CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of spatially distributed sensor nodes that are designed to monitor an environment as shown in Figure 1.1. The sensor nodes are limited battery powered sensor nodes with non rechargeable batteries. These sensor nodes are deployed randomly or with a static strategy in the desired remote territory to collectively fulfill the requirements of WSNs. All the sensor nodes form a network amongst themselves using different optimal mechanism(s) to establish a communication infrastructure which helps in transmission of the desired information like environmental and physical conditions as temperature, pressure, motion, sound, etc to a central authority efficiently. The central authority is referred to as sink or base-station (BS) and has ample power supply unlike constrained power of sensor nodes. This has increased the use and applicability of WSNs with unattended sensors in vivid areas like vehicular movement, weather monitoring, security-surveillance, industry applications, health monitoring, etc [1, 2].

Some restrictions with the sensor nodes used in WSNs other than limited power battery are limited processing capability, low bandwidth for communication and less memory space. Sensor nodes communicate over short distances using radio frequency channel to transmit the sensed information. Although each sensor has the capability of limited processing but collaborative efforts have the ability to analyze the desired environment in great details.

They sense the environment continuously or after every fixed quantum and transmit the information either periodically or on the occurrence of some event to the BS. The transmission of information to the sensor nodes to the BS can be done in a variety of ways as proposed by many routing protocols to increase the efficiency of the network.

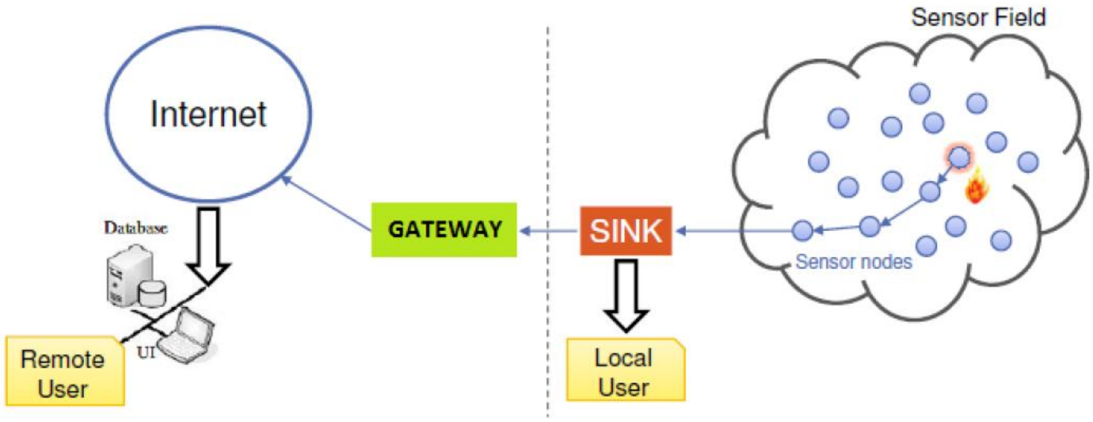


Figure 1.1: Wireless Sensor Network

WSNs form a subset of ad-hoc networks as shown in Figure 1.2. WSNs differ from ad-hoc networks in the way that sensor nodes impose many restrictions due to their limited energy source, memory storage capacity and processing capabilities. Thus, different protocols are required for WSNs. The issues that needs to be considered in WSNs:

- Scalability: WSNs consist of large number of sensor nodes as compared to ad-hoc networks. Hence, solutions must be scalable.
- Efficient Energy Usage: In WSNs, sensor nodes have limited power supply as compared to ad-hoc networks. Therefore, energy consumption is an important issue in WSNs.

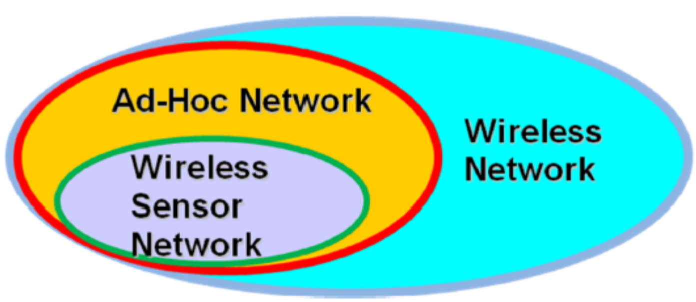


Figure 1.2: Wireless Sensor Networks as subset of Wireless Network

WSNs researchers work towards accomplishment of several requirements. Some of them are:

- There are bandwidth constraints that it should be sufficient for WSNs performance.

- Increasing reliability in the network.
- Efficient use of energy as sensors are battery powered with limited power supply.
- There is restriction at how much hardware resources can be used.
- Selection of routing protocol is also important.
- Efficient network architecture and topologies.
- To include security aspects like integrity, confidentiality and authentication.

1.1.1 Sensor Node Architecture

The sensing circuitry of the sensor node is capable of measuring ambient conditions related to surroundings of the sensor which is then transformed into an electric signal.

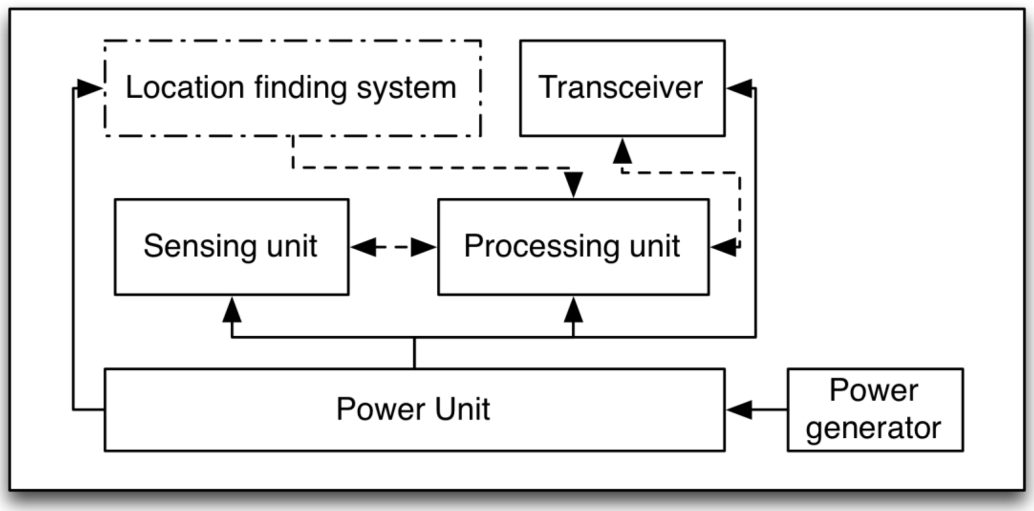


Figure 1.3: Components of a sensor node

Figure 1.3 represents the basic structure of a sensor node. Sensors generally consist of four basic components, viz. sensing unit, processing unit, transceiver unit and power unit. Some of the additional components which can be found in the sensor depending on the applications are mobilizer, location finding system and power generator. The sensing unit consists of two subunits, viz. sensors and ADC (Analog to Digital Convertors) as can be seen in the figure. Sensors produce the analog signals which are then sensed and converted in to digital signals by the ADC. The converted digital signals are then given as an input to the processing unit. The processing unit contains a small storage unit and handles the collaboration of the sensor nodes with each other. Transceiver is used to connect the node to the network. The power unit plays the most essential part as it drives all the other units to accomplish their respective tasks. Although in WSNs, nodes have limited power and are not rechargeable, there are some application dependent sources which can assist the power unit like solar cells. Location finding system is another optional but essential unit in sensor nodes as location of the respective node is generally required for various applications.

1.1.2 Sensor Network Protocol Stack

The sensor network protocol stack is different from the standard TCP/IP and along with the layers used in traditional protocols it has additional planes to handle the issues in the sensor nodes

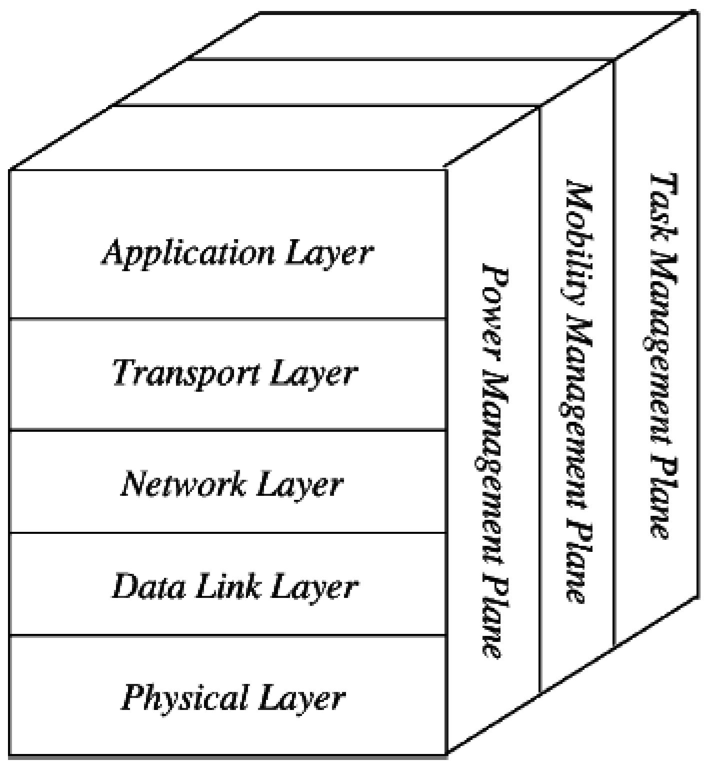


Figure 1.4: Sensor Network Protocol Stack

The front plane in the Figure 1.4 has five layers which work as traditional protocols. The application layer is involved in usage and development of application software(s) based on the specified sensing tasks. The transport layer provides the data to the network layer and it is the responsibility of the network layer to route the data as desired. Data link layer handles the noise and mobility of the nodes in the network and are power aware. Data link layer is also responsible for reducing the collision with the neighbors' broadcast. Transmission, modulation and receiving techniques are handled by physical layer of the protocol stack. The additional planes of power, mobility and task management are specially created to cater the needs of sensor node related to battery, movement and task distribution. Main objective of these planes is to carry out the collaborative sensing task as desired with least power consumption. The power management plane (PMP) has the responsibility to decide how a node will use its power like when it has to switch on their sensors and when they have to switch off them. Power management plane also takes care of when to transmit the information and when not to. Mobility management plane (MMP) is responsible to know the route back to the BS in case of movement of the sensor nodes from their initial positions and inform neighbors about the newly moved sensor node. With the updated knowledge of neighbors now they can contribute and use the power of the nodes efficiently. Task management plane (TMP) performs the balancing of sensing tasks amongst the sensor nodes. Thus TMP takes such decisions as per the application used in WSNs. So the additional planes are very essential for the objective of WSNs as without them a node will have no capability and will act as an individual transmitting the unwanted information and wasting the constrained power of the sensor node. These planes are essential for simple computations at the sensor nodes as per the changes in power, mobility and required tasks efficiently by the sensor nodes.

