COMPARITIVE STUDY ON FLOWER POLLINATION ALGORITHM AND OPTIMAL LINEAR WEIGHTED ALGORITHM IN COGNITIVE RADIO NETWORK

A PROJECT REPORT

Submitted by

ROSHNI.R	**311520106068**
SNEHASHREE.G	**311520106080**
YUVASHREE.B	**311520106105**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2024

COMPARITIVE STUDY ON FLOWER POLLINATION ALGORITHM AND OPTIMAL LINEAR WEIGHTED ALGORITHM IN COGNITIVE RADIO NETWORK

A PROJECT REPORT

Submitted by

ROSHNI.R	**311520106068**
SNEHASHREE.G	**311520106080**
YUVASHREE.B	**311520106105**

In partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE

ANNA UNIVERSITY::CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"COMPARITIVE STUDY ON FLOWER POLLINATION ALGORITHM AND OPTIMAL LINEAR WEIGHTED ALGORITHM IN COGNITIVE RADIO NETWORK"** is the bonafide work of **"ROSHNI.R (311520106068), SNEHASHREE.G (311520106080), YUVASHREE.B (311520106105)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Ms. SIJI SIVANANDAN,	Mrs. R. LAKSHMI,
M. Tech, MIETE	M.E,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
	ASSISTANT PROFESSOR
Electronics & Communication	Electronics & Communication
Engineering Department	Engineering Department
Meenakshi Sundararajan	Meenakshi Sundararajan
Engineering College,	Engineering College,
Kodambakkam,	Kodambakkam,
Chennai- 600 024.	Chennai- 600 024.

ANNA UNIVERSITY::CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"COMPARITIVE STUDY ON FLOWER POLLINATION ALGORITHM AND OPTIMAL LINEAR WEIGHTED ALGORITHM IN COGNITIVE RADIO NETWORK"** is the bonafide work of **"ROSHNI.R (311520106068), SNEHASHREE.G (311520106080), YUVASHREE.B (311520106105)"** who carried out the project work under my supervision.

Submitted for project viva voce of **BACHELOR OF ENGINEERING** in **ELECTRONICS AND COMMUNICATION ENGINEERING** held at **MEENAKSHI SUNDARARAJAN ENGINEERING COLLEGE** on

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

A great deal of time and effort has been spent in completing our project. We would like to extend our heartfelt thanks to all the special people who have guided us and have contributed significantly to our efforts. We take this opportunity to thank our Founder **Prof. K.R.Sundararajan**, our honourable Correspondent **Dr.K.S.Lakshmi M.Sc., M.Phil., Ph.D.**, our revered Secretary **Mr. N. Sreekanth B.Sc., B.E, M.S, Ph.D.,FIE**, our respected Principal **Dr. S. V. Saravanan BE, ME, Ph.D, MBA, D.Litt (H)** and our Dean of Academics **Dr.K.Umarani M.E, Ph.D.**, for being the foundational pillars of support and for providing us with excellent opportunities.

We feel elated in manifesting a deep sense of gratitude and thankfulness to our beloved Head of the Department **Ms. Siji Sivanandan M.Tech, MIETE** for everything she has done to us and for being a source of motivation.

We found our profound privilege to express our heartiest indebtedness towards **Dr.A.Babiyola M.E, Ph.D., MDST, MIEEE, MIETE, MISTE**, Professor and our project coordinators **Mr. S. Balasubramanian M.E, [Ph.D].,** Assistant Professor and **Mrs.N.Meenakshi M.E. [Ph.D].,** Associate Professor. We take immense pleasure in thanking our internal guide and project coordinator **Mrs. R. Lakshmi M.E,** Assistant Professor. We take this opportunity to thank all faculty members and supporting staff of our department for constantly motivating us to do our project with full interest and enthusiasm. Our library needs a special mention, and we extend our heartfelt thanks to our librarian for supporting us. We would like to thank our Parents for being the moral backbone and encouraging us to move forward. Last but perhaps, we would like to thank the Almighty God for showering us with his blessings all through the way.

ABSTRACT

In recent years, Cognitive Radio (CR) networks have emerged as a promising solution to efficiently utilize the limited spectrum resources by allowing secondary users to opportunistically access unused spectrum bands. In this study, we present a comparative analysis of two optimization algorithms, namely the Flower Pollination Algorithm (FPA) and an Optimized Linear Weighted Algorithm (OLWA), within the context of a CR network for wireless communication systems. The FPA is inspired by the pollination process of flowering plants and is known for its global exploration capabilities, while the OLWA incorporates a linear weighted approach to optimize spectrum allocation. The objective of this research is to evaluate the performance of these algorithms in terms of spectrum utilization efficiency, overall system throughput. We implement both algorithms in a simulated CR network environment and conduct extensive experiments to analyze their performance under various scenarios based on the number of users. Our results demonstrate the effectiveness of both algorithms in optimizing spectrum allocation and improving system performance in CR networks. Furthermore, we present graphical outputs depicting the relationship between Signal-to-Noise Ratio (SNR) and error probability, showcasing a decreasing curve and as the number of user increases there is a decrease in the error probability. This highlights the impact of spectrum sharing and user density on system reliability and performance. These findings provide valuable insights into the selection of optimization algorithms for CR networks, enabling network designers to make informed decisions to enhance spectrum utilization and overall system efficiency based on their implications of the algorithms depending on the applications of the developed Cognitive Radio Network (CRN).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
WC	Wireless Communication
PU	Primary User
SU	Secondary User
CRN	Cognitive Radio Network
RF	Radio Frequency
SDR	Software Defined Radio
CSS	Cooperative Spread Spectrum
FPA	Flower Pollination Algorithm
OLWA	Optimal Linear Weighted Algorithm
MATLAB	Matrix Laboratory
QoS	Quality of Service
FCC	Fusion center coefficient
BER	Bit Error Rate
SDF	Software Defined Fusion
SNR	Signal to Noise Ratio
FC	Fusion Center
AWGN	Additive White Gaussian Noise
CH	Cluster Head
TCP/IP	Transmission Control Protocol/ Internet Protocol
Db	Decibel
IIoT	Industrial Internet of Things

CHAPTER - I

INTRODUCTION

1.1 WIRELESS COMMUNICATION

1.1.1 WHAT IS WIRELESS COMMUNICATION?

Wireless communication involves the transmission of information over a distance without the help of wires, cables or any other forms of electrical conductors. In place of a physical connection, data travels through electromagnetic signals broadcast from sending facilities to intermediate and end-user devices.

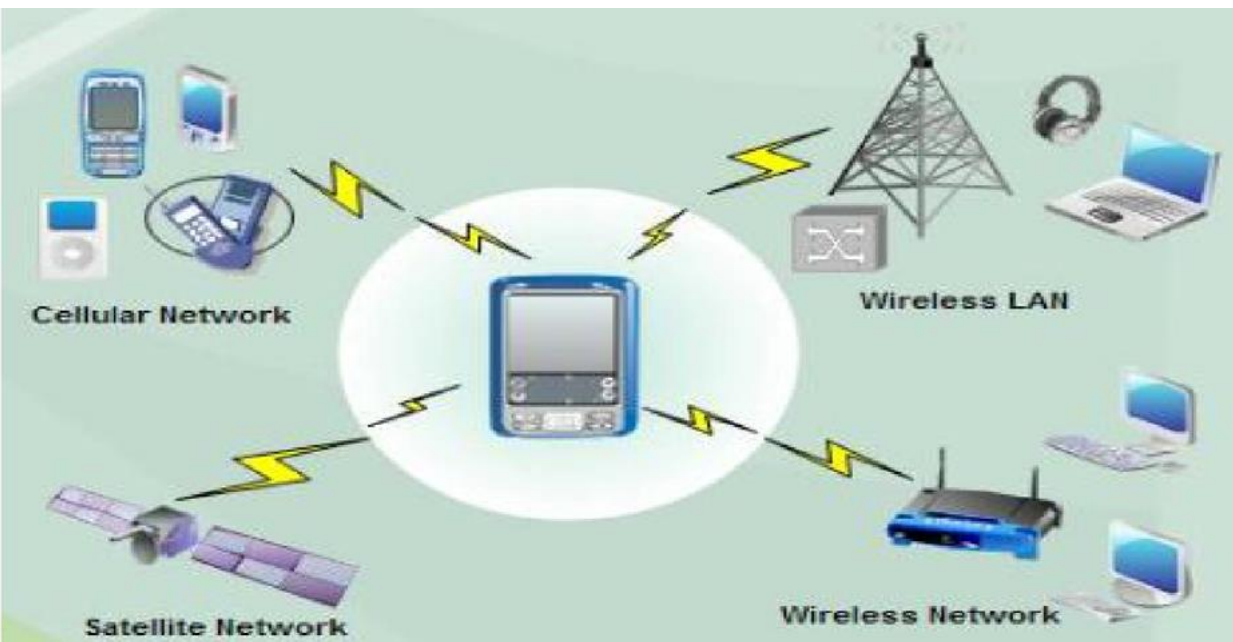


Fig.1 various sectors in wireless communication

As a medium, wireless communications has been around for more than a century. But it's only been in the past 15 years particularly after the ratification of the 802.11 ac and 4G standards that the technology evolved enough to permit the development of applications and services comprehensive enough for widespread enterprise and consumer adoption.

1.1.2 WHERE IT CAN BE USED IN SPECTRUM SENSING?

Wireless communication plays a crucial role in spectrum sensing, especially in cognitive radio networks where spectrum sensing is employed to detect and utilize available spectrum bands efficiently.

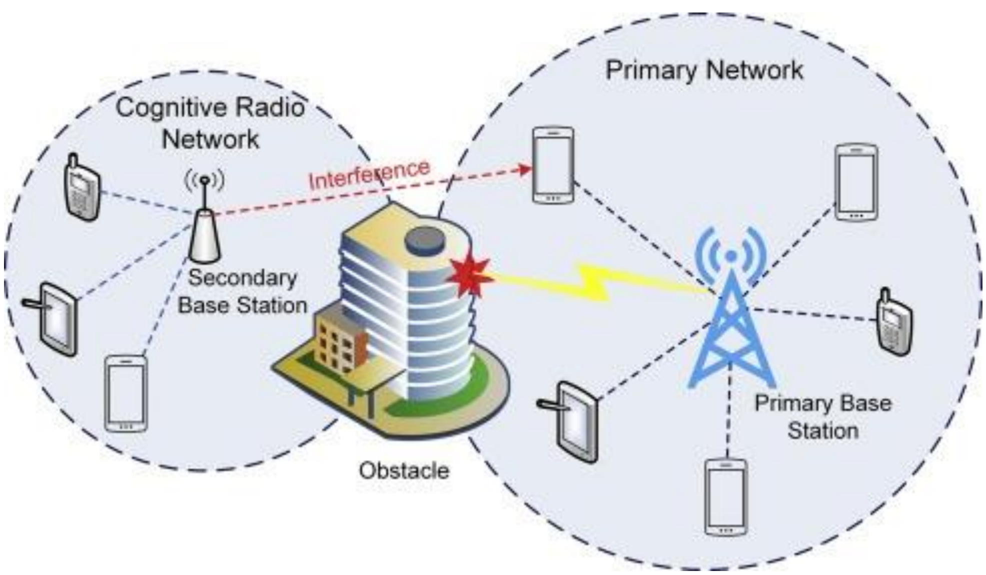


Fig.2 Sensing a PU from primary network for a CRN

1. **Primary User Detection:** Spectrum sensing involves detecting the presence of primary users (licensed users) in a particular frequency band. Wireless communication is utilized to sense and monitor the activity of primary users,

allowing secondary users (unlicensed users) to opportunistically access the spectrum when it's not in use by primary users.

2. **Channel Occupancy Sensing:** Wireless communication is used to sense the occupancy of specific frequency channels. By analyzing the wireless signals in those channels, cognitive radio devices can determine whether they are occupied by primary users or if they are available for secondary use.

3. **Dynamic Spectrum Access (DSA):** DSA enables secondary users to dynamically access spectrum bands that are not being used by primary users. Wireless communication is essential for secondary users to communicate with each other and with the cognitive radio infrastructure to coordinate spectrum access based on spectrum sensing results.

4. **Spectrum Handoff:** When primary users are detected in a spectrum band that secondary users are currently using, spectrum handoff procedures are initiated to vacate the occupied band and switch to an available one. Wireless communication facilitates the handoff process by enabling communication between secondary users and the cognitive radio network for coordination and control.

5. **Collaborative Sensing:** In collaborative spectrum sensing, multiple cognitive radio devices cooperate to improve the accuracy and reliability of spectrum sensing results. Wireless communication enables these devices to exchange sensing information and coordinate their sensing activities, enhancing spectrum awareness across the network.

1.1.3 FLOW OF WIRELESS NETWORK

1. **Data Generation:** The process begins with the generation of data by an application or device. This data could be generated by various sources such as sensors, user input, or software processes.

2. **Data Processing:** Before transmission, the data may undergo processing, such as encryption, compression, or formatting, depending on the requirements of the application and network.

3. **Data Transmission:** Once processed, the data is transmitted over the wireless medium. This transmission occurs in the following steps:

- **Data Segmentation:** Large data packets may be divided into smaller segments for transmission.

- **Frame Creation:** Each segment is encapsulated into frames with necessary header information added for routing and error detection.

- **Channel Access:** The sender device accesses the wireless channel using a protocol such as CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) or TDMA (Time Division Multiple Access) to avoid collisions with other transmissions.

- **Modulation and Transmission:** The digital data is modulated into radio waves suitable for wireless transmission and sent over the air.

4. **Wireless Medium:** The wireless medium carries the transmitted data through the air to the recipient device. This medium can be susceptible to interference, noise, and attenuation, affecting the quality and reliability of the transmission.

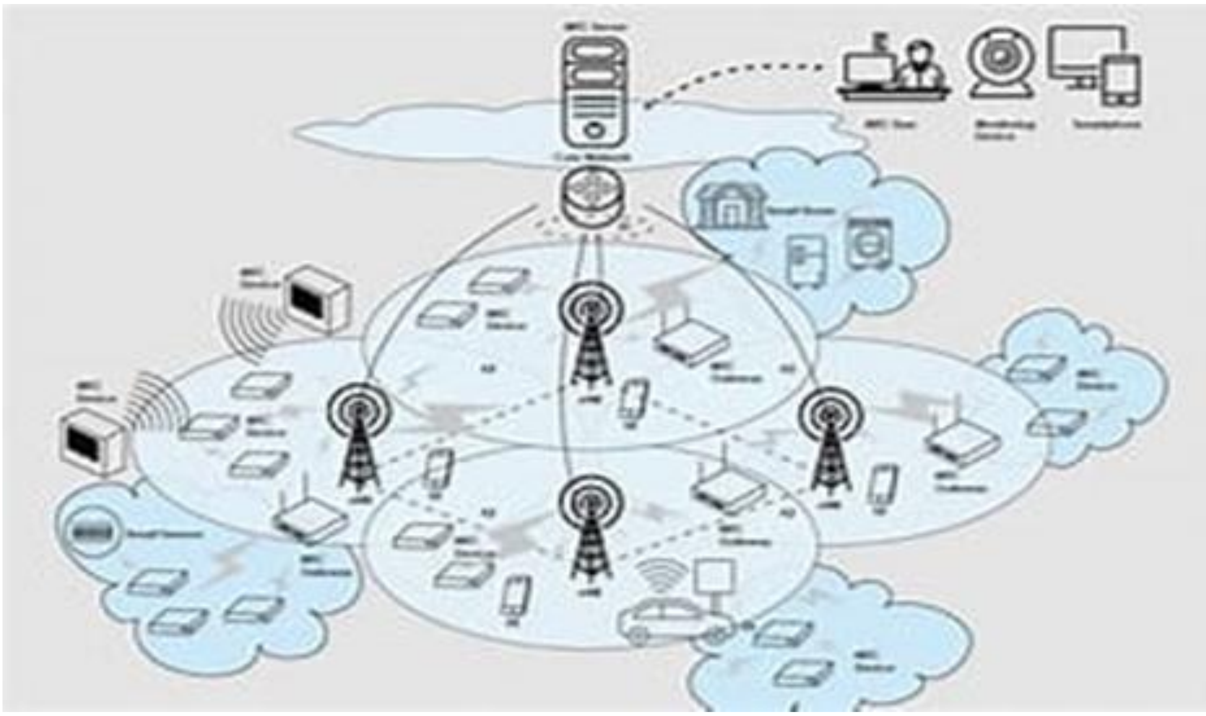


Fig.3 Flow of Wireless Network

5. **Data Reception:** The recipient device receives the transmitted data over the wireless medium. This reception process involves:

- **Signal Detection:** The recipient device detects and demodulates the received radio signals back into digital data.

- **Frame Decoding:** The received frames are decoded to extract the original data segments.

- **Error Checking:** Error detection mechanisms such as CRC (Cyclic Redundancy Check) are used to verify the integrity of the received data.

6. **Data Processing:** Upon reception, the received data may undergo further processing, such as decryption, decompression, or reassembly of segmented packets.

