Chapter 2

Literature Survey

Wireless sensor networks are primarily used for sensing and collecting the information from environment. This information is sent to sink node, where, it is processed and analyzed by the underlying application. Majority of energy consumption by sensor nodes in WSN, is due to data transmission. To enable the data transmission to BS from sensor nodes in energy efficient way, a variety of routing protocols are reported in literature. These protocols are either distributed in nature or centrally controlled by BS. Several existing centralized routing protocols are presented in detail in this chapter. The conventional techniques of clustering as well as the recent methods based on evolutionary computing is also discussed.

2.1 Basics of WSN

In this modern era of technology driven lifestyle, communication is omnipresent in every aspect. One of the applications of ad hoc networks is Sensor networks. Wireless sensor network (WSN) can be applied to broad range of applications for improving our lives in many ways. A typical sensor network is as shown in Figure 2.1. Such sensor nodes in WSN monitor the environment conditions and react accordingly by informing the same to the BS. The main objective (K. Sohraby and Znati) of a WSN is to detect an event and dissipate the same to other nodes reliably. In order to detect the forest fire, intrusion detection, seismic activity or tsunami like situations, a large number of sensor nodes, which can monitor the surrounding environment conditions, are densely deployed. Messages indicating alarming situations in these critical environments have to be transferred to the sink node as fast as possible. These
messages must not be lost. The sensor networks follow several traffic patterns based on the underlying application and the deployment strategy. Traffic pattern generally followed in sensor networks is many-to-one where, nodes report to one single base station. Sensor nodes are required to function with constrained resources like less energy utilization with a better lifetime.

Several components like Micro Electro-Mechanical Systems (MEMS) sensor technology, digital circuit, customized system integration for low power consumption and a low power radio frequency (RF) with related control circuitry form the hardware base of the sensor network. Sensor node is composed of micro-sensor technology, low power signal processing, lower power computation and low cost wireless networking capability. Sensor node is composed of micro-sensor technology, low power signal processing, lower power computation and low cost wireless networking capability. Figure 2.2 shows the architecture of sensor node along-with the communication infrastructure. Components of a sensor node are as follows:

- a sensing unit
- a processing unit
- a transceiver unit and
- a power unit
- application dependent component (additional and optional) Eg. a location finding system or a power generator and a mobilizer
Sensors and Analog to Digital Converters (ADCs) are subunits of a sensing unit. Sensors produce analog signals. These signals are converted to digital signals by the ADC. These digital signatures are fed to processing unit. Processing unit is associated with small storage unit that manages the collaboration with other sensor nodes for carrying out the assigned sensing tasks. A transceiver unit is responsible for transmitting and receiving information and hence, connects the node to the network. Power units may be supported by solar cells, capable of power scavenging. As such, sensor networks are a special case of wireless communications with certain characteristics.

2.2 Characteristics of WSN

Wireless Sensor Networks have relatively distinct features compared to other networks. As per the application requirements, the sensor nodes may be deployed in a large number. Due to the large size of such networks, global identification of every node can be difficult to manage at times. The nodes are deployed generally with random deployment pattern. These nodes may be dropped from high altitude or from a flying object. Sensor nodes are capable of self organization to create the necessary infrastructure to monitor especially unreachable areas. To learn the positions of the neighboring nodes, some localization methods are required. As these nodes are small in size, the resources are highly constrained. The available energy supply is very limited, storage and processing capabilities are also restricted. The commu-
communication capabilities of sensor nodes are not as high as that of wired networks. The communications is also unreliable. Due to limited energy supply, the nodes may die early or may be switched off momentarily for network longevity. Because of this on and off status of several nodes, the network experiences constant changes in topology. If the nodes are not stationary, frequent topological changes can be there. Hence, it becomes difficult to maintain already established information propagation paths. The network must continue working for the application even in the case of failure or death of some of the nodes.

Current research in WSN platforms enables the support to a wide range of sensors. Different radio components, processors and storage are present in products that offer sensors and sensor nodes. As the sensor hardware is different, it is a challenge to integrate multiple sensors on a WSN platform. Moreover, the problem gets intensified in processing raw data with limited resources in the sensor node. Various sensor hardware available are like Mica2, MicaZ, IRIS, iMote, Cricket, Sun SPOT, Telos B, Tmote Sky and others [Akyildiz and Vuran]. Operating system must be designed to handle these sensor platforms. Various sensor network operating systems which drive the node are TinyOS, Contiki, Mantis, SOS, LiteOS, Nano-RK and several others [Dong et al]. Each of these operating systems support different types of nodes with specific hardware availability. Designing platforms to support automatic management, optimized network longevity and distributed programming are the goals of research in this area.

2.3 Applications of WSN

The characteristics of nodes in WSNs make it suitable for use in wide variety of applications. The nodes can be deployed in the area to be monitored without visiting the area physically. The nodes can self organize themselves, creating the infrastructure on the fly. Although there are some limitations with these nodes, still, it has been widely adopted for various applications. WSNs can be used in applications like military target tracking and surveillance, natural disaster relief, biomedical health monitoring, hazardous environment exploration and seismic sensing [Yick, Mukherjee, and Ghosal].
2.3.1 Military Applications

In military scenarios, it is advantageous to transmit the sensor network’s collected data to the ultimate end-users via an Unmanned Aerial Vehicle (UAV) that acts as an airborne relay. Fixed infrastructure is infeasible in hostile environment or in inhospitable terrains deployed for group of soldiers in enemy territories. Sensor networks provide the required communication mechanism quickly for such applications. Other applications include cooperative target identification and tracking.

2.3.2 Monitoring Applications

Sensors currently under deployment sense temperature, humidity, visual and infrared light, acoustic, vibration, pressure, chemicals, mechanical stress, magnetic effect etc. to name a few. Wireless sensor nodes are often deployed for monitoring of vehicles, animals, machines, medical purposes, environment studies, structural health etc.

2.3.3 Emergency Operations

WSNs prove useful in situations of emergency like search or rescue operations during natural disasters. Few such disasters are earth quake, forest fire, flood etc. For example, to produce a temperature map of the area or to determine the perimeter of areas with high temperature from outside, sensor nodes can be deployed from an airplane over a wildfire in a forest. Major factors favoring WSNs for these tasks are self configuration of the system with minimal overhead, independent of fixed or centralized infrastructure, nature of the terrain of such applications, freedom and flexibility of mobility, and the unavailability of conventional communication infrastructure (Akyildiz and Vuran).

2.3.4 Health Care

Applications of WSNs in health-care are controversial and vary from post-operative and intensive care to long term surveillance of elder patients. In the first case, sensors are directly attached to patients, thereby, eliminating the need for cables. In the later case, automatic drug administration is achieved by embedding sensors into drug packaging. If the drug is administered to incorrect person, an alarm may be triggered.
2.3.5 Machine surveillance and preventive maintenance

Sensor nodes can be fixed at difficult to reach areas of machinery. These nodes can detect vibration patterns and conclude whether maintenance is required or not. Such nodes can be applicable in robotics or the axles of trains. Other applications in manufacturing units are obvious. WSNs can be utilized in these applications since they have cable-free operation, don’t have maintenance problems and are cheap and often retrofitted.

2.4 Issues in Wireless Sensor Networks

Issues affecting the design, deployment, and performance of Wireless Sensor Networks are described in this section.

2.4.1 Energy Constraints

Energy is an important constraint in Wireless Sensor networks. Nodes run on batteries. This implies that energy utilization must be controlled for the sake of network longevity [Perkins]. Energy is spent mostly during transmission and receiving of data packets. On an average, for a single bit communication, 1 mJ is consumed in transmitting and 0.5 mJ is consumed in receiving. If this consumption is compared to 0.8 mJ for 208 CPU cycles (approximately 100 instructions), it can be concluded that significant amount of energy can be conserved if efficient data transmission and receiving techniques are employed. Alternatively, number of data transfer operations can be also reduced [Heinzelman].