1.2 WSNs Application Areas

With the increase in growth and feasibility of WSNs, there are numbers of fields of applications which now use WSNs. Wide varieties of sensors are used for such applications. The sensors used can be thermal, visual, infrared, radar, seismic, low sampling rate magnetic and these sensors are capable of monitoring variety of diversified environmental conditions like temperature, pressure, soil makeup, vehicular movement, noise levels. Sensors are also used to detect the presence or absence of certain objects, to measure the mechanical stress levels on attached objects and to measure some of the dynamics characteristics like speed of the target object along with its direction and size.

As the numbers of applications of WSNs are enormous, one cannot list an exhaustive list of the application areas but can be briefly categorize in military, health, home, environment and commercial applications.

1.2.1 Military applications

Most extensive use of WSNs is done in military applications right from the inception of WSN. The category of military applications can be in computing, intelligence, surveillance, military command, control etc [3]. The characteristics of WSNs like rapid deployment, cheap sensors, self-configuration and fault tolerance makes WSNs the most suitable option in military operations as in the battlefield there is a need to quick deployment of the nodes so as to setup a network as fast as possible in high tension areas. The number of nodes may be subject to change frequently due to intrusions in the field and hence self-configuration is necessary to keep the network working. Cheap cost of sensors in WSN is another factor which is of importance to military applications as the sensors once deployed may be destroyed and can be subjected to redeployment quite frequently.

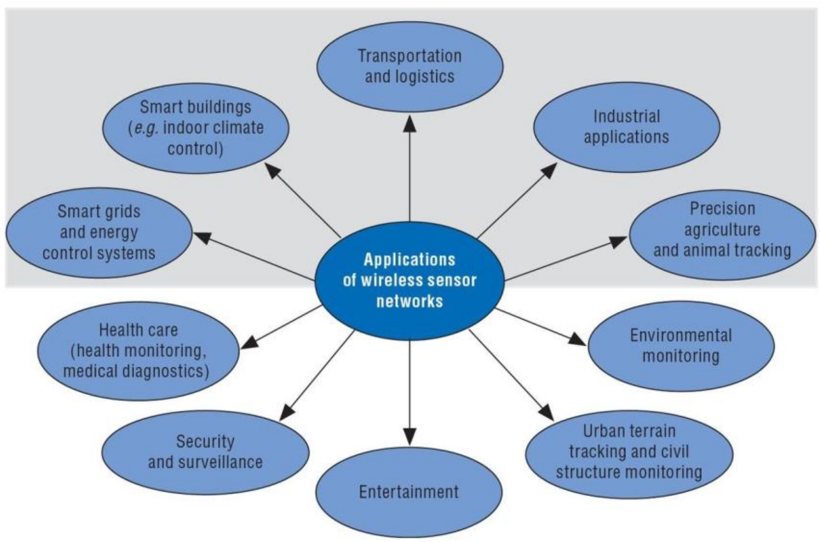


Figure 1.5: Applications of WSNs

Few military operations involving WSNs are targeting, battlefield surveillance, battle damage assessment, monitoring friendly forces, equipment and ammunition etc. Sensor networks are used in the guidance systems of intelligent ammunition. Battle damage assessment is done with the help of WSNs in the areas which are not in the home range. For the battlefield surveillance military install WSNs on the critical terrains and the areas far from the home range and can closely track the activities of the opposing forces. Other than these military can always use WSNs for monitoring of the troops of the friendly forces to keep a close watch on them.

WSNs are also used in managing the information related to the availability and conditions of the equipments used in the battlefield before and after the battles.

1.2.2 Health applications

Health industry has adopted the use of WSNs for variety of applications right from the simple diagnostics to providing interfaces for the disabled. Some other applications which are employing WSNs are patient monitoring, administration of drugs, monitoring the movements of internal processes for animals of interest, tele-monitoring of the physiological human data and also tracking of the doctors in the hospital [3].

In tele-monitoring of the physiological human data sensors give more freedom to the humans. It can retain the information from the sensed behavior of the individual which can be stored for long and can be used for exploration. Also, use of WSNs in tele-monitoring provides greater flexibility and comfort to the patient. WSNs used to track the doctors and patient is done by attaching sensor nodes to them. The sensor nodes attached has specific tasks which may be checking heart beat, blood pressure etc.

1.2.3 Environmental applications

WSNs play a major role in environmental applications specially related to natural calamities and disasters. Some of the other applications using WSNs are monitoring environmental conditions which can affect the growth of crops, tracking the movements of small animals, insects and birds, monitoring in soil, marine and other atmospheric contexts, bio-complexity and pollution studies.

Precision in agriculture can be maintained by WSNs as it can monitor the level of soil erosion, the level of pesticides in water and the amount of air pollution. Flood detection is another area where WSNs are used and one of the examples of such sensors is ALERT which is used by US. Several sensors like weather, water level and rainfall sensors are deployed and send's the data packets to central authority. The central authority then monitors the values and takes appropriate actions and keeps the data for later purposes. Another critical area of environmental applications where WSNs are used is forest fire detection. There are situations when forest fire can broke large and can cause a lot of disaster. Thus to avoid such incidents sensor nodes are deployed either statically or dynamically of the stretch of the forest with high density. In case of early stages of fire in forest the sensor node communicate the exact location of the origin of fire and thus necessary action can be taken to stop it from spreading. If the fire has expanded and is beyond controls the information from the sensors helps in evacuating the place and thus saves any human losses.

1.2.4 Commercial applications

Many commercial applications use WSNs for better flexibility and versatility. Some of the commercial applications which use WSNs are monitoring quality of the products, construction of smart office places, robot control, automation and control in factories, machine transportation. WSNs are deployed in security services also like monitoring and detecting thefts, tracking of vehicles. Various factory applications are modeled using WSNs like factory instrumentation, local control of actuators, instrumentation of semiconductor processing chambers, rotating machinery, wind tunnels, and anechoic chambers.

Environmental control in the office buildings is widely handled using WSNs. As the heat and air conditioning of most of the buildings is centrally controlled there are places with differences in temperature. One place in the room may be cooler than the other as the central control does not distribute the energy evenly. Thus distributed WSNs are deployed in such building to control the temperature and air flow in different parts of the building. Detection and monitoring of vehicle thefts is another commercial application which uses WSNs. Sensors are spread within a geographical region and with the help of internet these changes reach the users in the remote areas. Also the location in case of the theft can be directly sent to the BS and theft vehicle can be recovered. In inventory control each item is attached to a sensor node which can tell the location of the item. Thus in the inventory the user can locate each and every item and can keep a count of items. Remote user can also track the exact position of the items and can be ensured of the count of the items.

1.2.5 Characteristics

Most of the applications of WSNs share many common characteristics. Some of the similarities based on the interaction(s) between sources and BS are:

- Periodic monitoring: sensor can be made to periodically measure the values in the environment and send the measured values to the user.
- Event detection: Specific events can be detected by the sensor nodes. Such sensor are also applied to addresses the event of natural calamities. Some of the applications with event detection are grass fires, forest fires, volcanic eruptions etc. Classification of events is necessary in case of a sensor network used for more than one event detection.

- Location driven: Most of the applications work on the changes in the position of the sensors. Applications like theft detection, inventory management, detection of behavior of humans are monitored based upon the changes in the position of the sensors.
- Tracking: Some applications sense the data even after continuous change in position. In cases of surveillance if any threat is on the move the sensor nodes communicate the source to the BS along with the speed and direction. This helps to estimate the present position of the targets.

1.3 Research Issues

The research challenges posed by WSNs are quite different from general wireless networks. The reason is that there are various constraints and WSNs are highly application specific in nature. These constraints affect the communication properties of the system. Considering the size, power, cost constraints in WSN, the following issues affect the design and performance of a WSN:

- Deployment Strategy
- Localization
- Clustering
- Coverage Efficiency
- Efficient medium access control
- Efficient database centric design
- Quality of service implementation
- Acceptable security

We have restrained ourselves with the study of localization. This thesis concentrates mainly on localization.

1.4 Localization

The rapid growth of wireless and mobile devices has brought up the demand for applications in which location awareness is one of the most significant requirements. Nowadays, location based services like Bluetooth exists widely in wireless communication networks. Location information supports many fundamental network services like topology control, boundary detection, coverage, clustering, network routing. Here’s a brief overview as follows:

1. Topology Control

Topology control is used in wireless sensor networks for saving energy, reducing interference between nodes and extending the lifetime of the network. The reduction in interference and energy consumption can be achieved by adjusting network parameters (e.g., the transmitting range) while retaining some global network properties like connectivity. By using location information, we can apply geometry techniques (for e.g. Euclidean minimum spanning trees) to topology control.

