A

Dissertation

on

Node Deployment for Maximization of Coverage Area in Wireless Sensor Network

Submitted in Partial Fulfillment of the Requirement

For the Award of the Degree of

Master of Technology

in

Computer Science and Engineering

by

Dipayan Chatterjee

University Roll No. 2K13/CSE/04

Under the Esteemed Guidance of

Mr. Rajesh Kumar Yadav

Assistant Professor, Computer Science & Engineering Department, DTU

2013-2015

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

DELHI – 110042, INDIA

Acknowledgement

I express my heartfelt gratitude to my project supervisor, Mr. Rajesh Kumar Yadav, for allowing me to pursue a field of research of my choice. I am in debt to him for his constant support, guidance and mentorship over the course of this project, devoid of which the completion of this project would not have been possible.

I would also like to thank Prof. O.P. Verma, HoD, Department of Computer Science & Engineering, Delhi Technological University (formerly Delhi College of Engineering), New Delhi, for having provided me with an opportunity to work on this project. I take this opportunity to thank the Delhi Technological University library and staff members for providing the academic resources and environment that were essential for this work to be carried out.

Last but not the least a heartfelt thanks to all my friends for their huge support and encouragement throughout the course of this project.

Dipayan Chatterjee
Roll no: 2K13/CSE/04
M.Tech (Computer Science & Engineering)
Department of Computer Science & Engineering
Delhi Technological University
Delhi – 110042

CERTIFICATE

This is to certify that the dissertation titled *"Node Deployment For Maximization of Coverage Area in Wireless Sensor Network"* is submitted by **Mr. Dipayan Chatterjee,** 2K13/CSE/04 in partial fulfillment of the requirements for the award of degree of Masters in Technology in Computer Science & Engineering at Delhi Technological University. It embodies the work carried out by him under my supervision.

Mr. Rajesh Kumar Yadav, Date:

Assistant Professor, Place:

Dept. of Computer Science & Engineering,

Delhi Technological University.

Abstract

A Wireless Sensor Network is a collection of autonomous sensors that work in coordination with each other to achieve a predetermined objective. Today wireless sensor networks are being used in a gamut of application belonging to both civilian and military domains. One of the open problems with respect to wireless sensor networks is the problem of finding an optimal deployment sequence that covers the entire mission area so that the best quality of service can be provided with low cost.

In this dissertation, two techniques for dynamic node deployment based on the Artificial Bee Colony (ABC) algorithm in wireless sensor networks are proposed.

The first proposed algorithm incorporates two additional phases namely the crossover phase and the opposition based learning phase apart from those of the ABC algorithm. The crossover phase is used for replacing a certain amount of worse solutions with new solutions that were generated using multi-point crossover operation. The opposition based learning phase generates an opposite solution for each of the F food sources, hence doubling the number of available food sources to 2F after which we selected the best F food sources out of them. The opposition based learning phase was carried out only 30% of the time.

The second proposed algorithm uses a hybrid local search for striking a balance between the exploration and exploitation behavior of the algorithm. The hybrid local search constituted of three search strategies namely the perturbation based local search strategy of ABC, the global best-directed search and a variable step size local search. The algorithm uses all the three strategies for finding a new solution around an old solution and makes a greedy selection amongst the three. This algorithm too like the previous one uses a crossover phase to increase the probing performance of the algorithm.

Both of the proposed algorithms were simulated, and results were compared with the ABC algorithm, it was observed that both the algorithms outperformed the ABC algorithm in finding a deployment sequence for maximizing area coverage when executed under similar conditions.

Keywords: Wireless Sensor Networks, Node Deployment, Area Coverage, Artificial Bee Colony.

Table of Contents

List of Figures

List of Abbreviations

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
BBO	Biogeography Based Optimization
BS	Base Station
DE	Differential Evolution
GA	Genetic Algorithm
GPRS	Global Packet Radio Switching
HIS	Habitat Suitability Index
LTE	Long – Term Evolution
LB	Lower Bound
NP	Non Polynomial
NSGA	Nondominated Sorting Genetic Algorithm
PSO	Particle Swarm Optimization
QoS	Quality of Service
SA	Simulated Annealing
SGA	Stud Genetic Algorithm
UB	Upper Bound
VFA	Virtual Force Algorithm
WSNs	Wireless Sensor Networks

1.1 Wireless Sensor Networks

A sensor network is a deployment of several devices equipped with sensors for performing a collaborative measurement task. We can choose any type of sensor for a specific purpose. The fields of application of sensor networks include logistics, smart agriculture, smart environment industrial process control etc. The working process of a sensor network consists of three parts, first is the sending of data measured by sensors, second part is the use of any protocol to communicate the information and the third part constitutes the external systems that process the data.