7. **Data Consumption:** Finally, the recipient device delivers the processed data to the intended application or user for consumption or further processing.

1.1.4 FUTURE SCOPE OF WIRELESS COMMUNICATION

Future wireless networks will support 100 GB per second communication between people, devices, and the “Internet of Things,” with high reliability and uniform coverage indoors and out. The shortage of spectrum to support such systems will be alleviated by advances in massive MIMO and MMW technology as well as cognitive radios.

Wireless technology will also enable smart and energy-efficient homes and buildings, automated highways and skyways, and in-body networks for monitoring, analysis, and treatment of medical conditions. Breakthrough energy-efficiency architectures, algorithms, and hardware will allow wireless networks to be powered by tiny batteries, energy-harvesting, or over-the-air power transfer. Finally, new communication systems based on biology and chemistry to encode bits will enable a wide range of new micro and macroscale applications.

1.2 COGNITIVE RADIO

1.2.1 WHAT IS COGNITIVE RADIO?

Cognitive radio is a special kind of smart radio transceiver hardware that automatically detects all available wireless channels on a spectrum, facilitating changes to its reception or transmission parameters, which allows the concurrent transmission of multiple additional wireless communications in a location's given spectrum.

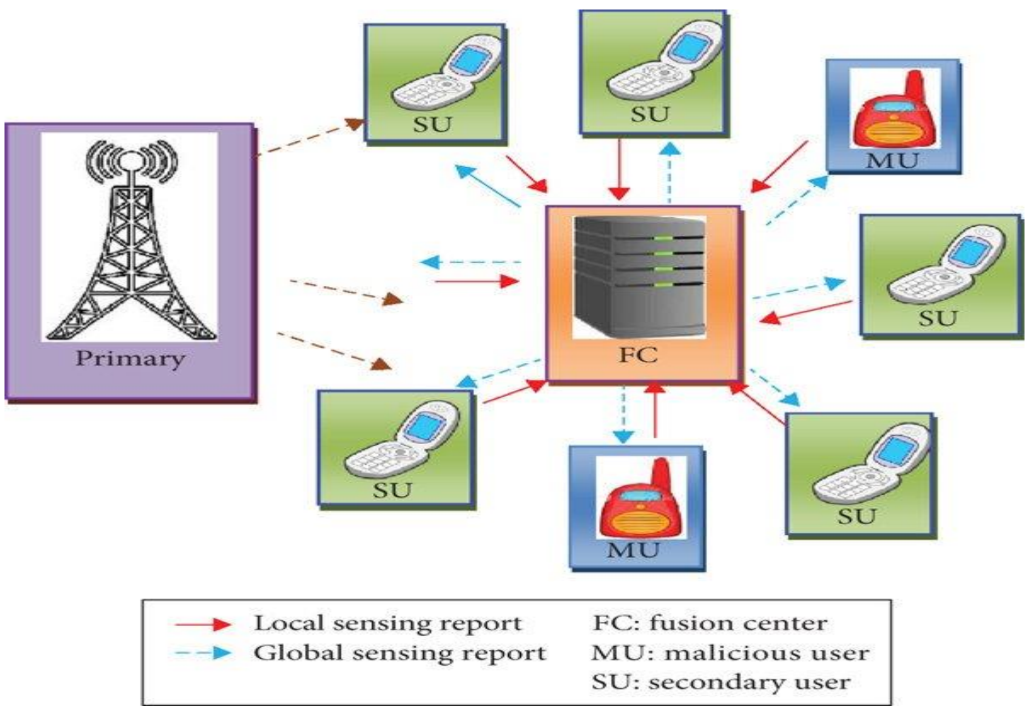


Fig.4 Illustrating Cognitive Radio with its components

A CR "monitors its own performance continuously", in addition to "reading the radio's outputs"; it then uses this information to "determine the RF environment, channel conditions, link performance, etc.", and adjusts the "radio's settings to deliver the required quality of service subject to an appropriate combination of user requirements, operational limitations, and regulatory constraints".

1.2.2 HOW IT OVERCAME SDR?

Cognitive radio (CR) enhances and surpasses existing software-defined radio (SDR) technology by incorporating advanced spectrum sensing, dynamic spectrum access, and intelligent decision-making capabilities. While SDR allows for flexible reconfiguration of radio parameters, CR takes this a step further by autonomously adapting to its environment based on real-time spectrum analysis, regulatory constraints, and user requirements.

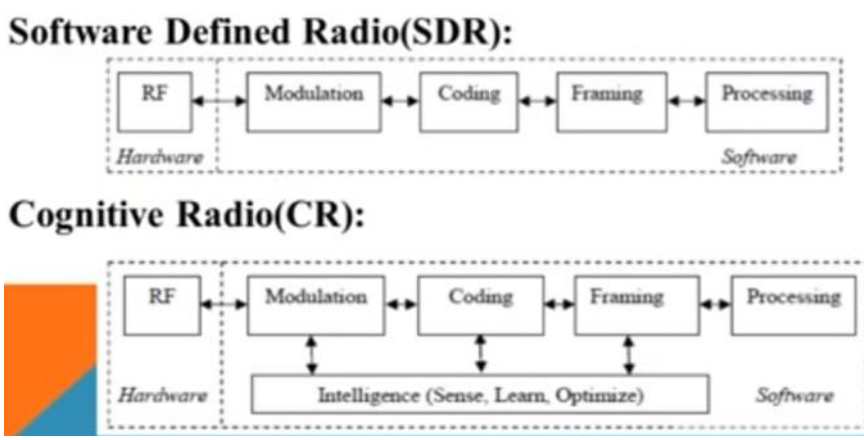


Fig.5 Block diagram of Cognitive radio and software defined radio

Unlike SDR, which primarily focuses on hardware flexibility, CR's cognitive capabilities enable it to intelligently sense, analyze, and exploit available spectrum opportunities, mitigating interference, optimizing spectrum utilization, and enhancing overall network performance. By dynamically adjusting transmission parameters, frequency bands, and modulation schemes, CR optimizes spectral efficiency while ensuring seamless coexistence with incumbent users, thus maximizing spectrum utilization and enabling efficient deployment of next-generation wireless communication systems.

1.2.3 FEATURES OF COGNITIVE RADIO

Cognitive radio is a radio that can be programmed and configured dynamically to use the best wireless channels in its vicinity to avoid user interference and congestion. The main features of cognitive radio are:

- A transceiver can determine its geographic location.
- It can identify and authorize its users.
- It can perform encryption and decryption.
- Able to sense nearby wireless devices.
- It can adjust its output.

1.2.4 COGNITIVE RADIO FACETS

The two main facets used in CR are spectrum sensing and spectrum database.

1. **Spectrum sensing:** CR devices track the spectrum bands in their neighborhoods to identify users licensed to operate in that band. They also look for unused portions of the RF spectrum known as white spaces or spectrum holes. These holes are created and removed dynamically and can be used without a license. Spectrum sensing may be cooperative or non-

cooperative. In the cooperative method, cognitive radio devices share spectrum information, while in the non-cooperative method, each CR device acts on its own.

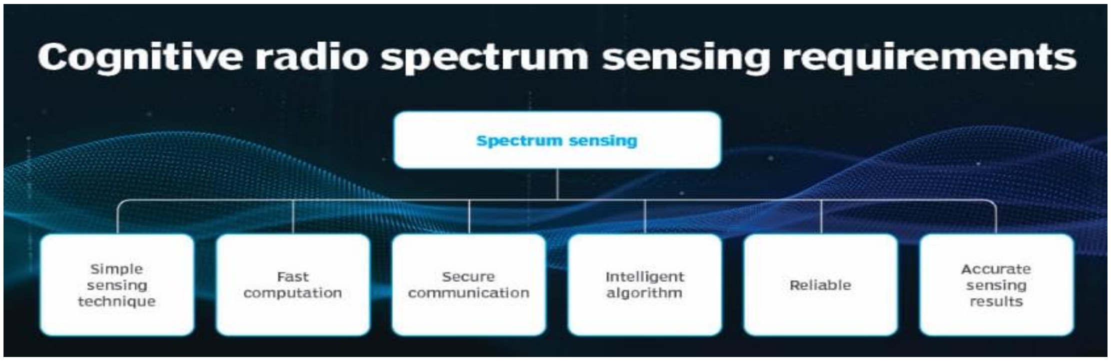


Fig.6 Need for spectrum sensing in CRN

2. **Spectrum Database:** TV stations update their next week's use of the RF spectrum in a database that the FCC maintains. Cognitive radio devices can seek information about free spectrum from this database, so they don't have to rely on complex, time-consuming and expensive spectrum sensing techniques. The drawback of this method is it's difficult for the database to update dynamic spectrum activity in real time. As a result, CR devices may miss out on opportunities to access unused spectrum. To support the growing number of devices that use the RF spectrum, a combined approach is useful. It ensures that devices can quickly and accurately detect unused spectrum and so improve QOS.

1.2.5 RELATION BETWEEN COGNITIVE RADIO AND SPECTRUM SENSING

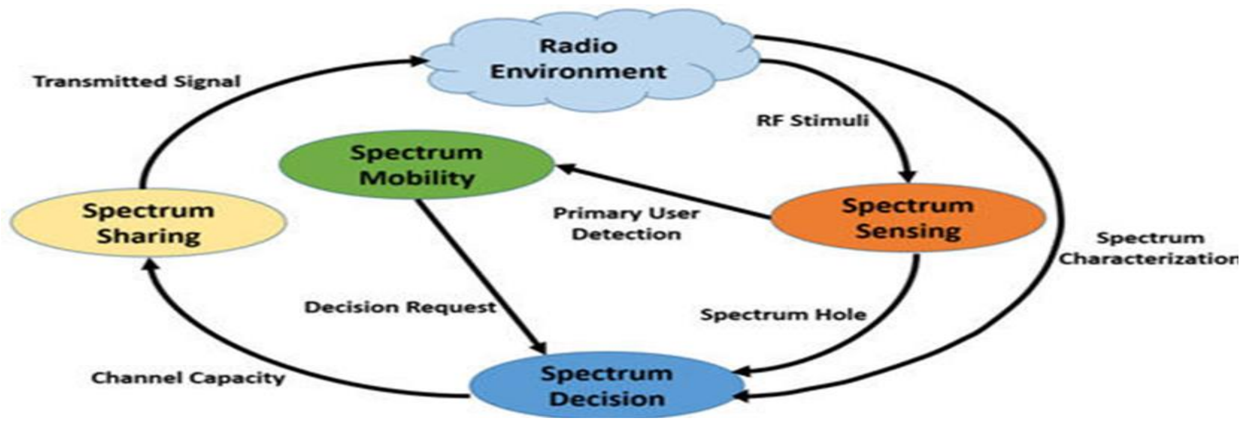


Fig.7 Spectrum sensing abilities of a cognitive radio

1.2.6 FUTURE SCOPE

Since the advancement being developed in each field, cognitive radio took an important place in communication area. In wireless communication system and mobile technology, cognitive radio plays an important role better in future. It is used in machine learning and with other fields and technologies as well.

1.3 COOPERATIVE SPECTRUM SENSING

1.3.1 WHAT IS COOPERATIVE SPECTRUM SENSING?

Cooperative Spectrum Sensing (CSS) is a fundamental concept in **cognitive radio (CR)** systems. In CSS the secondary users (cognitive users)

Collaborate, with each other to sense the spectrum.

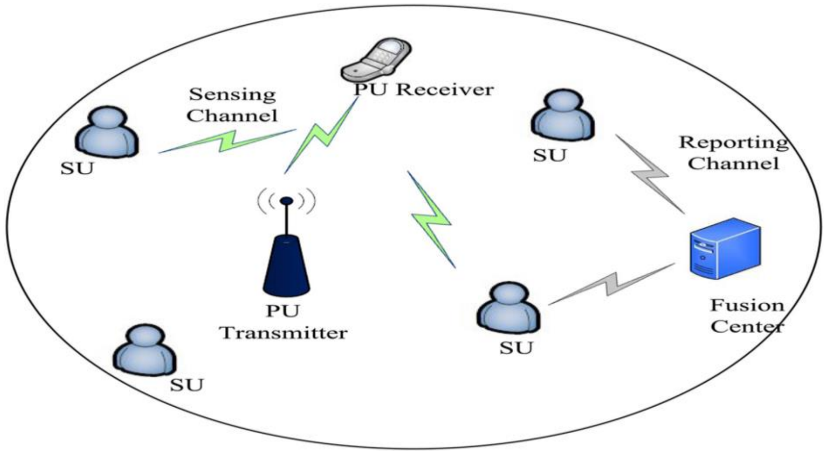


Fig.8 Cooperative Spectrum Sensing between a PU transmitter and PU receiver using FC

By working together, they enhance detection performance. CSS overcomes various challenges, including noise, fading, and transmission impairments. The cooperative approach improves reliability and accuracy in identifying spectral opportunities.

1.3.2 CLASSSIFICATION OF COGNITIVE RADIO NETWORK

Cooperative spectrum sensing involves multiple cognitive radio users that are collaboratively sensing the radio frequency spectrum to improve the reliability and accuracy of spectrum sensing. It can be classified into several categories based on the cooperation model and sensing techniques employed:

1. **Cluster-Based:** Cognitive radios are organized into clusters, and each cluster has a cluster head responsible for coordinating spectrum sensing within the cluster. Cluster heads collect sensing data from member radios, fuse the data locally, and then transmit the fused results to a central fusion center or neighboring clusters for further processing.

2. **Hierarchical:** Hierarchical cooperative sensing involves organizing cognitive radios into multiple layers based on their sensing capabilities and responsibilities. Lower layers perform local sensing and decision fusion, while higher layers aggregate information from lower layers to make final decisions. This hierarchical structure improves scalability and efficiency in large-scale cognitive radio networks.

3. **Distributed:** In distributed cooperative sensing, cognitive radios collaborate directly with neighboring radios without relying on a central coordinator or fusion center. Radios exchange sensing information and decisions with nearby peers to reach a consensus on spectrum occupancy. This approach reduces the overhead associated with centralized fusion and is more resilient to network failures.

1.3.3 TECHNIQUES OF COOPERATIVE SPECTRUM SENSING

Cooperative spectrum sensing involves multiple cognitive radio devices collaboratively sensing the radio frequency spectrum to improve the reliability and accuracy of spectrum sensing results. Several techniques are used in cooperative spectrum sensing.

1. **Fusion-Based Techniques:** Fusion techniques combine the local sensing decisions of multiple cognitive radios to make a global decision about spectrum occupancy. Common fusion techniques include:

 - Hard Decision Fusion: In this technique, each cognitive radio makes a binary decision (occupied or unoccupied) based on its local sensing results, and a fusion center aggregates these decisions using simple logic operations like AND, OR, or MAJORITY to make a final decision.

 - Soft Decision Fusion: Soft decision fusion involves transmitting not only the binary decision but also the confidence level associated with it. These confidence levels, represented as probabilities or reliability metrics, are combined at the fusion center to make a more informed global decision.

2. **Relay-Based Techniques:** In relay-based cooperative sensing, cognitive radios not only sense the spectrum but also relay their local sensing information to a fusion center or to other cognitive radios in the network. This allows for more comprehensive spectrum sensing and helps in overcoming fading, shadowing, and other propagation effects.

3. **Distributed Sensing:** In distributed sensing, cognitive radios collaborate in a decentralized manner without relying on a central fusion center. Each cognitive radio exchanges sensing information with its neighboring radios, and decisions are made collectively based on this distributed information. Distributed sensing reduces the communication overhead and improves scalability.

4. **Optimization-Based Techniques:** Optimization techniques aim to maximize the overall sensing performance by optimizing sensing parameters such as sensing time, sensing frequency, and sensing threshold. These techniques use mathematical optimization algorithms to allocate sensing resources efficiently and optimize the trade-off between sensing accuracy and energy consumption.

5. **Machine Learning-Based Techniques:** Machine learning algorithms, such as support vector machines (SVM), artificial neural networks (ANN), and clustering algorithms, can be employed to analyze the collected sensing data and make the predictions about spectrum occupancy. These techniques can adaptively learn from historical sensing data and improve the accuracy of spectrum sensing over time.

1.3.4 ROLE OF CLUSTERING IN COOPERATIVE SPECTRUM SENSING

Clustering plays a pivotal role in cooperative spectrum sensing by grouping cognitive radios into clusters based on proximity or similarity of sensing capabilities.it improves performance by grouping all secondary users to clusters. These clusters facilitate efficient data aggregation and decision-making processes, reducing the communication overhead and enhancing scalability of the network.

