CHAPTER 4: FUZZY LOGIC BASED UNEQUAL CLUSTERING HIERARCHY IN WSN

4.1 Introduction to Fuzzy logic:
Classical binary valued logic accepts the values in only two states. The value in-between these extreme ends cannot be represented in it. Also it does not admit the imprecision in truth. In engineering problems, building and defining a system with high precision will be requiring high cost and time. In classical set theory approach, for an element only one membership is given (i.e.) the element belongs to one set only. But our human reasoning supports various level of membership to the element. The element can belong to various sets at a time with varying membership values.

This basic human way of reasoning leads to the development of Fuzzy logic by Zadeh [94] in the year 1965. A fuzzy set contains the element which can be in other fuzzy set also but with varying membership values. For example, a student may be a member of set excellent with a degree of 0.7. Some of the advantages of fuzzy logic [95] are as follows.

- Fuzzy logic (FL) can approximate system behaviors well enough where there is no proper analytic model.
- Some conventional systems need only approximate results rather than highly accurate results for its immediate action. In those systems, fuzzy logic will play a major role in getting results much faster than other tedious time consuming techniques.
- Defining a very complex systems with too many inputs and outputs using the conventional analytical is hardly possible. These kinds of complex systems which are closer to human reasoning can be defined well using fuzzy logic.
• Uncertainties in a complex system cannot be captured using analytical model of it. These uncertainties in input, output, etc. of a system can be well captured using fuzzy logic.

4.1.1 Terminologies in Fuzzy Logic:

The following terms are important for understanding the fuzzy logic mechanism.

a) **Fuzzy linguistic variables:** It is the variables whose values are not numerical numbers but words like in a natural human language [96]. These values are less specific than numbers and more suitable to express uncertainties in the numerical values.

b) **Membership function:** Represents the degree of membership to a particular fuzzy set. Some of the commonly used membership functions are triangular membership, trapezoidal membership, Gaussian membership functions, etc.

c) **Fuzzification:** Process of converting crisp input values into the appropriate fuzzy linguistic variables using the given membership functions.

d) **Defuzzification:** Process of converting output fuzzy linguistic variable into crisp output value.

e) **Fuzzy Inference system:** A system which maps the input into corresponding output using fuzzy set theory. Mamdani [97] and Sugeno [98] are the most widely used FIS.

f) **Rule base:** Set of rules made of if antecedent and then consequent. (e.g.) if X then Y, where X and Y are fuzzy linguistic variables.
4.1.2 Fuzzy Inference System:

FL mainly consists of four significant parts as shown in Figure 4.1. It consists of a Fuzzifier, an inference system, a rule base and a Defuzzifier. The input values are usually crisp, which again converted into appropriate fuzzy linguistic variables. The fuzzified values are finally sent to Fuzzy Decision Block (FDB). FDB is made up of Inference system and Rule Base. It provides fuzzy output based on the rules. Then the fuzzy output is converted into crisp output using defuzzifier.

![Diagram of General fuzzy inference system](image)

Figure 4.1: General fuzzy inference system

4.2 Fuzzy Logic in WSN:

Fuzzy logic is being used in Wireless Sensor Network (WSN) for solving various challenges [99] like design, deployment, security, routing, clustering, data aggregation and sensor fusion. Since fuzzy logic works with reasonable number of linguistic variables it can easily process the input and output of a sensor node with less energy consumption.
Also it is found to be the most desirable technique for the energy constrained distributed WSN.

4.3 Distributed Unequal Clustering using Fuzzy approach (DUCF):

The proposed fuzzy based clustering algorithm DUCF increases the network lifetime of the WSN by forming appropriate sized clusters to balance energy consumption among the clusters. Residual energy, node degree and distance to Base Station (BS) are the three input variables used to elect the Cluster Head (CH) nodes.

- **Residual Energy**: Generally, a CH node has to spend more energy than a member node. CH involves in data collection, aggregation and transmission of aggregated data to the BS. So a CH is expected to be in higher energy level for executing all the above activities.

- **Node degree**: Node degree is the number of neighbor nodes for a particular node. The higher node degree for a CH reduces the intra cluster distance for a cluster.