2. Boundary Detection

Boundary detection is used to find out the boundary of the area being monitored by a WSN. The boundaries could be outer boundary or inner boundary. The outer boundary depicts the area which is under-sensed while the inner boundary depicts holes in the network deployment. The information about the boundary of the area being monitored facilitates the design of routing, network management and load balancing. The boundary detection also needs location information to identify the nodes at the boundary and hence, determine the network boundary

3. Coverage

Coverage means how well the sensor nodes observe or monitor the area. Coverage depends on the distance between the point of interest and the closest node. To find out this distance we require the location information of these nodes, therefore, location awareness is important to algorithms which examine the coverage of the network.

4. Clustering

Grouping sensor nodes into clusters to facilitate network management and organizing them hierarchically is known as clustering. There is a special node in each cluster known as cluster head which facilitates the inter-cluster communications, while the ordinary nodes communicates only with the nodes in the same cluster. The grouping of sensor nodes into clusters improves network scalability and generally achieves high energy efficiency and prolong network lifetime in large-scale WSN environments. The location-based approaches for clustering generate non-overlapping clusters and are more efficient. The location information can also be used to build clusters again locally when some new node joins the network or when a failure occurs in some node.

5. Routing

Routing is the process of finding the path or route along which the data could be sent. When a node receives a message, it forwards it according to the position of the destination and its neighboring nodes.

1.5 Research Objective

The objective of our research work can be identified as:

- To formulate localization in WSN as an optimization problem and to use Bat Algorithm to solve this problem
- To achieve higher accuracy and efficiency in localization using Bat Algorithm.
- To solve the flip ambiguity problem in localization.
- To improve the efficiency and precision of the localization algorithm.
- To localize nodes with only two references or three near-collinear references.

1.6 Thesis Organization

We start this dissertation with introduction in chapter 1. Chapter 2 presents the formulation of localization problem in wireless sensor networks and gives a literature review of previous research in WSN localization. In Chapter 3, we explain the Bat algorithm along with other bio-inspired algorithms and how they are used for optimization in localization. In Chapter 4, we explain our proposed approach to solve the localization problem using Bat algorithm with improvements to reduce flip ambiguity problem. Chapter 5 discusses the numerical simulation and results obtained. Finally Chapter 6 concludes the thesis with discussing future research possibilities.

CHAPTER 2

LITERATURE REVIEW

2.1 Localization

Localization means assigning geographical coordinates to each device with unknown position in the deployment region. Localization is required in WSN to report the origin of events, routing, assisting group querying of sensors and to answer questions on the network coverage. In WSN, there will be sensor nodes densely deployed at positions which may or may not be predetermined. In most cases, it's not feasible to deploy every node at predetermined locations as they may be randomly scattered in the region. In most applications of WSN, the information gathered by these sensor nodes will have no meaning if it is not accompanied by the position of the sensor nodes which gathered this information. A simple solution could be to have each node equipped with Global Positioning System (GPS). GPS is not practical because it imposes high cost and high power consumption. Also GPS fails in indoor locations, dense forests and underground applications. Therefore, we require a cost effective solution which can be deployed in diverse environments.

Another solution as an alternative to GPS is used known as Self-localization. In Self-localization, the sensor nodes estimate their positions by using some localization protocols. Most of these protocols use a few special nodes, called anchor nodes (also known as beacon nodes, seeds, references or landmarks), which are assumed to know their own locations (through manual configuration or GPS receivers). These beacon nodes provide position information, in the form of beacon messages, for the benefit of non-beacon nodes (also known as blind nodes or dumb nodes). These non-beacon nodes can use this position information to find their own coordinates.

2.2 Problem Formulation

A WSN consists of N sensor nodes deployed in a network area. A subset of nodes called anchors or beacons are aware of their locations (a_1, a_2, a_3, … a_M). The problem of sensor localization is to determine the locations of the remaining set of nodes (s_1, s_2, s_3, … s_{N-M}) given the pair wise distance measurements between the nodes.

2.3 Different Localization Approaches

There are different ways to characterize localization algorithms. Based on the nature of underlying algorithm, the scheme could be distributed or centralized. Based on the use of anchor nodes, the scheme could be anchor-based or anchor free. However, all existing localization approaches consist of two phases. These are:

1. Distance/Angle estimation (also known as Ranging phase)
2. Distance/Angle combining

1. Distance (or angle) estimation

In ranging phase, the geographical information is measured. Geographical information could be coarse grained (neighbour awareness) or fine grained (inter node ranging e.g. distance or angle). The nodes estimate their distances from anchor nodes using signal propagation time or strength of received signal. We determine distance or angle between nodes in first phase using following methods:

- **Received Signal Strength Indicator (RSSI):** This technique measures the power of the received signal. The power of radio signal diminishes during propagation. If the transmit power is known effective propagation loss can be calculated. This loss can then be translated into distance estimates using theoretical models. The radio signal strength decreases with distance according to the Power Law as shown in equation (2.1):

$$P(d) = P(do) - \eta 10 \log\left(\frac{d}{do}\right) + X_\sigma \tag{2.1}$$

where,

P(d) = received power at distance d,

P(do) = received power at some reference distance do,

η = path-loss exponent, and

X_σ = a log-normal random variable with variance σ^2 that accounts for fading effects.

- **Angle-of-Arrival (AoA):** In AoA, nodes estimate the angle at which the signals are received and use simple geometric relationships to calculate the positions of the nodes. Localization in AoA can be solved using triangulation. AoA localization is susceptible to noise in measurement and few additional problems. This technique requires nodes to be equipped with an array of antennas so that they can find out reference directions, these are expensive though.

- **Time Difference of Arrival (TDoA):** In TDoA, ultrasound/acoustic and radio signals are used to estimate distances by determining the Time Difference of Arrival (TDoA) of these signals. Figure 2.1 shows the idea behind TDoA. For example, RF and acoustic signal have different speeds and if transmitted will be received at different times T_1 and T_2 respectively. Thus time difference is

$$\Delta T = (T_1 - T_2) \tag{2.2}$$

Let the distance between transmitter and receiver be D and velocity of acoustic and RF be Vs and Vr. Then D is given by

$$D = \frac{\Delta T \times Vs}{(1-\frac{Vs}{Vr})} \tag{2.3}$$

As Vs<<Vr

$$D = \Delta T \times Vs \tag{2.4}$$

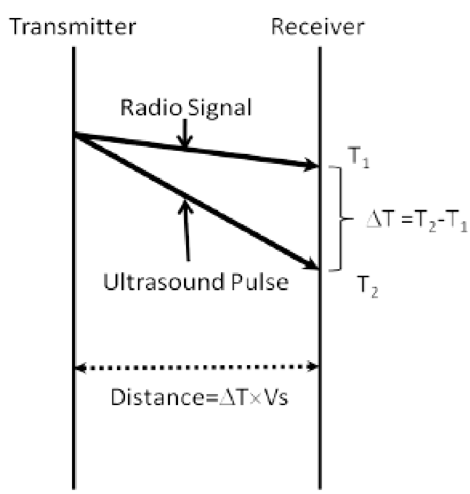
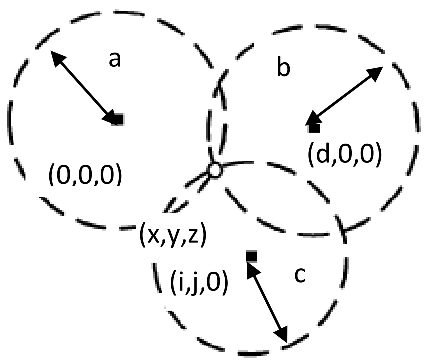


Figure 2.1: TDoA

2. Distance (or angle) combining

In phase 2, the data collected in phase 1 is used to determine (X, Y) coordinates of all the nodes. In second phase, position of the target nodes is estimated using ranging information from first phase. This is done by either solving a set of simultaneous equations or by using an optimization algorithm that minimizes the localization error. In iterative localization, the nodes localized serve as beacon nodes in the further rounds and this process is repeated until all nodes are localized or none can be localized. The techniques used are lateration, triangulation, bounding box, probabilistic approach and fingerprinting. The most common methods used in this phase are discussed below:

- **Hyperbolic Tri-lateration:** This process uses distances or absolute measurements of time-of-flight from three or more sites as shown in Figure 2.2. The equations of circles can be used to find the intersection of these circles i.e. (x, y, z).



$$a^2 = x^2 + y^2 + z^2,$$
$$b^2 = (x-d)^2 + y^2 + z^2, \ and$$
$$c^2 = (x-i)^2 + (y-j)^2 + z^2.$$

Figure 2.2: Hyperbolic Tri-lateration

- **Triangulation:** This process uses at least one distance measures and two measured angles. The angle measurement may be done by with receiver antenna diversity or phase comparison. As shown in Figure 2.3, Sine Theorem, i.e.

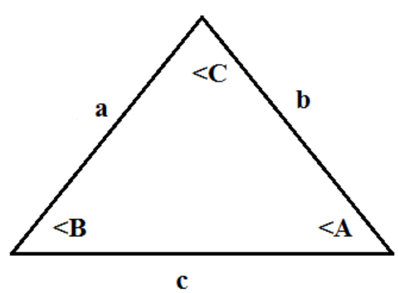


Figure 2.3: Triangulation

$$\frac{a}{sinA} = \frac{b}{sinB} = \frac{c}{sinC}$$

And Cosine theorem, i.e.

$$c^2 = a^2 + b^2 - 2ab.CosC$$
$$b^2 = a^2 + c^2 - 2ac.CosB$$
$$a^2 = b^2 + c^2 - 2bc.CosA$$

can be used. But these methods require at least two references for localization.

2.3.1 Centralized Localization

In centralized localization, the individual sensor collects inter-node ranging and connectivity data and then forwards their estimates to a central base station where all the processing takes place. The advantage of centralized algorithm is that it eliminates the need of computation in each node but it’s followed by certain drawbacks like traffic congestion due to central processing server and computational complexities.

2.3.2 Distributed Localization

In distributed localization, the sensors try to find their locations by using the sensing estimates from beacons. After the sensors have computed their locations, they can be used as anchors in further steps. This is known as Iterative Localization scheme. This scheme prevents the bottleneck problem due to a central processing server in large networks.