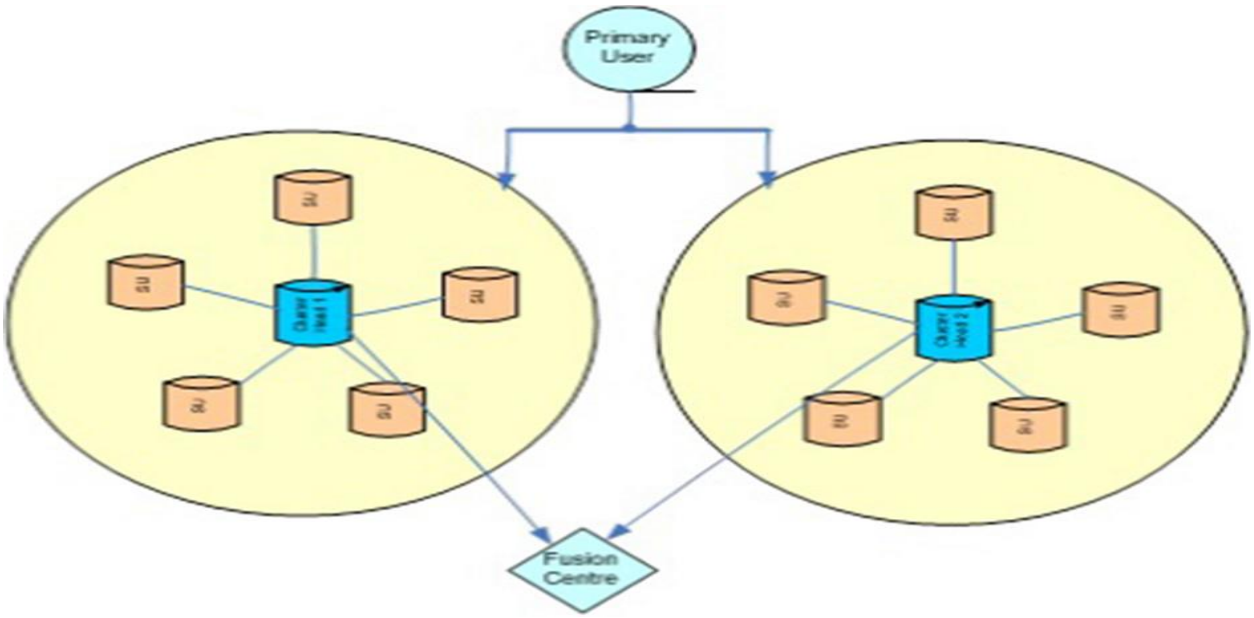


Fig.9 Cluster formation in CSS

Within each cluster, radios collaborate to collectively sense the spectrum and share their local sensing results, enabling more accurate and reliable spectrum occupancy detection. Clustering also enables dynamic reconfiguration of cluster memberships in response to changing network conditions, optimizing resource allocation and improving the overall performance of cooperative spectrum sensing in cognitive radio networks.

1.3.5 APPLICATIONS OF COOPERATIVE SPECTRUM SENSING

- Enhanced spectrum awareness
- Improved spectrum efficiency
- Reliable secondary user access
- Spectrum mobility and adaptation
- Dynamic spectrum access in TV white spaces

- Dynamic spectrum management
- Interference mitigation
- Stochastic modelling of spectrum sensing based applications
- Functional and non-functional metrics of CR applications
- Security of Cooperative Spectrum Sensing techniques.

CHAPTER – II

LITERATURE SURVEY

2.1 Optimal linear weighted cooperative spectrum sensing for clustered-based cognitive radio networks by Haiyan Ye and Jiabao Jiang by Springer

This paper proposes an optimized cooperative spectrum sensing method for clustered cognitive radio networks. By assigning weighted values based on signal-to-noise ratio and historical sensing accuracy, our approach enhances detection performance and reduces error probability. Simulation results demonstrate improved sensing efficiency, benefiting wireless communication applications.

2.2 Enhanced Cooperative Spectrum Sensing in Cognitive Radio Network Using Flower Pollination Algorithm by H. Asfandyar, N. Gul, I.Rasool, A. Elahi

This paper proposes a Flower Pollination Algorithm (FPA) based Cooperative Spectrum Sensing (CSS) scheme for Cognitive Radio Networks (CRNs). The FPA intelligently optimizes weighting coefficients for cooperative users' information to enhance spectrum sensing performance. Simulations demonstrate that the proposed FPA-based scheme outperforms traditional techniques like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Maximum Gain Combination (MGC), and Count methods, offering higher detection accuracy and lower error probability in varying channel conditions and user scenarios.

2.3 Spectrum Sensing Using Cognitive Radio Technology by M.Meena, F.Bhagari and V.Rajendran

This paper explores the utilization of spectrum in mobile communication through cognitive radio technology. It addresses challenges such as spectrum sharing and efficiency, proposing techniques like spectrum sensing, analysis, and allocation. Through MATLAB simulations, it demonstrates the identification of spectrum holes and efficient spectrum usage by primary and secondary users. The study highlights the potential of cognitive radio in enhancing spectrum utilization and management for future wireless communication systems.

2.4 Cognitive Radio Networking and Communications: An Overview by Ying-Chang Liang, Fellow, IEEE, Kwang-Cheng Chen, Fellow, IEEE, Geoffrey Ye Li, Fellow, IEEE, and Petri Mähönen, Senior Member, IEEE

Cognitive Radio (CR) technology offers a solution to the spectrum scarcity problem through dynamic spectrum access. However, achieving truly cognitive radios and networks involves a collaborative effort across various research disciplines. This paper provides a systematic overview of CR networking and communications, focusing on the key functions of the physical (PHY), medium access control (MAC), and network layers. It addresses signal processing techniques for spectrum sensing, cooperative spectrum sensing, and transceiver design at the PHY layer, sensing scheduling schemes, spectrum-aware access MAC, and CR MAC protocols at the MAC layer, and CRN tomography, spectrum-aware routing, and quality-of-service (QoS) control at the network layer. Additionally, emerging CRNs and spectrum-sharing economics are reviewed. The paper concludes by highlighting open questions and challenges in CRN design.

2.5: A Survey of Spectrum Sensing Algorithms for Cognitive Radio Applications by Tevfik Yucek and Huseyin Arslan

The increasing demand for higher data rates, driven by the shift from voice-only to multimedia applications, poses significant challenges to traditional frequency allocation schemes due to spectrum limitations. Cognitive radio emerges as a promising solution to address spectral congestion by enabling opportunistic spectrum usage. This paper explores the concept of cognitive radio, focusing on its ability to autonomously sense and adapt to the electromagnetic environment for efficient spectrum utilization. Spectrum sensing, a fundamental component of cognitive radio, involves obtaining awareness of spectrum usage and primary user presence. Various spectrum sensing methods, including geolocation, database utilization, and local sensing by cognitive radios, are discussed. Challenges associated with spectrum sensing, such as the hidden primary user problem and hardware requirements, are addressed. Additionally, enabling algorithms and multi-dimensional spectrum sensing concepts are explored. Cooperative sensing techniques and network traffic modeling for predicting primary user behavior are also investigated. Finally, the paper examines sensing features in modern wireless standards and provides conclusions on the advancements and challenges in spectrum sensing for cognitive radio applications.

2.6 IEEE 802.22: An Introduction to the First Wireless Standard based on Cognitive Radios Carlos Cordeiro, Kiran Challapali, and Dagnachew Birru and Sai Shankar N

The profound dependence of modern society on radio spectrum, exemplified by its pivotal role in mobile communications, public safety, Wi-Fi, and

TV broadcast, underscores the necessity for efficient spectrum utilization. Unlicensed bands have facilitated a surge in innovative applications, spurred by regulatory leniency, while licensed bands remain underutilized. Cognitive Radios (CRs) emerge as a solution, promising adaptive spectrum utilization without impinging on incumbent devices. The FCC's consideration to open further bands for unlicensed use reflects this paradigm shift. This paper explores the evolution of CRs within the IEEE 802.22 Working Group (WG), focusing on the development of an air interface for unlicensed operation in TV broadcast bands. Key aspects including incumbent service detection, coexistence, and the air interface are discussed, highlighting 802.22's pivotal role in shaping future CR developments. The organization of the paper and its significance are outlined, offering insights into the ongoing efforts to revolutionize spectrum utilization.

2.7 Sensing-based Spectrum Sharing in Cognitive Radio Networks Xin Kang, Ying-Chang Liang ,Hari Krishna Garg , and Lan Zhang

The proliferation of wireless devices in the 21st century has strained the traditional fixed spectrum allocation policy due to the increasing demand for radio spectrum. Cognitive radio (CR) and cognitive radio networks (CRNs) have emerged as solutions to this challenge. In this paper, we propose a new transmission model called sensing-based spectrum sharing, where secondary users (SUs) adapt their transmission parameters based on spectrum sensing results to access frequency bands allocated to primary users (PUs). Unlike opportunistic spectrum access or spectrum sharing, SUs in our model can access PU bands regardless of PU activity, adjusting transmission power accordingly to avoid interference. We analyze the ergodic capacity of the SU link under joint constraints on transmit and interference powers, demonstrating that our model achieves higher capacity compared to existing approaches. This paper contributes to the optimization of transmit power and sensing time for efficient spectrum utilization in CRNs, addressing practical constraints for enhanced wireless communications.

2.8 A Parallel Cooperative Spectrum Sensing in Cognitive Radio Networks by Shengli Xie, Yi Liu, Yan Zhang, and Rong Yu

In cognitive radio networks, spectrum sensing plays a critical role in identifying spectrum opportunities for secondary systems. Existing approaches focus on improving sensing accuracy but may limit efficiency. This paper proposes a novel parallel spectrum sensing scheme where multiple secondary users are optimally selected to sense different channels simultaneously, enhancing efficiency. An analytical model investigates the tradeoff between transmitted data and sensing overhead. Throughput maximization is formulated, considering parameters like the number of users performing parallel sensing. Both saturation and non-saturation scenarios are analyzed. Numerical examples demonstrate the scheme's ability to achieve higher throughput and lower delay compared to existing mechanisms. This work contributes to addressing spectrum scarcity challenges and enhancing spectrum utilization in cognitive radio networks.

2.9 Cooperative Spectrum Sharing in Cognitive Radio Networks With Multiple Antennas Raed Manna, Student Member, IEEE, Raymond H. Y. Louie, Student Member, IEEE, Yonghui Li, Senior Member, IEEE, and Branka Vucetic, Fellow, IEEE

Cognitive Radio (CR) technology offers a promising avenue for enhancing radio spectrum utilization by enabling secondary user (SU) networks to coexist with primary user (PU) networks through spectrum sharing. This paper explores three models for spectrum sharing: interweave, underlay, and overlay. While each model presents unique challenges and advantages, the overlay model, which involves simultaneous transmission of SUs and PUs with cooperative communication techniques, holds particular promise. We propose a novel overlay scheme where PUs lease half of their time slots to SUs, facilitating cooperative relaying of PU data. Unlike previous works primarily focused on PU performance, our scheme prioritizes enhancing performance for both PUs and SUs, resulting in significant rate and error performance improvements over conventional underlay systems. We derive closed-form expressions for bit-error-rate (BER) and rate, providing insights into system parameters' effects on PU performance. Monte Carlo simulations confirm the scheme's efficacy, underscoring its potential for enhancing both PU and SU performance in CR networks.

CHAPTER – III

MATLAB

3.1 WHAT IS MATLAB?

MATLAB stands for Matrix Laboratory. It is a high-performance language that is used for technical computing. It allows matrix manipulations, plotting of functions, Implementation of algorithms and creation of user interfaces.

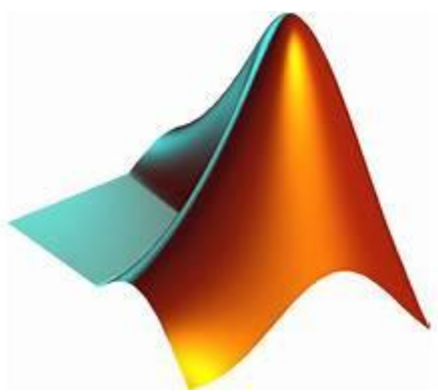


Fig.10 Icon of MATLAB software

The MATLAB application is built around the MATLAB programming language. Common usage of the MATLAB application involves using the "Command Window" as an interactive mathematical shell or executing text files containing MATLAB codes.

3.2 COMPONENETS PRESENT IN MATLAB

- **Command Window:** In this window one must type and immediately execute the statements, as it requires quick prototyping. These statements cannot be saved. Thus, this is can be used for small, easily executable programs.

- **Editor (Script):** In this window one can execute larger programs with multiple statements, and complex functions These can be saved and are done with the file extension '.m '

- **Workspace:** In this window the values of the variables that are created in the course of the program (in the editor) are displayed.

- **Command History windows**: This window displays a log of statements that you ran in the current and previous MATLAB sessions. The Command History lists the time and date of each session in the short date format for your operating system, followed by the statements from that session.

3.3 BASIC FUNCTIONS IN MATLAB

Function	**Description**
disp()	The values or the text printed within single quotes is displayed on the output screen

Function	**Description**
Clear	To clear all variables
close all	To close all graphics window
Clc	To clear the command window
exp(x)	To compute the exponential value of x to the base e
abs(x)	To compute the absolute value of x
sqrt(x)	To compute the square root of x
log(x)	To compute the logarithmic value of x to the base e
log10(x)	To compute the logarithmic value of x to the base 10
rem(x, y)	To compute the remainder of x/y
sin(x)	To compute the sine of x
cos(x)	To compute the cosine of x
tan(x)	To compute the tangent of x

Function	Description
atan2(x, y)	To compute the arctangent or inverse of y/x

Table.1 Functions in MATLAB

3.4 PLOTTING IN MATLAB

A plot is a graphical technique for representing a data set, usually as a graph showing the relationship between two or more variables. In a general sense, a plot refers to the sequence of events that make up a story or narrative. It encompasses the arrangement and development of characters, settings, conflicts, and resolutions within a work of fiction or non-fiction. The plot serves as the structural framework that organizes the unfolding of events, guiding the audience through the storyline and engaging them with its twists, turns, and developments.

A well-crafted plot typically includes key elements such as exposition (introduction of characters and setting), rising action (development of conflicts and complications), climax (the highest point of tension or drama), falling action (resolution of conflicts), and resolution (conclusion or outcome). These elements work together to create a cohesive and compelling narrative arc that captivates the audience's attention and elicits emotional responses. Overall, the plot is fundamental to storytelling, providing the framework upon which themes, character arcs, and messages are built.

3.5 STEPS TO PLOT OUR DATA IN MATLAB:

1. **Define the Data**: First, define the data you want to plot. You'll need two vectors: one for the x-coordinates (X) and another for the corresponding y-coordinates (Y). These vectors should have the same length if you're plotting points connected by line segments.

2. **Create the Plot**:

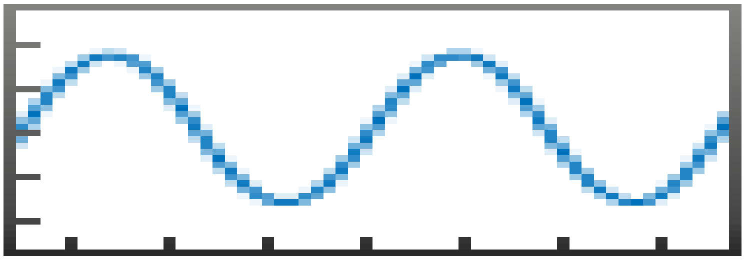
   - Use the plot(X, Y) function to create a 2-D line plot. This will display the data in Y against the corresponding values in X.

   - If you want to customize the line style, marker, and colour, you can use the plot(X, Y, LineSpec) syntax. Replace LineSpec with your desired specifications (e.g., 'r-' for a red solid line).

EXAMPLE PROGRAM FOR BETTER UNDERSTANDING

```
X = linspace(0, 2*pi, 100); % Create x-values from 0 to 2*pi
Y = sin(X); % Corresponding y-values (sine function)
% Create the plot
plot(X, Y, 'b-'); % Blue solid line
xlabel('X-axis');
ylabel('Y-axis');
title('Simple Sine Curve');
grid on; % Add grid lines
```

OUTPUT



3.6 FUNCTIONS IN MATLAB

A **function** is a block of statements that intend to perform a specific task. Functions allow the users to reuse the code frequently. MATLAB has several predefined functions which are ready to use such as **sin(), fact(), cos()** etc. MATLAB also allows the users to define their own functions.

Syntax:

function output_params = function_name(iput_params)

% Statements

End

- The function starts with the keyword function.
- Returning variables of the function are defined in output_params
- function_name specifies the name of the function
- input_params are input arguments to the function

3.7 STEPS TO WRITE A FUNCTION IN MATLAB

1. **Function Declaration**:

   - Begin by declaring the function name, inputs, and outputs. The syntax for function declaration is as follows:

   - function [y1, ..., yN] = myfun(x1, ..., xM)

   - Replace myfun with your desired function name.

   - Specify the input arguments x1, ..., xM and the output arguments y1, ..., yN.
   - This declaration statement must be the **first executable line** of the function.

   - Valid function names start with an alphabetic character and can contain letters, numbers, or underscores.

2. **Function Implementation**:

   - Write your MATLAB code inside the function.

   - You can save your function in either of the following ways:

- In a **function file**: This file contains only function definitions. The filename must match the name of the first function in the file.