The example of smart cities demonstrates the utility of sensor networks in modern lifestyle scenario. In these cities, sensor networks are used to monitor phenomenon like city pollution, decibel levels of noisy part of the city,

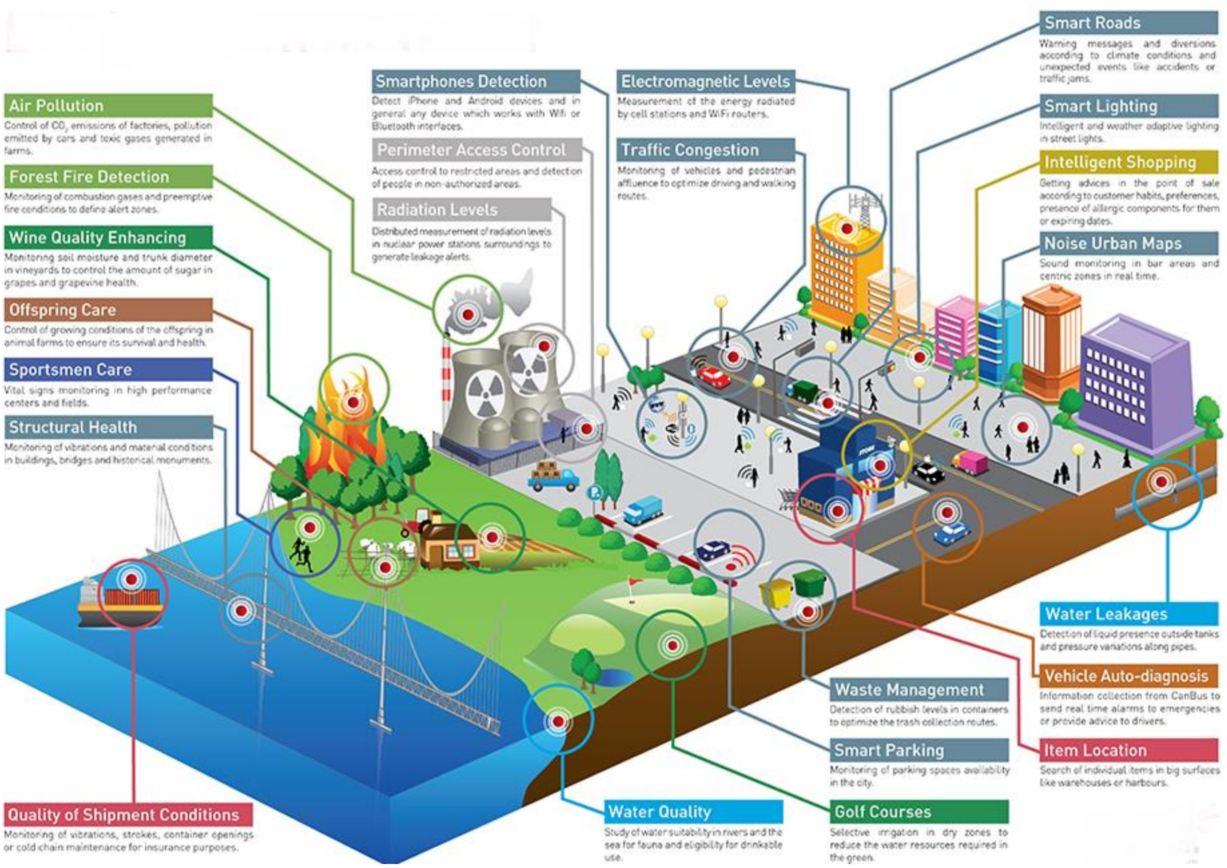


Figure 1.1

Figure 1.1 shows some of the applications of sensor networks, for example, sensor nodes can be used for monitoring the quality of shipment conditions, structural monitoring, health monitoring, forest fire detection, air pollution and radiation level detection and monitoring, traffic congestion detection, intelligent and adaptive street light management, electromagnetic level monitoring and many other endless possibilities.

1.1.1 Wireless Sensor Network elements

The different elements that comprise a WSN are:

- Node: It is an autonomous device equipped with a sensor.
- Data gatherer: It is a device that captures the data and relays it to the external systems using gateways.
- The external system: It is the data storage and management center.

