | 50 nJ/bit |
| Energy consumed by radio electronics in receiving mode(𝐸𝑅𝑥) | 50 nJ/bit |
| Energy consumed by the power amplifier on the free space model (𝐸𝑓𝑠) | 10 pJ/bit/ |
| Energy consumed by the power amplifier on the multi path model(𝐸𝑎𝑚𝑝) | 0.0013 pJ/bit/ |
| Energy consumed for data aggregation(𝐸𝐷𝐴) | 5 nJ/bit/signal |

1. DESIGN APPROACH AND DETAILS

**5.1 Design approach**

The proposed algorithm has been implemented with the help of the following specifications.

**Objective functions**

To select the optimal CH, the main parameters of WSN such as distance, delay and energy are utilized to perform the effective cluster head selection. The network performance enhances if it has high QoS which depends on parameters such as distance, delay and energy. The objective function to calculate the fitness of the algorithms are given by: -

OF = β x + (1 – β) ; 0 < β < 1 …. (5.1.1)

Where  is a constant which is set to 0.3.

= \* + \* + \* …. (5.1.2)

= …. (5.1.3)

The values of γ1, γ2, γ3 are taken to be 0.5,0.3,0.3. [6].

**Energy:** The energy utilized by the WSN is determined in Eq. 5.1.4. where EN(Di) and EN(CHi)indicates the energy of ith the normal node as well as the energy of jth the cluster head, respectively. fenergy(q), refers to the energy between the normal node as well as the CH and among the CH and the network’s BS, and fenergy(p) specifies the energy among two normal nodes.

=

…. (5.1.4)

Where,

(q) = …. (5.1.5)

= = …. (5.1.6)

(*p*) = …. (5.1.7)

Distance: The Eq. (5.1.8) defines the mathematical model of the distance parameter, where fdist(q) indicates the distance between the normal node as well as the CH and among the CH and the network’s BS, which is given in Eq. (5.1.9) and fdist(q) specifies the distance among two normal nodes, the value for which is in the range [0,1] as specified in [6].

…. (5.1.8)

…. (5.1.9)

.... (5.1.10)

**Delay:** The data transmission delay by nodes is specified in Eq. (5.1.11). in which the value of delay should be among in the range [0,1]. If the count of nodes in a cluster reduces, subsequently the delay also gets minimized substantially

.... (5.1.11)

**Conventional GSO**

The glow-worms are randomly distributed in the search space and each glow-worm expresses a luminescent quantity namely ‘luciferin’ with them. The strategy behind the interaction among glow-worms is as follows: the light intensity of glow-worms is directly proportional to associated luciferin, the glow-worms in the same neighbourhood interact with each other. Further, the luciferin intensity is greatly associated with the present fitness of location.

Generally, the GSO algorithm includes four stages: (i) Initialization (ii) Luciferin-update (iii) Movement (iv) Neighbourhood range Update. [5]

**Initialization:** The glow-worms are randomly distributed in the search space and are assigned identical luciferin intensities and similar decision domains.

**Luciferin-update:** The luciferin intensity of glow-worm is tremendously related to the location’s fitness. If the intensity value is greater, can attain the best position, and that is the best target value. Otherwise, the target is considered as poor. The glow-worm’s position gets varied as the iteration increases and the value of luciferin gets updated automatically.

The g glow-worm’s location at t time is Xg(t) and the associated objective function value at gth glow-worm location at t is J(Xg(t)).

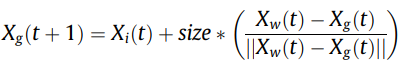
.... (5.1.12)

Where ν is the luciferin decay constant and η refers to the luciferin improvement constant.

**Movement:** In this phase, all the glow-worms choose their neighbour and follow it with a distinctive possibility. The neighbour glow-worm needs to satisfy two desires: The glow-worm is in the decision domain of the gth glow-worm and that the luciferin value should be greater than the luciferin value of gth glow-worm.

**** .... (5.1.13)

Once the movement of g glow-worm takes place, the location gets updated using Eq. 5.1.14 where size indicates the step size.

 .... (5.1.14)

**Neighbourhood range update:** After updating the location of the glow-worm, it follows the update of neighbourhood range. If the range of neighbourhood covers up only little glow-worm density, then the range of neighbourhood gets raised or else; the neighbourhood range gets minimized. λ indicates the constant parameter.

**** .... (5.1.14)

**Conventional FFOA**

The basis of FFOA models the behaviour of fruit flies at the time of food searching. The algorithm includes four steps. [2]

**Initialization phase:** In this phase, the fruit flies are randomly dispersed in the X-axis and Y-axis.

**Path construction phase:** This phase includes the determination of distance and smell concentration values for every fruit fly. SMiC indicates the smell concentration and is the reciprocal of the distance.

 .... (5.1.15)

 .... (5.1.16)

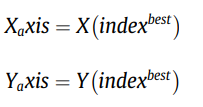
**Fitness evaluation phase:** The fitness formulation is defined

**** .... (5.1.17)

**** .... (5.1.18)

where smelli indicates the value of smell concentration of the distinctive fruit fly, smellbest and indexbest refers to the highest element and its respective indices with different dimensions of smell vectors, and max(smell) specifies the maximal smell concentration among fruit flies.