- In a **script file**: This file includes commands and function definitions. Functions must appear at the end of the file, and script files cannot have the same name as a function in the file.

- Note that functions are supported in scripts starting from MATLAB R2016b.

- values = [12.7, 45.4, 98.9, 26.6, 53.1];

- [ave, stdev] = stat(values)

3.8 TOOL BOXES IN MATLAB

- A **toolbox** is a collection of functions and/or classes that provide specialized tools for specific tasks or problem domains within MATLAB.

- These toolboxes enhance MATLAB's functionality by offering pre-built functions, algorithms, and utilities.

- Toolboxes can cover diverse areas such as signal analysis, image processing, control design, financial modelling, and more.

3.9 EXAMPLES OF INBUILT TOOLBOXES IN MATLAB

- **Simulink**: For modelling and simulating dynamic systems.

- **Sim scape**: For physical modelling and simulation.

- **State flow**: For event-based modelling.

- **Signal Processing Toolbox**: For signal analysis and processing.

- **Image Processing Toolbox**: For image manipulation and analysis.

- **Control System Toolbox**: For control system design and analysis.

- **Financial Toolbox**: For financial modelling and risk management.

- **Statistics and Machine Learning Toolbox**: For statistical analysis and machine learning.

3.9.1 COMMUNICATION TOOLBOX

Communications Toolbox™ provides algorithms and apps for the design, end-to-end simulation, analysis, and verification of communications systems. The toolbox includes a graphically based app that lets you generate custom- or standard-based waveforms. You can create test vectors to verify receiver performance or to create datasets for artificial intelligence (AI) applications by adding RF impairments to waveforms. The toolbox lets you model propagation channels statistically or with ray-tracing solutions that include terrain and buildings. You can compensate for the effects of channel degradations and use SDRs to verify your designs with over-the-air (OTA) testing.

Communications Toolbox facilitates modelling communications links from antenna to RF chain to bit processing (with Antenna Toolbox™ and RF Block set ™). You can accelerate BER simulations using the cloud or your local cluster (with Parallel Computing Toolbox™). The toolbox helps you solve communications problems using AI techniques (with Deep Learning Toolbox™).

3.9.1.1 APPLICATIONS OF COMMUNICATION TOOLBOX

- Waveform Generation
- RF Propagation and Channel Modeling
- End-to-End Simulation

3.9.2 SIGNAL PROCESSING TOOLBOX

Signal Processing Toolbox™ provides functions and apps to manage, analyse, pre-process, and extract features from uniformly and non-uniformly sampled signals. The toolbox includes tools for filter design and analysis, resampling, smoothing, de-trending, and power spectrum estimation. You can use the Signal Analyser app for visualizing and processing signals simultaneously in time, frequency, and time-frequency domains. With the Filter Designer app you can design and analyse FIR and IIR digital filters. Both apps generate MATLAB® scripts to reproduce or automate your work.

Using toolbox functions, you can prepare signal datasets for AI model training by engineering features that reduce dimensionality and improve the quality of signals. You can access and process collections of files and large datasets using signal data stores. With the Signal Label app, you can annotate signal attributes, regions, and points of interest to create label signal sets. The toolbox supports GPU acceleration in addition to C/C++ and CUDA® code generation for desktop prototyping and embedded system deployment.

3.9.2.1 APPLICATIONS OF SIGNAL PROCESSING TOOLBOX

- Filter Design and Analysis
- Spectral Analysis
- Time-Frequency Analysis
- Signal Exploration and Pre-processing
- Feature Extraction for Machine Learning
- GPU Acceleration and Code Generation

CHAPTER – IV

FLOWER POLLINATION ALGORITHM

4.1 WHAT IS FLOWER POLLINATION ALGORITHM?

The FPA was developed by Xing-she Yang in 2012 inspired by the pollination process of flowering plants. The flower pollination algorithm (FPA) is a novel heuristic optimization algorithm inspired by the pollination behaviour of flowers in nature. This algorithm has applied in different types of areas like optimization, numerical problems, image recognition, local search problem, economic and emission dispatch problems, Power System, scheduling problems, and combinatorial optimization problem. PA has been extended to multi-objective optimization.

4.1.1 STEPS FOLLOWED BY FLOWER POLLINATION ALGORITHM

- Biotic and cross-pollination can be considered processes of global pollination.

- For local pollination, abiotic pollination and self-pollination are used.

- The interaction or switching of local pollination and global pollination can be controlled by a switch probability.

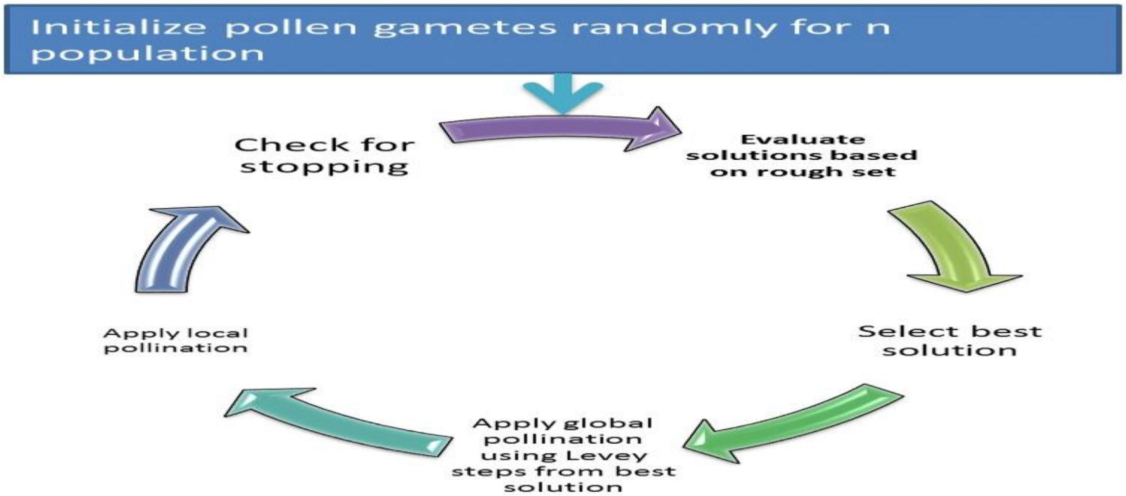


Fig.11 Process of FPA

When the FPA is applied to cognitive radio networks, it can enhance spectrum utilization, mitigate interference, and optimize resource allocation. By mimicking the diverse pollination behaviours, FPA enables dynamic adaptation to varying network conditions, leading to improved performance and robustness in cognitive radio systems. Overall, integrating FPA into cognitive radio networks facilitates intelligent spectrum management, ensuring efficient utilization of available resources while meeting quality of service requirements.

4.2 CHARACTERISTICS OF FLOWER POLLINATION ALGORITHM

- **Global and Local Search**: FPA employs a combination of global exploration and local exploitation through the movement of pollinating agents (flowers), allowing it to efficiently explore the search space while exploiting promising regions.

- **Adaptability**: FPA incorporates mechanisms for adaptive adjustment of search parameters, enabling it to dynamically respond to changes in the optimization landscape and environmental conditions.

- **Population-based Optimization**: FPA maintains a population of candidate solutions (flowers) and iteratively improves them through the exchange of information, promoting diversity and preventing premature convergence.

4.3 IMPLEMENTING FLOWER POLLINATION ALGORITHM IN COGNITIVE RADIOS

The proposed scheme of Flower Pollination Algorithm (FPA) intelligently finds optimum weighting coefficients against cooperative users' information and utilizes these weights in the global decision of the Soft Decision Fusion (SDF). This scheme is able to find optimum weights that lead to minimum false alarm, high detection and minimum error probability. The system is simulated for different numbers of the cooperative users and Signal-to-Noise Ratios (SNRs) that shows better sensing performance of the proposed FPA.

In the proposed work, all cooperative users employ energy detector that compares received signal energy of the channel with an adaptive threshold determined by the Flower Pollination Algorithm (FPA). As the cooperative users in the proposed work are considered at different geographical locations and experience independent Raleigh fading effects. Therefore, it is not suitable to treat their sensing performances equally in the global decision made by the FC.

Similarly, in the proposed method the FPA instead of keeping fixed threshold point for all sensing intervals determines an optimized threshold value. The weighted coefficient vector with optimum threshold point is selected by the proposed method. This leads to a minimum false alarm, high detection and low error probability at the FC. The final weighted results are further utilized by the SDF to reach to a final global decision at the FC. FPA finds the optimal set of weighted coefficient vector against the sensing reports received from all cooperative users. In the random normalized set of coefficient vector population the vector with low error probability results are elected as the optimal set of vector and is further utilized in the global decision of the SDF scheme.

The steps involved in optimization process are given below:

Step 1: Initial Population - The algorithm initializes the initial population by randomly generating N flower or pollen gametes. These values are normalized between the range of 0 and 1.

Step 2: Fitness of the pollen gametes - It determines the suitability of each coefficient vector by measuring their fitness scores. The population is arranged in the increasing order of their fitness measure.

Step 3: Global and Local Pollination - In this step, either global or local pollination is performed with the help of current best solution and global best pollens. The interaction or switching between global and local pollination is controlled by probability switch $p \in [0, 1]$ slightly biased towards local pollination .This process results in new population.

Step 4: New Population - The fitness of new population is determined in the same way as described in step 2. The results are then sorted in ascending order of their fitness and step 3 is repeated again.

Step 5: Stopping Criteria - FPA repeats step 2 time and again until the minimum *Pe* is not achieved or given number of iteration are not completed.

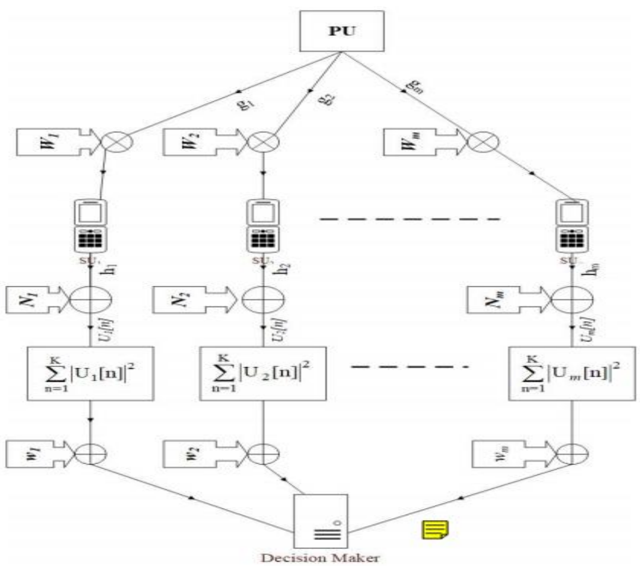


Fig.12 Implementing CSS model in FPA

In this diagram, FC receives statistical observations of the *M* SUs about the channel. The sensing users in the diagram operate similar to a forwarding relay that simply receive and forward the received PU signal to the FC. The final verdict regarding the presence of the licensed user is made at the FC using SDF based CSS that use received signal information of the SUs.

Here, the total number of samples is $2K = BTs$ that are considered large enough to make the energy distribution Gaussian. Here, *B,* is the signal bandwidth and, Ts , is the sensing time. $W\,ni[\ ]$ is the Additive White Gaussian Noise (AWGN) of channel between *i*th user and PU. Similarly, wi is the weight assigned to the ith sensing user. $\boldsymbol{W}$ =[w_1, w_2 … w_M] ^T , are the weighting coefficient vectors. These weights are then optimized to determine the appropriate threshold value β.

Assuming that the $Pf = Pm$, where Pm is the misdetection probability and $Pf = -1\ Pd$, therefore the total error probability Pe is determined as:

$$Pe=Pf+ Pm$$

The error probability is applied as fitness function and is highly dependent on the selection of the wρ. Therefore β is optimized for the selection of the weighting coefficients leads to a high detection, minimum false alarm, and low error probability.

4.4 IMPACT OF USING FLOWER POLLINATION ALGORITHM IN THE COGNITIVE RADIO NETWORKS

- **Dynamic Spectrum Management:** FPA can adaptively allocate spectrum resources based on changing network conditions and user requirements. It enables intelligent spectrum utilization, allowing cognitive radios to opportunistically access available spectrum bands while avoiding interference with licensed users.

- **Optimization of Resource Allocation:** FPA optimizes resource allocation by efficiently distributing available spectrum resources among cognitive radio users. It can balance competing demands for bandwidth, power, and other resources to maximize overall network performance and throughput.

- **Adaptive Channel Selection:** FPA facilitates adaptive channel selection by identifying and exploiting vacant spectrum bands for transmission. It can intelligently switch between available channels based on channel conditions, traffic load, and interference levels, enhancing communication reliability and quality of service.

- **Self-Organization and Adaptation:** FPA enables self-organization and adaptation in cognitive radio networks by allowing nodes to autonomously adjust their behaviour and parameters based on local observations and interactions. It promotes distributed decision-making and robustness to dynamic network conditions.

- **Energy Efficiency:** FPA promotes energy-efficient operation in cognitive radio networks by optimizing transmission power levels, channel utilization, and routing decisions. It can minimize energy consumption while maintaining satisfactory network performance and coverage.

CHAPTER – V

LINEAR WEIGHTED ALGORITHM

5.1 WHAT IS OPTIMAL LINEAR WEIGHTED ALGORITHM?

The Optimal Linear Weighted Algorithm in cognitive radio is a technique used to efficiently allocate spectrum resources. It prioritizes spectrum bands based on predefined criteria such as channel quality and interference levels. By assigning weights to these factors, the algorithm optimally selects the best available spectrum bands for transmission, maximizing throughput and minimizing interference. This approach enhances spectrum utilization, enabling cognitive radios to dynamically adapt to changing environmental conditions and efficiently utilize spectrum resources.

5.2 FACTORS TAKEN INTO CONSIDERATION FOR ASSIGNING WEIGHTS IN LINEAR WEIGHTED ALGORITHM:

1. **Signal-to-Noise Ratio (SNR) based weighting**: Prioritizes spectrum bands with higher SNR values for transmission.

2. **Interference-based weighting**: Assigns weights based on the level of interference in each spectrum band, favouring bands with lower interference.

3. **Quality-of-Service (QoS) based weighting:** Considers factors such as latency, reliability, and bandwidth requirements, assigning weights accordingly.

4. **Dynamic weighting**: Adjusts weights dynamically based on real-time measurements and environmental conditions to optimize spectrum allocation.

5. **Multi-objective optimization**: Considers multiple criteria simultaneously, such as throughput, fairness, and energy efficiency, using techniques like Pareto optimization.

5.3 WORKING OF LINEAR WEIGHTED ALGORITHM:

1. **Weight Assignment**: Different weight values are assigned to cooperative nodes based on their SNRs and historical sensing accuracy. SUs with better SNRs receive higher weights, emphasizing their contributions to the overall sensing process.

2. **Clustering**: SUs are clustered, and those with superior channel characteristics become cluster heads. These cluster heads gather local sensing information from other SUs within their clusters.

3. **Cooperative Sensing**: The weighted information from all SUs is combined to make informed decisions about spectrum availability. By considering both SNR and historical accuracy, the algorithm enhances detection probability and reduces error rates.

5.4 FUSION CENTRES AND ITS IMPORTANCE:

In the context of cognitive radio systems, a Fusion Centre plays a crucial role in aggregating information from multiple sources to make informed decisions regarding spectrum allocation. In the case of the Linear Weighted Algorithm, the Fusion Centre acts as a central entity responsible for collecting and processing data from various cognitive radios within the network.

- Firstly, the Fusion Centre gathers spectrum sensing data from individual cognitive radios distributed across the network. This data typically includes measurements of signal strength, noise levels, interference, and other relevant parameters.

- Secondly, the Fusion Centre utilizes the Linear Weighted Algorithm to assign weights to different spectrum bands based on the collected data. These weights reflect the suitability of each band for transmission, considering factors such as signal quality, interference levels, and user requirements.

- Thirdly, the Fusion Centre aggregates the weighted information and makes decisions regarding spectrum allocation. It selects the optimal spectrum bands for transmission based on the weighted criteria, aiming to maximize throughput, minimize interference, and meet Quality of Service (QoS) requirements.

Overall, the Fusion Centre serves as a central intelligence hub in cognitive radio networks, leveraging the Linear Weighted Algorithm to effectively allocate spectrum resources and optimize network performance in dynamic and heterogeneous environments.

5.5 LINEAR WEIGHTED ALGORITHM SYSTEM MODEL:

It is necessary to estimate the channel state before the SU sends sensing data in each intervals. In addition, in the node's clustering structure, the nearest cognitive users should be selected as the member nodes in the same cluster, and the channel state between them can be approximately considered to be ideal .The fusion results of each cluster will finally sent to the fusion centre (FC) by cluster heads (CHs) for fusion, and the FC uses OR fusion method for processing .Considering that a FC or base station and N cognitive users participate in cooperative spectrum sensing. The SUs will be organized into K clusters, and there are Kc cognitive users in the c-th cluster.