2.4.2 Fault Tolerance

Depending on the operational environment, some sensors may fail due to lack of power, physical damage, or environmental interference. Several motes in a region are turned off, and then on again. Some mote locations are swapped. Motes are displaced from their location. So accordingly, there can be transmission power adjustment, change in sensing rate and rerouting of packets through regions with more power.
Table 2.1: Interaction patterns between sources and sinks

<table>
<thead>
<tr>
<th>Event Detection</th>
<th>Sensors detect events like forest fires, grass fires and volcanic eruptions etc. Such events are required to be classified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periodic Measurements</td>
<td>Periodically, the measured values are reported. These event reports are triggered by detected events.</td>
</tr>
<tr>
<td>Function approximation and edge detection</td>
<td>Sampling is done for gathering the detected event at each node. The edge detection function finds edges when sampling value jumps across boundary of space-time region</td>
</tr>
<tr>
<td>Tracking</td>
<td>When source of event is mobile, sensors report the event sources position to sink with speed and direction estimation.</td>
</tr>
</tbody>
</table>

2.4.3 Information Processing

Sensor networks can be deployed for many different types of applications like event detection, target tracking, periodic measurements, function approximation or edge detection. As per the application requirements, the data gathering and processing could be done differently in these applications. Few typical patterns for interaction between sources and sinks are as shown in Table 2.1. Depending on the application, the data delivery model can be continuous sensing, query based or event driven. To ensure the scalability and increased efficiency of network operations, clustering approaches have become an emerging technology for building scalable, robust, energy-efficient WSN applications. On examining various functionalities of sensor networks, it is felt that communication holds a big share of
the consumed energy resources. With respect to energy, it is observed that the cost of transmitting data far exceeds the cost of processing it because, radio communication expends the maximum power. So, developing state of the art techniques to reduce the number of radio transmissions can greatly increase the life of the network thereby, delivering more data over a longer time period. Number of wireless communications can be reduced through in-network data processing. Researchers have contributed significantly to minimize energy consumption during communication. The process of routing data from all the nodes to sink node is also called convergecast. Many routing protocols are reported in literature.

2.4.4 Routing

Routing protocols intend to find a feasible path from a particular source to a destination based on hop length, minimum power required, minimum bandwidth requirement and lifetime of wireless link. For this, gathered information about power and bandwidth is used (Ilyas, Mahgoub, and Kelly). Routing in WSNs can be divided into flat, hierarchical and location-based routing (Al-karaki and Kamal). All nodes are assigned equal roles in flat routing, while in hierarchical routing, nodes play different roles in the network. In location-based routing, sensor nodes are addressed by their locations. There are design trade-offs between energy and communication overhead savings in every routing paradigm. Deciding about the routing policy is also crucial. Proactive routing policy (table driven) consumes more energy due to frequent updates compared to reactive (on demand) routing policy (Akkaya and Younis). Data aggregation is generally performed in the resource constrained networks. In such networks, data generated by sensing nodes is sent jointly by all nodes. This combined data is in-processed and forwarded to base station. Data aggregation intends to reduce the volume of data communicated. For this, aggregating nodes collect local data at intermediate nodes and forward the result of an aggregation operation to the sink node. This result may be in the form of min and max. Less energy is consumed due to in-processing, thereby, limited sensor resources are used. Whenever base station queries from the network, aggregator nodes receive data from sensor nodes. This aggregator node aggregates data received from sensor nodes and then forwards this aggregated information to base station. The processes of data aggregation and rout-
ing are almost tightly coupled, hence routing generally includes both of them as an integral routing function.

### 2.4.5 Aggregation Methods

Energy is very crucial resource in WSNs. Hence, aggregation intends to increase network lifetime and reduce energy consumption and bandwidth utilization. Routing techniques and data aggregation techniques are tightly coupled throughout the network. Several protocols are proposed by researchers to integrate routing and data aggregation. These protocols are categorized as tree-based and cluster-based data aggregation protocols. In order to reduce the latency inherent in tree based data aggregation, researchers proposed the cluster based data aggregation. In cluster based approach, sensor nodes are grouped into clusters so that data gets aggregated in each group for improved efficiency.

Tree and cluster based data aggregations are shown in Figures 2.3 and 2.4. Tree based data aggregation method introduces the delay in the communication if the network size is large. This delay is inherent to any tree based information processing method. Sensors in the network group themselves into several regions by sending short messages to each other in cluster based data aggregation technique. At any instance of time, one sensor in the network sends short messages within the network to all the other remaining sensors. This sensor is called Cluster-Head (CH). Other sensors decide to join those groups or not, depending on the signal strength of the messages sent by the cluster-heads. The cluster head decides the number of optimal cluster members it can handle. In order to transmit data from sensor nodes to CH
and CH to base station, a schedule is applied. Once the cluster head is selected for a region, data collection phase starts. All cluster members of that region send the collected or sensed data in their allotted slots to the CH as per the Schedule. CH now transmits this collected data in a compressed format to the base station. Hence, the collection phase is completed. Depending upon the adopted strategy, a new search is initiated for formation of cluster heads for a region.

![Cluster based data aggregation](image)

Figure 2.4: Cluster based data aggregation

### 2.5 Routing Protocols in WSN

There are numerous routing protocols as per the literature. Every routing protocol is generally designed considering the problem constraints, application requirements, hardware restrictions applicable to sensor network, network size and other factors. The most basic protocols designed for WSN are flat routing protocols like Flooding, Gossiping, SPIN, Directed Diffusion. Some of these flat routing protocols are described in the following subsection.

#### 2.5.1 Flat Routing

Sensor Protocols for Information via Negotiation (SPIN) is a family of protocols suggested by Heinzelman et al. (Heinzelman, Kulik, and Balakrishnan). SPIN efficiently disseminates information in a wireless sensor network. Flooding and Gossiping approaches when applied for data dissemination, uses communication and energy resources in excess. These two conventional techniques send redundant information in the network without considering the present conditions of the network resources. SPIN mitigates these problems of conventional methods of data dissemination. SPIN
applied data negotiation with resource adaptive mechanisms. It uses three message types, ADV, REQ and DATA. Whenever any node has large amount of data to send, it announces this with ADV message containing meta-data for actual data to its neighbors. Nodes interested in this data, reply with a REQ message demanding data. After the arrival of REQ message at the source node, the actual large data is sent to the demanding node. In this way SPIN reduces unnecessary messages and communication overhead, which is present in Flooding and Gossiping. SPIN also utilizes the present energy levels of nodes and accordingly adapts the protocol as per the remaining energy.

Directed Diffusion (Intanagonwiwat, Govindan, and Estrin) uses flooding to gather route information and deliver data. Base station sends query to sensor nodes by flooding some tasks, while sensors in SPIN advertise availability of data. The sink broadcasts interest through neighbours. This interest is a task description for data matching attributes. Each node receives interest and maintains an interest cache for later use. This process continues until gradients are set up from sources to base station. This functionality is illustrated in Figure 2.5 A gradient is a reply link to neighbour from which the interest was received, characterised by data rate, duration and expiration time which are derived from the received interests field. When interests fit gradients, multiple paths of information flow are formed. Best paths are reinforced by re-sending original interest through selected path from the sink with a smaller interval, to prevent further flooding. In this technique, there is no need for maintaining a global network topology. However, the same mechanism may not work
for demand driven applications. The performance of flat routing protocols deteriorates when number of nodes are increased. As there is too much of work to be done by every node in this category of routing protocols, it cannot be applied directly to large scale networks.

