CHAPTER 4

A NOVEL CLUSTERING TECHNIQUE USING BACTERIAL FORAGING ALGORITHM

4.1 SCOPE OF THIS CHAPTER

Nature inspired many algorithms design. A principle behind nature inspired algorithms is efficiency, interpreted as an individual’s to get sufficient energy source in least time called foraging which is crucial in natural selection, as animals with poor foraging strategies are eliminated, and those of the successful are propagated. So, to survive, an animal or a group must develop optimal foraging policies. Some successful foragers are bacteria like E.Coli, which uses chemical sensing organs to detect concentration of nutritive or noxious substances in environment to which it moves through tumbles and runs, avoiding noxious substances and getting closer to food areas in a process called ‘chemotaxis’. The bacterium also secretes a chemical agent to attract peers, resulting in indirect communication.

Inspired by E.Coli foraging strategy, Passino proposed Bacterial Foraging Optimization Algorithm (BFOA) for distributed optimization and control. BFOA is E.Coli foraging behaviour present in human intestines and already used in many engineering problems. BFOA is better than PSO and GA regarding convergence, robustness and precision.

There are WSN routing protocols that minimize energy used, extending WSN life subsequently. WSN routing must be improved. This chapter discusses an improved cluster head selection for efficient sensor
networks data aggregation. The proposed Bacterial Foraging Optimization (BFO) based algorithm is mapped to WSN.

4.2 OPTIMIZATION METHODS FOR ROUTING

Optimization is an activity in science and engineering. Much modelling, design, control and decision making problems are formulated regarding mathematical optimization. The classical optimization framework is minimization (or maximization) of objectives, given the constraints for problem to be solved. Many design problems are characterized by many objectives, where a trade-off among various objectives is made, leading to under/over-achievement of objectives. Also, some flexibility is present to specify problems constraints (Kaymak & Sousa 2001).

Network routing optimization problems include locating paths (simple and closed paths, tours, or walks) between network nodes efficiently. Such problems and solutions are essential to address many real-world decisions in applications as varied as logistics, transportation, computer networking and internet routing.

An optimization problem is generally recognized to be nondeterministic as well as fuzzy in nature and the nondeterministic condition is not only in the design variables, it can also be in the allowable limits. One can use the expected value and the chance constrained programming technique to transform the stochastic problem into its deterministic form.

Quality of Service for routing protocols deals with optimization of routing paths. Various optimization techniques find an available optimal path from source to destination. It is not necessary that an optimal path always be a short one, but it should be feasible and give the best possible route from source to destination.
4.3 METHODOLOGY

In this chapter, the BFO is mapped with the LEACH for selection of cluster-head. Hence forth, the protocol is optimized using BFO. Example mapping is also provided.

4.3.1 Bacterial Foraging Optimization

BFO is new biologically encouraged, stochastic global search technique mimicking E. coli bacteria’s foraging behaviour which locates, handles and ingests food. A bacterium during foraging exhibits two actions: tumbling or swimming (Thomas 2013).

When local bacteria forage, locomotion by a set of tensile flagella which help an E.coli bacterium to tumble or swim the two basic operations performed by a bacterium during foraging (Das et al 2009). When they rotate flagella clockwise, each flagellum pulls on the cell, resulting in the moving of flagella independently. Finally, the bacterium tumbles with lesser number of turns while in a harmful place it tumbles more to find a nutrient gradient. Moving flagella counter-clockwise helps it to swim fast. In the algorithm, the bacteria undergo chemotaxis, where they move to a nutrient gradient and avoid noxious environment. Generally bacteria move for longer distances in a friendly environment.

Chemotaxis

Chemotaxis is the foraging behaviour of bacteria where try to avoid noxious substances and search for nutrient rich substances by climbing up a nutrient concentration (Kaur & Joshi 2012). This involves two actions; either a run (in same direction as previous step) or tumble (in a different direction from previous step). To explore whole search space run steps in a particular direction are limited. Hence, they tumble after some run steps.
E. coli bacteria’s Chemotaxis foraging behaviour has common bacteria with a diameter of 1 $\mu m$ and length of about 2 $\mu m$ which under appropriate circumstances reproduces in 20 min. It is the ability to move from up to six rigid 100–200 rps spinning flagella, each driven by a biological motor. When flagella rotate clockwise, they are propellers and so an E.Coli runs or tumbles.