**Movement phase:** The fruit fly gives the best value of smell concentration and flies towards that location

 .... (5.1.19)

**Dynamic channel allocation**

Due to the characteristics of the spectrum holes and the behaviour of primary user is changing over time, the cognitive nodes need to adapt dynamically depending on the situation and at the same time ensure that the quality of communication is not compromised. Therefore, it is vital to seek an efficient spectrum decision making method. By analysing the network channel characterization, an adaptive spectrum decision framework and propose a joint channel selection and power control spectrum decision algorithm based on Q learning.

Channel characterization: In order to select an appropriate channel, the network nodes must describe the current characteristics of each channel and ensure the current status. The bandwidth, signal interference, false alarm rate of spectrum detection, and the idle time of band is considered. [10]

1. **Channel bandwidth** (Wd): In cognitive techniques nodes can detect the whole communicative spectrum and find the idle channel, but those channels may belong to different frequency bands. However, the channel division for different frequencies may be different, so the bandwidth is different and the channel capacity of idle channel is also different. The network node must consider the channel selection and power control according to different channel bandwidths.
2. **Signal interference** (Id,t): is the interference size of received signal at channel 𝑑 in time 𝑡, and it contains white noise interference and the interference of other nodes. Nodes can make detection according to the current channel interference, and Id,t shows that the greater the value means the worse the channel condition.
3. **Band last free time** (TdIdle): is the time interval between primary user appearances on band d. The network node will tend to select the channel of large idle time.
4. **Spectrum sensing of false alarm rat**e (𝑃d): Due to the fact that the primary user behaviour is unpredictable, the spectrum sensing cannot be ensured completely. Different communication frequency shows different characteristics for shadow and fading hence we assume a value for false alarm rate.

 .... (5.1.20)

ε1, ε2, ε3 are weighting factors in the range [0,1].

**Q learning**

Q learning is a Reinforcement learning algorithm that considers environmental feedback as the input and learns through constant interaction with the environment, then uses the feedback signal to find the optimal action which adapted to the current environment. The system consists of an environment that changes depending on the actions taken and each action gives respective feedback to the agent that uses it to learn from its mistakes. The feedback also called the reward value can either be a positive value or a dynamic value. As a model irrelevant learning algorithm, Q learning mainly cares about the evaluation value 𝑎t and selects the optimal state action according to 𝑄(𝑠, 𝑎). Usually, we consider 𝑄\*(𝑠, 𝑎) as the optimal evaluation value and 𝜋\*(𝑠, 𝑎) as the optimal strategies.

Assuming that the state set is 𝑠𝑡 and the action set is 𝑎𝑡, the evaluation value at the next time 𝑡+1 can be calculated by the formula as follows:

.. (5.1.21)

where 𝛾 is the discount factor, 𝛼t is the learning rate, 𝑟t+1 is the return value at next time, and 𝑄𝑖,𝑡(𝑠𝑡, 𝑎𝑡) is the value function of state action for node 𝑖; its means are the sum of the return value by executing action 𝑎t at state 𝑠t.

State: The network state st is a set of available bands, st=[Dt], where Dt=d1,d2,d3,…dk

Action: Each state has one or more associated actions. Any change of a network property is considered as an action.The action at=[dt,pt] where dt is the selected band and pt is the communication power. Each network node can switch to a better channel or adjust the transmission power of the current channel.

Reward: Rewards are assigned with the intention to reinforce specific state-action pairs and can be positive or negative. Due to the fact that network nodes can adjust the action immediately according to the received rewards, choosing and defining the rewards are challenging at times. In this algorithm, the reward function is given below.

**Collision with PU:** Due to the fact that the behavior of primary user is unpredictable and spectrum detection has a certain error, network node may conflict with the primary user when it selects a channel for communication. Then, we define the reward value as the lowest −0.5

**Channel switching:** When the interference of current communication channel increased or the primary user suddenly appeared, nodes require switching of the channel. But switching the channel frequently will lead to excessive energy consumption, Hence channel switching should be avoided and the reward value is set as −0.1.

**Normal communication:** When the channel does not switch and the communication happens normally in the same channel as before when the PU does not exist, we give a positive value to the reward function. For every normal communication, theta is given as the respective reward.

* 1. **Codes and Standards**

The pseudo code for the proposed algorithm is given below:

|  |
| --- |
| * + Initializing the parameters for WSN creation     - Number of nodes, BS, clusters     - Area     - Range     - Location of BS, PU’s     - Location of nodes     - Energy initializations     - Maximum number of iterations     - Alpha, beta, rmax, packet size, epsilon, decay rate, learning rate, discount factor, state set, action set, Q table.   + Creation of clusters and CH     - Run a loop for 1000 iterations and randomly select CHs     - Calculate the distance between the CHs     - Select the best position to be the initial CH     - Create a table for each cluster consisting details such as indices, position of nodes sensing nodes and CHs and CH indices.     - Calculate the distance between CH and BS.   + Calculation of the mathematical operations     - Calculate the delay matrix     - Initialize the initial energy for the nodes     - Calculate the distance evaluation using Eq- of the base paper     - Calculate the energy of the network using Eq – of the base paper     - Calculate the objective function and evaluate the fitness functions for the initial clusters.   + Spectrum Sensing     - For each PU, check if the PU is being used or not, find the signal interference value, assign the bandwidth.     - Calculate the channel idle time   + CH selection using FGF     - Initialization of GSO parameters: position, luciferin densities, decision range, population.     - Create functions for fitness function and run the functions to get the fitness     - Update the luciferin value according to te fitness function.     - Check if the best luciferin value has an index lesser than five       * GSO update: Movement Phase         + Iterate through each glowworm and find the distance between current position and that of all the glowworms in the cluster.         + Identify the indices of the glowworms that are within the decision range and have a higher LU value.         + Calculate the probability of movement using the selection formula         + Update the position of the glowworm after movement and update the decision range of the glowworms     - If the best luciferin value has an index greater than five       * FFOA update:         + Evaluate the distance and smell concentration of the nodes         + Fitness of the nodes are calculated along with the best smell, best index.         + Update the position of the glowworm after movement and update the decision range of the glowworms     - Q learning       * Each node obtains the network state information for the available bandwidths.       * Calculate the comprehensive value according to formula ()       * If the network state changes         + According to the current available channels, take an action         + Calculate the reward value and update the Q table.       * If the network state does not change         + Normal communication can take place, calculate the reward and update the Q table.       * If the PU appears back again, go to the action function.     - Calculate the energy dissipated, residual energy for each iteration and plot the graphs.     - Calculate the interference vector for each iteration and plot the graph. |

**FIRST ORDER RADIO MODEL**

In this model as depicted in Fig. , both the free space and multi-path fading channels are used depending on the distance between the transmitter and receiver. When this distance d is less than a threshold value d0, then the free space model with a power loss of d2 is used, otherwise, the multipath model with a power loss of d4 is used. In the algorithm, Eelec, Efs and Eamp is assumed to be the energy required by the electronics circuit, the amplifier in free space and multipath respectively. Then the energy required by the radio to transmit an k-bit message over a distance d is given as follows:

.... (5.2.1)

….. (5.2.2)

And for receiving a message, the radio consumes,

….. (5.2.3)

….. (5.2.4)

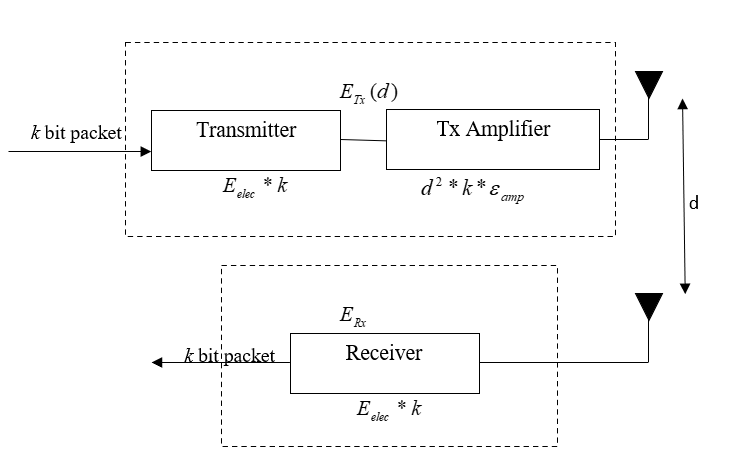
**

Fig 5.2.1 First Order radio model

The sensor nodes will always sense the environment at a fixed rate and thus always has data to be send to the end user. For the above parameter values, receiving a message is not a low-cost operation, the protocols should thus try to minimize the transmit distances between the cluster head and the sink and also within the cluster. The algorithm assumes that the energy required to transmit from a node to a CH is the same as the energy required for the transmission between the CH and the node.

The following assumptions were taken into consideration before the start of the project:

1. When the PU is communicating, its transmission power is really high and the transmission power of the SU is relatively small.
2. Different cognitive nodes can be in the same channel for communication but must adjust their power accordingly to avoid interference.
3. The channel rewards and the channel state transmission probabilities are unknown at the beginning and the nodes will have to therefore learn the channel probabilities and find the θ values.
   1. **Constrains and trade offs**

The proposed algorithm consists of certain constraints that can be implemented in the future to model the real world better.

1. The spectrum sensing algorithm used in this model is random, which impacts the value of channel idle time and signal interference which act as key parameters for the comprehensive value (θ)
2. This algorithm uses the epsilon greedy algorithm where all actions are chosen coequally which would lead the best and worst action being chosen with the same probability during the random search phase. However, a soft-max approach can be implemented in the future for better efficiency of the learning algorithm.
3. SCHEDULE, TASKS AND MILESTONES

The project had been carried out in a period of four months with some initial research work that took place for a month until early December 2022. Then followed the basic implementation of the WSN network followed by the implementation of the initial code for the FGF algorithm by mid-January. The first order radio model was implemented and the number of alive nodes were then plotted. The focus was then shifted into the implementation of cognitive radio using Q learning until mid-February. After this was done, a more comprehensive algorithm for Q learning was developed by mid-March. The code for the FGF algorithm was then updated and the final implementation of the base paper carried out.

Major milestones in this project were

1. Implementation of the initial clustering
2. Implementation of the proposed algorithm and the initial implementation of the first order radio model.
3. The implementation of the cognitive radio technology using Q learning.
4. PROJECT DEMONSTRATION

The proposed algorithm was compared with glowworm optimization algorithm, fruit fly optimization algorithm, Ant lion optimization algorithm and Artificial bee colony algorithm. The parameters used for comparison are alive nodes Vs number of rounds, normalized energy Vs time, throughput Vs Number of rounds and the distance analysis of the cluster head with respect to GSO and FFOA were done.