Due to limited energy availability, the information processing methods must be designed to adapt the changing energy map of the network. There is a trade-off between reaching a node directly using higher power and reaching a node via multiple hops using lower power. There can be hundreds or thousands of sensor nodes deployed depending on the size of the area to be monitored. Performance of the designed protocols should not degrade with increasing network size or increasing node density; ensuring a scalable network [Ilyas, Mahgoub, and Kelly] [Al-karaki and Kamal]. Unlike ad hoc network where there are typically around 100 nodes, sensor network is highly scalable with thousands of nodes deployed. Routing must be designed to scale to support several thousands of sensor nodes in sensor field [Akkaya and Younis]. Following section analyzes the recent work with respect to various clustering strategies suggested by researchers as per the various application requirements with application specific assumptions. The clustering process is formally described here.

2.6 Clustering Technique

As reported in literature, there are mainly three approaches for clustering in WSN. First one is direct transmission of data from cluster-heads to BS, other one is multi-hop where cluster-heads act as relay nodes and send data to BS. Third one is the combination of both direct and multi-hop transmissions. There are $N$ number of nodes represented by set $S$, to be clustered in $k$ clusters having one cluster-head in each group. Clustering can then be defined as, creating collection of $k$ subsets such that, following three conditions are satisfied [Wu].

\[ C \subset S, C \neq \emptyset \] (2.1)

\[ C_1 \cup C_2 \cup C_3 \ldots \cup C_k = S \] (2.2)
Clustering technique is applicable to a variety of domains, including image processing, pattern recognition, data engineering, robotics, medical, and many others. Clustering generally segregates the given raw data set into correlated data points, with certain similarity measures applied as per the application domain. The problem of clustering is generally categorized as an NP-hard problem (Khalil and Attea).

### 2.7 Clustering in WSN

Hierarchy, clustering and location-awareness techniques are proposed to cope up with scalability in large sensor networks. Hierarchical Routing is the well-known technique with special advantages related to scalability and efficient communication. In clustered hierarchical architecture, higher energy nodes can be used to process and send the information, while low-energy nodes can be used to perform the sensing in the proximity of the target. Several clustering techniques are proposed by researchers to cope with scalability issues (Pan et al.) (Younis and Fahmy, “Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach”). The most popular hierarchical clustering protocol among researchers is LEACH (Heinzelman, Chandrakasan, and Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”). LEACH protocol creates hierarchical clusters, which operates the network in an energy-efficient way.

![Figure 2.6: TDMA Slot- Setup and Steady State](image)

The operating process of LEACH protocol is shown in Figure 2.6. LEACH

\[
C_i \cap C_j = \emptyset, i \neq j \quad (2.3)
\]
protocol operates in rounds till the nodes are alive. Each round has two phases as follows:

- **Setup phase:** cluster-heads are selected and clusters are created in distributed way.
- **Steady state phase:** Cluster members transmit data to the respective CH node and CH node then sends aggregated data to BS.

\[
T(s) = \begin{cases} 
  p & \text{if } s \in G \\
  1 - p \cdot \frac{(r \mod p)}{p} & \text{otherwise} \\
  0 & \text{otherwise}
\end{cases}
\]  

(2.4)

The nodes are elected as cluster-head as per the Equation 2.4. A threshold is computed by every node and it is compared with a random value generated by the node. If the node has been cluster-head in the last r rounds, it decides threshold value of zero. If it has not been cluster-head in last r rounds, it may become cluster-head with the probability that random value generated is larger than the threshold computed as per Equation \ref{eq:2.4}. Here, the role of cluster-head is assigned with randomized rotation among the nodes so that the energy usage in the network is balanced (Heinzelman, Chandrakasan, and Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”). In this way, the energy load of cluster-head is evenly distributed among the nodes over the time. LEACH is considered as the most popular routing protocol that use cluster based routing in order to minimize the energy consumption. LEACH protocol does not provide uniform placement of CH nodes in the network.

Despite the significant overall energy savings, LEACH suffers from several drawbacks. There is no mechanism to ensure that the elected cluster-heads will be uniformly distributed over the network. Hence, there is a possibility that all cluster-heads will be concentrated in one part of the network. It also assumes that the amount of energy consumed by cluster-heads in every cluster round is constant. This assumption is however not realistic. Data rate generation from a region is not considered in LEACH during cluster-head selection process. Hence, non-CH nodes belonging to the regions having higher data rate generation and expected to transmit frequently,
dissipate more energy in transmitting data to a remote CH. So, uneven energy dissipation takes place over the network, resulting into reduced network lifetime. Secondly, LEACH assumes that whenever a node becomes CH, equal amount of energy is dissipated. Cluster-heads located far from the base station spend more energy in transmitting data than those located near the base station. So, it cannot be said that equal amount of energy is dissipated by CH. To ensure an even energy load distribution over the whole network, additional parameters including the residual energy level of candidates relative to the network and their data rate generation value should be considered.

Results show that the proposed protocol by Hussain et al. (Hussain and Matin) performs better than the LEACH protocol. Here, the network clusters are managed by a set of associates called head-set. Using the head-set approach, energy consumption is optimized by reducing number of elections. The long-range transmissions are evenly distributed among the network nodes. Half of the nodes report sensor readings while the remaining half are used for management purpose. Cluster management and sensor data collection is managed by head-set members. It is observed that, whenever the number of alive nodes in a network become less than 60%, maintenance cost of head-set size increases. The head-set size is directly proportional to the cluster size. Energy efficiency is increased by adjusting head-set with respect to number of transmissions and percentage of alive nodes.

In DEEC (Qing, Zhu, and Wang), the cluster-heads are elected by a probability based on the ratio between residual energy of each node and the average energy of the network. Traditional clustering algorithms are unable to treat each node discriminatorily in terms of the energy discrepancy. DEEC achieves longer network lifetime and more effective messages than other classical clustering algorithms in two-level heterogeneous environments. Moreover, DEEC is also fit for the multilevel heterogeneous networks. To control the energy expenditure of nodes by means of adaptive approach, DEEC use the average energy of the network as the reference energy.

Mainly two types of clustering approaches are found in literature, distributed clustering and centralized clustering. Distributed clustering can reduce the energy consumption during the clustering process significantly, as nodes will not send information to BS during setup phase. However, the message transmissions for electing
cluster-heads, join requests and TDMA schedule transmission by CHs is an extra burden on the nodes. Moreover, distributed clustering does not place cluster-heads uniformly as centralized clustering does. A centrally controlled clustering algorithm can produce better clusters compared to distributed approach of clustering. Hence, LEACHC is a centralized clustering algorithm. A variant of LEACH, LEACHC exploits base station for clustering purpose.