5.6 CLUSTERING FORMATION IN LINEAR WEIGHTED ALGORITHM:

During the clustering formation, channels for secondary users should be selected primarily. The selection of candidate channels should meet the following requirements: the candidate nodes should be closer to the fusion centre, and the candidate nodes are also be closer to other secondary users. Then, the residual Secondary users are equally divided into several clusters formed into clusters according to the process of clustering formation. If the distance between the cooperative secondary users in a cluster is far, relatively small number of members in a single cluster will be. It will result in low performance of cooperative spectrum sensing of the cluster, and the decision result of the cluster may be inaccurate. The main idea of clustering is to organize the adjacent secondary users into a same cluster. The cluster-based cooperative spectrum sensing can be divided into two parts: spectrum sensing and intra cluster data fusion. All secondary users in each cluster need to sense the PU's signal independently .Then, the channel receives the sensing observations from all member nodes in the cluster, and decides the authorized user's state. Compared with the typical cooperative spectrum sensing, the clustered-based cooperative spectrum sensing can make more reasonable use of the spatial diversity of nodes in different geographical locations, and reduce the error of decision information sent by secondary users. For simplicity, we define the Euclidean distance dis si, sj between i-th node and j-th node, and assumes that the number of nodes in each cluster is an integer.

The specific steps of clustering process are as follows:

- **Step 1:** The distance from all SUs to the FC is calculated, and the 2C SUs with the shortest distance will be selected as candidate CHs;

- **Step 2:** The distance between those candidate CHs and the centroid degree of all cooperative SUs is will be estimated. The optimal nodes with the smallest distance are added into the CHs set {CH1,CH2,...,CHC} , and the number of optimal nodes is C;

- **Step 3:** Initialize the member nodes set of the clusters, cluster centre ˆ mc and the number of nodes in the cluster as Kc . The total number of residual SUs is denoted as Nres = N −C.

- **Step 4:** Calculate the distance between the SUs from residual nodes set and cluster centroid. For a SU, if it satisfies with c = arg min{dis(si, CHc)} , the node should be joined into c-th cluster and the cluster centroid will be updated. Then, the number of member nodes in c-th cluster plus one, i. e., Kc = Kc + 1 and the total number of residual SUs will be decreased by Nres = Nres − 1;

- **Step 5:** If Kc = N−C C , it shows that the c-th cluster is at full length, and subsequent nodes are no longer joined into the cluster

- **Step 6:** if Nres > 0 , return to step 4 and continue execution

- **Step 7:** The distance from all SUs in each cluster to the FC is calculated, and the nearest SU can be determined as the CH. The CH assigns ID to each member node, and the formation of cluster ends.

Here, SU means secondary users and CH means cluster heads.

5.7 COMPARISON OF LINEAR WEIGHTED ALGORITHMS WITH OTHER ALGORITHMS:

The other algorithms that are used to compare the optimized linear weighted algorithm are:-

- Channel Assignment Algorithm
- Leader Election Algorithm
- Selective Weight Allocation Algorithm

1. **Channel Assignment Algorithm:** Channel assignment algorithms are typically used in wireless communication networks to allocate available channels to different communication links or devices. The goal is to minimize

interference and maximize throughput. These algorithms often involve graph colouring techniques or optimization methods to allocate channels efficiently.

2. **Leader Election Algorithm:** In distributed systems, leader election algorithms determine a single process or node to act as a leader, coordinating the activities of other nodes. These algorithms ensure fault tolerance and efficient coordination among distributed components. Common techniques include the Bully Algorithm or the Ring Algorithm.

3. **Selective Weight Allocation Algorithm:** This algorithm dynamically assigns weights to components based on their performance or relevance in the current context. It's often used in adaptive systems where priorities or conditions change over time. For example, in load balancing systems, selective weight allocation algorithms adjust weights based on the current workload of each component.

5.7.1 RESULTS OBTAINED BY COMPARISON OF THE 4 ALGORITHMS:

- **Efficiency**: Linear weighted algorithms are often efficient because they have a straightforward calculation process. Channel assignment algorithms can also be efficient if designed properly, but their complexity may increase with the size of the network. Leader election algorithms and selective weight allocation algorithms may involve more communication overhead and computation, especially in large-scale distributed systems.

- **Scalability**: Linear weighted algorithms are generally scalable since they can handle a large number of components without significant performance degradation. Channel assignment algorithms may face scalability issues in dense wireless networks due to increased interference and limited channel availability. Leader election algorithms and selective weight allocation algorithms may also encounter scalability challenges in large distributed systems due to communication overhead and synchronization requirements.

- **Flexibility**: Selective weight allocation algorithms offer flexibility by dynamically adjusting weights based on changing conditions. Linear weighted algorithms can also be flexible if the criteria for weight assignment are adjustable. Channel assignment algorithms and leader election algorithms may have limited flexibility since they typically operate based on predefined rules or constraints.

- **Optimization Objective**: The choice of algorithm depends on the optimization objective. Linear weighted algorithms optimize based on linear combinations of features, while channel assignment algorithms optimize channel utilization and interference reduction. Leader election algorithms optimize for fault tolerance and coordination, and selective weight allocation algorithms optimize for adaptive resource allocation.

5.8 WHY OPTIMIZING A LINEAR WEIGHTED ALGORITHM TO OBTAIN CLUSTERS IS BETTER?

- **Resource Efficiency**: Clustering allows for the organization of nodes into groups or clusters, reducing the overhead of communication and coordination among a large number of nodes. In a cognitive radio network, where spectrum resources are dynamically allocated and managed, clustering can help streamline the process of resource allocation and utilization. By grouping nodes with similar characteristics or requirements, the linear weighted algorithm can be applied within each cluster more efficiently, optimizing resource usage.

- **Distributed Optimization**: Clustering facilitates distributed optimization by enabling local decision-making within each cluster. Instead of applying a global linear weighted algorithm across the entire network, which may require extensive communication and coordination, each cluster can independently perform optimization based on its own local information and objectives. This decentralized approach can improve scalability and adaptability in dynamic cognitive radio environments.

- **Interference Mitigation**: Clustering can help mitigate interference in cognitive radio networks by spatially grouping nodes with compatible transmission characteristics. By assigning different frequency channels or time slots to different clusters, interference among neighbouring nodes can be minimized, enhancing overall network performance and reliability. The linear weighted algorithm within each cluster can then optimize resource allocation considering interference constraints specific to that cluster.

- **Fault Tolerance**: Clustering enhances fault tolerance in cognitive radio networks by isolating faulty or malfunctioning nodes within clusters. In the event of node failures or communication disruptions, the impact can be limited to the affected cluster, preventing cascading failures across the entire network. The linear weighted algorithm can adapt to changes in cluster membership or topology, ensuring continued optimization even in the presence of faults or failures.

- **Dynamic Spectrum Access**: Cognitive radio networks rely on dynamic spectrum access to opportunistically utilize available spectrum bands. Clustering with a linear weighted algorithm can facilitate dynamic spectrum sharing by coordinating spectrum usage within and across clusters. By dynamically adjusting weights based on spectrum availability, usage patterns, and quality of service requirements, the algorithm can optimize spectrum utilization while minimizing interference and ensuring fairness among cluster members.

To conclude, Clustering with a linear weighted algorithm offers a more scalable, efficient, and adaptable approach to resource management in cognitive radio networks compared to a normal linear weighted algorithm. By leveraging the benefits of clustering, such as resource efficiency, distributed optimization, interference mitigation, fault tolerance, and dynamic spectrum access, cognitive radio networks can effectively utilize spectrum resources and meet the diverse communication requirements of modern wireless applications.

5.9 BENEFITS OF LINEAR WEIGHTED ALGORITHM

- **Improved Sensing Performance**: The proposed scheme outperforms traditional equal-weight methods by leveraging individual SU characteristics.

- **Enhanced Spectrum Utilization**: Dynamic access to idle spectrum intelligently optimizes resource utilization.

- **Reduced Channel Congestion**: By avoiding static spectrum division, the algorithm mitigates congestion issues.

CHAPTER – VI

CLUSTERING

6.1 WHAT IS CLUSTERING?

Clustering in wireless communication refers to the process of organizing a network of wireless devices into groups or clusters to improve network efficiency, scalability, and management.

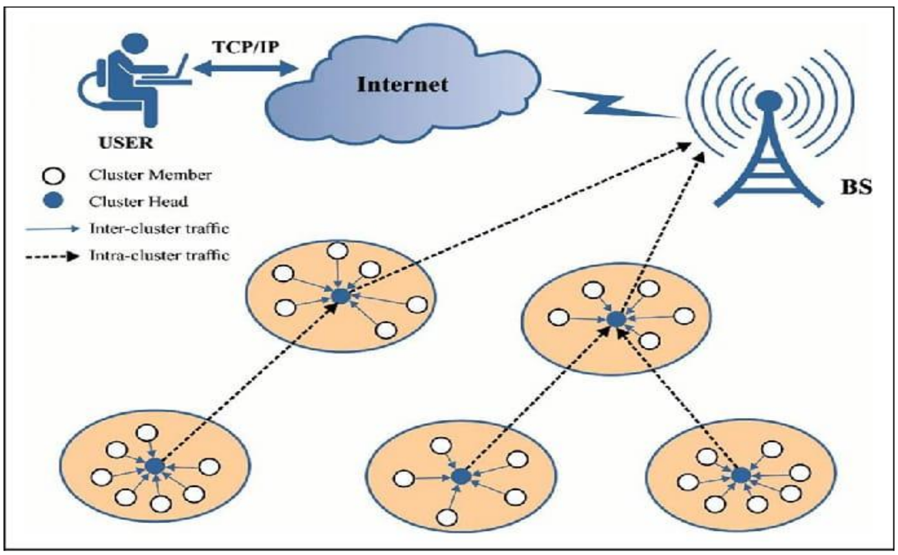


Fig.13 Base Station Accessing information from cluster heads

In clustering, nodes within each cluster communicate directly with each other or with a central node called the cluster head, which then communicates with other cluster heads or a base station. This hierarchical structure helps in reducing energy consumption, optimizing resource utilization, and enhancing network performance.

6.2 WHAT IS CLUSTERING IN COGNITIVE RADIO?

Clustering in cognitive radio refers to the grouping of cognitive radio nodes into clusters to efficiently manage spectrum usage and communication. These clusters typically consist of nodes with similar characteristics or objectives, allowing for more coordinated spectrum access and better utilization of available resources.

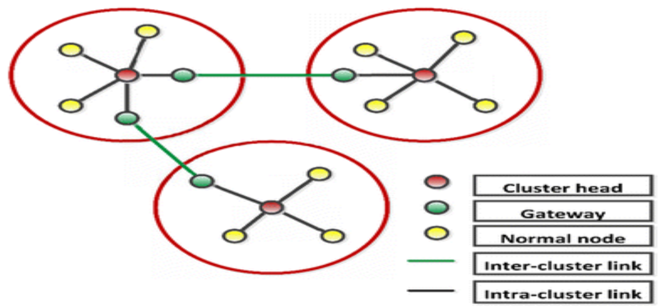


Fig.14 Connecting clusters using gateways

6.3 WHY DO WE DO CLUSTERING IN CRN?

Clustering in Cognitive Radio Networks (CRNs) serves several purposes:

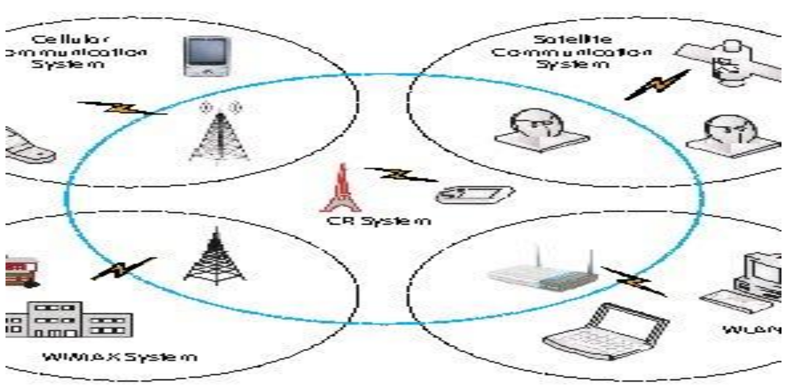


Fig.15 Purpose of clustering in CRN

- **Spectrum Management:** By clustering nodes with similar characteristics or objectives, CRNs can better manage spectrum access and utilization. Clusters can coordinate to allocate spectrum dynamically, minimizing interference and maximizing efficiency.

- **Cooperative Communication:** Clustering facilitates cooperative communication among nodes within the same cluster. Nodes can share spectrum sensing information, coordinate spectrum access, and collaborate on data transmission, leading to improved network performance.

- **Interference Mitigation:** Clustering helps mitigate interference by organizing nodes into groups that can coordinate their transmissions to avoid or mitigate interference with other users or systems sharing the spectrum.

- **Dynamic Spectrum Access:** Clustering facilitates dynamic spectrum access by enabling efficient coordination and negotiation among clusters of cognitive radio nodes to access available spectrum bands opportunistically.

- **Quality of Service (QoS) Management:** Clustering allows for better management of QoS metrics such as latency, throughput, and reliability by grouping nodes with similar communication requirements and dynamically adjusting resource allocation within each cluster.

- **Scalability:** Clustering can improve the scalability of CRNs by dividing the network into manageable clusters, allowing for easier network management and control as the network size grows.

Overall, clustering in CRNs enables more efficient spectrum usage, better network coordination, and enhanced performance in dynamic and heterogeneous wireless environments.

6.4 CLUSTERING IN LINEAR WEIGHTED ALGORITHM

Clustering in wireless communication often involves organizing network nodes into groups based on certain criteria, such as proximity or similarity. In a linear weighted algorithm, the importance of each node or feature is weighted linearly. This approach can help optimize resource allocation and communication efficiency within the network. The lack of spectrum resources restricts the development of wireless communication applications.

In order to solve the problems of low spectrum utilization and channel congestion caused by the static division of spectrum resource, an optimal linear weighted cooperative spectrum sensing for clustered-based cognitive radio networks can be used.

In this network different weight values will be assigned for cooperative nodes according to the SNR of cognitive users and the historical sensing accuracy. In addition, the cognitive users can be clustered, and the users with the better channel characteristics will be selected as cluster heads for gathering the local sensing information.

CHAPTER – VII

PARAMETERS

7.1 PARAMETER ANALYSIS

By utilizing the below parameters, we can gain insights of the FPA's and OLWA's behaviour and identify optimal settings for cognitive radio network optimization:

- **Population Size (N):** Investigate the effect of varying the population size (N) on the convergence speed and solution quality. Test different values such as N = 50, 100, 150, and 200 to find the optimal population size.

- **Number of Pollens per Flower (M)**: Explore how the number of pollens per flower affects the algorithm's performance. Try different values of M, such as M = 5, 10, 15, and 20, to determine its impact on convergence and exploration-exploitation balance.

- **Number of Iterations**: Analyse the influence of the number of iterations on convergence and solution quality. Experiment with different iteration counts, such as 5, 10, 15, and 20 iterations, to observe how it affects the algorithm's performance.

- **Probability Switch (p)**: Assess the sensitivity of the algorithm to the probability switch parameter (p). Explore different values of p, such as p =

0.6, 0.7, 0.8, and 0.9, to understand its impact on the algorithm's ability to balance exploration and exploitation.

- **Sensing Interval**: Investigate the effect of changing the sensing interval on the algorithm's performance. Test different sensing intervals, such as 0.5 ms, 1 ms, 2 ms, and 5 ms, to see how it influences convergence speed and accuracy.

- **Initial SNR Range**: Analyse the impact of the initial SNR range (-25 dB to +10 dB) on the algorithm's ability to converge to optimal solutions. Experiment with narrower and wider SNR ranges to observe how it affects the algorithm's performance.

- **Clustering Parameter:** The Linear Weighted Algorithm incorporates clustering with 3 clusters to facilitate weighted decision-making based on sensing results.

- **Number of Samples:** Each sensing interval comprises 200 samples, ensuring sufficient data for robust spectrum sensing and decision-making.

7.2 ANALYSING THE RESULTANT PARAMETERS

The observed trend in the figures indicates a notable correlation between the number of users and the error probability across different signal-to-noise ratio (SNR) ranges in the cognitive radio network. As the number of users increases, there is a consistent decrease in the error probability across all schemes, irrespective of the SNR range. This phenomenon can be attributed to the collective intelligence and collaborative capabilities of the expanding user base within the network. With more users actively participating in spectrum sensing and decision-making processes, the cognitive radio network benefits from a broader spectrum of observations and insights into the radio environment.

Considering the SNR range, the decrease in error probability with increasing users suggests that the collaborative efforts among users contribute significantly to improving the network's ability to accurately detect and characterize the primary user signals amidst varying levels of noise and interference. In lower SNR ranges, where the signal strength relative to noise is weaker, the added diversity and redundancy introduced by more users aid in distinguishing the primary user signals from background noise, thereby reducing error probability. Similarly, in higher SNR ranges, where the signal is stronger relative to noise, the collaborative sensing and decision-making among a larger user population further enhance the network's ability to make accurate detections and mitigate false alarms, leading to a decrease in error probability.