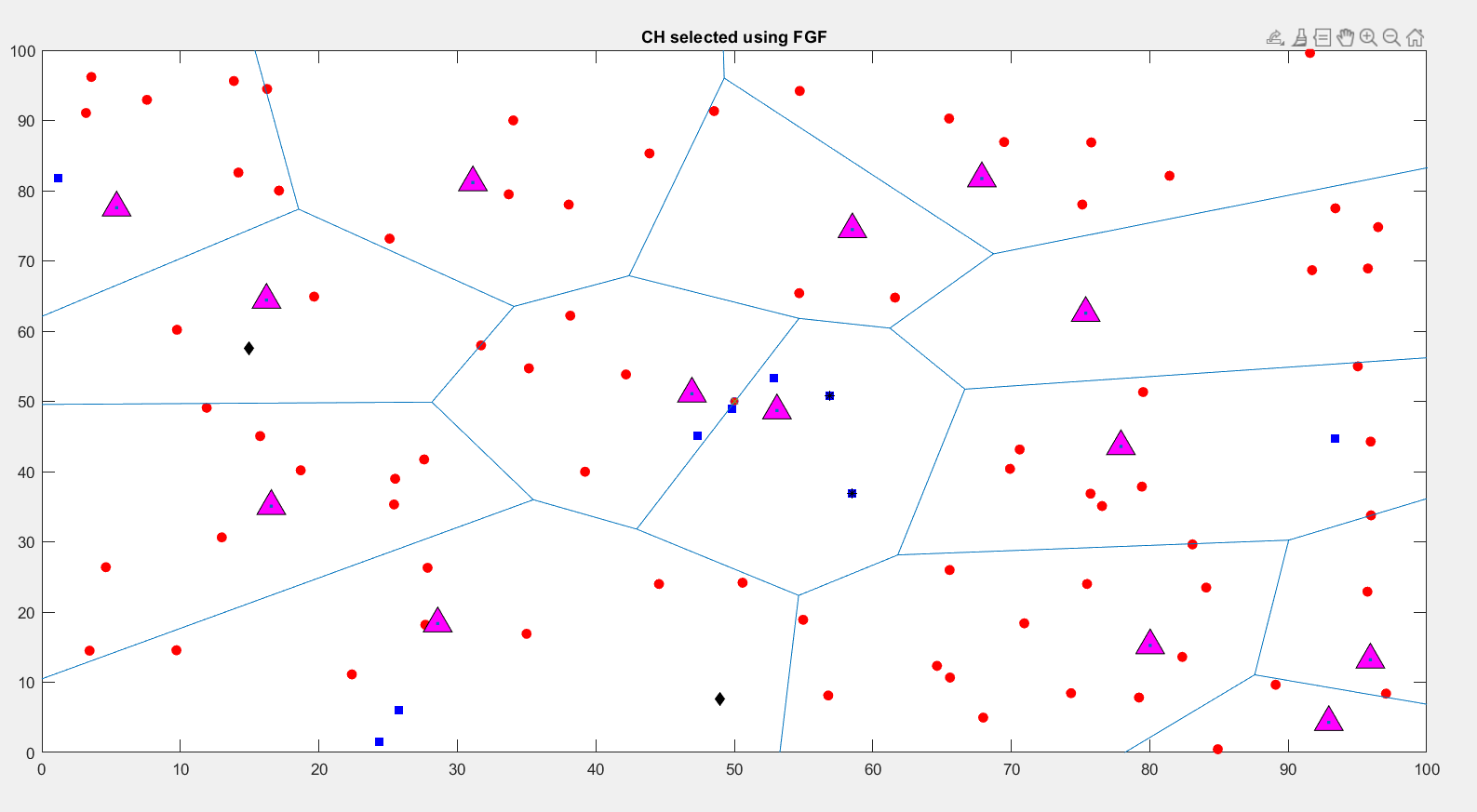


Fig 7.1 Network after 1500 rounds

* 1. **Number of alive nodes Vs Number of rounds**

The algorithm was simulated in the following scenarios

1. 100 nodes in an area of 100x100
2. 200 nodes in an area of 200x200

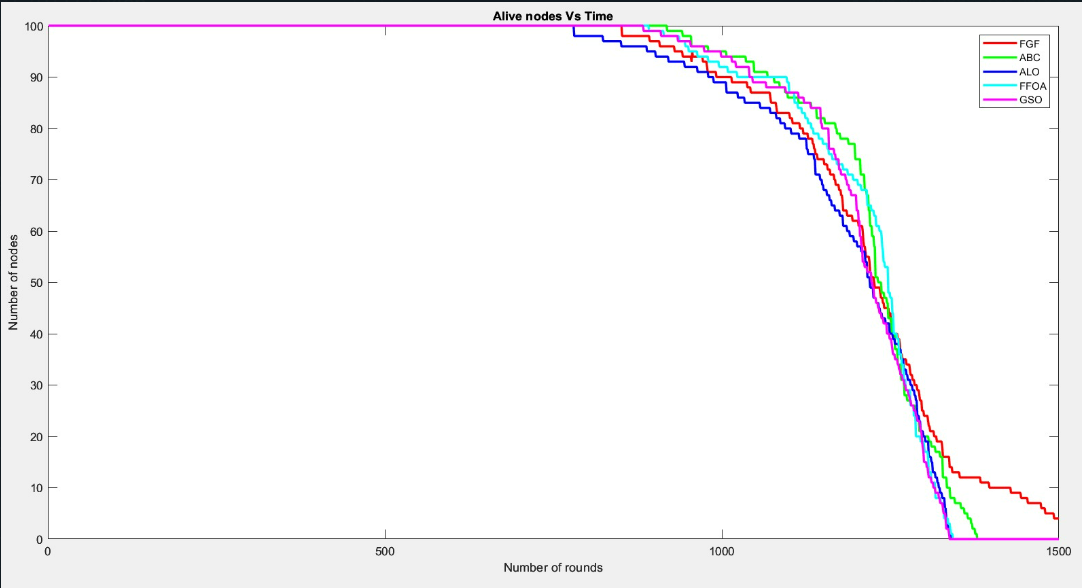


Fig 7.1.1 Analysis of alive nodes Vs Number of rounds for 100 nodes

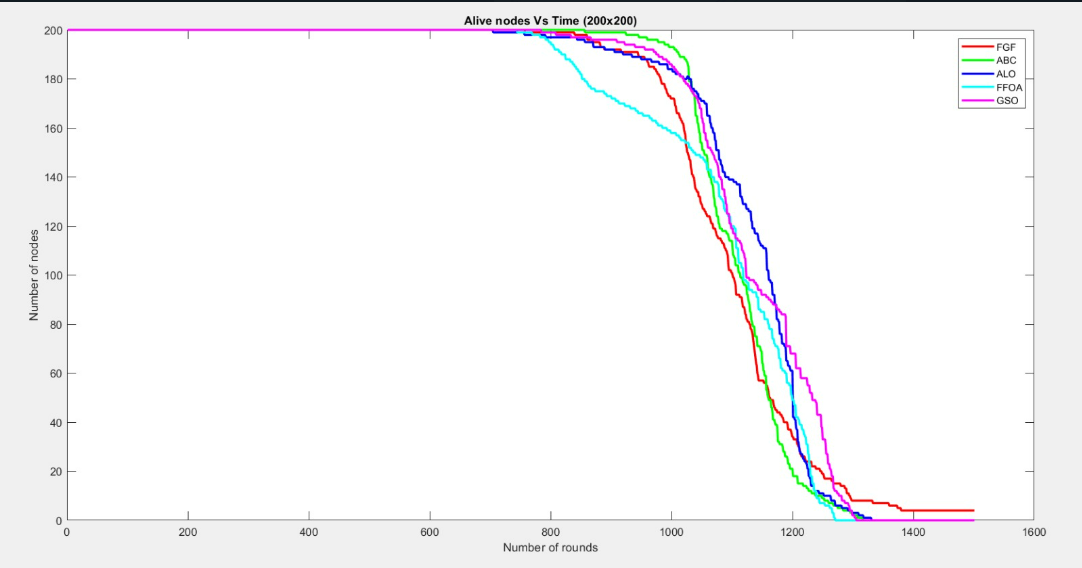


Fig 7.1.2 Analysis of alive nodes Vs Number of rounds for 200 nodes

The table below gives a brief analysis of the death of the nodes.

Table 7.1.1 Alive nodes analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | 100 X 100 with 100 nodes | | 200 X 200 with 200 nodes | |
|  | First node death | Last node death | First node death | Last node death |
|  |
| GSO | 885 | 1339 | 786 | 1307 |
| FFOA | 893 | 1344 | 746 | 1272 |
| ABC | 920 | 1380 | 858 | 1318 |
| ALO | 782 | 1342 | 706 | 1332 |
| FGF | 853 | 1587 | 759 | 1549 |

Fig 7.1.1, Fig 7.1.2 along with Table 7.1.1 shows the alive nodes analysis for the proposed and compared algorithms. At the 1300th round, 25%, 20%, 20%, 17%, 19% of the nodes are alive for FGF, ABC, ALO, GSO, FFOA respectively. The proposed algorithm has a better network lifetime compared to the other algorithms.

* 1. **Throughput of the Network**

The throughput of the algorithm has been analyzed for the following scenarios:

1. 100 nodes in a 100x100 m2 field
2. 200 nodes in a 200x200 m2 field

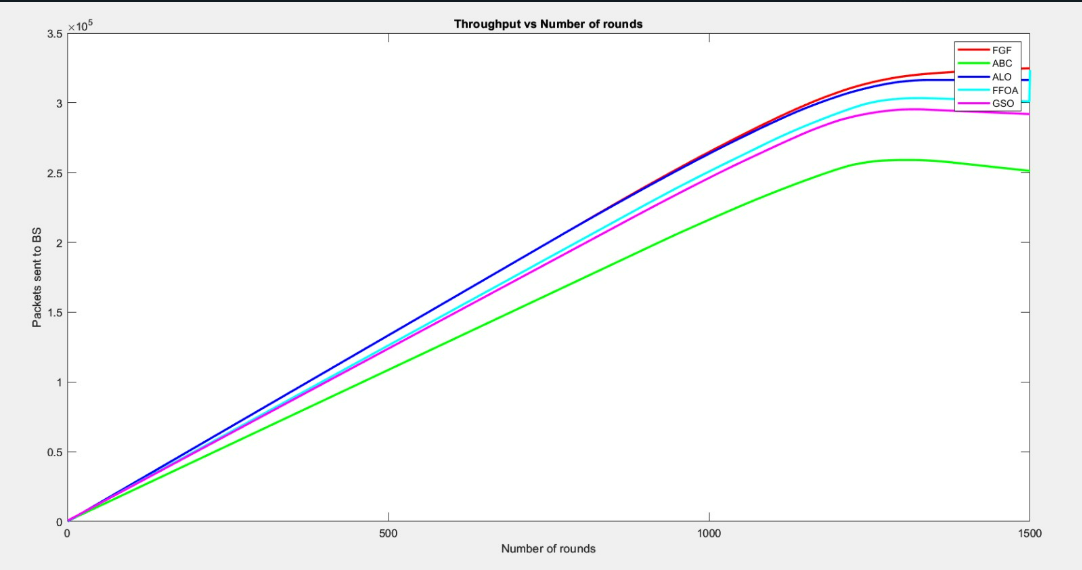


Fig 7.2.1 Analysis of throughput Vs Number of rounds for 100 nodes

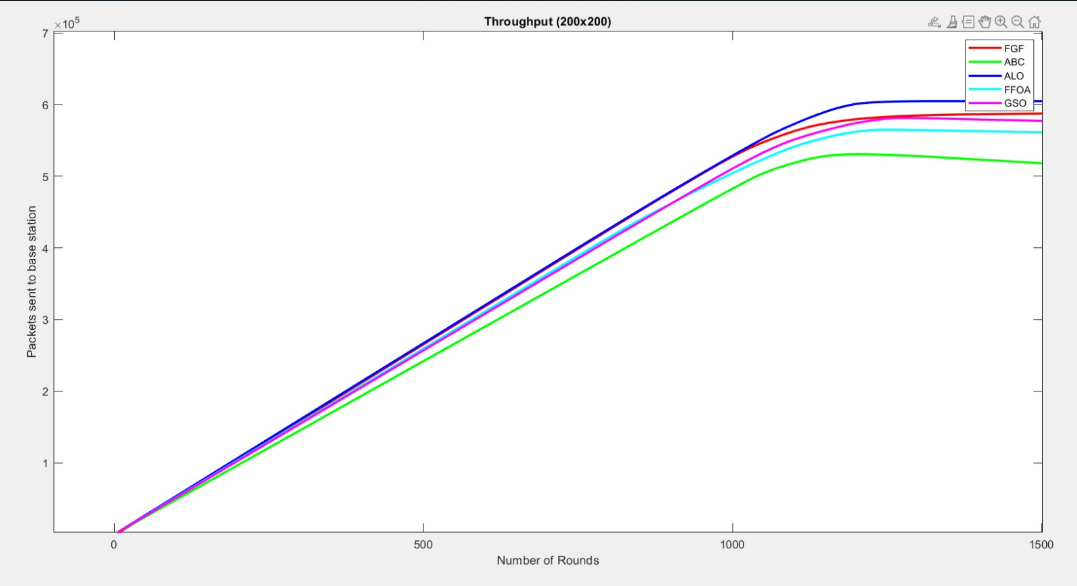


Fig 7.2.2 Analysis of throughput Vs Number of rounds for 200 nodes

Table 7.2.1 Throughput of the network

|  |  |  |
| --- | --- | --- |
| Algorithm | Throughput for 100 nodes | Throughput for 200 nodes |
|  |  |  |
|  |  |  |
| GSO | 291973 | 577188 |
|  |  |  |
| FFOA | 300949 | 561286 |
|  |  |  |
| ALO | 316330 | 604824 |
|  |  |  |
| ABC | 251231 | 518101 |
|  |  |  |
| FGF |  | 587546 |
|  | 324746 |  |

From Fig 7.2.1, Fig 7.2.2 and table 7.2.1, we can observer that the proposed algorithm gives 11.21% more throughput than GSO, 7.9% more throughput than FFOA, 2.66% more than ALO and 29.2% more than ABC algorithm in an area of 100x100 for 100 nodes

Whereas it is 1.79% more throughput than GSO, 4.6% more throughput than FFOA, 13.40% more throughput than ABC but the ALO algorithm gives the most throughput when we consider 200 nodes in an area of 200x200m2. Hence, we can say that this algorithm is most efficient for smaller number of nodes in a smaller area.