- **Distance to BS**: Distance to BS is a very important factor in controlling hot spot problem which is prevalent in unequal clustering. The nearby CHs to the BS should have less number of member nodes, so that the conserved energy can be spent for relaying data from distant CHs to BS. The distant CHs may accommodate more number of members compared to nearby CHs.

DUCF has two output fuzzy linguistic variables: (i) chance and (ii) size.

- **Chance**: The ability of a node to take responsibility as CH is represented as chance.
• **Size**: Size represents the maximum number of member nodes can be accepted by a CH. DUCF assigns size based on the input variables residual energy, node degree and distance to BS.

As depicted in Figure 4.2, DUCF has two phases: (i) Cluster formation phase and (ii) Data collection phase.

![Figure 4.2: DUCF operational diagram](image)

**4.3.1 Cluster Formation phase**: 

The unequal clusters are formed at the end of cluster formation phase. The CH nodes are elected using the fuzzy if-then rules. Cluster formation phase is again divided into two sub phases, *CH election sub phase* and *Cluster building sub phase*.

**4.3.1.1 CH Election sub phase**: 

During the start of the CH election sub-phase all the nodes will be in the Probationary CH state. The input variables will be converted into appropriate fuzzy linguistic variables and this process is termed as fuzzification. The first input variable, residual energy has three fuzzy linguistic variables: *Low, Medium and High*. The second input variable, node
degree has *Enormous, Average* and *Less* as fuzzy linguistic variables. The third input variable, distance to BS has *Distant, Reachable and Nearby* as fuzzy linguistic variables. Trapezoidal membership function is used for the linguistic variables *Low, Less, Nearby, High, Enormous, Distant* and triangular membership functions for the variables *Medium, Average* and *Reachable*. Figure 4.3, Figure 4.4 and Figure 4.5 depict the membership function for the input variables residual energy, node degree and distance to BS respectively.

![Figure 4.3: Membership function for Residual Energy](image-url)
Very High, High, Rather High, High Medium, Medium, Low Medium, Rather Low, Low and Very Low are the nine fuzzy linguistic variables for the output variable chance.
Trapezoidal membership function is used for linguistic variables *Very High* and *Very Low*. Triangular membership function is used for all the remaining seven output linguistic variables. Figure 4.6 depicts the membership functions for chance.

![Membership Function for output variable ‘Chance’](image1)

Figure 4.6: Membership Function for output variable ‘Chance’

![Membership function for output variable ‘Size’](image2)

Figure 4.7: Membership function for output variable ‘Size’

Seven fuzzy linguistic variables like *Very small, Small, Rather Small, Medium, Rather big, Big and Very Big* are used for second output variable size. *Very Small, Very Big* takes trapezoidal membership function and triangular membership function for remaining
five linguistic variables. The ranges for the membership functions are chosen carefully based on iterative experimental analysis. These ranges ensure that all the nodes will be a part of clustering architecture. Figure 4.7 depicts the degree of membership for size.

DUCF’s Fuzzy Inference System (FIS) converts the given crisp input values into appropriate fuzzy linguistic variables using the membership functions. The fuzzified input variables are evaluated through the fuzzy if-then rule base. The fuzzy if-then rules are developed based on Mamdani [97] method. Mamdani method is used for its simplicity and efficiency. In total, 27 rules are there based on the combination of the three different input linguistic variables. DUCF fuzzy if-then rules is mentioned in Table 4.1. The output given by FIS is again a fuzzy linguistic variable. Center of Area (CoA) method is used to defuzzify the output to a crisp value of chance and size.
Table 4.1: DUCF Fuzzy if-then rules