1.1.2 Working Basis of Sensor Networks

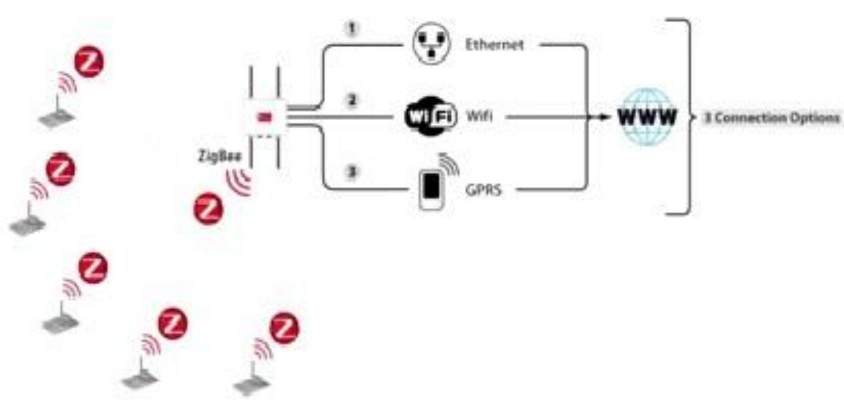


Figure 1.2

Figure 1.2 shows the skeletal structure of the working of a sensor network. On the left-hand side of the figure, we can see the sensor-equipped devices measuring the monitoring the sensing area. These devices transmit the data to the data gatherer using standard radio channel, the data gatherer transmits the data to the external systems using any wireless of wired communication protocol with Ethernet, Wi-Fi and GPRS being the most commonly used protocols.

1.1.3 Node Structure of a Sensing Device

A sensing device is typically made up of the following components

- Microcontroller
- Sensors
- Battery
- Communication modules
- Storage

Figure 1.3

Communication Modules: Sensors offer the flexibility of using a single communication module or multiple modules with each module supporting a standard communication protocol. Some of the modules offered by the modern sensors are:

- 802.15.4 / ZigBee / Digi Mesh
- Bluetooth
- GPRS/3G/LTE
- Wi-Fi

Storage: The standard storage options offered by sensing devices are, the first being a non-volatile EEPROM (Electrically Erasable Programmable Read Only Memory) of 3-4KB memory capacity and the second is an external storage device like SD cards whose storage capacity are on the upwards of 2GB.

1.1.4 Data Gatherer

The data gatherer is a multiprotocol router with several interfaces on it. These interfaces are used to transmit the data gathered from the sensor nodes to the external systems. A data gatherer generally supports most if not all of the communication modules mentioned above. Most modern day data gather come with a software application that helps the end user to configure the data gatherer to their required specification.

Figure 1.4

1.1.5 Data Communication and Monitoring

There are two approaches of transmission of data from the sensing device to the external system, the first approach is the direct communication between the sensing device and the external system using any standard wireless communication protocol, Figure 1.4 shows one such scenario where the sensor nodes directly communicate with the base station using GPRS or the 3G communication module, the second approach of data communication is to send the sensed data to a data gatherer

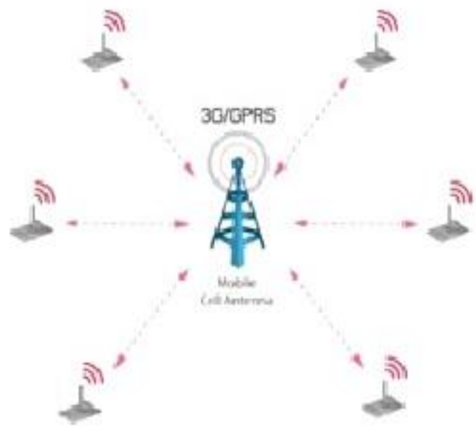


Figure 1.5

which stores the data, synchronizes processes and carries out some optimizations before transmitting the data to the external system, Figure 1.5 depicts one such scheme where Wi-Fi modules are used for communicating information to an access point (router). Every access point has a lightweight database like Tiny DB, MySQL etc. associated with it. The database is usually configured with the help of the software application that comes installed with the data gatherer.