Figure 4.1  Flow chart of basic BFO algorithm
Chemotaxis Actions are:

(A1) In neutral medium, alternate tumbles and runs ⇒ search.

(A2) If swimming (in a nutrient of gradient or out of noxious substances), swim longer (climb up nutrient gradient or down noxious gradient) ⇒ seek increasingly favourable environments.

(A3) when swimming down nutrient gradient (or up noxious substance gradient), then search ⇒ to avoid unfavourable environments.

Reproduction Operation

All biological evolution goes through natural laws of survival of the fittest. The remains are improved solutions, while those ousted are poor solutions after a system runs for a time. To maintain population size, remains split into two to supply population size; a process called reproduction operation in BFO (Wu 2013).

Elimination and Dispersal Operation

Elimination and Dispersal operations are that a bacterium eliminates or disperses to a new location due to gradual/sudden changes in a local environment where a bacterium population lives. To simulate this in BFO, some bacteria are killed randomly with a very small probability while new replacements are initialized over search space randomly. This enhances the algorithm’s global search capability.
Bacteria swarm S behaves as follows (Montes & Ocana 2008):

1) Bacteria are distributed in nutrients map randomly.

2) Bacteria move to high-nutrient regions. Those in noxious substances regions or low-nutrient regions die/disperse, respectively. Bacteria in convenient regions reproduce (split).

3) Bacteria located in promising nutrients map regions try to attract others by generating chemical attractants.

4) Bacteria are now located in highest-nutrient region.

5) Bacteria disperse to look for new nutrient map regions.

The BFO is controlled by a number of external parameters. These are the number of bacteria \((S)\), maximum number of chemotactic loop \((N_c)\), maximum number of reproduction \((N_{re})\), maximum number of elimination and dispersal events \((N_{ed})\), dimension of the search space \((p)\), divisor of the step size \((d_s)\), probability for the elimination dispersal event \((P_{ed})\), swim length \((N_s)\), counter for swim length \((m)\), and chemotactic step sizes \(C(i)\), \(i = 1, 2, \ldots, S\).

The bacterium is pointed in a random direction after a tumble. To represent a tumble, a unit length random direction \(\hat{\delta}(j)\) is produced; this is used to determine the direction of movement after a tumble:

\[
\zeta(i + 1, k, l) = \zeta(i, k, l) + C(i)\hat{\delta}(j)
\]

(4.1)

where \(\zeta(i, k, l)\) is the \(i^{th}\) bacterium at \(j^{th}\) chemotactic, \(k^{th}\) reproductive and \(l^{th}\) elimination and dispersal step.
The procedures implemented are:

\[ f = \beta \cdot f_1 + (1 - \beta) \cdot f_2 \]  

(4.2)

Where \( f_1 \) is maximum average Euclidean distance of nodes to associated cluster heads and \( f_2 \) is ratio of total initial nodes energy to total energy of cluster-head candidates expressed as follows:

\[ f_1 = \text{MAX}_{k=1,2,3,...,K} \left( \frac{\sum_{i \in \mathcal{C}_{p,k}} d(n_i, \text{CH}_{p,k})}{|\mathcal{C}_{p,k}|} \right) \]

(4.3)

\[ f_2 = \frac{\sum_{i=1}^{N} E(n_i)}{\sum_{i=1}^{N} E(C_{p,k})} \]

(4.4)

Where \( N \) is number of nodes of which \( k \) is elected as cluster-heads. \(|\mathcal{C}_{(p,k)}|\) is number of nodes belonging to cluster \( C_k \) in particle \( p \), ensuring that nodes with above average energy resources are elected as cluster-heads and average distance between nodes and cluster-heads is minimum.

### 4.3.2 Proposed Fitness function

A novel fitness function using link quality and available energy is proposed in this work.

\[ f(i) = \alpha \left( \frac{\text{energy}_i}{\sum_{i=1}^{n} \text{energy}_i} \right) + (1 - \alpha) \left( \frac{1}{\sum_{j=1}^{m} L_{i,j}} \right) \]

(4.5)

where \( \text{energy}_i \) is the residual energy in node \( i \), \( n \) is the number of nodes in the network, \( L_{i,j} \) is the link quality between nodes 1 which is being evaluated for
becoming cluster-head and its two hop neighbours, $\alpha$ is a constant whose value lies between $0 < \alpha \leq 1$. As alpha increases the energy savings are increased, however if QoS is required, the value of alpha is reduced. Experiments were conducted for alpha=0.5.