<table>
<thead>
<tr>
<th>S.No</th>
<th>Input Variables</th>
<th>Output Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Residual Energy</td>
<td>Node degree</td>
</tr>
<tr>
<td>1</td>
<td>High</td>
<td>Enormous</td>
</tr>
<tr>
<td>2</td>
<td>High</td>
<td>Enormous</td>
</tr>
<tr>
<td>3</td>
<td>High</td>
<td>Enormous</td>
</tr>
<tr>
<td>4</td>
<td>High</td>
<td>Average</td>
</tr>
<tr>
<td>5</td>
<td>High</td>
<td>Average</td>
</tr>
<tr>
<td>6</td>
<td>High</td>
<td>Average</td>
</tr>
<tr>
<td>7</td>
<td>High</td>
<td>Less</td>
</tr>
<tr>
<td>8</td>
<td>High</td>
<td>Less</td>
</tr>
<tr>
<td>9</td>
<td>High</td>
<td>Less</td>
</tr>
<tr>
<td>10</td>
<td>Medium</td>
<td>Enormous</td>
</tr>
<tr>
<td>11</td>
<td>Medium</td>
<td>Enormous</td>
</tr>
<tr>
<td>12</td>
<td>Medium</td>
<td>Enormous</td>
</tr>
<tr>
<td>13</td>
<td>Medium</td>
<td>Average</td>
</tr>
<tr>
<td>14</td>
<td>Medium</td>
<td>Average</td>
</tr>
<tr>
<td>15</td>
<td>Medium</td>
<td>Average</td>
</tr>
<tr>
<td>16</td>
<td>Medium</td>
<td>Less</td>
</tr>
<tr>
<td>17</td>
<td>Medium</td>
<td>Less</td>
</tr>
<tr>
<td>18</td>
<td>Medium</td>
<td>Less</td>
</tr>
<tr>
<td>19</td>
<td>Low</td>
<td>Enormous</td>
</tr>
<tr>
<td>20</td>
<td>Low</td>
<td>Enormous</td>
</tr>
<tr>
<td>21</td>
<td>Low</td>
<td>Enormous</td>
</tr>
<tr>
<td>22</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>23</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>24</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>25</td>
<td>Low</td>
<td>Less</td>
</tr>
<tr>
<td>26</td>
<td>Low</td>
<td>Less</td>
</tr>
<tr>
<td>27</td>
<td>Low</td>
<td>Less</td>
</tr>
</tbody>
</table>
4.3.1.2 State Transformations in a Node:

The sensor nodes will be in any one of the below mentioned states during the execution of DUCF algorithm,

(i) Initial state

(ii) Probationary CH state

(iii) Member state

(iv) Final CH state

If there is no higher chance value for the neighboring nodes within the radius $R_c$, the Probationary CH node enters Final CH state or otherwise it withdraws from CH competition and enters Member node state. Figure 4.8 depicts the state flow. During the starting of next re-clustering process, again all the nodes come back to the Initial state.

4.3.1.3 Cluster Building sub phase:

All the probationary CH nodes calculate its chance to act as CH node using its FIS. The calculated chance value is then broadcasted to all the neighbor nodes within

![Figure 4.8: Different states of a node](image-url)
communication radius $R_c$ using $CH_{CANDIDATE}$ message. Only the details like chance value and node id will be mentioned in $CH_{CANDIDATE}$. Among the neighbors, the probationary CH node having higher chance value is elected as final CH node. Then the newly elected final CH nodes will broadcasts its election through $CH_{WON}$ message in its communication range $R_c$. Since the algorithm is executed in distributed fashion, a probationary CH node may receive more than one $CH_{WON}$ message. But the probationary CH node will send $CM_{JOIN}$ message to the nearest final CH node. In the case of certain probationary CH, it will receive only one $CH_{WON}$, so these nodes will send $CM_{JOIN}$ message only to that concerned final CH node. On receiving the $CM_{JOIN}$ messages, the final CH will check its size (i.e.) second fuzzy output variable for accepting the member nodes. If the number of already existing members is less than the crisp value of size, the new members can be allowed to join the cluster otherwise the request have to be rejected. $CM_{ACCEPTANCE}$ message is used for giving permission to join the cluster whereas $CM_{REJECTION}$ message to indicate no space for the new member node. In $CM_{ACCEPTANCE}$ message the spread code of the respective cluster is attached. The spread code is used for avoiding inter cluster interference in the network. The nodes receiving $CM_{REJECTION}$ message will try with next nearby CHs by sending $CM_{JOIN}$ message till joining a final CH node within communication radius $R_c$. In the worst case, when a non-CH node cannot join any CH within its $R_c$, it elects itself as final CH. This mechanism ensures that no nodes are left to be away from the cluster framework.
4.3.1.4 Message Complexity in Clustering:

If there are \( N \) number of nodes in the network, \( N \) number of CH_CANDIDATE message broadcasts will be there. Out of the \( N \), \( M \) number of nodes gets elected as final CH and broadcasts its election through \( M \) number of messages. The probationary CH node tries to join nearby final CH by sending \( N-M \) number of CM_JOIN messages. Corresponding to CM_JOIN messages the final CH responds with CM_ACCEPTANCE or CM_REJECTION which is also equal to \( N-M \).

\[
N + M + 2p(N - M) \quad \text{where } p \leq q
\]  

where \( p \) is the number of join request issued by a member node and \( q \) is the number of final CH nodes within the communication radius \( R_c \) of the member node. Equation (4.2) is the expansion of Equation (4.1)

\[
N + M + 2pN - 2pM
\]  

\[
T = (1 + 2p)N + (1 - 2p)M = O(N)
\]

\( T \) is the total number of message transmissions in the network in Equation (4.3). Also from Equation (4.3), it is clear that in DUCF, the order complexity of control message transmissions is in the order of 1 which is same as in LEACH.

4.3.2 Data Collection phase:

The CH nodes will generate TDMA schedule for its members. A member node can report the data to their respective CH only during the specified time slot for it. In other time slots, it will be in sleep state and should not involve in transmission. The length of the frame is depending on the number of cluster members and time allotted for each member for data transmission. Since a homogenous network is assumed, all the members will be given the same amount of time slot. A member has to send one data per frame. Data
gathered by the CHs may contain redundant information since the member node’s data will be highly correlated. The CHs aggregates the received data packets from members into a single packet. At the end of the frame, the aggregated data are sent to BS in multi-hop fashion.

In some cases, when the members transmit the data to its CH, the data may be reaching the other nearby CHs too since the radio is broadcast in nature. In order to avoid this inter-cluster interference spread codes are used. When a CM sends the data to its respective CH node, the spread code of the concerned cluster is attached to avoid inter-cluster interference.

4.3.2.1 Energy Consumption in Data Collection phase:

Intra-cluster communications and inter-cluster communication are the two main activities happening during the data collection phase. Also, these two activities are the major source of energy consumption. The intra-cluster transmission is made up of three components as given in the Equation (4.4),

\[
E_{\text{intra-cluster}} = E_{\text{member to CH}} + E_{\text{CH Reception}} + E_{\text{DA}}
\]  

(4.4)

where \(E_{\text{member to CH}}\) is the energy consumed at CM to transmit data to CH, \(E_{\text{CH Reception}}\) is the receiving energy spent at the CHs and \(E_{\text{DA}}\) is the energy for data aggregation.

\[
E_{\text{member to CH}} = \sum_{i=1}^{k} \sum_{j=1}^{m_i} E_{\text{Tx}}(j, CH_i)
\]  

(4.5)

Equation (4.5) depicts the energy consumption at the member nodes during intra-cluster communication, where \(k\) is number of clusters in the network, \(m_i\) represents number of members within the \(i^{th}\) cluster and it will be varying for each cluster, and \(E_{\text{Tx}}(j, CH_i)\)
represents the transmission energy between node \( j \) to its CH in the \( i^{th} \) cluster in the network.

\[
E_{CH\text{ Reception}} = \sum_{i=1}^{k} m_i \cdot E_{Rx} \quad (4.6)
\]

Equation (4.6) shows the energy consumption at CHs due to data reception from its member nodes where \( m_i \) is the number of member nodes in \( i^{th} \) cluster, \( k \) is number of clusters and \( E_{Rx} \) is the receiving energy.