* 1. **Deviation in CH Distance**

At each iteration, the CH differs on the basis of energy as well as distance. The CH distribution varies with the difference in a number of rounds and difference in algorithms.

|  |  |
| --- | --- |
|  |  |

Fig 7.3.3 Distance analysis of CH with respect to GSO and FFOA

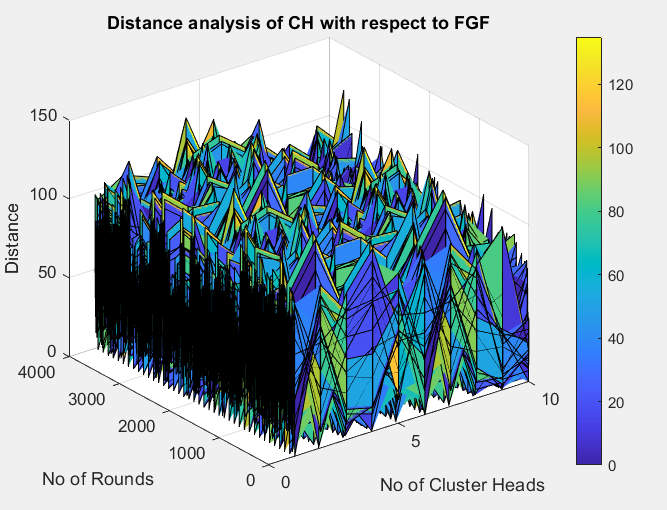


Fig 7.3.4 Distance analysis of CH with respect to FGF

Table 7.3.1 Statistics related to distance

|  |  |  |  |
| --- | --- | --- | --- |
| Statistics | FFOA | GSO | FGF |
| Best | 117.6 | 204.8 | 126.47 |
| Worst | 4532.7 | 4587.3 | 4453 |
| Mean | 1667.2 | 1688.8 | 1728.9 |

* 1. **Interference Vector Analysis**

In Q learning, the model learns by making mistakes, here we have taken the number of channels that SU is interfering Vs Number of rounds. The analysis was done by taking two values of epsilon

1. ε=0.8
2. ε=0.4

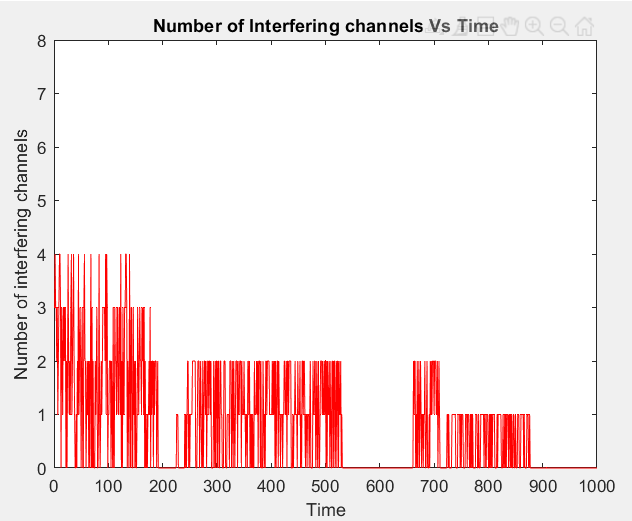


Fig 7.4.1 Number of interfering channels Vs Time (ε=0.8)

None of the channels are being interfered on 54.5% of the time, one channel is being interfered 22.6% of times, two channels are being interfered on 16.6%, three channels are being interfered on 37% and four channels are being interfered on 8% of times.

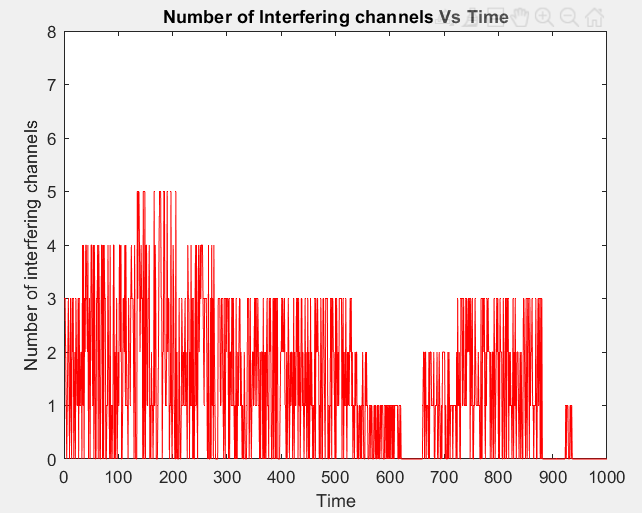


Fig 7.4.2 Number of interfering channels Vs Time (ε=0.4)

None of the channels are being interfered on 38% of the time, whereas one channel is being interfered on 22% of the time, two channels are being interfered on 16% of the time, and three channels are being interfered on 7% of times, four channels are being interfered 11% of the time and five channels are being interfered on for 6%.

When the strategy is not being allowed to explore more, the number of interfering channels increase. When the ε value was 0.8, 545 channels that were selected for communication had no presence of PU whereas when the ε value was 0.4, 381 channels were selected for communication that had no presence of PU.

1. RESULTS AND DISCUSSION

Wireless sensor networks have applications in almost all the fields of today and there is a need to constantly make sure that the energy is being utilized efficiently and finding a balance in utilizing energy for the opportunistic use of the licensed spectrum. In this project, a CH selection algorithm was proposed along with the implementation of a dynamic channel allocation using Q learning and the following were observed.

While considering the alive nodes analysis, at the 1300th round, 25%, 20%, 20%, 17%, 19% of the nodes are alive for FGF, ABC, ALO, GSO, FFOA respectively. The proposed algorithm has a better network lifetime compared to the other algorithms for 100 nodes and while considering the throughput, the proposed algorithm gives 11.21% more throughput than GSO, 7.9% more throughput than FFOA, 2.66% more than ALO and 29.2% more than ABC algorithm in an area of 100x100 for 100 nodes

When we look into the dynamic channel allocation, we consider two different values of the exploration-exploitation model. When the strategy is not being allowed to explore more, the number of interfering channels increase. When the ε value was 0.8, 545 channels that were selected for communication had no presence of PU whereas when the ε value was 0.4, 381 channels were selected for communication that had no presence of PU for 1000 rounds.