2.3.3 Anchor Based Localization

In this type of localization, there are some nodes which know their locations as either they are deployed at known locations or are equipped with GPS devices. They propagate their locations to target nodes. The target nodes then find their locations using any of the localization techniques. Some anchor based approaches are discussed below.

- **Centroid Method**

In this method [4], the most likely position of the node is at the centroid of neighbor anchors. The centroid method depends on the number of anchors available within the proximity of node.

- **APIT**

Approximate Point in Triangle test [5] finds the location of the target nodes from the region (triangle) it resides in. Three anchors are chosen from the neighboring anchors of the target node and PIT test is performed to find whether the node resides inside triangle formed by joining the three anchors. APIT repeats the test for all the possible combinations of the anchors till there are no combinations left or desired accuracy is achieved. The APIT then finds the center of gravity of the intersection of all the triangles in which the node resides as the estimated position of the node. The advantage of APIT is that it is simple and easy to implement but it requires high density of anchors.

- **Gradient Method**

The gradient method [6] uses the radio range along with proximity information. In this method, the target node calculates its distance from anchor node by multiplying the radio range with hop count. After finding the distances, it can use multi-lateration to estimate its location.

- **DV Hop Method**

In this method [7], first the average distance per hop is estimated. Then each target node determines its distance from the anchor node by multiplying the hop count by the estimated average distance. The target nodes find their location using triangulation approach.

2.3.4 Anchor-Free Approaches

In anchor-free approaches, the relative locations of the nodes are found using the proximity information, angle information or range information.

MDS-MAP [8] based on multidimensional scaling (MDS) is a a centralized range-free algorithm. Here the nodes are able to find their relative positions by using the information from neighboring nodes. This scheme works well in situations where exact locations are not required. The MDS based approach the MDS based approach includes three steps:

1. In the first step, the distance matrix is built using the distances between the anchors and the target nodes in the network. When anchors are not present, the distance is obtained by multiplying the hop count with the radio range.
2. In the second step, an initial relative map of the nodes is created using Singular Vector Decomposition (SVD).
3. The last step, the necessary flip, rotation and scaling is performed according to the distances between anchors.

When the positions of an adequate number of anchor nodes are known (3 anchor nodes for 2-D localization and 4 anchors for 3-D), the absolute coordinates of all nodes in the map is estimated. MDS-MAP generates the most accurate location information among range-free techniques. However, it suffers from high time complexity, requires large bandwidth and large number of computation to estimate the locations when there are a large number of sensors.

CHAPTER 3

BIO-INSPIRED SYSTEMS

Recent researches in the WSNs field are directed towards the biological and animal behaviors in nature. These are known as bio-inspired systems. Bio-inspired systems try to mimic the problem solving and information gathering and processing ways as observed in nature. Animals show many kinds of behaviors in nature. Their behavior can be used in solving many problems. Therefore, many algorithms are being developed to mimic these biological and animal behaviors [9].

In most cases of animal species, there's a leader who leads all other group members. For example lions, elephants, deer, monkeys, wolves, etc. Then there are also animals that don't have any leaders like birds, ants, fishes, etc. In these animals, they show organized behavior that allows their movements through the environment. They don't have any knowledge about the group and the environment in which they move. Their movements are guided by the information shared by the closest members. The bio-inspired systems also imitate the biological models like human immune system.

The growing interest in the areas of bio-inspired systems is due to the following reasons.

- Their capability to adapt to the changes in the environment.
- High resistance to failures.
- Their capability to address and implement complex situations and behaviors using basic set of rules.
- Management of limited resources efficiently.
- Self-organization of individuals in a fully distributed manner achieving efficient collaboration.

Because of the above mentioned reasons, bio-inspired algorithms have attracted a lot of attention of researchers. We will describe some popular bio-inspired algorithms in following section.

3.1 Ant colony optimization

The Ant Colony Optimization (ACO) [10] meta-heuristic is one of the best-known examples of swarm intelligence systems. It was proposed by Marco Dorigo in 1992.

The inspiration for using ant colony arose from observing ants (Figure 3.1) and their foraging. In ACO, the problem is represented as a search in a graph for a minimum cost path. The artificial ants walk through the graph with each path being a candidate for potential solution.

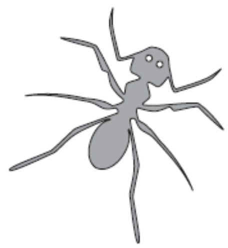


Figure 3.1: Ant

While walking the graph, the ants leave pheromone trails. The ants that come later sense the pheromones on path. They are attracted to the path with high levels of pheromone trails. This kind of indirect communication via the local environment is called stigmergy. It intensifies the search around the paths which appears to be more promising.

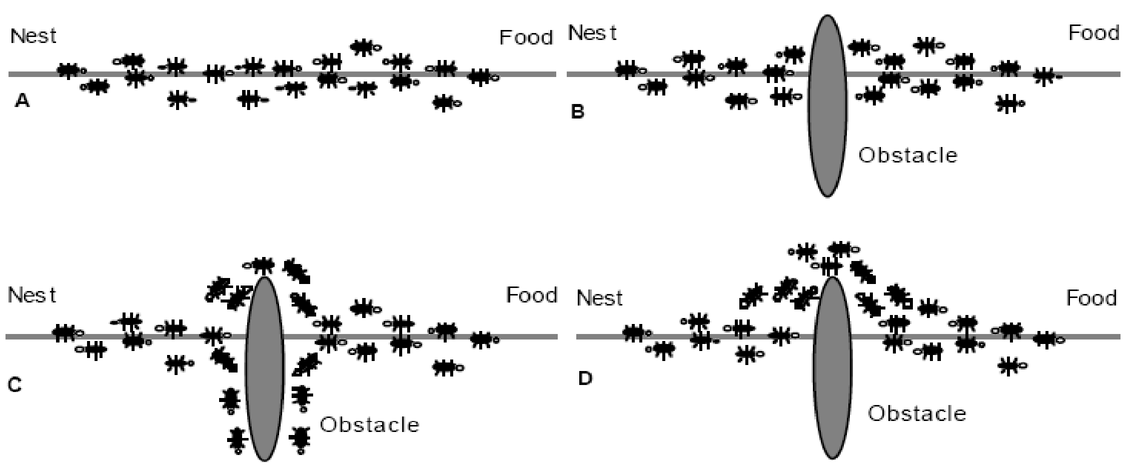


Figure 3.2: (A) Real ants follow a path between nest and food source.
(B) An obstacle appears on the path.
(C) Pheromone is deposited more quickly on the shorter path.
(D) All ants have chosen the shorter path.

As shown in Figure 3.2, initially ants choose any of the available paths with equal probability. The ants which choose the shorter path reaches food early and returns to their nest

ALGORITHM 1: Pseudo code of the ACO algorithm

Input: ACO population of particles$(x_{i1}, x_{i2} \dots x_{iD})^T$ for i = 1…N_p

Output: The best solution x_{best} and its corresponding value f_{min} = min(f(x)).

1. Set parameters
2. Initialize pheromone trails
3. Evaluate the new population.
4. fmin= find the best solution (x_{best})
5. **while** termination condition not met **do**
6. **for** each ant **do**
7. Generate new solution
8. Evaluate the new solution
9. Update pheromones
10. **end for**
11. Choose the ant with the best fitness value of all as f_{min}

earlier than the ones choosing longer path. Therefore, the pheromone trail concentration increases on the shorter path. And eventually more ants start taking the shorter path.

Goss et al. [11] developed a model of the observed behavior: assuming that at a given moment in time $m1$ ants have used the first path and $m2$ the second one, the probability $p1$ for an ant to choose the first path is:

$$p1 = \frac{(m1+k)^h}{(m1+k)^h+(m2+k)^h} \tag{3.1}$$

where parameters k and h are fitted according to the experimental data.

The basic steps of the ant colony algorithm are illustrated as a pseudo-code in Algorithm 1.The ant colony algorithm starts by generating the initial population of ants. This population is then evaluated using an evaluation function. The best solution is found with maximum value.

The new route is selected for each ant in colony and evaluated. If a better route is found, the old route is replaced by the new route. Finally, the pheromone level is updated according to the evaporation rate.

The best global solution is then searched among all the solutions. Two main issues need to be solved in order for this algorithm to be fully operable:

- probability of route selection
- evaporation rate of the pheromone.

However, both issues are influenced by the problem to be solved. The ways in which these issues have been tackled has led to the emergence of several variants of ant colony algorithms. The ant colony optimization algorithms are used for many real-world applications. Shelokar et al. [12] applied an ant colony for clustering. Dowsland and Thompson [13] colored graphs using ant colony optimization.

3.2 Particle swarm optimization

Particle swarm optimization (PSO) is a swarm intelligence algorithm presented by Russ Eberhart and James Kennedy in 1995 [14]. PSO is inspired by the social foraging behavior of some animals such as flocking behavior of birds (Figure 3.3) and schooling behavior of fish.

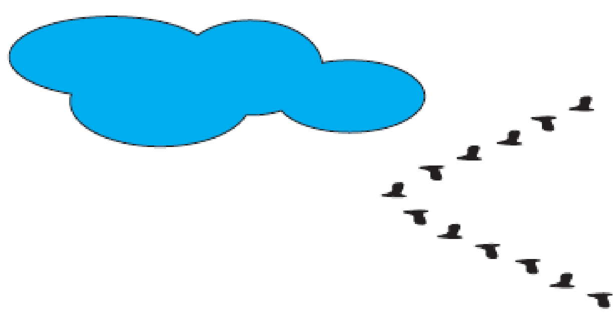


Figure 3.3: PSO

PSO consists of a population of candidate solutions (particles) that move around in the search space. Each particle in the algorithm has its local best known position and is guided towards the best known positions in the search space. The best positions in search space are updated by all the particles in the search space [15]. The pseudo code for PSO is given in Algorithm 2.