Figure 1.6

A typical record associated with a stored sensor frame in a data gatherer's database comprises of the following attributes namely:

- ID: This attribute serves as the primary key of the record
- Date: A timestamp value denoting the date of insertion of the record

- Sensor ID: Contains the logical ID of the sensor that transmitted the data
- Serial ID: It is a ID that is hardwired with the sensor and cannot be reconfigured
- Frame Type: It indicates the type of packet sent by the sensor device
- Frame Number: Acts as the sequence number of the received packets
- Value: The data measured, recorded by the sensor

1.2 Problem Formulation

The problem of dynamic node deployment in wireless sensor network can be formulated as follows:

"*For a given network with certain number of sensor nodes how to deploy these nodes such that maximum area is covered* ".

Our goal was to find a dynamic deployment of sensors using swarm-based meta-heuristic such that the maximum portion of the service area is covered. The meta-heuristic was used to guide the movement of sensor nodes inside a 2-dimensional bounded area of interest, $A \subset A^2$. The sensor network consists of M number of dynamic nodes designated as $S_1 \ldots S_M$ .Every sensor S_i has a position given by P (X_i, Y_i) and a constant sensing radius R and can sense any point K if d(S_i, K) < = R, where d(S_i, K) denotes the Euclidian distance between sensor S_i and point K.

1.3 Motivation

WSNs today are being deployed for supporting a wide range of applications. An important measure for being able to deploy a WSN is to find precise position of sensor nodes. For most applications a random deployment of sensor nodes fail to achieve the predefined objectives and consequently such a deployment strategy becomes intractable. Hence coverage of a network is one of the key parameters for determining the QoS of a network.

The area coverage problem is one of the fundamental problems concerning with wireless sensor network, till date this problem remains an open problem with a great amount of work already done on it in the past as well as in the present day by researchers. Area coverage is not monolithic in nature but the problem is classified into sub categories, more on this in chapter 2 but the bottom line is, if a network can't monitor the entire mission space, it cannot provide good quality of service. A good deployment strategy not only helps in covering the entire mission space but it also increases the network lifetime as well as facilitates easy communication between the sensors thereby reducing the cost. Though coverage is studied as a separate topic, in reality coverage is implicitly related to another parameter of network quality, connectivity.

1.4 Organization of Thesis

The dissertation begins with a bird's eye view of wireless sensor networks, their utility, the various sub components that constitute a sensor network and other intrinsic details etc. all covered in chapter 1. Chapter 2 provides a detailed view of the problem laying the foundation for the upcoming chapters, also covered in this chapter are some of the approaches studied and suggested by researchers pertinent to the problem under study. Chapter 3 starts with a brief introduction to the ABC algorithm and its application for solving the node deployment problem, followed by our proposed approach. In chapter 4 the performance of our proposed approach is compared with that of the standard ABC algorithm when applied to the node deployment problem. We conclude the text with a brief summary of our findings and a few other approaches for extending our work all of this is covered in chapter 5.

A WSN typically consists of numerous sensing devices, positioned either arbitrarily or by using some formulation like complex geometric approaches, statistical distribution, and swarm intelligence meta- heuristic etc. over the area to be monitored. A sensor is a small device that operates on a battery and hence its work efficiency is constrained. The limited power source hinders the signal processing capabilities, transmission capabilities as well as data processing capabilities of a sensing device, as a result of which a sensor can sense only a small portion of the entire monitoring area. The resource constrained nature of a sensing device directs us to the idea of using multiple sensor nodes for achieving the objectives efficiently.

The efficiency of a deployed WSN in accomplishing its objectives is governed by how efficiently we are able to position the sensors of the network. The most intuitive approach for node deployment is to randomly deploy the sensors either manually or drop the sensors over the desired area from an airborne vehicle like aircraft etc. The bright side of such random deployment techniques is that they are easy to achieve, but on the other side, such deployments fail to satisfactorily achieve the objectives like maximizing area coverage, minimization of communication cost, fault tolerance etc. It is for this reason that the problem of node deployment has piqued the interest of researchers.

Today there exist a myriad of approaches that researchers had come up with for tackling the problem. All these approaches can be broadly classified into two categories namely static and dynamic deployment. In static deployment once the sensor is positioned in the area to be monitored its position remains the same throughout the lifetime of the network. The position of the sensors can be determined either randomly or in a deterministic way. The dynamic deployment approaches are more robust than their static counterpart and hence are generally the preferred one. In dynamic deployment the sensor nodes are mobile thereby enabling them to change their positions if required in cases where the topology of the network changes (due to the failure of a node).

There are multiple goals of the node deployment problem which are as follows:

- **Maximizing coverage area**: The coverage problem is WSN is related to issues such as energy consumption, communication cost etc. The goal of this problem is to position the sensor nodes in the area such that every point in the area is covered by at least one sensor node.