1. REFERENCES

**Base paper:**

[1] Dattatraya, K. N., & Rao, K. R. (2022). Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN. Journal of King Saud University-Computer and Information Sciences, 34(3), 716-726.

[2] Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems, 26, 69-74.

[3] Dey, A., Sarkar, T., & Ali, S. (2016, December). Fruit Fly algorithm based clustering protocol in wireless sensor networks. In 2016 9th international conference on electrical and computer engineering (ICECE) (pp. 295-298). IEEE.

[4] Aljarah, I., & Ludwig, S. A. (2013, June). A new clustering approach based on glowworm swarm optimization. In 2013 IEEE congress on evolutionary computation (pp. 2642-2649). IEEE.

[5] Kaipa, K. N., & Ghose, D. (2017). Glowworm swarm optimization: theory, algorithms, and applications (Vol. 698). Springer.

[6] Kumar, R., & Kumar, D. (2016). Multi-objective fractional artificial bee colony algorithm to energy aware routing protocol in wireless sensor network. Wireless Networks, 22, 1461-1474.

[7] Chopra, N., Mehta, S., 2015. Multi-objective optimum generation scheduling using Ant Lion Optimization, 2015 Annual IEEE India Conference.

[8] Li, M., Xu, Y., & Hu, J. (2009, November). A Q-learning based sensing task selection scheme for cognitive radio networks. In 2009 International Conference on Wireless Communications & Signal Processing (pp. 1-5). IEEE.

[9] Hosey, N., Bergin, S., Macaluso, I., & O'Donoghue, D. (2009). Q-learning for cognitive radios. In Proceedings of the China-Ireland Information and Communications Technology Conference (CIICT 2009). ISBN 9780901519672. National University of Ireland Maynooth.

[10] He, J., Peng, J., Jiang, F., Qin, G., & Liu, W. (2015). A distributed Q learning spectrum decision scheme for cognitive radio sensor network. International Journal of Distributed Sensor Networks, 11(5), 301317.

[11] Bakshi, M., Chowdhury, C., & Maulik, U. (2021). Energy-efficient cluster head selection algorithm for IoT using modified glow-worm swarm optimization. The Journal of Supercomputing, 77, 6457-6475.

[12] Sakthy, S. S., & Bose, S. (2019). Optimising residual energy transmission head with SNR value in multiple clusters. Wireless Personal Communications, 108, 107-120.

[13] El-Mougy, A., & Ibnkahla, M. (2014). A cognitive framework for WSN based on weighted cognitive maps and Q-learning. Ad hoc networks, 16, 46-69.

[14] Poluru, R. K., & Kumar R, L. (2021). An Improved Fruit Fly Optimization (IFFOA) based Cluster Head Selection Algorithm for Internet of Things. International Journal of Computers and Applications, 43(7), 623-631.

[15] Bongale, A. M., Nirmala, C. R., & Bongale, A. M. (2019). Hybrid cluster head election for WSN based on firefly and harmony search algorithms. Wireless Personal Communications, 106, 275-306.

[16] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications, 1(4), 660-670.

[17] Najimi, M., Ebrahimzadeh, A., Andargoli, S. M. H., & Fallahi, A. (2014). Lifetime maximization in cognitive sensor networks based on the node selection. IEEE sensors Journal, 14(7), 2376-2383.

[18] Malanchini, I., Cesana, M., & Gatti, N. (2009, November). On spectrum selection games in cognitive radio networks. In GLOBECOM 2009-2009 IEEE Global Telecommunications Conference (pp. 1-7). IEEE.

[19] RejinaParvin, J., & Vasanthanayaki, C. (2015). Particle swarm optimization-based clustering by preventing residual nodes in wireless sensor networks. IEEE sensors journal, 15(8), 4264-4274.

[20] Yau, K. L. A., Komisarczuk, P., & Teal, P. D. (2009, October). Cognitive radio-based wireless sensor networks: Conceptual design and open issues. In 2009 IEEE 34th Conference on Local Computer Networks (pp. 955-962). IEEE.

[21] Siddavaatam, P., Sedaghat, R., & Sharma, A. T. (2017, December). Intel-leach: an optimal framework for node selection using dynamic clustering for wireless sensor networks. In 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 359-364). IEEE.

[22] Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, 8(1), 687-697.

[23] Haykin, S. (2005). Cognitive radio: brain-empowered wireless communications. IEEE journal on selected areas in communications, 23(2), 201-220

[24] Pan, W. T. (2012). A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowledge-Based Systems, 26, 69-74.

[25] G. P. Joshi, S. Y. Nam, and S. W. Kim, “Cognitive radio wireless

sensor networks: applications, challenges and research trends,”

Sensors, vol. 13, no. 9, pp. 11196–11228, 2013.

[26] Malanchini, I., Cesana, M., & Gatti, N. (2009, November). On spectrum selection games in cognitive radio networks. In GLOBECOM 2009-2009 IEEE Global Telecommunications Conference (pp. 1-7). IEEE.