4.3.3 Proposed BFO Technique

For a sensor network with N nodes and k number of clusters, the sensor network can be clustered as follows:

**Step 1:** Initialize population, each individual containing k randomly selected cluster-heads.

**Step 2:** Evaluate the fitness function of each individual, where:

i. For each node $n_i$ in the network:

   a. Calculate distance $d$ between $n_i$ and all cluster heads $C_k$

   b. Assign $n_i$ to $C_k$, where distance($n_i$, $C_k$) = min$_{k=1,2,…,k}$\{distance ($n_i$, $C_k$)\}

ii. Compute the fitness function.

**Step 3:** Perform the position update by the optimization algorithm.

**Step 4:** Repeat steps 2 to 4 until the maximum number of cycle is reached.

4.4 EXPERIMENTAL SETUP

An improved cluster-head selection for efficient data aggregation in sensor networks based on BFO is proposed. Table 4.1 shows the experimental setup with parameters.
Table 4.1 Experimental setup parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of the network</td>
<td>3000 by 3000 m</td>
</tr>
<tr>
<td>Transmission power of each node</td>
<td>0.005 watt</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>50-175</td>
</tr>
<tr>
<td>Alpha</td>
<td>0.5</td>
</tr>
<tr>
<td>S</td>
<td>50</td>
</tr>
<tr>
<td>$N_c$</td>
<td>100</td>
</tr>
<tr>
<td>$N_s$</td>
<td>50</td>
</tr>
<tr>
<td>$N_{re}$</td>
<td>10</td>
</tr>
<tr>
<td>$N_{ed}$</td>
<td>2</td>
</tr>
<tr>
<td>$P_{ed}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$d_s$</td>
<td>1.9</td>
</tr>
<tr>
<td>$C(i), i = 1, 2, \ldots, S$</td>
<td>0.02</td>
</tr>
</tbody>
</table>

4.5 RESULT AND DISCUSSION

The results of PDR, End to End Delay and remaining energy in joules are shown graphically in Figure 4.2 to 4.4 respectively. Table 4.2 to 4.4 tabulates the same.

Table 4.2 Packet delivery ratio

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>M-LEACH</th>
<th>GA-LEACH</th>
<th>Proposed BFO technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.8642</td>
<td>0.8787</td>
<td>0.9068</td>
</tr>
<tr>
<td>75</td>
<td>0.7874</td>
<td>0.8099</td>
<td>0.8854</td>
</tr>
<tr>
<td>100</td>
<td>0.7327</td>
<td>0.767</td>
<td>0.8129</td>
</tr>
<tr>
<td>125</td>
<td>0.7036</td>
<td>0.7368</td>
<td>0.7915</td>
</tr>
<tr>
<td>150</td>
<td>0.6611</td>
<td>0.6961</td>
<td>0.6499</td>
</tr>
<tr>
<td>175</td>
<td>0.5663</td>
<td>0.5999</td>
<td>0.6084</td>
</tr>
</tbody>
</table>
From Figure 4.2 and Table 4.2 it is observed that the PDR achieves in a better way. The proposed BFO technique achieves PDR by 4.93% than M-LEACH and by 3.2% than GA-LEACH with 50 nodes. The proposed BFO technique achieves PDR by 7.43% than M-LEACH and by 1.42% than GA-LEACH with 175 nodes.

**Table 4.3 End to end delay**

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>M-LEACH</th>
<th>GA-LEACH</th>
<th>Proposed BFO technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.0275</td>
<td>0.022</td>
<td>0.0358</td>
</tr>
<tr>
<td>75</td>
<td>0.0301</td>
<td>0.067</td>
<td>0.0339</td>
</tr>
<tr>
<td>100</td>
<td>0.0151</td>
<td>0.021</td>
<td>0.0203</td>
</tr>
<tr>
<td>125</td>
<td>0.053</td>
<td>0.0357</td>
<td>0.0726</td>
</tr>
<tr>
<td>150</td>
<td>0.1378</td>
<td>0.1204</td>
<td>0.1715</td>
</tr>
<tr>
<td>175</td>
<td>0.1349</td>
<td>0.1401</td>
<td>0.2041</td>
</tr>
</tbody>
</table>
From Figure 4.3 and Table 4.3 it is observed that the End to End Delay performance is not obtained in a better way. The proposed BFO technique increases End to End Delay by 30.18% than M-LEACH and by 62.73% than GA-LEACH with 50 nodes. The proposed BFO technique increases End to End Delay by 51.3% than M-LEACH and by 45.68% than GA-LEACH with 175 nodes.