- **Network Connectivity**: The objective of maximizing coverage and network connectivity are complementary to each other and not two different concepts altogether, if we simply have coverage but no connectivity then the information associated with the sensed phenomenon can't be relayed by the sensing node to the data center (sink node) as result of which the information is as good as lost
- **Network Lifetime**: This objective deals with the problem of reducing the energy consumption of the nodes in operations related to data transmission so that the service ceiling of the network in increased. This objective concerns with the need of finding better ways of routing and aggregating data etc. Collision, overhearing and control overhead are the chief reason of energy wastage.

Till date the problem of node deployment is an unsolved or open problem. This problem has been shown to be NP-Hard in nature. Today a myriad of approaches do exist related to finding optimal positions for sensor nodes , the approaches range from geometrical, positional solutions to meta-heuristics. Swarm based meta–heuristics have gained immense popularity and has become an important area of study, these meta-heuristics are generally inspired by the natural phenomenon, or they mimic the complex social behavior of species of the animal kingdom. These meta-heuristics are quite potent in solving or optimizing complex mathematical functions. Researchers through their work have demonstrated how these meta-heuristics outperform previous known approaches which lead to the proliferation of the application of meta-heuristics on well-known NP-Hard problems like the travelling salesman problem, finding shortest path in a graph, process scheduling problems, job shop problems etc.

Coverage is vague term and, thus, has multiple interpretations, coverage is classified into the following categories:

- Blanket Coverage: In this type of coverage every point inside the monitoring area is a point of interest and it should be covered by at least one sensor, ideally it is desirable to cover the entire area with the minimum number of sensors.

- Barrier Coverage: In this type of coverage a chain of sensors is deployed across the sensing area with the aim of detecting breaches in the barrier of sensors, this type of coverage aims at detecting intrusions with the chain of sensors acting as fence or trip wire.
- Target Coverage: This type of coverage is generally associated with surveillance based applications both civilian and military. In this case specific pockets in the sensing area is of greater importance than the others.

In our work, we have used a variant of target coverage, where in the sensors instead of covering the entire monitoring area, the sensors try to cover a maximum number of predefined preferential target points. The fitness of a deployment is measured in terms of the number of such target points covered.

The remainder of the chapter contains the essence of some of the work carried out by researchers till date which are pertinent to the problem under study. The objective of this chapter is not to be a comprehensive guidebook for the research done in the past, but to give the reader a bird's eye view of the various different approaches that have been suggested and can be used for tackling the problem of node deployment.

I. Maleki et al [3] formulated a new approach for optimizing the area coverage problem using hybrid PSO and Differential evolution (DE) algorithm. The hybrid algorithm used the PSO algorithm for positioning of the sensor nodes so as to cover a maximum of the preferential target points in the sensing area. The DE is used to carry out the crossover and mutation operations on the solutions generated by the PSO algorithm. The mutation and the crossover operations further try to find better positions of some of the sensors in the deployment sequence that was generated by PSO. The objective of the algorithm was to cover maximum target points and increasing the lifetime of the sensor network at the same time.

W. Huang et al [4] proposed a new approach for solving node deployment problem using Van Der Waals Force. In contrast to Virtual Force Algorithm (VFA) where every node is perceived as points that experience repulsive and attractive forces exerted by other points in the vicinity and the movement of the particles is calculated using the force equation, the Van Der Waals Force based approach introduces the force exerted by friction as a parameter into the force equation, the forces exerted by different points is calculated which in turn is used to calculate the acceleration with which the points move the adjacency of nodes in determined using Delaunay Triangulation. The fitness of a solution is gauged according to a new function called the correlation function. The results showed that the proposed algorithm exhibits higher convergence rate as well as a better result in comparison to VF Algorithms.

X Wang et al [13] put forward a new approach for energy efficient coverage in WSN using distributed PSO and Simulated Annealing (SA).The authors used a network model comprising of both static and mobile sensor nodes, the fitness of a solution was gauged on the parameters of coverage and energy consumption. For determining coverage, the authors used a grid exclusion algorithm and Dijkstra’s shortest path algorithm for calculating the lowest cost path for data transmission and communication. For reducing the energy intake, the authors proposed the use of a hybrid algorithm comprising of PSO and SA. The local best and the global best solutions of the PSO are calibrated and corrected using SA which is performed on the nodes of the sensor network. The published results showed that the approach put forward by the authors was able to find deployments that had good coverage as well as low communication costs.