2.7.1 LEACH Centralized

LEACHC protocol is also having the same two phases like setup phase and steady state phase. Similar to LEACH, LEACHC uses the same steady-state protocol. However the setup phase is different than that of LEACH. During the set-up phase of LEACHC, in the beginning of each round, all nodes send their current location and energy level to the BS. BS computes the average residual node energy, and elects cluster heads (CH) having energy above this threshold for the current round. Simulated Annealing algorithm is used by BS to find clusters \cite{Murata and Ishibuchi}. This algorithm tries to minimize the amount of energy required by the member nodes to transmit their data to the cluster head. Sum of squared distances between all non-cluster head nodes and the closest cluster head is kept minimum during clustering process. During each iteration, next state consisting of a set of nodes in $C'$ is determined. Based on current state and given set of nodes in $C$, $x$ and $y$ coordinates of nodes $c$ in $C$ are randomly perturbed and new coordinates $x'$ and $y'$ are determined. The nodes closest to $(x', y')$ become new set of cluster head nodes $c'$, eventually, they make up set $C'$. Given the current state at iteration $k$, represented by the set of cluster-head nodes $C$ with cost $f(C)$, the new state, represented by the set of cluster-head nodes $C'$ with cost $f(C')$ will become the current state with probability shown in Equation 2.5 \cite{Heinzelman, Chandrakasan, Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”}. The probability of new set selection or old one for clustering is decided by Equation 2.5.

$$f(x) = \begin{cases} e^{-(f(C')-f(C))/\alpha_k} & : f(C') \geq f(C) \\ 1 & : f(C') < f(C) \end{cases}$$

(2.5)
Where the term $\alpha_k$ represents the control parameter as per Equation 2.6,
\begin{equation}
\alpha_k = 1000 \times e^{k/20} \tag{2.6}
\end{equation}
and function $f(\ )$ represents distance cost shown by Equation 2.7,
\begin{equation}
f(C) = \sum_{i=1}^{n} \min(d^2(i,c)) \tag{2.7}
\end{equation}
Here, $d^2(i, c)$ in Equation 2.7 is the distance between member node $i$ and candidate cluster-head node $c$. The BS broadcasts messages that contain the cluster-head ID and member nodes for each cluster, once the cluster heads and associated cluster members are determined. If the cluster-head ID of a node matches its own ID, the node is itself a cluster-head; otherwise, TDMA slot is determined by the node for data transmission. Node now goes to sleep state until its slot for transmission arrives.

### 2.7.2 PEGASIS and HPEGASIS

PEGASIS (Lindsey, Raghavendra, and Sivalingam) and Hierarchical PEGASIS (Lindsey and Raghavendra) are enhancements of LEACH protocol. Here, chain of sensor nodes is formed to transmit to and receive from a close neighbour for static and homogeneous network. Cluster heads are elected turn-wise for transmitting to base station, thereby reducing average energy spent by each node. Construction process is as shown in Figure 2.7. In PEGASIS, node $n0$ sends its data to node $n1$, node $n1$ aggregates the received data with its own data and sends it to node $n2$, a leader node. Similar process is followed for right hand side neighbors of leader node. Each such chained aggregated message is sent to BS by leader node. On expiry of nodes, chain is recreated considering the energy levels of node in deciding the leader node. PEGASIS is far more efficient than LEACH. However, it introduces significant overhead and
CHAPTER 2. LITERATURE SURVEY

propagation delay for the last nodes in the chain. In addition, it needs dynamic topological adjustments, as the nodes in the chain need to know the energy level of the neighbors. Accordingly data is routed to neighbors by a particular node. Presence of only one leader node can be the performance bottleneck. Another drawback of this scheme is that, if the thresholds are not reached, the nodes will never communicate. Objective of Hierarchical PEGASIS, an enhancement to PEGASIS, is to decrease delay in transmitting packets to base station. HPEGASIS tries to reduce the message latency of PEGASIS by establishing multiple parallel communication groups, using CDMA along-with spatial transmissions. HPEGASIS improves simultaneous transmissions of data messages. Figure 2.8 depicts the data gathering chain of Hierarchical PEGASIS. A hierarchy of nodes is created where, lower level node transmits data to upper level node. The node at highest level in the tree sends data to BS. HPEGASIS reduces the delay present in PEGASIS by considerable amount.

![Data gathering in chain of Hierarchical PEGASIS](image)

Figure 2.8: Data gathering in chain of Hierarchical PEGASIS

2.7.3 TEEN and APTEEN

Threshold sensitive Energy Efficient sensor Network protocol (TEEN)\cite{Manjeshwar} and the AdaPtive Threshold sensitive Energy Efficient sensor Network protocol (APTEEN) are two proposals for time-critical applications. TEEN, based on hierarchical approach uses data-centric mechanism whereby, clusters are formed by close nodes, repeating the process until base station is reached. Architecture of TEEN/APTEEN is as shown in Figure 2.9. After formation of clusters, cluster-heads broadcast two thresholds (hard and soft thresholds for sensed attributes) to other nodes. Hard threshold is the benchmark for triggering the node to transmit attribute information to the cluster head. Hence, information is transmitted only
when attribute value is in interest range. Soft threshold reduces transmissions when there is very little or no change in sensed attribute. TEEN fails when deployed in applications which require periodic reporting. APTEEN is a hybrid protocol and an enhancement of TEEN which changes periodicity or threshold values depending on application type and user needs. APTEEN supports historical queries, one-time queries and persistent queries to monitor an event for a period of time. Timeline for TEEN and APTEEN is as shown in Figure 2.10. Disadvantage of TEEN and APTEEN is overhead and complexities involved in creating multi-level clusters.

2.7.4 HEED

Hierarchical Energy Efficient Distributed clustering (Younis and Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks") applies hybrid approach for cluster head selection. It considers residual energy of the nodes as well as the nodes closeness with its neighbor or the neighbor degree of the node. This protocol extends network lifetime by way of balanced energy usage with uniform distribution of cluster-heads in the deployed network. It is scalable over large network sizes and performs load balancing within clusters. However, the requirement of frequent computation of communications cost and broadcasting among neighbors degrades performance of HEED. It can only work in a satisfactory way when the network topology is uniform in nature. HEED also does not guarantee
Figure 2.10: Time line for the operation of TEEN and APTEEN: (a) TEEN when cluster-heads are to change, new values of the hard threshold and soft threshold are broadcast; (b) APTEEN the cluster-heads broadcast attributes, thresholds, schedule and count time
the number of selected cluster head. If the energy of all nodes is similarly low, most
nodes can become cluster head.

2.7.5 K-means based Clustering

K-means is the simplest clustering method which partitions the data set into k number
of clusters. K-means algorithm is used as clustering method in several application
areas like image processing, artificial intelligence and machine learning. K-means
algorithm uses Euclidian distances of the points to be clustered. Cluster head selection
process is based on residual energies of nodes. Sasikumar et al. (Sasikumar and
Khara) have proposed K-means based clustering wherein, the objective function to
be minimized is given by Equation 2.8,

\[ F = \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| x_{i}^{(j)} - c_j \right\|^2 \]  \hspace{1cm} (2.8)

where \( \left\| x_{i}^{(j)} - c_j \right\|^2 \) is a chosen distance measure between a data point \( x_{i}^{(j)} \) and the
cluster centre \( c_j \). It is an indicator of the distance of the \( n \) data points from their
respective cluster centres. The clustering process adopted by the authors is as follows:

K-Means Algorithm (Sasikumar and Khara)

a. Take \( k \) number of centroids initially at random places

b. Calculate the Euclidian distance from each node to all centroids

c. Recalculate the positions of centroids in each cluster and check for the change
   in position from the previous one

d. If there is change in position of any centroid, then go to STEP 2, else the clusters
   are finalized and the clustering process ends

Tan et al. (Tan, Gong, and Chen) have proposed balanced parallel K-means clus-
tering (BPK-means) algorithm, wherein they strictly assign fixed number of members
to the cluster during clustering process. Here, balanced clustering means every clus-
ter is assigned equal number of members. The same concept is formally presented by
Equation 2.9

\[ \frac{N}{K} - \delta \leq |C_j| \leq \frac{N}{K} + \delta, \delta = 0 \]  \hspace{1cm} (2.9)
where $C_j$ means number of cluster members in a cluster $j$. While doing this kind of clustering to maintain equal number of members in every cluster, sometimes nodes far away from cluster-head may also be assigned to cluster-head. BPK-Means is a clustering protocol where total spatial distance between cluster heads and member nodes has been optimized. Initially, base station announces random set of nodes as temporary cluster heads (TCH). Then after, the TCHs find their neighbors and cluster center. Then TCHs exchanges this cluster structure information with its peer TCHs. Now the TCHs again select members as per the distance of the member nodes from the computed cluster center. This process is repeated for predefined number of iterations. Final CHs are elected depending on the residual energy and distance from cluster center. BPK-means does balanced clustering so that every cluster is assigned almost equal number of member nodes so that, the cluster-heads or cluster-members are not overloaded with the sensing, sending, receiving or computing duties. The authors have also applied the optimal clustering strategy (Smaragdakis, Matta, and Bestavros) proposed by Smaragdakis et al.