The particle i has a position x_{id} and moves with a velocity v_{id} in dimension d. The particle's fitness is calculated using a fitness function and it depends on the position of the particle in the search space. The closer the particle is to the optimum position the better the fitness is. The local best of a particle is given by pBest and gBest gives the best of all the pBests. The gBest represents the best solution found.

At iteration k, the velocity and position of particle i is updated as follows.

$$v_{id}(k+1) = c_0 . v_{id}(k) + c_1 . rand_1 . (pBest_{id} - x_{id}) + c_2 . rand_2 . (gBest_{id} - x_{id}) \quad (3.2)$$

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1) \quad (3.3)$$

In these equations, rand1 and rand2 are random numbers between 0 and 1.

ALGORITHM 2: Pseudo code of the PSO algorithm

Input: PSO population of particles $(x_{i1}, x_{i2} \dots x_{iD})^T$ for i=1…N_p

Output: The best solution x_{best} and its corresponding value f_{min} = min(f(x)).

Initialize particles velocity and positions.
while maximum iterations or minimum error criteria is not attained **do**
  for each particle **do**
    Calculate fitness value
    If the fitness value is better than its personal best
    Set current value as the new **pBest**
  end for
  Choose the particle with the best fitness value of all as **gBest**
  for each particle **do**
    Calculate particle velocity according equation (a)
    Update particle position according equation (b)
  end for
end while

3.3 Bat Algorithm

Bats represent a biological system that uses decentralized decision making as well as coordinated movement in order to survive. They usually live in colonies. Bats use echolocation as their primary mechanism for orientation towards a surface, although not all the species are blind. Additionally, this mechanism serves as a tool for finding their prey and discriminating between different types of insects. As matter of fact, bats are not the only creatures that use echolocation because there are also other remarkable animals that use it. For example dolphins, shrew, oil birds and toothed whale also use echolocation.

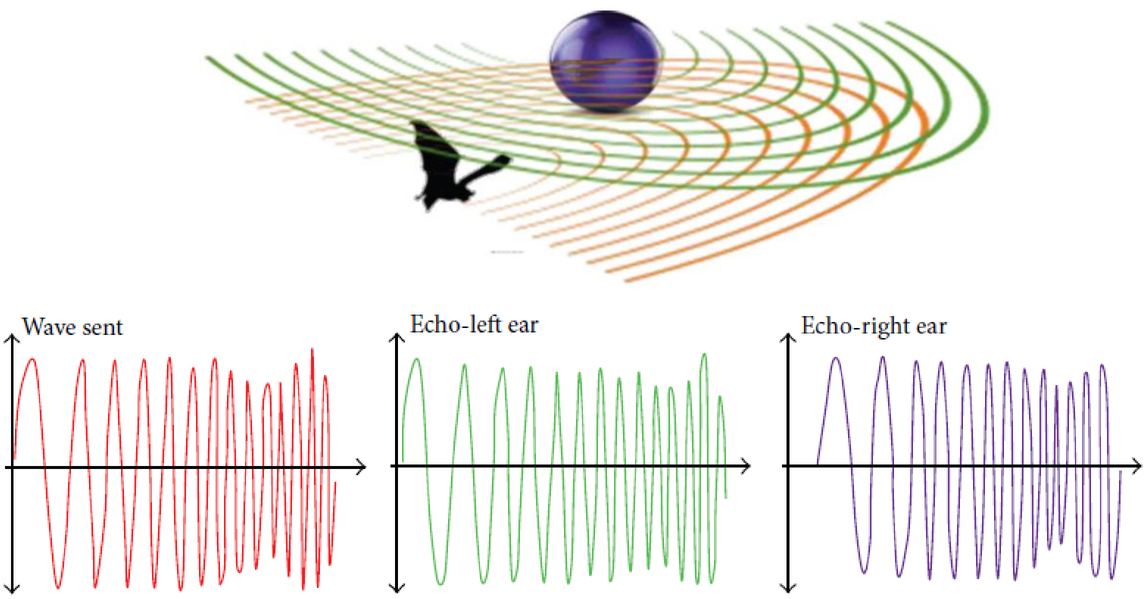


Figure 3.4: A bat emitting and receiving a wave to detect an object and the perceived wave that the animal has for each ear.

3.3.1 Biological foundations of bats

Bats are the only mammals that can fly. They hunt their prey in the dark and avoid obstacles using a special navigation system called Echolocation [16]. Echolocation is the phenomenon of bouncing back of sound waves after hitting some object or obstruction. Bats use sonar to find their prey and also to avoid obstacles. The bats can emit sound waves as high as 110 decibels which fall in the ultrasonic range.

The delay between the transmission and reflection, the time difference between the two ears and the variations in the echoes are used to find out how far the prey or obstacle is and also to create a three dimensional view of their surroundings. Figure shows a bat emitting sound waves and the echoes from the object and that how the bat perceives those waves. The bats are able to differentiate between their preys by using Doppler Effect which is induced by the differences in the wing flutter rates of the insects.

3.3.2 Bat Algorithm

The bat algorithm (BA) was developed by Xin-She Yang in 2010 [17, 18, 19]. The inspirations for these works were micro-bats and their echolocations. The author wanted to mimic their natural behavior and succeed in creating a powerful algorithm which could be applied to almost all areas of optimization. The following simple rules idealize the characteristics of micro-bats:

1. All bats use echolocation to sense distance and they also know the difference between their prey and their surroundings.
2. They fly randomly with velocity v_i at position x_i with varying wavelength λ , frequency f_{min} and loudness A_0 while searching for prey.
3. The loudness is assumed to vary from a maximum value A_0 to minimum value A_{min}.

The pseudo-code of BA is illustrated in Algorithm 3. Bats' behavior is captured within the fitness function of the problem to be solved.

3.3.3 Bat Motion

The virtual bats move in the d-dimensional search space by updating positions x_i and velocities v_i according to following rules:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{3.4}$$

$$v_i^{t+1} = v_i^t + (x_i^t - x*)f_i \tag{3.5}$$

$$x_i^{t+1} = x_i^t + v_i^t \tag{3.6}$$

where β Є [0,1] is a random vector drawn from a uniform distribution. Where x* is a current global best location which is located after comparing all the solutions among all the n bats at each iterations t.

3.3.4 Loudness and Pulse Emission

The loudness A_i and the rate r_i of pulse emission are updated as the iterations proceed. As the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience [18].

3.3.5 Applications of the Bat Algorithm to Localization

Goyal and Patterh [20] presented an algorithm for localization in WSN using Bat algorithm. The purpose of WSN node localization is to estimate the positions of N target nodes as much as possible by using M beacons with known locations in a distributed range-based way. The steps in

ALGORITHM 3: Pseudo code of the Bat Algorithm

Input: Bat population $(x_{i1}, x_{i2} \dots x_{iD})^T$ for i=1…N_p

Output: The best solution x_{best} and its corresponding value f_{min} = min(f(x)).

1. Initialize bat population x_i (i=1,2,…n) and v_i
2. Define pulse frequency f_i at x_i
3. Initialize pulse rates r_i and loudness A_i
4. **while** number_of_iterations<Max number of iterations **do**
5. Generate new solutions by adjusting frequency and updating velocities and locations according to equations 2 to 4
6. **if** rand>r_i
7. Select a solution among the best solutions
8. Generate a local solution around the selected best solution
9. **end if**
10. Generate a new solution by flying randomly
11. If rand<A_i andf(x_i)
12. Accept the new solutions

the algorithm proposed by them are as follows:

1. There are N target nodes and M anchor nodes. These are randomly deployed in a two-dimensional sensor field having transmission radius, R. Anchors possess location awareness and transmit their coordinates frequently. The target nodes that get settled at the end of iteration serve as beacons during the next iteration.
2. The target node that falls within the transmission range of three or more non-near-collinear references is considered as a localizable node.

3. The measured distance from a localizable node to each of its neighboring references is simulated as the actual distance plus a Gaussian additive white noise which can be expressed as $\widehat{d_i} = d_i + n_i$ Here, $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the actual distance and (x,y) and (xi,yi) are the locations of the target node and the i^{th} references, respectively. The measurement noise n_i has a random value uniformly distributed in the range $d(1 \pm P_n/100)$. P_n is the percentage noise that affects the result of localization.
4. Each localizable node runs the improved Bat algorithm to localize itself by finding the coordinates (x, y) that minimize the value of the objective function given as

$$f(x,y) = \frac{1}{M}\sum_{i=1}^{M}(\sqrt{(x - x_i)^2 - (y - y_i)^2} - \widehat{d_i})^2$$

5. Above steps are repeated until either all target nodes are settled or no more nodes can be localized.
6. The total localization error El is computed as the mean of squares of distances between the actual node locations (xi, yi) and the computed locations $(\hat{x}_i, \hat{y}_i)$ $(i = 1, 2, \ldots, N_L)$ determined by the improved Bat algorithm, as in (4.4). N_L is the number of localized nodes.

$$E_l = \frac{\sum_{i=1}^{N_L}((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)}{N_L}$$

CHAPTER 4

PROPOSED WORK

This chapter gives the understanding of the proposed work along with the details of the new features introduced. Here we will describe a distributed localization algorithm for wireless sensor networks. This localization algorithm is range-based. The prerequisite for this algorithm is the availability of ranging information using any of the techniques like RSSI, TOA or TDOA as described in chapter 2.

4.1. Network model

For the proposed algorithm, the network consists of N unknown nodes which need to be localized and M anchor nodes. Deployment of the nodes is done randomly in D X D area with the sensor nodes deployed randomly. After the random deployment of the nodes, they are immobile. Each node has a transmission range of R units. The anchor nodes send beacon messages to transmit their location coordinates to the unknown nodes. Therefore, all nodes are available with the pair-wise distances from the anchor nodes.

This is the network model which is taken into consideration for the development of the proposed protocol.