Table 4.4 Remaining energy in joules

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>M-LEACH</th>
<th>GA-LEACH</th>
<th>Proposed BFO technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>354</td>
<td>365</td>
<td>361</td>
</tr>
<tr>
<td>75</td>
<td>298</td>
<td>327</td>
<td>320</td>
</tr>
<tr>
<td>100</td>
<td>286</td>
<td>301</td>
<td>295</td>
</tr>
<tr>
<td>125</td>
<td>232</td>
<td>258</td>
<td>254</td>
</tr>
<tr>
<td>150</td>
<td>205</td>
<td>231</td>
<td>228</td>
</tr>
<tr>
<td>175</td>
<td>190</td>
<td>204</td>
<td>200</td>
</tr>
</tbody>
</table>
From Figure 4.4 and Table 4.4 it is observed that the remaining energy in Joules increases in a better way. The proposed BFO technique increases remaining energy in joules by 1.98% than M-LEACH but decreases by 1.1% than GA-LEACH with 50 nodes. The proposed BFO technique increases remaining energy in joules by 5.26% than M-LEACH but decreases by 5.26% than GA-LEACH with 175 nodes.

Table 4.5 Lifetime of nodes

<table>
<thead>
<tr>
<th>Number of rounds</th>
<th>M-LEACH</th>
<th>GA-LEACH</th>
<th>Proposed BFO technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>200</td>
<td>90</td>
<td>92</td>
<td>96</td>
</tr>
<tr>
<td>300</td>
<td>65</td>
<td>85</td>
<td>90</td>
</tr>
<tr>
<td>400</td>
<td>40</td>
<td>63</td>
<td>72</td>
</tr>
<tr>
<td>500</td>
<td>20</td>
<td>54</td>
<td>63</td>
</tr>
<tr>
<td>600</td>
<td>0</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>700</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>800</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Figure 4.5 Lifetime computation

From Figure 4.4, it is observed that the proposed BFO technique increases lifetime of the network. It is observed that by 600 rounds, the nodes using M-LEACH and GA-LEACH are dead whereas all the nodes die in 700 rounds with the proposed BFO method. The proposed BFO selects optimal cluster-heads more efficiently, thus increasing the network lifetime.

Table 4.6 Number of clusters formed

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>M-LEACH</th>
<th>GA-LEACH</th>
<th>Proposed BFO technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>75</td>
<td>15</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td>100</td>
<td>19</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>125</td>
<td>22</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>150</td>
<td>25</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>175</td>
<td>30</td>
<td>27</td>
<td>28</td>
</tr>
</tbody>
</table>
It is observed from the results that the proposed BFO achieves improved performance than M-LEACH and GA-LEACH. BFO is preferred for optimizing as it is better than PSO and GA regarding convergence, robustness and precision.

4.6 SUMMARY

BFO is more feasible and efficient algorithm than GA. Fast and optimized results are achieved and nodes selection is easy by BFO because of the fast conversion property of BFO algorithm. This study proposed an improved cluster-head selection for sensor networks efficient data aggregation using BFO. PDR improves with the proposed BFO technique by 4.93% than M-LEACH and by 3.2% than GA-LEACH with 50 nodes.

The proposed BFO technique improves PDR by 7.43% than M-LEACH and by 1.42% than GA-LEACH with 175 nodes. The proposed BFO technique increases remaining energy in joules by 1.98% than M-
LEACH but decreases by 1.1% than GA-LEACH with 50 nodes.

The proposed BFO technique increases remaining energy in Joules by 5.26% than M-LEACH but decreases by 5.26% than GA-LEACH with 175 nodes. The network lifetime is significantly improved due to the proposed BFO technique. Though, the proposed BFO technique increases End to End Delay by 30.18% than M-LEACH and by 62.73% than GA-LEACH with 50 nodes.

The proposed BFO technique increases End to End Delay by 51.3% than M-LEACH and by 45.68% than GA-LEACH with 175 nodes. Further work is to be carried out to improve BFO algorithm to avoid local minima.