\[
E_{DA} = \sum_{i=1}^{k} l \cdot m_i \cdot E_{per\text{DataBit}} \quad (4.7)
\]

where \( l \) is the number of bits, \( E_{per\text{DataBit}} \) is the energy consumption for single bit data aggregation as specified in Equation (4.7). Equation (4.8) shows the energy consumption if there is no intermediate node for communication between a CH node and BS,

\[
E_{inter-\text{cluster}} = \sum_{i=1}^{k} E_{TX}(CH_i, BS) \quad (4.8)
\]

In Equation (4.9) \( E_{MH} \) depicts the energy consumption if there is any intermediate nodes for communication between a CH node and BS,

\[
E_{MH} = E_{TX}(CH, v) + \sum_{j=2}^{w} E_{TX}(CH_{(j-1)}, CH_j) + E_{TX}(z, BS) \quad (4.9)
\]

where \( v \) is the next hop intermediate node from CH to BS, \( w \) is the number of intermediate nodes between CH to BS and \( z \) is the nearest CH node to BS in the path of distant CH to BS.
4.3.3 Time complexity of DUCF:

During the execution of DUCF $n$ fuzzy computation occurs. Each node decided itself to be CH for current round or not and it takes $n$ computations. In the worst case, a member node will be sending $Join_{Req}$ for $k$ times and $mk$ computations occurs. Similarly for accepting a member or rejecting its request other $mk$ processing is done.

$$T(n) = n + nx + mk$$  \hspace{1cm} (4.10)

$$T(n) = n + nx + mk = O(n)$$  \hspace{1cm} (4.11)

where $n =$ number of nodes in the network, $x=$number of neighbors, $k =$ number of cluster heads, $m =$ number of members in the cluster.

Figure 4.9 shows the algorithm of DUCF, $calculateFuzzy()$ is a fuzzy function executed in all the nodes during the start of CH election sub phase.
DUCF Algorithm:

1. \( N = \) Total number of nodes
2. \( \text{this} = \) pointer to current node
3. for \( k = 1 : N \)
4.   \( \text{state} (k) = \) probationary CH
5.   \( \text{ND} (k) = \) number of nodes within communication radius \( R_c \)
6.   \( \text{DBS} (k) = \) distance of the node with BS
7.   \( \text{RE} (k) = \) residual energy of the node.
8.   chance, size = \text{calculateFuzzy}(\text{ND}(k), \text{DBS}(k), \text{RE}(k))
9.   Send \( CH\_CANDIDATE \) to all neighbor nodes
10. end
11. for \( n = 1 : N \)
12.   \( x (n) = \) list of all \( CH\_CANDIDATE \) from neighbor nodes
13.   \( \text{status} = 1 \)
14.   for \( k = 1 : |x (n)| \)
15.     if (this.chance > chance(k))
16.       continue
17.     else
18.       \( \text{status} = 0 \)
19.       break
20.     end
21.   end
22.   if \( \text{status} == 1 \)
23.     advertise \( CH\_WON \)
24.     \( \text{state} (n) = \) Final CH
25.   else
26.     \( \text{state} (n) = \) CM
27.   end
28. end
29. for \( z = 1 : N \)
30.   \( \text{FCH}(z) = \) list of all Final CH nodes within \( R_c \)
31.   sort distance to all members in \( \text{FCH}(z) \) in ascending order
32.   for \( h = 1 : \text{size}(y) \)
33.     send \( CM\_JOIN \) to \( \text{FCH}(h) \)
34.     acceptance = 0
35.     if (\( \text{FCH}(h) \) accepts)
36.       \( \text{FCH}(h) \) sends \( CM\_ACCEPTANCE \)
37.       acceptance = 1
38.     break
39.     else
40.       \( \text{FCH}(h) \) sends \( CM\_REJECTION \)
41.     continue
42.     end
43.   if (acceptance == 0)
44.     \( \text{state} (z) = \) Final CH
45.   end
46. end

Figure 4.9: DUCF algorithm
4.4 Simulation Setup:

Using simulation setup, DUCF is compared with three other distributed fuzzy based algorithms Cluster Head Election mechanism using Fuzzy (CHEF), Energy aware distributed Clustering Protocol using Fuzzy approach (ECPF) and Energy Aware Unequal Clustering using Fuzzy approach (EAUCF). Also Low Energy Adaptive Clustering Hierarchy (LEACH) algorithm is simulated, since it is the base for all the distributed clustering algorithms in WSN.