### 2.7.6 Fuzzy C-Means based Clustering

Fuzzy c-Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. Fuzzy C-Means algorithm was originally proposed by (Bezdek, Ehrlich, and Full). FCM based clustering method can be also extended for wireless sensor networks for centralized clustering protocol. In (Hoang, Kumar, and Panda, “Realisation of a cluster-based protocol using fuzzy C-means algorithm for wireless sensor networks”), authors have applied Fuzzy C-Means clustering technique to create clusters in the network. Degree of membership is assigned to each node to associate with different cluster heads instead of the node being a member of just one cluster. Hence, nodes which are close to the boundary of any cluster may become members of neighboring clusters as per the assigned degree of membership. Required tolerance measures can be followed depending on the accuracy of the clustering in practice. Number of iterations completed by the FCM algorithm determines the accuracy of the degree of membership. Measure of accuracy is computed using the degree of membership in every iteration. The largest of these values across all data points in all clusters is
considered. When the difference between coefficients in two iterations is less than a threshold or specified number of iterations are reached, convergence is said to have achieved (Hoang, Kumar, and Panda, [Fuzzy C-Means clustering protocol for Wireless Sensor Networks](#)).

For the first round, base station chooses cluster head of each cluster; subsequently, the current cluster head selects the next cluster head among the other nodes. Sensor nodes transmit residual energy information along with data packets to the cluster head. CH choose the node with the highest residual energy and the nearest to cluster center to be the cluster head in the next round. This protocol is as good as fixed clustering. Deployment region is divided into similar sized sub-regions. Each sub-region is fixed throughout the lifetime of the protocol. This may not be a suitable method as, when the nodes start expiring, the overall topology of the network will change. So, the fixed clustering will not adapt to this topological changes happening over the time. Also, during the data transmission phase in every message, energy information is sent as part of data packet, which unnecessarily consumes energy.

Fuzzy C-Means algorithm is used for centralized clustering algorithms. In this algorithm, base station decides cluster-heads and cluster members according to the location information and residual energy of nodes. The aim of cluster formation is to optimize the objective function (Raghuvanshi et al., [Optimal number of clusters in wireless sensor networks: An FCM approach](#)) shown in Equation 2.10.

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} \delta_{ij}^m ||x_i - c_j||^2,$$  \hspace{1cm} \text{where} \hspace{1cm} 1 \leq m < \infty \hspace{1cm} (2.10)$$

where degree of membership of a node is given by Equation 2.11.

$$\delta_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{||x_i - c_j||}{||x_i - c_k||} \right)^{m-1}} \hspace{1cm} (2.11)$$

The norm $||x_i - c_j||$ measures the similarity (or closeness) of the data point $x_i$ to the centre vector $c_j$ of cluster $j$. Cluster centers are computed with

$$C_j = \frac{\sum_{i=1}^{N} \delta_{ij}^m x_i}{\sum_{i=1}^{N} \delta_{ij}^m} \hspace{1cm} (2.12)$$

The degree of membership for data point $i$ to cluster $j$ is initialized with a random value $\theta_{ij}$, $0 \leq \theta_{ij} \leq 1$, such that $\sum_{j}^{C} \delta_{ij} = 1$. Initially, the degree of membership
is assigned randomly. As the algorithm progresses, clusters are formed iteratively. The algorithm converges with termination conditions, least overlapping membership of members with neighboring clusters and the fuzziness coefficient of the membership. Once cluster centers are found, actual nodes near to that cluster center can be assigned the role of cluster-head depending on the residual energy available with the node. In (Raghuvanshi et al., "Optimal number of clusters in wireless sensor networks: a FCM approach"), authors have used FCM method for clustering. To calculate the optimal number of clusters, authors have used FCM algorithm for partitioning the sensor nodes into clusters and thereby determining the optimal number of clusters utilizing Euclidian distance norm. Xie and Benis (XB) index has been used for validation of the optimal number of clusters. Authors have also considered the optimal clustering suggested in (Smaragdakis, Matta, and Bestavros) by Smaragdakis et al.

In (Chattopadhyay, Pratihar, and Sarkar), authors have compared Fuzzy-C-means clustering with Entropy-based Fuzzy Clustering (EFC). In FCM, the cluster centers are chosen at random. So they might be out of the data set and are virtual. Cluster centers and membership values of the data points with them are updated through a few iterations. Entropy-based Fuzzy Clustering (EFC) algorithm works based on a similarity-threshold value. In EFC, the cluster centers are chosen from data points and are real. In (Medhat, Ramadan, and Talkhan), authors have applied FCM clustering to create overlapping cluster layers wherein, the member node may be associated with one cluster for measuring one parameter while it may be part of other cluster(s) if it is able to measure and report other parameter(s) as well. Authors have shown that their FCM clustering with multimodal approach outperforms LEACH with respect to various performance parameters.

K-Means and FCM clustering are the conventional techniques to form clusters. Center point of the cluster is defined by minimizing distance between the nodes and the nearest cluster centers. Both algorithms initialize $k$ cluster centers randomly. Nodes are then assigned to a cluster which is having its cluster center nearest to the node under consideration. In K-Means, hard partitioning of nodes is performed, whereas, in FCM, one node can be associated to multiple clusters as per the degree assigned by FCM. These conventional clustering techniques are sensitive to starting points. They also have tendency to frequently converge at local optima or it may
diverge altogether. Hence, if search starts near the local optima, global optima may be missed (Chong and Zak).

2.8 Evolutionary Algorithms for Clustering

Evolutionary algorithms (EA) eliminate the difficulties related to conventional techniques. EAs can be used as an alternative to conventional techniques. Evolutionary algorithms aim to find an optimal solution for an optimization problem with trial-and-error in an efficient way in terms of convergence time (De Jong). Clustering is considered to be an NP-Hard problem. Hence, evolutionary algorithms have been successfully applied to such problems.

2.8.1 Genetic Algorithms

Encoding of optimization objective into arrays of either character strings or binary strings (chromosome representation) forms the essence of Genetic Algorithms. The population of GA contains various set of these encoded strings called chromosomes. Genetic operators include crossover, mutation and inversion. Crossover operator exchange segments of two solutions. The crossover operation swaps the substring of the two parent solutions under consideration at the specified crossover point. The inversion operation reverses the substring specified by the starting and ending point for the solution under consideration and generates new solution. The mutation operation randomly selects one or more bits in the given solution and complements the existing bit value and thus generates new solution. Mutation operator deals with identifying and altering one or more specific bits randomly in the individual string and changing their values (Iyengar and Brooks). Genetic operators are performed on candidate solutions. In each iteration cycle, genetic operators are performed to produce a population of new candidate solutions within pre-defined operation times. Initial population is evolved by generating new generation individuals using crossover and mutation of bits. Crossover of two randomly selected parent chromosomes and the mutation of some random bits is performed with predetermined probability. Fitness value is used to decide whether a new individual is to be selected or not. According to their corresponding fitness values, good solutions are retained and unqualified ones are screened out. The above process does not stop until either the algorithm
converges upon a particular solution or the allocated execution time is exhausted.