G. Wang et al [14] suggested a dynamic deployment strategy for coverage control in sensor networks using Biogeography Based Optimization (BBO) meta-heuristic. The authors used the binary sensing model and a network model consisting of both static and mobile sensor nodes. The BBO algorithm mimics the immigration and emigration of species from islands in search of better-matched islands. In BBO, the initial solutions are called ‘habitats ‘and the fitness function is called the Habitat Suitability Index (HIS). New solutions are generated by using two operations called the migration operation and the mutation operation. The published results demonstrate that he proposed algorithm outperforms other contemporary algorithms like the VFCPSOA, Homo -H-VFCPSO and SGA.

Ozturk et al [15] demonstrated the use of ABC for probabilistic deployment of sensors in an area under observation. The authors used a network model that consisted of both static and mobile sensors. The static sensors were first randomly deployed after which the mobile nodes were used to cover the areas that remained exposed. The authors used the ABC algorithm as the underlying meta-heuristic for guiding the movements of the mobile nodes. The authors discouraged excess overlapping of the sensing area of adjacent nodes using the probabilistic sensing model and a controlling parameter called the coverage threshold. The results showed that the ABC algorithm outperformed the PSO algorithm by generating a deployment sequence that covered more area.

The results did show that the PSO converged faster than the ABC algorithm but was not able to find better solutions than those found by the ABC algorithm.

J. Wang and S. Medidi [17] in their work proposed an energy efficient coverage strategy using sensors with variable sensing radius. The radius is optimized using Delaunay Triangulation for minimizing the energy expedited thereby increasing the lifetime of the network. The coverage was enhanced with the help of triangulation of sensors into planar graphs and each triangle was required to be covered by each of the three adjacent sensors. The radius of the sensors was first optimized locally each sensor chose a radius that could enable it to cover all the adjacent triangles in order to attain global optimality with respect to coverage area the authors used Delaunay triangulation.

Banimelhem et al [18] proposed a GA based algorithm for reducing the number of holes left after the random deployment of static sensor nodes. They suggested an algorithm that not only guides the mobile nodes to cover the holes by calculating their best position, but also determines the minimum number of mobile sensors required to achieve the objective. The fitness function used by them inhibits overlapping as the function counts the total number of targets covered for a mobile node only if the targets covered by this sensor are not covered by any other sensor both mobile and static. The results published by them showed that the proposed algorithm was able to cover more area by finding the minimum number of mobile nodes required to fill the holes.

Babaie et al [19] proposed a “hole healing algorithm” whose aim was to fill up the holes left after the initial random deployment of WSN, the key challenge in any hole healing algorithm is the detection of the presence of a hole in the coverage followed by determining the size of the hole. In their work, the authors used the triangular oriented diagram for detecting a hole. The authors connected the centers of three adjacent sensors thus forming a triangle. Once the presence of a hole and its size is ascertained, the stationary sensors should decide the number of mobile sensors that are required to fill the hole. If the size of the hole is less than the sensing radius of a sensor then a circumcircle center is chosen for target location on the contrary if the size of the hole is larger than the sensing radius then incircle center is chosen for target location. The authors claimed that their method was simpler than approaches that were based on Voronoi diagram based strategy also the exact size of the hole could be determined and not just an approximation of it.

Z Chen et al [23] proposed a memetic based multi-objective optimization of the coverage problem in sensor networks. In their work, the authors propose the use of multiple local searches to find better deployment sequences that had high area coverage, efficient node utilization, and increase network lifetime. The authors included a hybrid local search in the standard ABC algorithm along with a multipoint crossover operation akin to GA. The algorithm uses multiple local searches for finding new solutions and then making a greedy selection amongst them as a replacement for the old solution. The different local search strategy used were namely tabu search, hill climbing search, and an adaptive directional search. The result showed that the algorithm was successfully able to solve the coverage problem under the given constraints.

Jameii et al [24] proposed an energy efficient multi-objective coverage control scheme. The objective of the proposed algorithm was firstly to maximize the network coverage area, secondly reduce the number of nodes active. Their result showed that their proposed algorithm were able to find solutions that were better than other algorithms.

J. Jia et al [25] in their work showed how sensors with variable sensing radius employed to tackle the problem of optimizing coverage control in WSN using an improvised version of NSGA-II algorithm. The network model used by the authors for simulation was heterogeneous in nature. The authors introduced a cluster based scheme for controlling the coverage in wireless sensor networks.