7.3 THE IMPACT OF INCREASING THE NUMBER OF USERS ON ERROR PROBABILITY IN FPA SCHEME FOR COGNITIVE RADIO NETWORKS

1. **Effect of Interference Mitigation:** As the number of users increases, the cognitive radio network gains more opportunities for interference mitigation. With a larger user population, there are more opportunities for cooperation among users, leading to better interference management and reduced error probability.

2. **Diversity in Sensing Results:** With a greater number of users independently sensing the primary user (PU) channel, there's a higher likelihood of obtaining diverse sensing results. This diversity in sensing outcomes allows for better decision-making in spectrum access, resulting in reduced error probability.

3. **Enhanced Cooperative Diversity:** Cooperative diversity, facilitated by collaboration among multiple users, becomes more effective as the number of users increases. Cooperative diversity allows the cognitive radio network to exploit spatial diversity to combat fading and interference, resulting in more reliable communication and lower error probability.

4. **Increased Spectrum Sensing:** With a larger user population, there's an increase in the diversity of spectrum sensing approaches and capabilities. Different users may employ varying sensing techniques or have access to different portions of the spectrum, leading to a more comprehensive view of the available spectrum and improved error probability performance.

Thus, increasing the number of users in the cognitive radio network enhances the network's ability to exploit diversity, collaborate for interference mitigation, and make informed decisions regarding spectrum access, thereby reducing error probability and improving overall system performance. Here, both the flower pollination algorithm and the linear weighted algorithm shows a decrease on increasing the number of users in various scenarios .From this we come to know that by using all the above mentioned parameters the performance analysis can be made to increase the ability of the cognitive radio network to provide a better result.

CHAPTER – VIII

COMPARISON BETWEEN FLOWER POLLINATION ALGORITHM AND LINEAR WEIGHTED ALGORITHM

PURPOSES	FLOWER POLLINATION ALGORITHM	LINEAR WEIGHTED ALGORITHM
PRINCIPLES	Inspired by the pollination behaviour of flowering plants, where flowers exchange information to improve genetic diversity. It mimics natural processes and operates in a stochastic manner.	Based on mathematical calculations involving assigning different weights to input values and computing a weighted sum. It follows deterministic principles and relies on linear relationships.
	• Designed for optimization problems seeking global optima, especially for complex and	• Primarily used for linear regression and data analysis, it's well-suited for problems with linear

OPTIMIZATION APPROACH	nonlinear scenarios where traditional methods may struggle. It's capable of handling diverse problem spaces and can adapt to different types of optimization tasks. • FPA is stochastic and mimics natural processes.	relationships between variables. • Linear weighted algorithm is deterministic and relies on mathematical calculations. Linear weighted algorithm is more commonly used for linear regression and data analysis.
FLEXIBILITY	Flexible and adaptable, capable of handling nonlinear, multimodal, and high - dimensional optimization problems. It's not restricted by linearity and can explore diverse solution spaces efficiently.	Limited to linear relationships and may not perform well in scenarios with nonlinear or complex data patterns.
	Known for its ability to converge to global optima efficiently, even in challenging optimization	Convergence may be slower, especially in complex problem spaces with multiple local optima. Performance

CONVERGENCE AND PERFORMANCE	landscapes with multiple local optima. It often exhibits faster convergence rates, making it suitable for a wide range of optimization tasks.	depends on the linearity of the problem and the appropriateness of the weights assigned.

Table.2 Comparison of Flower pollination algorithm and Optimal linear weighted algorithm

In summary, while both algorithms serve different purposes and operate on distinct principles, the flower pollination algorithm generally offers greater flexibility and robustness for tackling complex optimization problems compared to the linear weighted algorithm.

CHAPTER – IX

RESULTS AND APPLICATIONS

EXECUTABLES

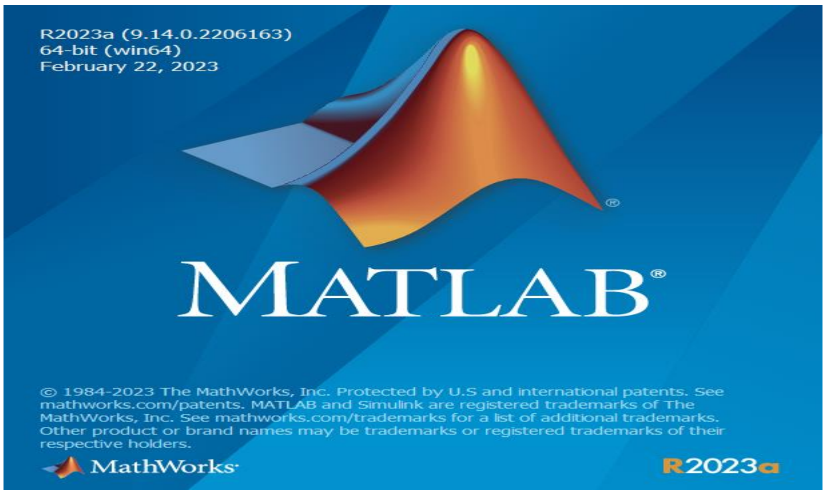
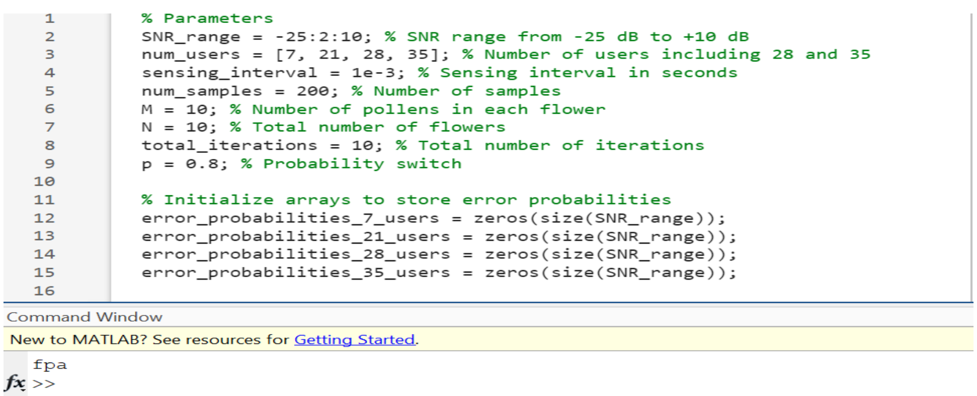


Fig.16 Opening image of Matlab software

FLOWER POLLINATION ALGORITHM



```
% Parameters
SNR_range = -25:2:10; % SNR range from -25 dB to +10 dB
num_users = [7, 21, 28, 35]; % Number of users including 28 and 35
sensing_interval = 1e-3; % Sensing interval in seconds
num_samples = 200; % Number of samples
M = 10; % Number of pollens in each flower
N = 10; % Total number of flowers
total_iterations = 10; % Total number of iterations
p = 0.8; % Probability switch

% Initialize arrays to store error probabilities
error_probabilities_7_users = zeros(size(SNR_range));
error_probabilities_21_users = zeros(size(SNR_range));
error_probabilities_28_users = zeros(size(SNR_range));
error_probabilities_35_users = zeros(size(SNR_range));

Command Window
New to MATLAB? See resources for Getting Started.
  fpa
fx >>
```

Fig.17 Compilation of FPA

Name	Value
calculate_Pf	@(SNR)qfunc...
calculate_P...	@(SNR,num_...
error_prob	0.0040
error_prob...	1x18 double
error_prob...	1x18 double
error_prob...	1x18 double
error_prob...	1x18 double
fitness_sco...	[0.0542,0.391...
i	18

initial_pop...	10x10 double
iter	10
j	4
M	10
N	10
new_popul...	10x10 double
num_sam...	200
num_user	35
num_users	[7,21,28,35]

num_sam...	200
num_user	35
num_users	[7,21,28,35]
p	0.8000
Pf	0.0024
Pm	0.0016
population	10x10 double
sensing_in...	1.0000e-03
SNR	9

Pf	0.0024
Pm	0.0016
population	10x10 double
sensing_in...	1.0000e-03
SNR	9
SNR_range	1x18 double
sorted_fitn...	[0.0251,0.049...
sorted_ind...	[6,5,1,3,2,9,10...
total_iterat...	10

Fig.18 Workspace values of FPA

OUTPUT

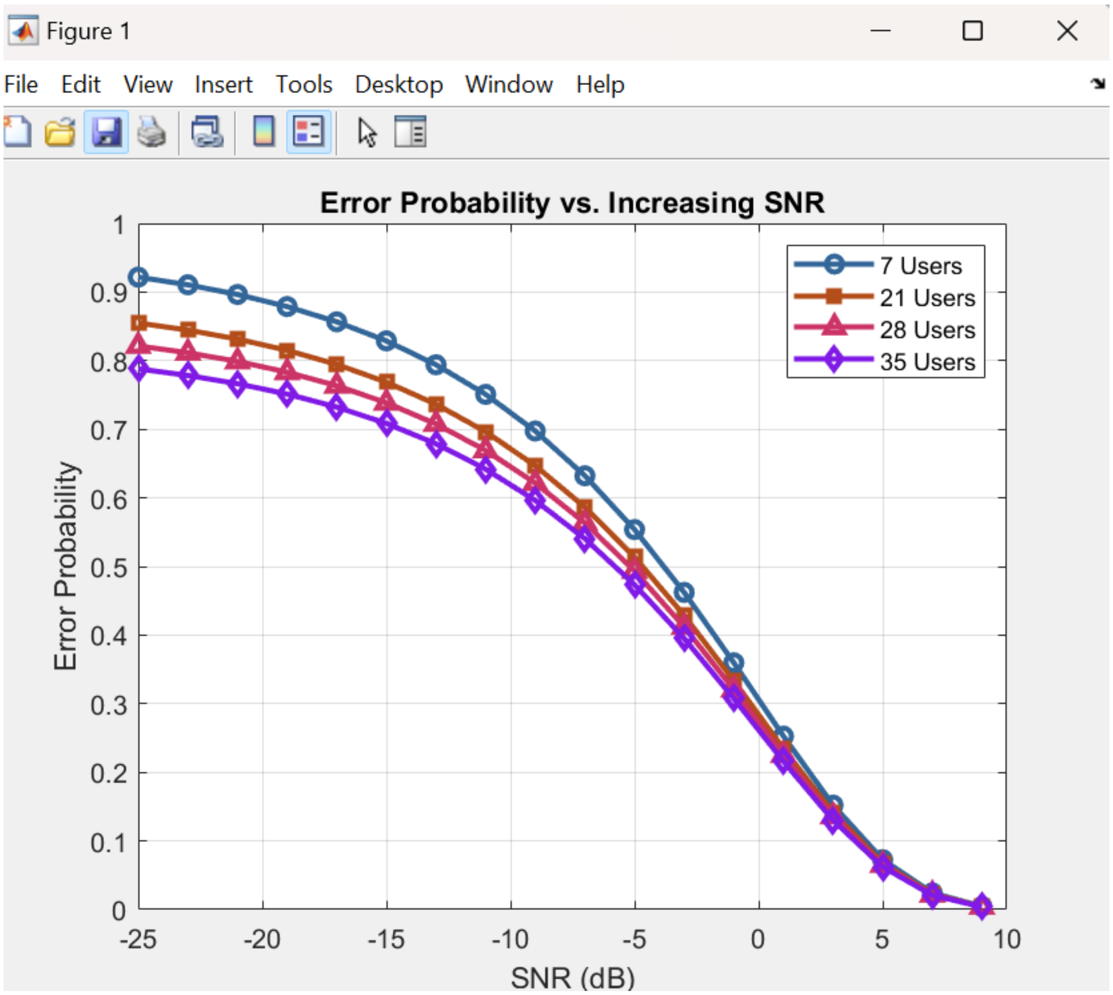
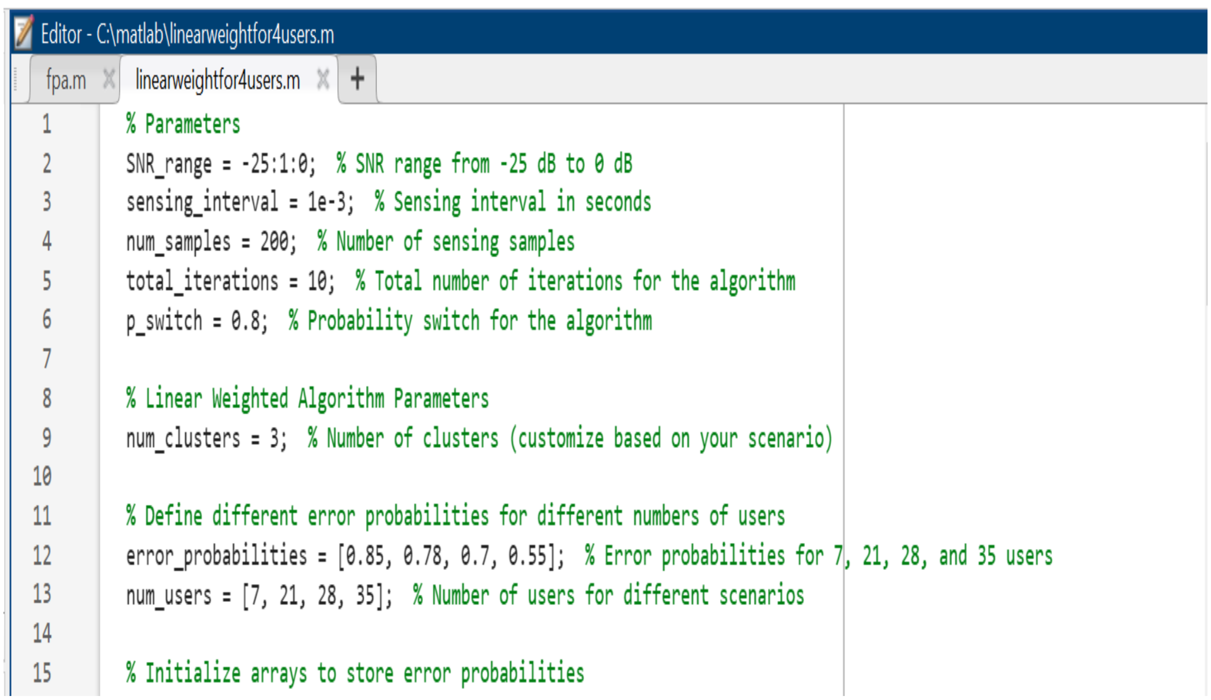


Fig.19 Graphical output of FPA

In the output graph, an analysis is conducted on the relationship between error probability, signal-to-noise ratio (SNR), and varying user counts across four distinct scenarios: 7 users, 21 users, 28 users, and 35 users. As the number of users escalates, a we can see that a progressive decline in error probability compared with a corresponding increase in the signal-to-noise ratio, traversing the spectrum from negative to positive values. This observation serves as a pivotal insight into the dynamics of network performance under increasing user loads. The signal-to-noise ratio is spanning from -25dB to 20dB.. This increase in SNR underscores the network's capacity to maintain flexibility and clarity in data transmission amidst escalating user demands. As the error probability decreases, the reliability of the networks increases and the algorithm can allot channels easily to the vacant users without disturbing the occupied channels.