4.2 Application of Bat Algorithm in WSN localization

Bat Algorithm is a kind of optimization algorithm which is applied to range-based methods. The algorithm needs to find the distance between the target nodes and the anchor nodes. The network can be considered as a two-dimensional coordinate system. The nodes are considered as two-dimensional coordinate points. Let T be a target node with coordinates (x,y) and (x_i,y_i) be the neighboring anchor node A_i (i=1,2,….M). Then the problem of localization can be considered as an optimization problem whose purpose is to find the solution (x,y) which minimizes the following objective function

$$f(x,y) = \frac{1}{M}\sum_{i=1}^{M}\left(\sqrt{(x-x_i)^2-(y-y_i)^2}-\widehat{d_i}\right)^2 \tag{4.1}$$

where, $\widehat{d_i}$ is the noisy distance between the target node and i^{th} anchor node.

The target nodes that get settled are considered as anchor nodes in the next iteration. The localization process is repeated until all the nodes get localized or there are no more localizable nodes available.

4.2.1. Flip Ambiguity

The main problem in the localization is the problem of flip ambiguity. The flip ambiguity occurs due to the reference nodes being collinear or near-collinear. The flip ambiguity phenomenon is shown in Figure 4.1.

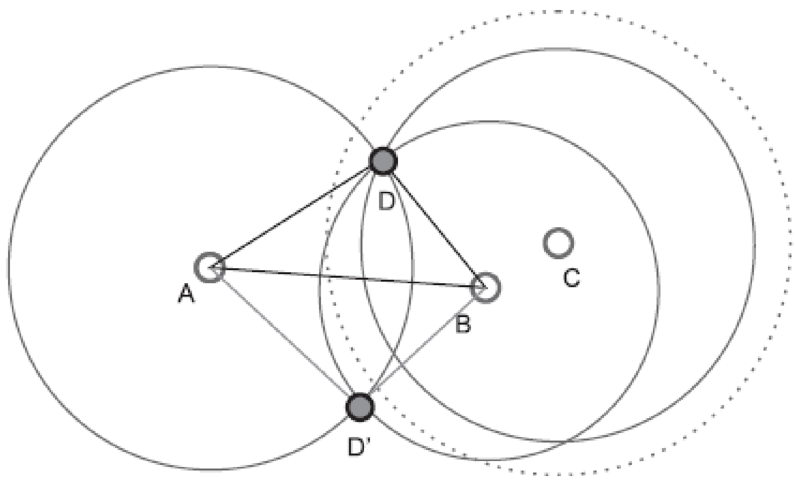


Figure 4.1: Flip ambiguity phenomenon.

As can be seen in the figure, the target node D has three neighbor reference nodes viz. A, B, C. These reference nodes are near-collinear. Due to this, the node D can be localized either at position D or D′. This problem is known as flip ambiguity. In our approach, we will provide a solution to alleviate this problem.

4.3 Proposed Approach

In this section, a distributed localization algorithm for wireless sensor networks is described.

4.3.1. Initial Search Space

The Bat algorithm as proposed in [17] employs a set of feasible solutions within the search space, called the bat population with random initial locations. In our proposed approach, we limit the initial search space by using a bounding box method. An example of bounding box method is described in Figure 4.2. Node n is an unknown node with three reference nodes n_1, n_2, and n_3 in its transmission range. The range of node is represented by a circle with radius r. But here we used squares with sides of length $2r$ each to limit the initial search space of node n. This helps in avoiding any floating point operations. The centers of these squares are the nodes n_1, n_2, and n_3, respectively. The initial search space of node n for the bat algorithm is represented by the overlapping portion of these squares as shown in figure 4.2 (the shaded portion). Suppose the coordinate of reference node n_i is (x_i, y_i); then the shaded portion can be expressed as:

$$([\max(x_i - r), \min(x_i + r)], [\max(y_i - r), \min(y_i + r)]) \quad (4.2)$$

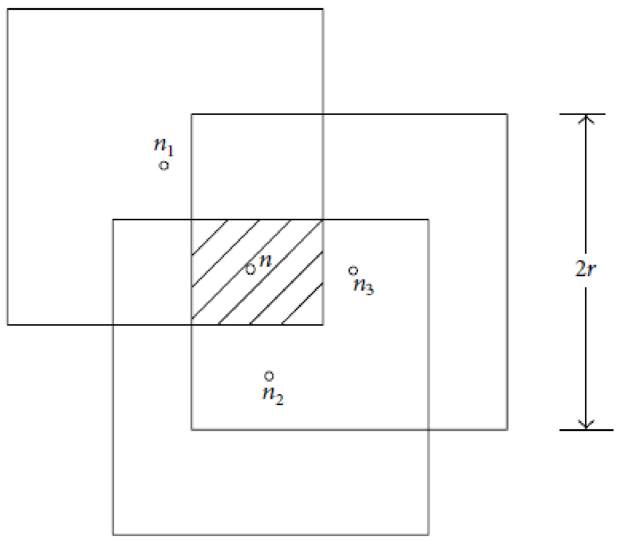


Figure 4.2: Bounding Box Method

Every target node has its unique search space determined by the reference nodes in its transmission range. Thus, instead of using the whole solution space for the search, this reduced search area is used. The proposed approach, therefore, reduces the computational time and the possibility of flip ambiguity.

4.3.2. Anticipation of Nearly Collinear References

To prevent occurrence of flip ambiguity phenomenon, there's need to find out whether the references are near-collinear or not in case there are a small number of references available. This is because the probability of occurrence of flip ambiguity increases when less number of references is available. As shown in Figure 4.3 node A is located by using reference nodes B, C, D, and E which are near-collinear and therefore, there are chances that A is estimated as its flipped location A'. We use the following rules for finding whether the references are near-collinear.

1. If there are only three reference nodes available, any two of them should not be too close to each other. When the distance between any two of them is less than a threshold value, they are considered to be near-collinear.
2. Calculate the distance of any one of the references from the line joining the other two references. If the distance is less than a given threshold value then the references are said to be near-collinear.

The above rules are used in combination to prevent the occurrence of flip ambiguity. If the reference nodes of a target node are found to be near-collinear, the target node is not localized in the iteration. Its localization is delayed for further iterations when it may have more references, thus making its localization possible without flip ambiguity.

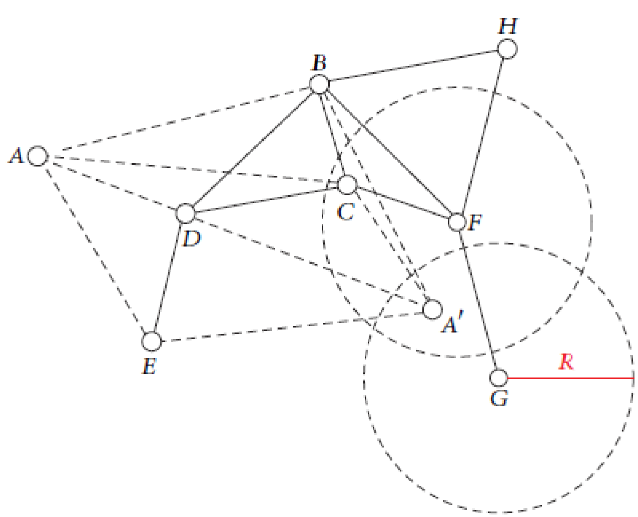


Figure 4.3: Identification of Flip Ambiguity

4.3.3. A Refinement Phase to Correct the Error due to Flip Ambiguity.

In the refinement phase, the nodes which have flip ambiguity problem are identified and are considered for re-localization. To find out which nodes have flip ambiguity problem, we analyze the relationship between the target node and its neighbor reference nodes. As can be seen in Figure 4.3, if node A is localized in wrong location A', then it can be seen that the neighborhood of A' is different as compared to A. This property can be used to identify if flip ambiguity problem occurs. As can be seen in Figure 4.3, node A' falls in the wrong neighborhood of anchor nodes F and G. In the refinement phase, such nodes are found and they are re-localized by adding an extra term (4.3) to the objective function (4.1).

$$\sum_{j=1}^{Q} \delta\big(r - \hat{d}_j\big)(r - \hat{d}_j)^2 \tag{4.3}$$

Here, (.) is unit step function; Q is the number of non-neighbor reference nodes.

4.4 Algorithm.

The main purpose of WSN node localization is to estimate the positions of N target nodes as much as possible by using M beacons with known locations in a distributed range-based way. The process of our proposed approach is as follows.

1. There are N target nodes and M anchor nodes. These are randomly deployed in a two-dimensional sensor field having transmission radius, R. Anchors possess location awareness and transmit their coordinates frequently. The target nodes that get settled at the end of iteration serve as beacons during the next iteration.
2. The target node that falls within the transmission range of three or more non-near-collinear references is considered as a localizable node.
3. The measured distance from a localizable node to each of its neighboring references is simulated as the actual distance plus a Gaussian additive white noise which can be expressed as $\hat{d_i} = d_i + n_i$ Here, $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ is the actual distance and (x,y) and (xi,yi) are the locations of the target node and the i^{th} references, respectively. The measurement noise n_i has a random value uniformly distributed in the range $d(1 \pm P_n/100)$. P_n is the percentage noise that affects the result of localization.
4. Each localizable node runs the improved Bat algorithm to localize itself by finding the coordinates (x, y) that minimize the value of the objective function (4.1).
5. In the refinement phase, some of the nodes localized in the previous step should be re-localized by adding an extra term (4.2) to the objective function (4.1) owing to their estimated position falls into the wrong neighborhood.
6. Above steps are repeated until either all target nodes are settled or no more nodes can be localized.
7. After those target nodes which have at least three non-near-collinear references are all localized, the remaining ones that only have two references or three near-collinear references can be localized by using the refinement phase at last.