Genetic Algorithm is useful to find global and local optima in applications areas such as solving combinatorial and nonlinear optimization problems with complicated constraints or non-differentiable objective functions, thereby avoiding combinatorial explosion. For improving performance of Genetic Algorithms, literature suggests improvements in decoding setting, choice of initial population, fitness function, genetic operators, control parameters \((p_c, p_m)\) and redefined algorithm structures. The pseudo code for Genetic Algorithm \cite{Yang} is as shown in Algorithm 1. Initially GA

\begin{algorithm}
\caption{Genetic Algorithm Pseudo-code}
\begin{algorithmic}[1]
\STATE Objective function \(f(x), x = (x_1, \ldots, x_n)^T\)
\STATE Encode the solution into chromosomes (binary strings)
\STATE Define fitness \(F\) (eg, \(F \propto f(x)\) for maximization)
\STATE Generate the initial population
\STATE Initialize Probabilities of crossover \((p_c)\) and mutation \((p_m)\)
\WHILE \((t < \text{Max number of generations})\) \DO
\STATE Generate new solution by crossover and mutation
\IF \((p_c > \text{rand})\) \THEN
\STATE crossover
\ENDIF
\IF \((p_m > \text{rand})\) \THEN
\STATE mutate
\ENDIF
\STATE Accept the new solutions if their fitness increase
\STATE Select the current best individuals for new generation
\ENDWHILE
\STATE Decode the results and visualisation
\end{algorithmic}
\end{algorithm}

based clustering for WSN was proposed by Jin et al \cite{Jin}. Authors attempted clustering with minimizing total network distance. A binary individual, \(I\), represents the role of the node. If \(I\) is set to 1, the node is cluster-head and the node is a regular member if \(I\) is zero. The fitness of \(I\) is computed with Equation \ref{eq:2.13}.

\begin{equation}
F_{GA}(I) = \omega \times (TD - D) + (1 - \omega) \times (N - N_{CH}) \tag{2.13}
\end{equation}
Here, \(TD\) is the total distance of all member nodes to BS. Distance \(D\) indicates the sum of distances from member nodes to CHs and sum of distances of CHs to BS. \(N\) is total number of nodes and \(N_{CH}\) is number of cluster-heads. Lower value of \(D\) or \(N_{CH}\) is preferred choice for longer network lifetime. Objective function is maximized here for good solution. Hierarchical Cluster based Routing protocol (HCR) [Matin and Hussain] adds standard deviation of cluster distances, estimated transfer energy and number of transmissions for fitness calculation. Fitness function, \(F_{HCR}\), is calculated as per Equation 2.14,

\[
F_{HCR} = \sum_i \alpha(w_i, f_i), \forall f_i \in \{TD, DD, E, SD, T\} \tag{2.14}
\]

Here, \(w_i\) is the preassigned arbitrary weight for adjusting the fitness parameters. \(TD\) represents the sum of the distances from sensor nodes to cluster-heads and the distance from cluster-heads to the sink. The term \(DD\) is the direct distance to the sink and it is the sum of the distances from nodes to the sink. Third term \(E\) represents the energy consumed to transfer the aggregated message from the clusters to the sink. \(SD\) shows the standard deviation in cluster distances which actually depends on the deployment type. Last term \(T\) is the number of transmissions assigned by the base station considering the current network conditions and energy levels. Mudundi et al. proposed Genetic Clustering Algorithm (GCA) [Mudundi and Ali] for dynamic formation of clusters. The network lifetime is improved in GCA with minimization of energy dissipation. The fitness function is computed using Equation 2.15,

\[
F_{GCA}(I) = \omega \times N_{CH} + (1 - \omega) \times ND \tag{2.15}
\]

The term \(ND\) represents the Euclidian distance of all nodes in every cluster to their respective CH. Fitness function of GA [Shakshuki, Malik, and Sheltami] proposed by Shakshuki et al. considers two more parameters as compared to fitness (Eq. 2.14) of HCR. In Equation 2.16, \(RE\) represents the remaining energy of sensor nodes and \(FT\) indicates number of frames received by BS. The fitness function for this GA is as per Equation 2.16,

\[
F_{GA}(I) = \sum_i \alpha(\omega_i, f_i), \forall f_i \in \{TD, DD, E, SD, T, RE, FT\} \tag{2.16}
\]

In EAERP (Khalil and Attea), authors applied GA for dynamic clustering for centralized routing. An individual solution, \(I^k\), \((\forall k \in \{1, ..., n\})\), is evaluated using the
fitness function given by Equation (2.17).

\[ F_{EAERP}(I^k) = \left( \sum_{i=1}^{nc} \sum_{s \in c_i} E_{TX_{s,CH_i}} + E_{RX} + E_{DA} \right) + \sum_{i=1}^{nc} E_{TX_{CH_i,BS}} \]  

(2.17)

Here, \( nc \) represents number of CHs, \( s \) is member node associated with \( i^{th} \) cluster-head and \( E_{TX_{s,CH_i}} \) is the energy required to transmit information from member node to cluster-head. Energy dissipated during receiving data is represented by \( E_{ER} \) and \( E_{DA} \) is the energy dissipated during data aggregation process. Initially random \( I^k \) solutions are generated and manipulated with crossover and mutation operations. Authors have focused entirely on reducing energy consumption.

Kuila et al (Kuila, Gupta, and Jana) applied GA for centralized evolutionary protocol. Authors have tried to balance the load on the cluster-head nodes represented as gateway nodes. However, the energy assignment to these gateway nodes is biased. Compared to regular members, the gateway nodes are assigned five times more energy. ERP is (Attea and Khalil) a centralized evolutionary routing protocol that uses GA for clustering process.

\[ F_{ERP} = \omega \times f1 + (1 - \omega) \times f2 \]  

(2.18)

where \( f1 \) is Compactness/\( d_{min} \) and \( f2 \) is number of CHs. The compactness is defined as shown in Equation (2.19).

\[ Compactness = \sum_{i=1}^{CHs} \sum_{n \in C_i} d(n, CH_i) \]  

(2.19)

The Minimum inter-cluster distance \( d_{min} \) is represented as,

\[ d_{min} = \min_{\forall C_i, C_j, C_i \neq C_j} \{d(CH_i, CH_j)\} \]  

(2.20)

Authors have emphasized on intra-cluster distance and inter-cluster distance. But the residual energy parameter has been neglected during clustering, which is an important parameter for these types of protocols.

The crossover operator design is one of the key factors which affects the performance of GA. A well designed crossover operator can assure the retention of desired segments of old solutions, which can be inherited by the new generation solutions. Another factor influencing GA performance is probabilities of crossover and mutation
Searching results are affected by the probability of crossover ($p_c$). If probability of crossover is too low, it will result into inefficient evolution, so generally large value of $p_c$ is preferred. If probability of mutation is high, there will be oscillations around the optimal solution. As there is no unified formulation of fitness function, literature suggests different objective function fitness value settings. GA algorithm complexity increases with increasing number of nodes since GA is centralized algorithm [Kulkarni, Forster, and Venayagamoorthy]. Search capability in GA is reduced due to lack of diversification; thereby, leading to premature convergence and higher probability of obtaining a local optimum [Fogel].

### 2.8.2 Swarm Intelligence

Swarm Intelligence implements the Artificial Intelligence by modelling the group behaviour of biological species, such as colonies of ants, groups of fishes, bees and flocks of birds. The Ant Colony Optimization algorithm [Dorigo] is one class of swarm intelligence algorithms. It was introduced as a heuristic method for solving combinatorial optimization problems. It is inspired by the behaviour of ant colonies finding the shortest path between their nest and a food source. The principle of ACO algorithms is to find the optimal solution through the exchange of information between individuals and mutual cooperation [Dorigo and Birattari].