LINEAR WEIGHTED ALGORITHM



Editor - C:\matlab\linearweightfor4users.m

fpa.m | linearweightfor4users.m

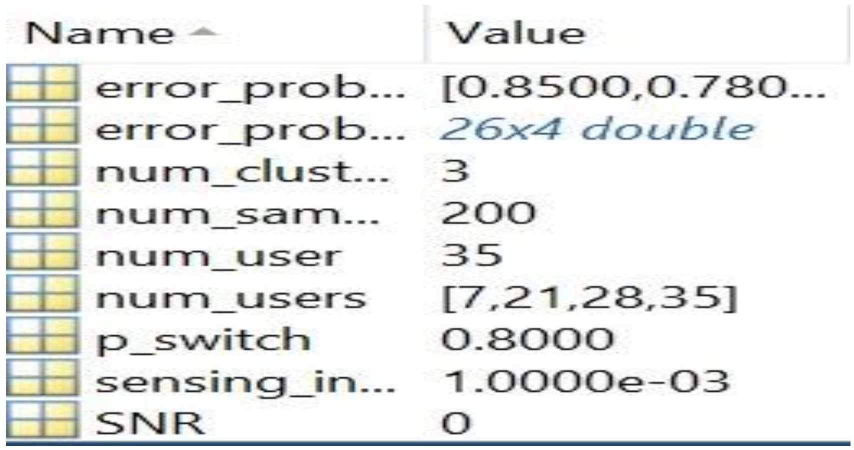
```
% Parameters
SNR_range = -25:1:0;  % SNR range from -25 dB to 0 dB
sensing_interval = 1e-3;  % Sensing interval in seconds
num_samples = 200;  % Number of sensing samples
total_iterations = 10;  % Total number of iterations for the algorithm
p_switch = 0.8;  % Probability switch for the algorithm

% Linear Weighted Algorithm Parameters
num_clusters = 3;  % Number of clusters (customize based on your scenario)

% Define different error probabilities for different numbers of users
error_probabilities = [0.85, 0.78, 0.7, 0.55];  % Error probabilities for 7, 21, 28, and 35 users
num_users = [7, 21, 28, 35];  % Number of users for different scenarios

% Initialize arrays to store error probabilities
```

Fig.20 Compilation of OLWA



Name	Value
error_prob...	[0.8500,0.780...
error_prob...	*26x4 double*
num_clust...	3
num_sam...	200
num_user	35
num_users	[7,21,28,35]
p_switch	0.8000
sensing_in...	1.0000e-03
SNR	0

Fig.21 Workspace values of OLWA

OUTPUT

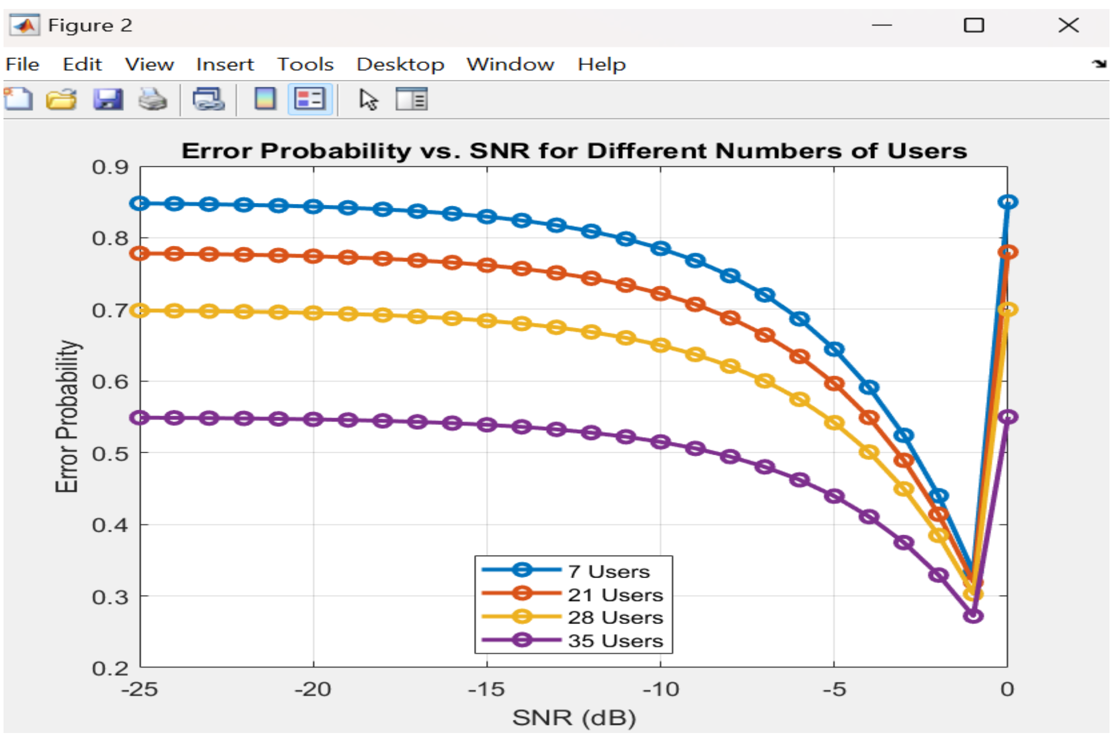


Fig.22 Graphical output of OLWA

In this output graph, the error probability and signal to noise ratio is compared for four different scenarios- 7 users, 21 users, 28 users and 35 users respectively. We can see that as the number of user increases, the error probability decreases. We can see that the error probability has seen a sudden decrease as the number of user increases. This is because the users are allocated based upon the random weights generated and as the weight increases, many users cannot come and occupy the channel. The main difference between the normal linear weighted algorithm and the optimal linear weighted algorithm is that, the optimal linear weighted algorithm is optimized by introducing the cluster heads which are customized (3 in our case) and the number of users are divided into 3 clusters where the cluster having more number of users is given priority.

APPLICATIONS

In scenarios characterized by decreasing error probability with increasing SNR, the implications for real-time applications in CRNs are profound. As SNR rises, the fidelity and reliability of data transmission improve, enabling more accurate detection and decoding of signals even amidst noise and interference. This directly translates to enhanced performance and throughput in cognitive radio systems, making them better equipped to support critical real-time applications such as:

1. **Emergency Communication Systems**: In crisis situations where reliable communication is paramount, the ability of CRNs to operate efficiently at high SNR levels ensures swift and dependable dissemination of information among emergency responders and affected populations.

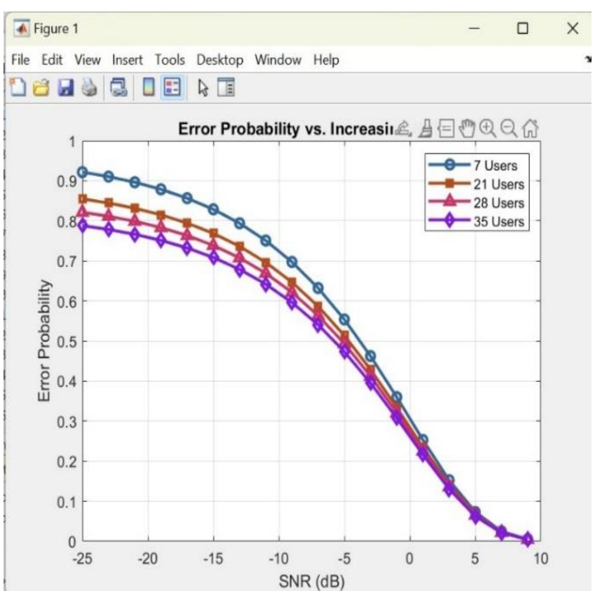
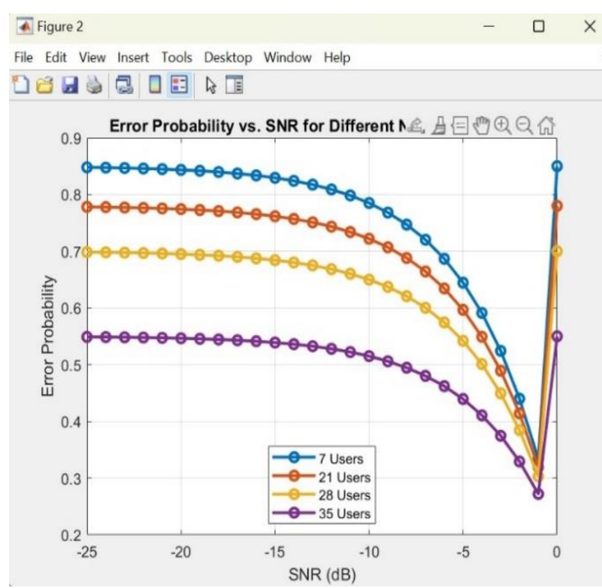
2. **Industrial IoT (IIoT) Applications:** Real-time monitoring and control systems in industrial settings benefit from CRNs' ability to maintain low error probabilities at high SNR levels, enabling seamless data exchange and decision-making processes critical for optimizing operations and ensuring worker safety.

3**. Smart Grid Management**: In smart grid environments, where timely and accurate data transmission is essential for grid stability and energy management, CRNs can leverage decreasing error probabilities at higher SNR levels to facilitate efficient energy distribution, fault detection, and load balancing in real-time.

4. **Healthcare Telemetry**: Remote patient monitoring systems rely on CRNs' ability to transmit medical data with minimal errors, particularly in environments with high SNR levels. This enables healthcare providers to monitor patient vitals and respond promptly to emergent medical conditions, enhancing patient care and outcomes.

CONCLUSION

The Flower Pollination Algorithm (FPA), drawing inspiration from the pollination behavior of flowering plants, and the Optimal Linear Weighted Algorithm (OLWA), a mathematical optimization technique, offer distinct strategies for channel allocation and resource management in Cognitive Radio Networks (CRNs). Upon examining two comparative graphs, it becomes evident that the error probabilities in the Linear Weighted Algorithm closely approximate those in the Flower Pollination Algorithm.

<table>
<tr><th>OUTPUT OF FLOWER POLLINATION ALGORITHM</th><th>OUTPUT OF LINEAR WEIGHTED ALGORITHM</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

For instance, with 7 users in both scenarios, the error probability in FPA stands at 0.92, while in the Linear Weighted Algorithm, it is marginally lower at 0.85. This trend persists across increasing user counts, indicating a convergence of performance between the two algorithms. As the number of users increases, the observed decrease in error probability underscores the network's growing reliability. Interestingly, the Linear Weighted Algorithm demonstrates increasing flexibility, attributed to its optimization through cluster heads.

This adaptability is a significant advantage, suggesting that for many cases, the complexity of algorithms like FPA may be unnecessary, especially when considering the comparable performance and enhanced flexibility offered by OLWA. In real-time scenarios, this comparative analysis between FPA and OLWA enables researchers to discern the algorithm best suited for optimizing spectrum utilization and mitigating errors across varying Signal-to-Noise Ratio (SNR) conditions. By identifying the most effective algorithm for specific applications, CRNs can achieve heightened efficacy and reliability in real-time operations.

FUTURE SCOPE

The domain of this project can be expanded into artificial intelligence. Machine learning techniques can be incorporated to analyze the algorithm’s parameters by increasing the number of users in different scenarios. Also, OWLA can be analyzed for dynamic spectrum allocation since OWLA till now performs fixed channel assessments. As security and privacy concerns still need the usage of complex algorithms such as FPA, new innovations in OWLA can be proposed by developing robust encryption and automation techniques. Extending the comparison to include multi-objective optimization criteria, such as maximizing throughput while minimizing interference or maximizing network lifetime while minimizing energy consumption. This could involve adapting existing algorithms or developing new ones tailored to multi-objective optimization in cognitive radio networks. By researching for advancement and development, the project comparing FPA and OLWA in cognitive radio networks can contribute to the ongoing efforts to improve spectrum efficiency, enhance network performance, and enable the realization of next-generation communication systems.

APPENDIX

FLOWER POLLINATION ALGORITHM:

```
% Parameters
SNR_range = -25:2:10; % SNR range from -25 dB to +10 dB
num_users = [7, 21, 28, 35]; % Number of users including 28 and 35
sensing_interval = 1e-3; % Sensing interval in seconds
num_samples = 200; % Number of samples
M = 10; % Number of pollens in each flower
N = 10; % Total number of flowers
total_iterations = 10; % Total number of iterations
p = 0.8; % Probability switch
```

Figure A.1: Parameter assigning

Figure A.1 represents the parameters that are tested and initialized in a specific scenario for four users where the sensing interval would be 1e-3.

```
% Initialize arrays to store error probabilities
error_probabilities_7_users = zeros(size(SNR_range));
error_probabilities_21_users = zeros(size(SNR_range));
error_probabilities_28_users = zeros(size(SNR_range));
error_probabilities_35_users = zeros(size(SNR_range));
```

Figure A.2: Array initialization

Figure A.2 represents the initializations of arrays to store the values that are obtained after computing the error probability that is occurred after testing the scenarios of 7 users, 21 users, 28 users and 35 users respectively.

```
% Function to calculate Pf (Probability of false alarm)
calculate_Pf = @(SNR) qfunc(sqrt(10^(SNR/10)));

% Function to calculate Pm (Probability of misdetection)
calculate_Pm = @(SNR, num_users) qfunc(sqrt(10^(SNR/10))) * (1 - (num_users / 100));
```

Figure A.3: Calculation of Pf and Pm

Here, in figure A.3, the probability factors of false alarm and misdetection is calculated by using formulas where the allocation of users varies, for example, it is 0.0040 for 35 users.

```
% Simulate for different SNR values
for i = 1:length(SNR_range)
    SNR = SNR_range(i);
    for j = 1:length(num_users)
        num_user = num_users(j);
%Step 1: Initial Population
        initial_population = rand(M, N);
```

Figure A.4: FPA process initialization

Figure A.4 represents the for loop that is used to initialize the population for FPA

```
% Step 2: Fitness of the pollen gametes
fitness_scores = rand(1, N);
[sorted_fitness, sorted_indices] = sort(fitness_scores);

% Step 3: Global and Local Pollination
population = perform_pollination(initial_population, p);

% Step 4: New Population
new_population = generate_new_population(population);

% Step 5: Stopping Criteria
iter = 1;
```

Figure A.5: Steps 2 to 5

In this figure, the2nd step represents the fitness which is randomly generated for the pollen gametes, where the fitness score is used to evaluate the shortest distance available. In the 3rd step, the pollination is decided as local or global based upon the user allocated and the 4th step is used to calculate the the new population of flowers generated after the pollination process. The final step is used to allocate the time at which the iteration must end.

```
while ~check_stopping_criteria(iter, total_iterations)

    % Update population
    population = new_population;

    % Perform global and local pollination
    population = perform_pollination(population, p);

    % Generate new population
    new_population = generate_new_population(population);

     % Increment iteration counter
    iter = iter + 1;
end
```

Figure A.6: While loop for updation

The while loop is used to perform functions to update the initial population to new population based upon the number of users arrived.

```
% Calculate error probabilities based on Pf and Pm
Pf = calculate_Pf(SNR);
Pm = calculate_Pm(SNR, num_user);
error_prob = Pf + Pm;
```

Figure A.7: Error probability calculation

In this figure, the error probability is calculated based upon the sum of the probabilities of false alarm and misdetection respectively.

```
        % Store error probabilities
        switch num_user
            case 7
                error_probabilities_7_users(i) = error_prob;
            case 21
                error_probabilities_21_users(i) = error_prob;
            case 28
                error_probabilities_28_users(i) = error_prob;
            case 35
                error_probabilities_35_users(i) = error_prob;
        end
    end
end
```

Figure A.8: Storage of error probabilities

In this figure, the error probabilities calculated previously (figure A.7) is saved in arrays that are declared in the figure A.2.

```
% Plotting for 7 users
plot(SNR_range, error_probabilities_7_users, '-o', 'LineWidth', 2, 'Color', [0.2 0.4 0.6]);
hold on;

% Plotting for 21 users
plot(SNR_range, error_probabilities_21_users, '-s', 'LineWidth', 2, 'Color', [0.7 0.3 0.1]);

% Plotting for 28 users
plot(SNR_range, error_probabilities_28_users, '-^', 'LineWidth', 2, 'Color', [0.8 0.2 0.4]);

% Plotting for 35 users
plot(SNR_range, error_probabilities_35_users, '-d', 'LineWidth', 2, 'Color', [0.5 0.1 0.9]);
```

```
xlabel('SNR (dB)');
ylabel('Error Probability');
title('Error Probability vs. Increasing SNR');
grid on;
legend('7 Users', '21 Users', '28 Users', '35 Users');
ylim([0 1]);
```

Figure A.9: Plotting function declaration

Here, the plotting functions are declared in which the error probabilities are plotted. The x axis, y axis and the legends are declared and the limits are in the range of 0,1.

```
 %Function to check stopping criteria
function stop = check_stopping_criteria(iter, total_iterations)
    stop = (iter >= total_iterations); % Stop if iteration limit reached
end
% Function to perform global and local pollination
function new_population = perform_pollination(population, p)
    % Apply global pollination with probability p
    global_pollination = population + p * (rand(size(population)) - 0.5); % Example of global pollination
    % Apply local pollination with probability (1-p)
    local_pollination = population + (1 - p) * (rand(size(population)) - 0.5); % Example of local pollination
    % Combine global and local pollination
    new_population = (global_pollination + local_pollination) / 2;
end
```

Figure A.10: functions for stopping criteria and pollination

The coding in the above figure can be explained in 2 steps:

- In the first part, the stopping criteria is used to determine the limit of iterations. In our project, the lint is set to 1.
- In the second part of coding, the new population calculated is brought and the decision making based upon the shortest path and the user allocation of whether local pollination pr global pollination can be done is analysed using formulas and the decision is made whether local pollination or global pollination can be made.

```
% Function to generate new population
function new_population = generate_new_population(population)
    new_population = population;
end
```

Figure A.11 Final population generation

The new population generated after the fitness score calculation and error probability calculation in the previous figures (figures A.7, A.8, A.10 respectively).

OUTPUT PARAMETERS OF FLOWER POLLINATION ALGORITHM:

Name	Value
calculate_Pf	@(SNR)qfunc...
calculate_P...	@(SNR,num_...
error_prob	0.0040
error_prob...	*1x18 double*
error_prob...	*1x18 double*
error_prob...	*1x18 double*
error_prob...	*1x18 double*
fitness_sco...	[0.0542,0.391...
i	18

Figure A.12: Probability of False Alarm and Misdetection

The probability of false alarm and misdetection are calculated and the error probability is based upon the allocation of users which is 0.0040 in this case since the probability of false alarm is lesser and misdetection is not possible due to the intelligent detection of the cognitive radio network. The fitness score is used to analyse the suitability of the users on the basis of their coefficient vectors. The population of users is allocated based upon the fitness score.