8. The total localization error El is computed as the mean of squares of distances between the actual node locations (xi, yi) and the computed locations ($\hat{x}_i$, $\hat{y}_i$) (i = 1, 2, . . . , N_L) determined by the improved Bat algorithm, as in (4.4). N_L is the number of localized nodes.

$$E_l = \frac{\sum_{i=1}^{N_L}((x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2)}{N_L} \tag{4.4}$$

CHAPTER 5

SIMULATION RESULTS AND ANALYSIS

Simulation is considered as efficient and flexible tool to evaluate the performance of the algorithm working under vivid environmental conditions. In this chapter, the improved Bat algorithm for localization proposed in chapter 4 is evaluated on a simulation platform. The performance of the approach is compared with other conventional protocols in terms of localization error, number of nodes localized.

5.1 Simulation Setup

We used MATLAB as the tool for simulation and performance evaluation of the proposed localization using Bat algorithm. N=50 target nodes and M=10 anchor nodes are randomly deployed in a sensor field of 100*100 square units. All the nodes are deployed randomly in the 100*100 field dimension network. Each node has a transmission radius of R=25 units. The initial deployment of the nodes is same for all the algorithms. The parameters of the proposed node localization algorithm are set as follows:

Parameters	Parameters value
Population size	20
Number of generations	100
Loudness	0.95
Pulse rate	0.6
Minimum Frequency	0
Maximum Frequency	0.1

Table 5.1: The parameters value of experiment

5.2. Performance Evaluation

In order to show the advantage of our algorithm more intuitively, we made some contrast figures to analyze each step independently as follows.

Firstly, we use methods described in section 4.3 to avoid flip ambiguity phenomenon. An example in a trial shows the different localization results between the experiments with and without method 4.3.

As shown in Figure 5.1 (the blue circle represents the location anchor node, the green circle represent the actual location and red cross represent the estimated location of the localized node using Bat algorithm, the black circle is for non localized target node and the red straight line represents the localization error that is the distance between the actual node location and the estimated location), flip ambiguity phenomenon leads to large localization errors.

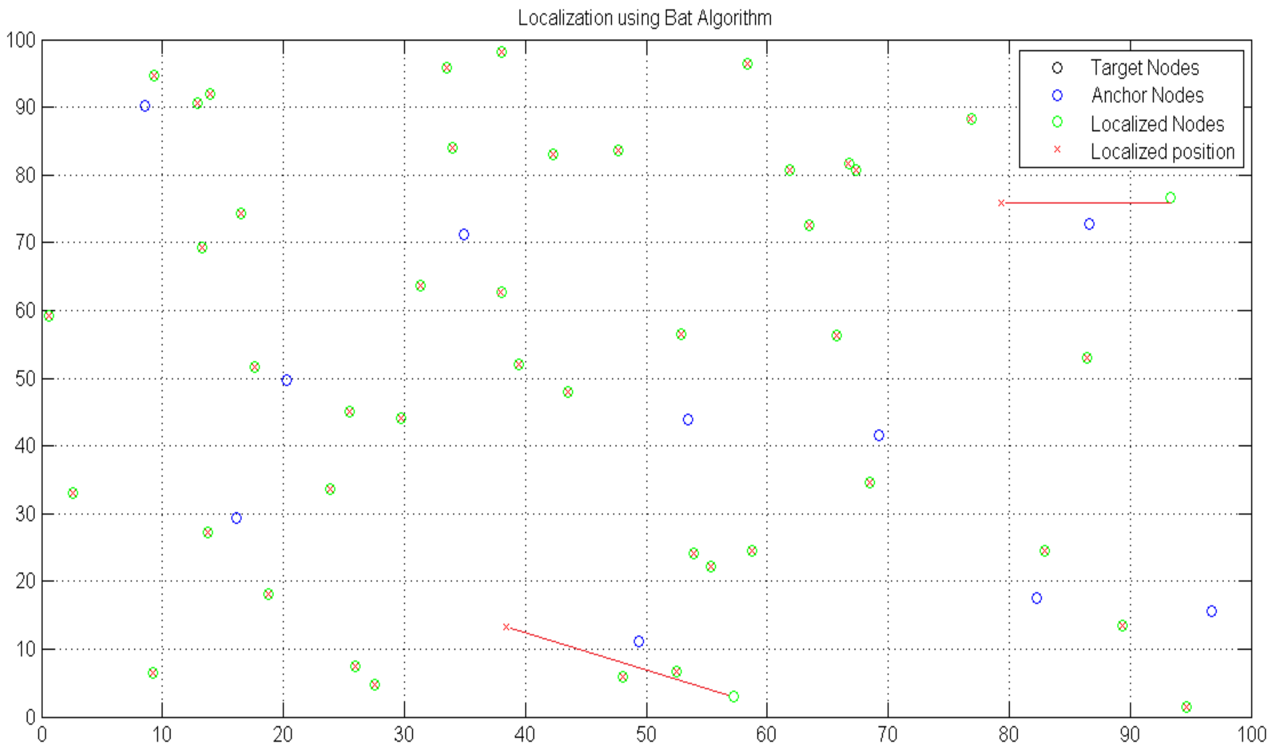


Figure 5.1: Result with flip ambiguity phenomenon

To avoid such problems, we use methods described in Section 4.3. As a result large localization errors have been corrected as shown in Figure 5.2.

Figure 5.3 shows the distance between the actual location of the target node and the one estimated by the Bat algorithm. The red curve represents the distance between the estimated position and the actual position of target node when method mentioned in Section 4.3 was not used and the blue curve represents when method mentioned in Section 4.3 was used.

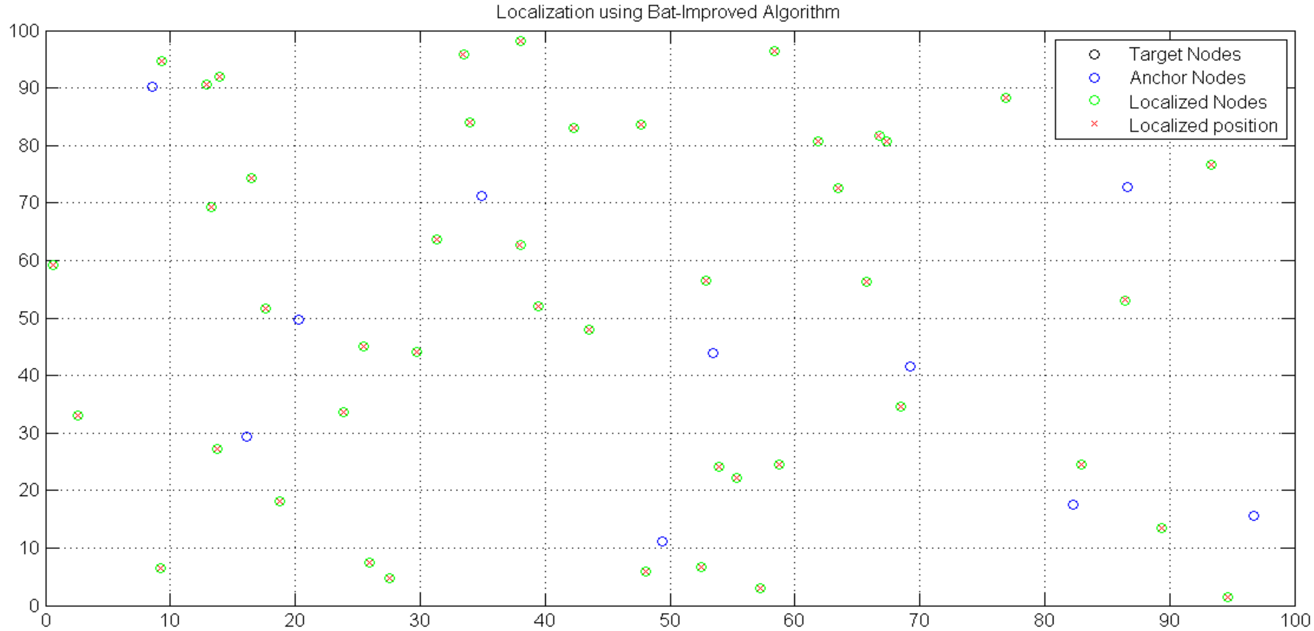


Figure 5.2: Result without flip ambiguity phenomenon

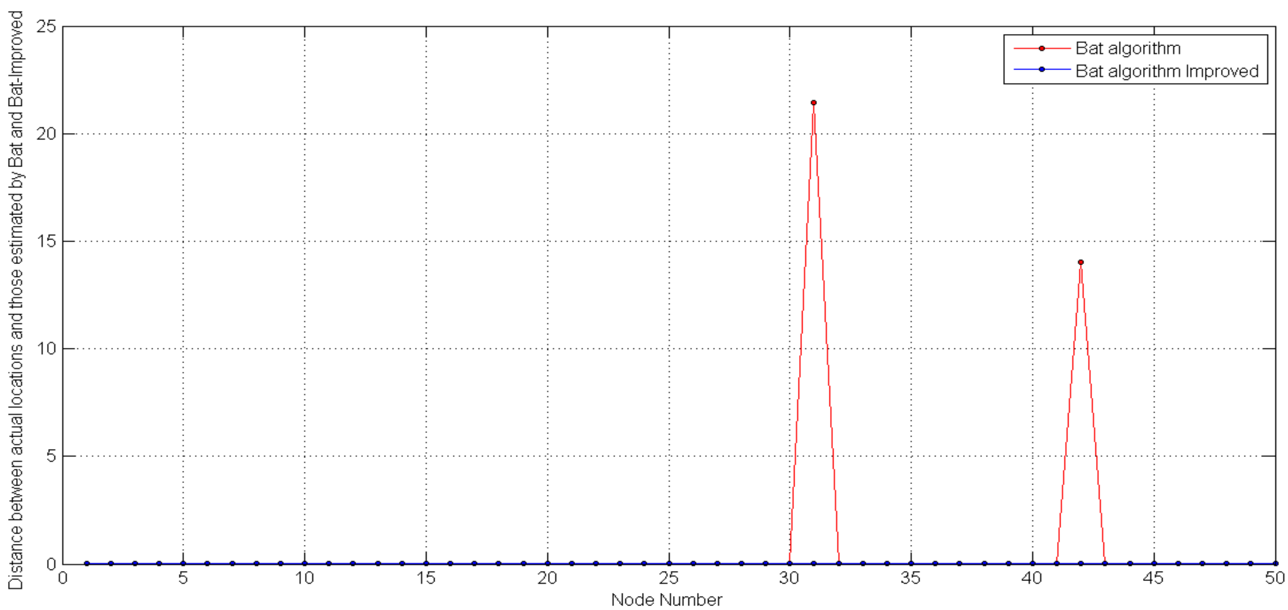


Figure 5.3: Distance between actual locations and those estimated by Bat and Bat-Improved algorithm

Obviously, the variation between the actual location and the estimated one for the first 30 nodes is reasonable. However, the localization of 31^{st} node causes flip ambiguity which leads to the huge difference between the estimated location and the actual location of the node. Similarly, in case of 43^{rd} node, flip ambiguity results huge difference between the estimated location and the actual location of the node. But on the blue curve we avoid huge and reasonable error with methods mentioned in Section 4.3 and receive an acceptable location estimate.