Jameii et al [26] proposed a non-dominated sorting genetic algorithm (NSGA II) for the multi-objective optimization of coverage. In their work the authors used a network model with static nodes having variable transmission radius but same sensing radius also each sensor is aware of its position as well as of those sensors that are within its transmitting radius. The objective of the proposed algorithm was to maximize the area coverage at the same time minimizing the number of active sensor nodes at the same time minimize the disturbed energy ingestion. The results showed that with the same number of active sensors the suggested algorithm could cover more area as compared to other algorithms.

J Chen and H Qian [27] proposed a node deployment strategy in wireless sensor networks based on the viscous fluid model which draws its inspiration from the field of fluid dynamics in classical physics. The authors in their work assumed the sensors as fluid particles and the movements of these sensors were guide by the rules of fluid motion. The proposed deployment strategy is adaptive because the algorithm doesn’t need to be provided with complete information about the environment all that the algorithm needs is the size of the deployment area and the algorithm calculates the minimum number of nodes required to get the desired coverage. The authors used the concept of viscidity that exists between the motions of micelles of the fluid to regulate the motion of the sensor nodes.

Dhillon and Chakroborty [28] proposed two algorithms for supporting distributed wireless sensor network. Their algorithm tackles the problem of coverage and optimizes subjected to constraints such as uncertainty in detection and irregularity of terrain. They proposed an approach of the placement of sensors such that each sensor transmits only a minimal amount of data but the placement of the sensors ensures that the collective data contains adequate information for the base station so that it can later query a limited number of sensors asking them to relay the detailed information. The authors divided the mission area into grids. The case of coverage of preferential grid points in a search space was also covered by their work.

Nematy et al [29] proposed a computational geometry-based method for solving the node deployment problem. They proposed the use of Voronoi diagram. The authors used stationary nodes with the same sensing radius which they deployed randomly in the monitoring space. Once the stationary nodes were placed the monitoring space was divided into cells using Voronoi diagram after which the authors used GA for placing additional mobile sensor nodes inside the cells, the fitness function used penalized the deployments with overlaps in order to increase the area coverage.

Chapter 3 Proposed Work

3.1 Introduction

As you might have gathered from the previous chapters that node deployment in sensor networks still remains one of those open problems that have intrigued researchers. The previous chapter covers some of the approaches suggested by researchers for tackling the problem. One of the recent swarm based approaches that has piqued the interest of many scientists for solving complex combinatorial problems is the Artificial Bee Colony algorithm.

Artificial Bee Colony algorithm is fairly young meta-heuristic, conceived by Karaboga and Akay [1] in the year 2005. This algorithm draws its inspiration from the foraging behavior of honey bees. A single honey bee may be quite primitive or incapable to carry out complex tasks, but the intelligence of the swarm as a whole emancipates it to achieve complex goals. The complex social interaction among the members of the animal kingdom has been studied extensively, and a result of which is that today we have a host of swarm based meta-heuristic algorithms.

In a bee hive there are three types of bees, the first type is the explorer. The job of the explorer bee is to wander around the hive in search of food. Once a food source is found, the bee memorizes its path from the hive, it also factors in the distance to be travelled to reach this source, the amount of nectar available, the effort required to reach the food source etc.

Once a food source is found the bee then returns to its hive, where the second type of bee is waiting. The role of the second type of bees is to exploit the food sources found out by the explorer bees. The explorer bee communicates with these bees and informs them about what they have found, the explorer bees then grade the quality of available food sources for exploitation as found out be explorer bees and then decide on to which food source to exploit. The area in the hive where this proliferation of information takes palace is called the dancing area. Each explorer bee moves around in a random fashion, a movement which the authors termed as" waggle dance". It is generally noticed that the better the quality of the food source the longer the explorer bee dances. The other bees after deciding on a food source for exploitation embark on a journey to find new food sources around the vicinity of the already found food source.

Lastly there is the last class of bees whose sole responsibility is to wander around in hunt of new unexplored food sources which are to be used as replacement for those food sources that have been exhausted or due to external factors no longer remain as profitable as before.

Karaboga et al [1] developed a mathematical model of this behavior and captured the interactions amongst the bees in the form of simple linear equations. The authors classified the bees into three categories namely:

- Employee bees
- Onlooker bees
- Scout Bees

The "employee bee" acts like the explorer bee and looks around for new food sources, for simplicity only the nectar value of the food source is used to determine the quality of the food source. The employee bee carries out a local search around a found food source in search of better sources. The local search around a food source is modelled using equation (1).

$$X_{ij} = Y_{ij} + \varphi * (Y_{kj} - Y_{ij}) \qquad (1)$$

Where X_{ij} is the new food source, Y_{ij} is the current food source,' k' is a real number that is not equal to 'i' and 'φ' is a uniformly distributed random number in the range (-1,1). The random number 'k' determines the magnitude of the movement from the current position where as 'φ' determines the axis of the coordinate systems along which the movement will take place.