There are certain advantages of using Ant Colony Optimization algorithms. Individuals work in a distributed way, hence there is no requirement of centralised control in the system. Therefore, the ultimate solution of the whole system may be unaffected if one or some individuals are non-functioning or failed. Individuals work in an independent and cooperative way. It provides a more extensible way to build the system, since the individuals communicate with each other in an indirect way. So the communication costs will not rise much when the group scale increases. It provides a convenient way of obtaining realisations of the system because the function is simple and the execution time of each individual is short [Zhe et al.]. Another advantage of ACO is that, it is highly adaptive to topology change and the method for updating routing table when changes happen. The most important aspect of any evolutionary method is the fitness function, which determines selection criterion of a problem. Pseudo code of basic steps of Ant Colony Optimization is as shown in Al-
For routing, ants follow the paths with higher pheromone concentrations.

**Algorithm 2** The pseudo code of an ant colony optimization algorithm

1: Objective function $f(x, x = (x_1, \ldots x_n)^T$

2: Define pheromone evaporation rate $\gamma$

3: **while** (t\text{Max number of iterations}) **do**

4: **for** loop over all $n$ nodes **do**

5: Generate new solutions

6: Evaluate the new solutions

7: Mark better routes with pheromone $\delta\phi_{ij}$

8: Update pheromone: $\phi_{ij} \leftarrow (1 - \gamma)\phi_{ij} + \delta\phi_{ij}$

9: **end for**

10: Select the current best solution $\text{best.route}$

11: **end while**

12: output the best solution $\text{best.route}$ and pheromone distribution

The probability of ants at a node $i$ choosing route from node $i$ to node $j$ (among $n_d$ nodes) is given by Equation 2.21

$$p = \frac{\phi_{ij}^\alpha s_{ij}^\beta}{\sum_{i,j=1}^{n_d} \phi_{ij}^\alpha d_{ij}^\beta} \quad (2.21)$$

Here,

- $\alpha, \beta (\alpha, \beta > 0)$ are predefined parameters that control the influence of heuristic information. Their typical values are $\alpha \approx \beta \approx 2$

- $\phi_{ij}$ is the pheromone concentration on the route between $i$ and $j$

- $s_{ij}$ is the distance of the same route

- $d_{ij}$ is the desirability of the same route, $d_{ij} \propto \frac{1}{s_{ij}}$, which implies that shorter routes will be selected due to their shorter travelling time. This results in higher pheromone concentration on these routes.

Pheromone concentration ($\phi_{ij}$) varies exponentially with respect to time for a constant rate ($\gamma$) of pheromone evaporation as shown in Equation 2.22

$$\phi(t) = \phi_0 e^{-\gamma t} \quad (2.22)$$
If $\gamma t << 1$, $\phi(t) \approx (1 - \gamma t)\phi_0$. For the unitary time increment $\Delta t = 1$, evaporation is approximated as $\phi_{ij}^{t+1} \leftarrow (1 - \gamma)\phi^t$. Thus the pheromone update formula is as per Equation 2.23:
\[
\phi_{ij}^{t+1} = (1 - \gamma)\phi_{ij}^t + \delta\phi_{ij}^t
\]
Here,

- $\phi_0$ is the initial concentration of pheromone,
- $t$ is the current time, $\gamma \in [0, 1]$,
- $\delta\phi_{ij}^t$ is the increment of the amount of pheromone deposited at time $t$ at route $i$ to $j$ when an ant travels a distance $L$. Usually $\delta\phi_{ij}^t \propto 1/L$, only if there are no ants on a route, $\delta\phi_{ij}^t$ will be equal to 0.

The advantages of ACOs are self-adaptation, self-organization, flexibility, robustness, parallel computing, no need of prior information. ACO has been successfully applied to combinatorial optimization problems such as Internet routing problem, travelling salesman problem, assignment problems, scheduling problems and vehicle routing problems (Yang). There are three important issues which affect the performance of ACO algorithms, the population size, the probability of choosing a route and the evaporation rate of the pheromone. The researches have integrated ACO with other clustering algorithms or alternatively use different overall strategies with different pheromone attributes to solve clustering problems (Zhe et al.). Ant colony algorithms have a slow speed of convergence if there is little or no information in the initial searching (Yao, Liu, and Wang).

Inspired by movement of organisms in a bird flock or fish school, Kennedy et al. (Kennedy and Eberhart) developed Particle Swarm Optimization (PSO). Each solution in PSO represents a particle in the search space. These particles fly through the given problem space by following the local best known position. During this flight, every particle fine tune its position as per its own experience, and that of neighboring particles. While adjusting this way, particles use the best position found so far by the particle under consideration and its neighbors. Set of particles neighboring the particle and its experience defines the swarm direction. PSOs are unable to maintain
desired levels of diversity in population. Also PSOs are not good at maintaining balance between global and local searches. Hence, suboptimal solutions are obtained prematurely. Random components though present in PSO, do not add adequate diversity during optimization process. Frequent collisions of particles especially on to the leader in the search space can be detected (Montalvo et al.). Hence, effective population size is reduced in comparison to actual, which reduces the effectiveness of the PSO algorithm. Performance of PSO algorithm is also influenced by several tuning parameters.

2.8.3 Harmony Search Algorithm

Harmony Search Algorithm was initially proposed by Geem and Kim (Geem, Kim, and Loganathan, “A New Heuristic Optimization Algorithm: Harmony Search”). It is analogous to how a musician plays music. Musician generally plays a famous piece of music or something similar to known harmony or a completely new harmony is generated by him. These three options are correlated to memory based solutions or mutated existing solutions or randomized solutions respectively. With this concept, HSA based optimization method has been suggested by Geem and Kim (“A New Heuristic Optimization Algorithm: Harmony Search”). It has been used in many applications as reported in the literature for function optimization problem in various problem domains by researchers. The domains include engineering problems including structural designs, water network designs, dam scheduling, groundwater management, soil stability analysis, vehicle routing, ecological conservation, project scheduling, heat exchanger designs, Internet routing, cell phone networking and clustering in WSN as well. As per (Manjarres et al.), HSA has been used for various optimization problems in engineering, robotics, telecommunications, medical, power and energy, construction etc. HS algorithm features a great potential and efficiency while seeking near-optimal solutions to computationally hard optimization problems.

HSA has fewer mathematical requirements. It does not require initial value settings of decision variables. HSA generates a new solution vector after considering all the existing vectors, while GA only considers two parent vectors. HSA does stochastic random search. Hence, derivative information is not required (Geem, “Novel derivative of harmony search algorithm for discrete design variables”). Traditional
calculus-based derivative gives information of search direction and step size at certain
single vector for a function which has continuous variables. The stochastic derivative
in HSA gives information of probabilistic inclination to select certain discrete point
based on multiple vectors stored in HM for a function which has discrete variables.
With increasing iterations, optimal and neighboring values have higher chances to
be selected. HSA was applied for clustering process in WSN by Hoang, Kumar and
Panda (Hoang et al., “A Robust Harmony Search Algorithm Based Clustering Proto-
col for Wireless Sensor Networks”). Authors find the best clusters using intra cluster
distance and residual average energy. Nodes with higher residual energy are elected
as cluster head along-with consideration of intra cluster distance of cluster members
from cluster heads. Harmony Memory is defined as matrix HM (Eq. 2.24) having
randomly selected member elements in every row. Each vector in the matrix HM
indicates that elements in the row may be cluster head after the algorithm converges.
These vectors are partially altered or replaced altogether with harmony improvisation
logic.