In the ROI of 200 m x 200 m square area, 100 sensor nodes are deployed. As mentioned in Chapter 3, the algorithms are tested in three different scenarios based on the location of BS. In scenario 1, BS is located at (100, 100), BS at (200, 200) in scenario 2 and in scenario 3 BS is located at (100, 250).

The desired percentage of CH in LEACH is set at 0.1 in all the scenarios. The $P_{opt}$ value in CHEF and EAUCF is set at 0.3 as specified in the respective algorithms. The communication radius $R_c$ of sensor nodes is set as 40m since it ensures that all nodes in the network join a nearby CH. The higher $R_c$ value than this results in some member nodes not joining with any CH due to size restriction. In certain cases, higher $R_c$ value leads to the formation of very big clusters. Again, these big sized clusters may increase the overhead for the CH.

All the algorithms are simulated in MATLAB environment, since it provides a coherent environment for execution and comparison. Fuzzy toolbox environment in MATLAB is used to create fuzzy inference system of CHEF, ECPF, EAUCF and DUCF.

The clustering algorithms are analyzed based on the following aspects:
(a) **Average Energy consumption per round**: Energy spent for one round of data collection activities. It is made up of intra-cluster and inter-cluster communication cost.

(b) **Network lifetime**: Estimated number of rounds till the death of nodes is referred as network lifetime. Stability period represents the number of rounds till the death of first node (FND) in the network. And Half Node Die (HND) is the number of rounds completed till the death of half the number of nodes in the network. Most Node Die (MND) corresponds to death of 80% of initial sensor nodes deployed.

The results presented in next section are based on total number of 60 simulations. For each scenario, 20 simulations are conducted by varying network deployment. In order to get the reliable results, 20 simulation runs are conducted in each scenario. In 95% confidence level, the obtained experimental results are within 2-8% around the average value.

### 4.5 Results and Analysis:

#### 4.5.1 Average Energy Consumption:

The average energy consumption of one time of data collection is depicted in Figure 4.10 for LEACH, CHEF, ECPF, EAUCF and DUCF.
From the Figure 4.10, it is observed that LEACH consumes more energy than other clustering algorithms because of two reasons: (i) The CH election in LEACH is not based on any standard metric and (ii) Communication between CH and BS is single hop which costs more energy for CHs. CHEF shows improvement than LEACH because of residual energy based CH election. But still, CHEF lags behind ECPF, EAUCF and DUCF in terms of single round energy consumption. The problem stated for LEACH (i.e.) directly communicating with the BS is applicable for CHEF also. This direct communication affects the distant CH to the BS a lot compared to closer CH nodes. Because of multi-hop transmission from CHs to the BS, ECPF shows considerable improvement than LEACH and CHEF. But the usage of improper input variables in ECPF makes it to lag behind EAUCF and DUCF. EAUCF shows much improvement than LEACH and CHEF. Also EAUCF is better than ECPF due to its better input variables. The clusters formed in EAUCF are of unequal sized to balance the energy consumption among them. But the
competition radius in EAUCF is only based on residual energy and not considering the number of neighbors. In certain cases like more number of neighbors, it may lead to increased intra-cluster communication cost. In all the scenarios, DUCF shows reduced energy consumption than LEACH, CHEF, ECPF and EAUCF. DUCF achieves this energy conservation by forming unequal clusters considering the distance between CH and BS and also the number of neighbors for a CH. In DUCF, the number of members to a CH is restricted using the output fuzzy variable size. The size is the maximum number of members a CH can accommodate. Multi-hop communication is also an important feature in DUCF for conserving per round energy consumption.