5.3 Comparison with PSO algorithm

The parameters used for the PSO algorithm are set as follows:

Parameters	Parameters value
Population size	30
Number of iterations	150
Acceleration constants, $c_1 = c_2$	2.0
Inertial weight, ω	Varied linearly from 0.9 to 0.4
Limits on particle position, X_{min}	0
Limits on particle position, X_{max}	100

Table 5.2: The parameters value for PSO

The inertia weight, ω is decreased linearly from $\omega_{max} = 0.9$ in the first iteration to $\omega_{min} = 0.4$ in the last iteration as described in (5.1). Here, k is the number of iterations and k_{max} is the maximum number of iterations.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{k_{max}} \times k \qquad (5.1)$$

30 trial experiments of PSO-based localization and our proposed algorithm are conducted with the same initial deployment for the purpose of comparison. The noise ratios used are $P_n = 2$ and $P_n = 5$. Average of total localization error, E_l defined in (4.4) in each iteration in 30 runs is computed. Similarly, 30 trial experiments of our proposed algorithm are run with the same initial deployments. The results are summarized in Figure 5.4.

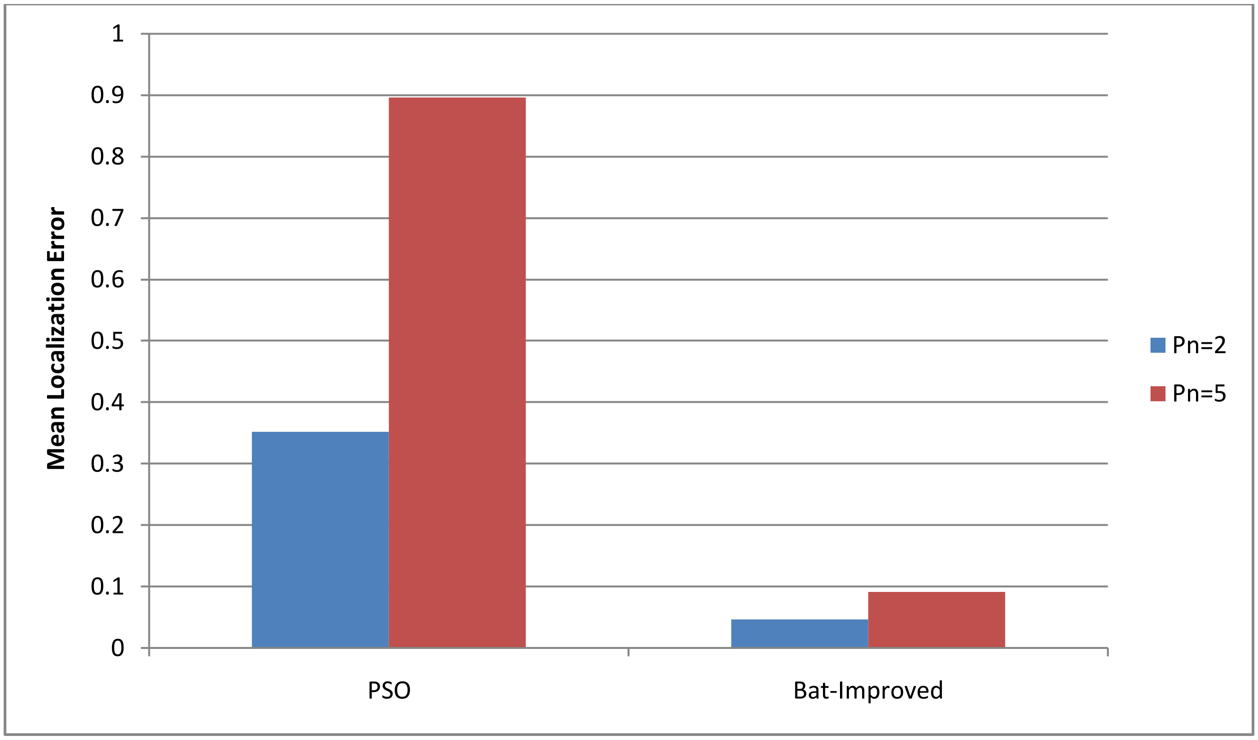


Figure 5.4: Mean Localization Error for P_n=2 and P_n=5

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this dissertation, we worked on finding solutions to perform distributed localization in WSNs with better accuracy .We presented an improved Bat algorithm to solve the distributed localization problem in WSNs, which can be regarded as a multidimensional optimization problem. To improve the localization precision, we presented some methods in Section 4.3. Their effectiveness has been proved by the simulation results. In order to show the advantage of our algorithm more intuitively, we made some contrast figures to analyze each method independently. The experimental results show that our method which can localize unknown (or target) nodes with higher precision is superior to the other two methods.

In this dissertation, we considered the nodes to be localized in two-dimensional search space. However, in real world, sensor nodes are distributed randomly in three-dimensional search space. Therefore, some research will be taken on our proposed work so as to consider the localization problem in three-dimensional search space. Further directions for research are the energy consumption as the sensor nodes have limited power supply and controlling the error from propagating iteratively. Moreover, we have considered nodes to be static. So, the proposed algorithm may be extended for mobile sensor nodes. Further, the proposed algorithm may be implemented for centralized localization and a comparison may be made for energy awareness. A hybrid stochastic algorithm may be proposed to achieve more accuracy.

References

1. Milenkovi´c, C. Otto, and E. Jovanov, “Wireless sensor networks for personal health monitoring: Issues and an implementation,” *Computer Communications* (Special issue: Wireless Sensor Networks: Performance, Reliability, Security, and Beyond), vol. 29, pp. 2521–2533, 2006.

2. T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, and L. Gu, “Energy-Efficient Surveillance System Using Wireless sensor networks” In *MobiSys’04,* Boston, June 6–9, 2004.

3. Th. Arampatzis, J. Lygeros, Senior Member, and S. Manesis, “A survey of applications of wireless sensors and wireless sensor networks,” In *Proc. 13 th Mediterranean Conference on Control and Automation*, Limassol, pages 719–724, 2005.

4. N. Bulusu, J. Heidemann, D. Estrin. GPS-less low-cost outdoor localization for very small devices[J]. *IEEE Personal Communications*, 2000, 7(5): 28 – 34.

5. T. He, C. Huang, B. M. Blum, et al. Range-free localization and its impact on large scale sensor networks[J]. *ACM Transactions on Embedded Computing System*. 2005, 4(4): 877 – 906.

6. R. Nagpal, H. Shrobe, J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network[C]// *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*. New York: ACM, 2003: 21 – 32.

7. D. Niculescu, B. Nath. Ad hoc positioning system (APS)[C]// *Proceedings of the IEEE Global Telecommunications Conference*. New York: IEEE, 2001: 2926 – 2931.

8. Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, “Localization from mere connectivity”, In *Proceedings of ACM Symposium on Mobile Ad Hoc Networking and Computing*

(MobiHoc'03), June 2003, Annapolis, Maryland, USA, pp. http://dx.doi.org/201-212. 10.1145/778415.778439.

9. Sendra, S., Parra, L., Lloret, J., & Khan, S. (2014). Systems and Algorithms for Wireless Sensor Networks Based on Animal and Natural Behavior Keywords :,(iv). http://doi.org/10.1155/2015/625972.

10. Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26, 29–41.

11. Goss, S., Aron, S., Deneubourg, J. L., & Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, *76*(12), 579–581. http://doi.org/10.1007/BF00462870

12. P. Shelokar, V. K. Jayaraman, and B. D. Kulkarni. An ant colony approach for clustering. Analytica Chimica Acta, 509(2):187-195, 2004.

13. K. A. Dowsland and J. M. Thompson. An improved ant colony optimization heuristic for graph colouring. Discrete Applied Mathematics, 156(3):313-324, 2008.

14. J. Kennedy and R. Eberhart. Particle swarm optimization. In Neural Networks, 1995. Proceedings., IEEE International Conference on, volume 4, pages 1942-1948. IEEE, 1995.

15. R. Poli, J. Kennedy, and T. Blackwell. Particle swarm optimization. Swarm intelligence, 1(1):33-57, 2007.

16. E. Hagen. Ask a biologist: Echolocation. http://askabiologist.asu.edu/echolocation. Accessed 15 February 2015.

17. X.-S. Yang. A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), pages 65-74, 2010.

18. X.-S. Yang. Bat algorithm for multiobjective optimization. International Journal of Bio-Inspired Computation, 3(5):267-274, 2011.

19. X.-S. Yang and X. He. Bat algorithm: literature review and applications. International Journal of Bio-Inspired Computation, 5(3):141-149, 2013.

20. Savithri, G., Sujathamma, P., Padmavati, S., & Visvavidyalayam, M. (2013). Wireless sensor network localization based on BAT algorithm. *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, (1961), 329–333.

21. Wang, J., Ghosh, R. K., & Das, S. K. (2010). A survey on sensor localization. *Journal of Control Theory and Applications*, *8*(1), 2–11. http://doi.org/10.1007/s11768-010-9187-7