Name	Value
initial_pop...	10x10 double
iter	10
j	4
M	10
N	10
new_popul...	10x10 double
num_sam...	200
num_user	35
num users	[7,21,28,35]

Figure A.13: User interaction

The initial population is set to be 100 where 100 secondary users are allocated for the example. Even more than 100 can be used. 10 iterations are done to allocate the users and process the algorithm based upon the number of users allocated by the clusters. Here, 4 different users 7,21,28,35 users are analysed and implemented. Here, for 35 users, the samples parameter is shown.

Workspace

Name	Value
Pf	0.0024
Pm	0.0016
population	10x10 double
sensing_in...	1.0000e-03
SNR	9
SNR_range	1x18 double
sorted_fitn...	[0.0251,0.049...
sorted_ind...	[6,5,1,3,2,9,10...
total_iterat...	10

Figure A.14: Allocation of users

Here, the path allocated to the user that is present inside the majority of users of a cluster is identified using short distance. The sensing interval of the users is at 1e-3. The fitness scores and the indices are calculated and sorted according to the distance in a period of 10 iterations.

LINEAR WEIGHTED ALGORITHM:

```
% Parameters
SNR_range = -25:1:0;  % SNR range from -25 dB to 0 dB
sensing_interval = 1e-3;  % Sensing interval in seconds
num_samples = 200;  % Number of sensing samples
total_iterations = 10;  % Total number of iterations for the algorithm
p_switch = 0.8;  % Probability switch for the algorithm

% Linear Weighted Algorithm Parameters
num_clusters = 3;  % Number of clusters
```

Figure A.15: Parameter initialization

Figure A.15 represents the parameters that are tested and initialized in a specific scenario for four users where the sensing interval would be 1e-3.

```
% Define different error probabilities for different numbers of users
error_probabilities = [0.85, 0.78, 0.7, 0.55];
num_users = [7, 21, 28, 35];  % Number of users for different scenarios

% Initialize arrays to store error probabilities
error_probabilities_all_users = zeros(length(SNR_range), length(num_users));
```

Figure A.16: Error probabilities

Figure A.16 represents the initialization of error probabilities based upon the number of users(7,21,28,35 in our scenario) and the initialization is done with 0s since an empty array would store garbage values.

```matlab
% Loop over SNR values
for snr_index = 1:length(SNR_range)
    SNR = SNR_range(snr_index);

    % Loop over different number of users
    for user_index = 1:length(num_users)
        num_user = num_users(user_index);

         % Calculate error probability based on SNR and number of users
         if SNR < 0
             % SNR is less than 0 dB
             error_probabilities_all_users(snr_index, user_index) = 0.2 + (error_probabilities(user_index) - 0.2)
         else
             % SNR is greater than or equal to 0 dB
             error_probabilities_all_users(snr_index, user_index) = error_probabilities(user_index);
         end
     end
 end
```

Figure A.17: Calculation of error probabilities

In this figure, the length of the shortest path based upon the number of users and the SNR range is calculated and the error probability is determined.

```matlab
% Plotting for different number of users
figure;
for user_index = 1:length(num_users)
    plot(SNR_range, error_probabilities_all_users(:, user_index), 'o-', 'LineWidth', 2);
    hold on;
end
xlabel('SNR (dB)');
ylabel('Error Probability');
title('Error Probability vs. SNR for Different Numbers of Users');
legend('7 Users ', '21 Users ', '28 Users ', '35 Users', 'Location', 'best');
grid on;
hold off;
```

Figure A.18: Plotting the parameters

Here, the plotting functions are declared in which the error probabilities are plotted. The x axis, y axis and the legends are declared and the error probability and the corresponding SNR values are plotted.

OUTPUT PARAMETER ANALYSIS FOR LINEAR WEIGHTED ALGORITHM:

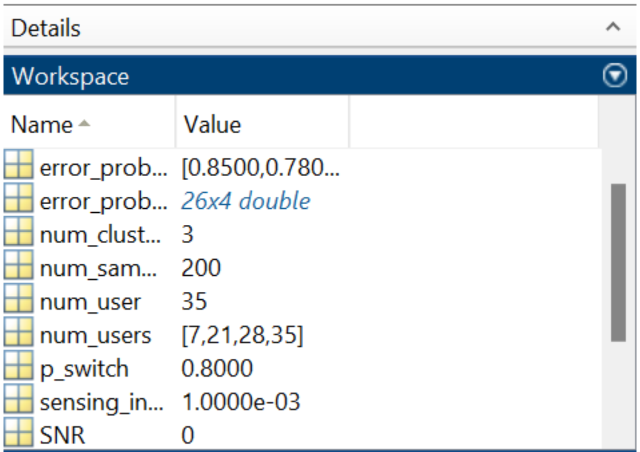


Figure A.19: parameters in workbench

Here, we can see that the error probability is calculated based upon the number of clusters and here the switch probability is assigned to be 0.8. the sensing interval is same as FPA, since the algorithm is optimized. As the number of users increases, the signal to noise ratio decreases and the last point of SNR occurance will be 0.

REFERENCES

1. S. Haykin "cognitive radio: brain-empowered wireless communication," IEEE J. Set. Areas communication. vol.23, no2, pp.201-220, 2005.

2. C.-H. Hwang, G.-L. Lai, and S.-C. Chen, “Spectrum sensing in wideband OFDM cognitive radios,” IEEE Trans. Signal Process., vol. 58, pp. 709– 719, Feb. 2010.

3. O.B. Akan, O.B. Karli, O. Ergul, Cognitive radio sensor networks. IEEE Netw. 23(4), 34–40 (2018).

4. H. Rif-Pous, M.J. Blasco, C. Garrigues, Review of robust cooperative spectrum sensing techniques for cognitive radio networks. Wirel. Pers. Commun. 67(2), 175–198 (2012).

5. Y. Chen, H. Oh, A survey of measurement-based spectrum occupancy modeling for cognitive radios. IEEE Commun. Surv. Tuts 18(1), 848–859 (2016).

6. G. Manish, Y. Gaurav, Improved weighted cooperative spectrum sensing algorithm based on reliability in cognitive radio networks. In: Proceedings of 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology, pp. 609–612 (2016)

7. A.A. El-Saleh, M. Ismail, M.A.M. Ali, I.H. Arka, Hybrid SDF-HDF cluster-based fusion scheme for cooperative spectrum sensing in cognitive radio networks. KSII Trans. Internet Inf. Syst. 4(6), 1023–1041 (2010).

8. N. Gul, I. Mansoor, Q. Aqdas, N. Atif, and E. Imtiaz, "Secured Soft Combination Schemes Against Malicious Users in Cooperative Spectrum Sensing," Wirel. Pers. Commun., no. 0123456789, 2019.

9. M. Akbari, M. R. Manesh, S. A. R. T. Zavareh, and P. Shahabi, "Maximizing the Probability of Detection of Cooperative Spectrum Sensing in Cognitive Radio Networks," Prog. Electromagn. Res. Symp. Proceedings, KL, MALAYSIA, pp. 123–126, 2012.

10. M. Akbari, M. R. Manesh, A. A. El-Saleh, and M. Ismail, "Improved soft fusion-based cooperative spectrum sensing using particle swarm optimization," IEICE Electron. Express, vol. 9, no. 6, pp. 436–442, 2012.
X. Yang, Nature-Inspired Optimization Algorithms.

11. A. A. . El-Saleh, M. . Ismail, M. A. M. . Ali, and M. K. . Hossain, "Biologically-inspired soft fusion scheme for cooperative spectrum sensing in cognitive radio networks," 7th Int. Conf. Inf. Technol. Appl. ICITA 2011, no. Icita, pp. 239–244, 2011.

12. A. Jovicic and P. Viswanath, "Cognitive radio: An information-the oretic perspective," IEEE Trans. Inf. Theory, vol. 55, no. 9, pp. 3945–3958, Sep. 2009.

13. O.Simeone, I.Stanojev,S.Savazzi, Y. Bar-Ness, U. Spagnolini, and R. Pickholtz, “Spectrum leasing to cooperating secondary ad hoc net works,” IEEE J. Sel. Areas Commun., vol. 26, no. 1, pp. 203–213, Jan. 2008.

14. M.A. McHenryand D.McCloskeyk, “Spectrum occupancy measurements:Chicago, IL,Nov.16–18,2005,”SharedSpectrumCo.,Vienna, 904 VA,2005,SharedSpectrumCo.Rep.

15. J. Mitola and G. Q. Maguire, “Cognitive radio: Makeing software radios more personal,” IEEE Pers. Commun., vol. 6, no. 6, pp. 13–18, Aug. 1999.

16. A. Ghasemi and E. S. Sousa, “Fundamental limits of spectrum-sharing in fading environments,” IEEE Trans. Wireless Commun., vol. 6, no. 2, pp. 649–658, Feb. 2007.

17. Federal Communications Commission, “Notice of proposed rule mak ing and order: Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies,” ET Docket No. 03-108, Feb. 2005.

18. S. Geirhofer, L. Tong, and B. Sadler, “Dynamic spectrum access in the time domain: Modeling and exploiting white space,” IEEE Commun. Mag., vol. 45, no. 5, pp. 66–72, May 2007.

19. P. Wang, L. Xiao, S. Zhou, and J. Wang, “Optimization of detection time for channel efficiency in cognitive radio systems,” in Proc. IEEE Wireless Commun. and Networking Conf., Hong Kong, Mar. 2007, pp. 111–115.

20. “Cooperative spectrum sensing in cognitive radio networks,” in Proc. IEEE Int. Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore, Maryland, USA, Nov. 2005, pp. 137–143.

21. A. Ghasemi and E. Sousa, “Optimization of spectrum sensing for opportunistic spectrum access in cognitive radio networks,” in Proc. IEEE Consumer Commun. and Net.

22. D. Datla, R. Rajbanshi, A. M. Wyglinski, and G. J. Minden, “Paramet ric adaptive spectrum sensing framework for dynamic spectrum access networks,” in Proc. IEEE Int. Symposium on New Frontiers in Dynamic Spectrum Access Networks, Dublin, Ireland, Apr. 2007, pp. 482–485.

23. S. Mishra, A. Sahai, and R. Brodersen, “Cooperative sensing among cognitive radios,” in Proc. IEEE Int. Conf. Commun.,vol.2, Istanbul, Turkey, May 2006, pp. 1658–1663.

24. T. Y¨ ucek and H. Arslan, “Spectrum characterization for opportunistic cognitive radio systems,” in Proc. IEEE Military Commun. Conf., Washington, D.C., USA, Oct. 2006, pp. 1–6.

25. J. Mitola, “Cognitive radio-An integrated agent architecture for software-defined radio,” Ph.D. dissertation, Roy. Inst. Technol., Stock holm, Sweden, May, 2000.

26. Q.Zhao and B.M.Sadler,“ A survey of dynamic spectrum access,” IEEE Signal Process. Mag., vol. 24, no. 3, pp. 79–89, May 2007.

27. X. Y. Wang, A. Wong, and P. H. Ho, “Extended knowledge-based rea soning approach to spectrum sensing for cognitive radio,” IEEE Trans. Mobile Comput., vol. 9, no. 4, pp. 465–478, Apr. 2010.

28. P. Mähönen, M. Petrova, J. Riihijärvi, and M. Wellens, “Cognitive wire less networks: Your network just became a teenager,” in Proc. 26th IEEE INFOCOM, Poster Session, 2006.

29. D. Niyato and E. Hossain, “Competitive spectrum sharing in cogni tive radio networks: A dynamic game approach,” IEEE Trans. Wireless Commun., vol. 7, no. 7, pp. 2651–2660, Jul. 2008.

Vision and Mission of the Institute

Vision:

To impart state-of-the art technical education, including sterling values and shining character, producing engineers who contribute to nation building thereby achieving our ultimate objective of sustained development of an unparalleled society, nation and world at large.

Mission:

Meenakshi Sundararajan Engineering College, Chennai constantly strives to be a Centre of Excellence with the singular aim of producing students of outstanding academic excellence and sterling character to benefit the society, our nation and the world at large.

To achieve this, the college ensures

- Continuous up gradation of its teaching faculty to ensure a high standard of quality Education and to meet the ever-changing needs of the society.
- Constant interaction with its stakeholders.
- Linkage with other educational institutions and industries at the national and international level for mutual benefit.
- Provision of research facilities and infrastructure in line with global trends.
- Adequate opportunities and exposure to the students through suitable programs, to mould their character and to develop their personality with an emphasis on professional ethics and moral values.

Vision and Mission of the Department

Vision:

To emerge as a centre of excellence in offering quality education to produce students technically competent, socially responsible and industry ready graduates in electronics and communication engineering.

Mission:

The above vision will be achieved by

M1: Ensuring effective teaching learning methodologies.

M2: Inculcating creative thinking through innovative and group work exercises.

M3: Developing and motivating research ability among students by establishing research linkage with leading industries.

M4: Equipping faculty and students with the latest developments in Electronics and Communication and to face the challenges.

Program Educational Objectives (PEOs)

PEO1: To provide appropriate knowledge in applying the concepts of basic electronics & Communication engineering.

PEO2: To be able to identify, analyse and solve engineering problems in the field of electronics & communication engineering.

PEO3: To provide an opportunity to work in multidisciplinary groups and research environments.

PEO4: To educate and inculcate professional ethics, human values, self-confidence and awareness of societal needs.

PEO5: To motivate the students in intellectual pursuits, lifelong learning in order to develop different perspectives of technological developments.

Program Outcomes (POs)

PO1: Engineering Knowledge Apply the knowledge of Engineering Mathematics, Basic Sciences, Engineering Fundamentals, and Engineering Specialization to the solution of complex Information Science and Engineering problems.

PO2: Problem analysis Identify, formulate, review research literature, and analyse complex engineering problems of Information Science and Engineering reaching substantiated conclusions using first principles of Engineering Mathematics and Engineering Sciences.

PO3: Design/development of solutions Design solutions for complex Information Science problems and design system components or processes of Information Science and Engineering that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. **PO4: Conduct investigations of complex problems** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions in Information Science and Engineering.

PO5: Modern tool usage Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations in Information Science and Engineering.

PO6: The engineer and society Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice in Information Science and Engineering.

PO7: Environment and sustainability understand the impact of the professional engineering solutions in Information Science and Engineering in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics Apply ethical principles and commit to professional ethics and responsibilities and norms of the Information Science and Engineering practice.

PO9: Individual and team work Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication Communicate effectively on complex Information Science engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning Recognise the need for and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Demonstrate principles of basic electronic circuits, digital electronics, microprocessor and signal processing.

PSO2: Design systems for applications in the areas of communication, networking and embedded systems.

PSO3: Design low cost quality, energy efficient and eco-friendly products.

Course Outcomes

EC 8811/ C413 PROJECT WORK

TITLE OF PROJECT: COMPARITIVE STUDY ON FLOWER POLLINATION ALGORITHM AND OPTIMAL LINEAR WEIGHTED ALGORITHM IN COGNITIVE RADIO NETWORK

At the end of the course, the students will be able to:

C413.1	Understand the concept of CNN architectures.
C413.2	Apply the concept of CNN in detecting malarial cells.
C413.3	Create the code for malaria detection using CNN architectures in deep learning.
C413.4	Analyze the results obtained from the training process.
C413.5	Evaluate the malaria-infected cells and uninfected cells using trained model.

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	POI0	PO11	PO12
C413.1	3	3	2	3	3	2	2	3	3	3	3	2
C413.2	3	3	3	2	3	2	2	2	3	3	3	2
C413.3	3	2	3	3	3	2	2	3	2	2	2	3
C413.4	3	2	2	2	3	2	2	2	3	2	3	2
C413.5	3	3	2	2	2	3	2	3	3	3	2	2
C413	3	2.6	2.4	2.4	2.8	2.2	2.0	2.6	2.8	2.6	2.6	2.2

CO-PSO Mapping

CO	PSO1	PSO2	PSO3
C413.1	1	2	2
C413.2	2	2	1
C413.3	2	2	2
C413.4	2	1	1
C413.5	2	2	2
C413	1.8	1.8	1.6

CO-PO Calculation

BATCH	CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3		REG NO.	INTERNAL	AURESULT	CREDIT
	C413.1	3	3	2	3	3	2	2	3	3	3	3	2	1	2	2	DIRECT ATT CALCULATION	6068			
	C413.2	3	3	3	2	3	2	2	2	3	3	3	2	2	2	1		6080			
	C413.3	3	2	3	3	3	2	2	3	2	2	2	3	2	2	2		6105			
	C413.4	3	2	2	2	3	2	2	2	3	2	3	2	2	1	1		**Target**			
	C413.5	3	3	2	2	2	3	2	3	3	3	2	2	2	2	2		**% of Students securing >target**			
	C413	3	2.6	2.4	2.4	2.8	2.2	2.0	2.6	2.8	2.6	2.6	2.2	1.8	1.8	1.6		**Attainment Level**			
	CO-POATT																	**Attainment (20% Internal & 80% University)**			