4.5.2 Network Lifetime:

On comparing the different clustering algorithms, FND, HND and MND are used. The nearby nodes in the same cluster may generate the same redundant data. So, the death of a single node would not affect the entire sensing phenomena but the quality of the information starts degrading. But after HND, the quality of the network degrades a lot and becomes useless in most of the applications. But to analyze the network performance MND is used. From the experimental results, it is observed that DUCF performs well than all other compared clustering algorithms in terms of FND, HND and MND. Figure 4.11, Figure 4.12 and Figure 4.13 depict the lifetime of WSN for clustering algorithms in scenario 1, 2 and 3 respectively. From Figure 4.11, it is observed that LEACH lacks behind all other algorithms due to ineffective CH election. The fuzzy based algorithm CHEF increases the lifetime in terms of FND compared with LEACH. But when HND is considered, CHEF founds to be lacking behind LEACH. The reason behind this degradation may be due to orphan nodes in the network. The orphan nodes are the node
which cannot join any CH and elect itself as CH and starts transmitting directly to the BS. ECPF and EAUCF show promising increase in lifetime than LEACH and CHEF. The main reason behind it is the selection of proper CHs and multi-hop communication between CH and BS. DUCF extends the death of its first node roughly by around 300 rounds than LEACH, 150 rounds than CHEF and around 100 rounds than ECPF and EAUCF. DUCF achieves this improvement due to two important aspects, (i) considering node degree for CH and (ii) assigning appropriate number of member nodes to a CH according to its capacity.

From Figure 4.12, it is observed that CHEF and LEACH lifetime is close to each other. The performance percentage of ECPF, EAUCF and DUCF is same as in scenario 1. DUCF shows around 80 more rounds than EAUCF, 100 more rounds than ECPF and more than 300 rounds than LEACH and CHEF.
It is observed from Figure 4.12 and Figure 4.13, when the distance between CHs and BS increases, the lifetime of the network decreases proportionally. In the case of LEACH and CHEF where the single hop communication between CH and BS is followed, the lifetime becomes worse. But ECPF, EAUCF and DUCF somewhat manages due to multi-hop communication between CH and BS. In scenario 3, DUCF leads EAUCF by at least 30 rounds, ECPF by 73 rounds and more than 300 rounds than LEACH and CHEF in terms of FND metric.
The stability period is time till the death of the first node in the network. FND of all the algorithms are depicted in Figure 4.14. It is interpreted from Figure 4.14 that DUCF is better by 71.87% than LEACH, 31.91% than CHEF, 18.96% than ECPF and 16.07% than EAUCF in scenario 1. Considering scenario 2, DUCF lifetime increases by 127.14% than LEACH, 116.72% than CHEF, 17.82% than ECPF and 14.95% than EAUCF. In scenario 3, DUCF is better than LEACH by 133.47%, better than CHEF by 150%, better than ECPF by 14.83% and better than EAUCF by 5.01%.
Figure 4.15 depicts the number of rounds of all algorithms considering HND condition. DUCF increases lifetime of WSN by 14.30% than LEACH, 31.78% than CHEF, 19.65% than ECPF and 16.74% than EAUCF in scenario 1. Considering scenario 2, DUCF is better by 54.10% than LEACH, 51.74% than CHEF, 5.78% than ECPF and 2.35% than EAUCF. In scenario 3, DUCF increases lifetime by 95.38% than LEACH, 91.84% than CHEF, 14.20% than ECPF and 6.72% than EAUCF.

Figure 4.15: Half Node Die (HND) of clustering algorithms in various scenarios

Figure 4.16 depicts the performance of the algorithms at MND condition. In scenario 1, LEACH comes close to DUCF. But when it comes to scenario 2 and scenario 3 DUCF achieves higher number of rounds than others.
4.6 Conclusion:

A fuzzy logic based unequal distributed clustering algorithm DUCF is presented in this chapter. The proposed algorithm DUCF is compared with LEACH, CHEF, ECPF and EAUCF in different scenarios. The experimental results are also analyzed in detail. DUCF gives increased lifetime than LEACH and other fuzzy based clustering algorithms such as CHEF, CHEF and EAUCF in all the scenarios. DUCF achieves this increased lifetime by assigning proper number of members to CH according to its ability and routing the aggregated data through multi-hopping mechanism. This way of assigning unequal number of members for the CHs ensures balanced energy conservation among the CHs in the network. This balanced energy conservation leads to increased network lifetime of WSN.