\[
HM = \begin{bmatrix}
V_1^1 & V_1^2 & \ldots & V_1^k \\
V_2^1 & V_2^2 & \ldots & V_2^k \\
\vdots & \vdots & \ddots & \vdots \\
V_{HMS}^1 & V_{HMS}^2 & \ldots & V_{HMS}^k
\end{bmatrix}
\begin{bmatrix}
F_1^1 \\
F_2^2 \\
\vdots \\
F_{HMS}^{HMS}
\end{bmatrix}
\] (2.24)

Here, \(F^i\) represents the calculated fitness of the solution vector \(i\). Once the HM is
derived, a new harmony is generated. The new Harmony from HM is derived using
Harmony Memory Consideration Rate (HMCR). A new harmony may be directly
selected from the candidates present in the HM with probability HMCR or it may
be generated fresh from the pool of all the sensor nodes which are alive using the
probability 1 - HMCR.

\[
V_j' \left\{ \begin{array}{l}
V_j' \in HM(j) \quad \text{with probability HMCR} \\
V_j' \in V_{\text{candidate}} \quad \text{with probability } (1 - \text{HMCR})
\end{array} \right.
\] (2.25)

If the new harmony has been derived from HM itself, it is further fine-tuned with
Pitch Adjustment Rate (PAR). With probability PAR, member elements in the new
harmony may be mutated with other members in the same position in the Harmony Memory having residual energy higher than the average energy. Otherwise the new harmony will be left as it is with the probability 1-PAR. Pitch Adjustment for selected $V_j'$ is,

$$V_j' \leftarrow \begin{cases} 
V_j^n \in HM & \text{with probability PAR} \\
V_j' & \text{with probability } (1 - PAR)
\end{cases}$$

(2.26)

The algorithm 3 shows how new harmony is improvised iteratively.

**Algorithm 3** The procedure of improvising a new harmony

1: \textbf{for} $j = 1$ to $k$ \textbf{do}
2: 
3: \quad //Harmony Memory consideration
4: \quad \textbf{if} $U(0, 1) \leq HMCR$ \textbf{then}
5: \quad \quad Choose a harmony $V_i$ from HM randomly, $i \in [1..HMS]$
6: \quad \quad $V_j' = V_i^j$
7: \quad \textbf{end if}
8: \quad //Pitch Adjustment
9: \quad \textbf{if} $U(0, 1) \leq PAR$ \textbf{then}
10: \quad \quad $V_j' = \text{Neighboring element of } V_j'$ in HM
11: \quad \textbf{end if}
12: \quad \textbf{else}
13: \quad \quad $V_j' = \text{Pick random element from pool of all nodes}$ //random selection
14: \quad \textbf{end if}
15: \textbf{end for}

The objective function to be minimized is as follows,

$$f_{\text{obj, min}} = \alpha \times f_1 + (1 - \alpha) \times f_2$$

(2.27)

where,

$$f_1 = \max_{j \in (1,k)} \left\{ \frac{\sum_{\forall \text{node}_i \in C_j} d(\text{node}_i, CH_j)}{|C_j|} \right\}$$

(2.28)

and

$$f_2 = \sum_{j=1}^{k} \left\{ \frac{\sum_{\forall \text{node}_i \in C_j} V_i^{res}}{V_{CH_j}^{res}} \right\}$$

(2.29)

The terms in the above equations are explained here in brief.
a. $f_1$ is the maximum of the ratio of Euclidean distance of nodes with cluster heads to the number of nodes of cluster $C_j$, $|C_j|$;

b. $\sum_{\forall \text{node}_i \in C_j} d(\text{node}_i, CH_j)$ is the sum of the distance between the nodes and their cluster-heads, and $d(\text{node}_i, CH_j)$ is the distance between $\text{node}_i$ and all cluster-heads $CH_j$, $j \in [1, k]$;

c. $k$ is the number of cluster-head nodes;

d. $f_2$ is the ratio of the energy of all the alive nodes in the network with the current energy of cluster-heads in the current round.

The function $f_1$ finds the maximum of the average member distance with the cluster head of every cluster. The function $f_2$ finds the effect of residual energy of CH on the sum of total energy of every member in every cluster. The objective function is designed in such a way that this term’s contribution is minimized, thereby ensuring that members with higher residual energy become cluster-heads. Hong et al. (Hoang et al., “A Robust Harmony Search Algorithm Based Clustering Protocol for Wireless Sensor Networks”) have suggested HSACP for clustering in WSN. Authors compared performance of HSA with various protocols and proved that HSA performs better than FCM, K-Means and several other evolutionary algorithms like Genetic Algorithms and Particle Swarm Optimization. Hoang et al. (Hoang et al., “Real-Time Implementation of a Harmony Search Algorithm-Based Clustering Protocol for Energy-Efficient Wireless Sensor Networks”) applied HSA to real-time implementation on sensor nodes and compared HSA with FCM based centralized clustering (FCMCP) and LEACHC protocols. Realtime HSACP shows improved lifetime compared to FCMCP and LEACHC. HSLATC (Nikravan and Jameii) applies HSA for improving connectivity of nodes using topology control, thereby reducing the energy consumption of sensor networks. DCHS (Alia et al.) is also an application of HSA for designing energy efficient protocol. In DCHS, authors have integrated FCM clustering approach with Harmony Search Algorithm for clustering and cluster-head election is done dynamically. Authors maximize the objective function as per Equation (2.30):

$$CH_{obj} = \max_{\forall cd_i \in CD_c} \left\{ \frac{E_{cd_i} \times q}{\alpha \times f_1 + (1 - \alpha) \times f_2} \right\}$$ (2.30)
\[ f_1 = \sum_{j=1}^{n} \| node_{jc} - cd_i \| \]  
(2.31)

\[ f_2 = \| cd_i - BS \| \]  
(2.32)

Where \( f_1 \) is the Euclidian distance of nodes from candidate CH nodes and \( f_2 \) indicates the distance of candidate CH nodes with Base Station.

### 2.9 Summary

The modern search algorithms like Simulated Annealing, Evolutionary Algorithms, Genetic Algorithms, Ant Colony Optimization, Particle Swarm Optimization and Harmony Search Algorithm can be broadly categorized as metaheuristic algorithms (Yang). These algorithms need strong consideration of two factors while applying them to specific application. First is intensification and the other is diversification. Intensification helps in exploration of all neighboring nearly optimal or sub-optimal solutions; whereas diversification focuses on randomization while searching for the solution. If the algorithm intensifies more, it might get trapped in local optima. On the other hand, if it does strong diversification, convergence time may be very large. High quality feasible solution from local region may be acceptable in the situation like time critical applications of sensor networks. Harmony Search Algorithm incorporates both the intensification as well as diversification with pitch adjustment and random harmony memory selection respectively. Search space is explored widely and efficiently in HS algorithm through randomization. Pitch adjustment ensures that the new solution is reasonably good, or not too far from good solutions present in HM. Therefore, randomization ensures that HS comes out of the local minima while, convergence to the value surrounding global minima is intensified using pitch adjustment (Hoang et al., “A Robust Harmony Search Algorithm Based Clustering Protocol for Wireless Sensor Networks”). It can be concluded that, HSA avoids the problem of premature convergence of GA. It also avoids the shortcoming of PSO, like inability in maintaining desired levels of population diversity and the balance between local and global searches. Performance of HSA is also less sensitive to the selected control parameters. Some of the existing centralized protocols like LEACHC, KMCP, FCMCP and HSACP were implemented and analyzed. The performance comparison
of these protocols was done to examine how these protocols behave with different network sizes and different BS locations. The next chapter describes the implementation and analyzes performance of these conventional and evolutionary protocols in detail.