Once a new food source is found around an existing food source, the quality of both sources is evaluated and a greedy selection is made between the two. The fitness all the found food sources is normalized in a range between 0 to 1, using equation (2)

$$P_i = \frac{fitnessi}{\sum_{j=1}^{No.\ of\ food\ sources} fitnessj} \qquad (2)$$

The "onlooker bee" is used for exploitation of found food sources as well as searching new sources. Each onlooker bee makes a stochastic selection of one of the food sources found by employer bees and works around the selected food source. Once an onlooker bee has selected a food source its behavior is same as that of the employee bee. Onlooker bee makes use of the same equation for local search.

The third class of bees called the “scout bees” are used to find new food sources that will be used to replace exhausted or abandoned food sources. In general the number of scout bees in a hive is 5-6% of the total number of bees. In this algorithm, the authors have limited the number of scout bees to a single bee.

There are three parameters of the Artificial Bee Colony algorithms namely the number of food sources, the colony size of the bee and limit. The authors in their work proposed an equal number of employee bees and onlooker bees in the population of bees, the size of the population is limited to half of the total number of food sources. The last parameter called the limit is a positive integer that is used to decide as to when a food sources should be abandoned. As mentioned earlier both the employee bee and the onlooker bee look for new food sources around the old source and on finding one they make a greedy selection between the two, if the previous food source was better than the new one then we increment the trail count associated with the food source, we check if the trial count has exceeded the limit, if it has then we abandon the food source and the scout bee replaces the old source with a new food source and setting the trial count of the new food source to 0. The limit parameter is helpful in getting out of suboptimal solutions when the algorithm gets trapped in local maxima or minima.

Pseudo code of the Artificial Bee Colony Algorithm

1. Initialize the Control Parameters of the ABC Algorithm: initialize the number of food sources, number of bees, max cycle, limit etc.
2. Generate food sources using random initialization and evaluate fitness of each food source
3. Memorize the best food source found so far
4. Set iteration =0
5. Repeat
   a) Send employee bees and look for new food sources using equation (1).
   b) Calculate the fitness of each food source and normalize its fitness in the range (0, 1) given by $prob_i$ using equation (2)
   c) Generate random number r and if r <= $prob_i$, send the onlooker bee at food source F_i
   d) Check the trail count of each food source, if for any food source F $trail_F$ = limit, abandon the food source and generate a new food source as its replacement.
   e) Memorize the best food source found in the current iteration

f) iteration = iteration+1
until (iteration = max cycle)

The authors demonstrated the potency of this new meta-heuristic by applying it to known benchmark functions like the Sphere function, Rastrigin function, Rosenbrock function etc. The results showed that ABC outperformed other well-known swarm based heuristics like the particle swarm optimization (PSO), ant colony optimization (ACO) as well as evolutionary algorithm like the genetic algorithm (GA).

The Artificial Bee Colony algorithm today has been applied to numerous complex optimization problems spanning different domains of computing and has shown to be well capable of optimizing them. The performance of ABC drew the attention of many researchers who have gone on studying the algorithm further, identifying the shortcomings in the algorithm as well as addressing them.

Today a wide variety of hybrid and improved ABC algorithms have been proposed each proposal overcomes some deficiency in the conventional algorithm and thereby improving on the original algorithm.

This chapter contains two such techniques based on the ABC algorithm for solving the dynamic node deployment problem in wireless sensor networks. The remainder of the chapter is organized as follows section 3.2 contains the model used for calculating the area coverage of the deployment, section 3.3 contains the different sensing models commonly used for detecting events, section 3.4 contains the approach adopted by Ortuz et al for applying ABC on the problem followed by section 3.5 that contains the two proposed modifications to the conventional ABC algorithm.

3.2. Coverage calculation strategy

Though there exist multiple ways of calculating area coverage in literature ranging from geometric random graph-based methods, coverage determined by minimal, maximal exposure paths, counting holes etc., for our implementation we have adopted the grid-based approach of WSN. In this approach, the entire area was divided into square grids of equal size and the four endpoints of a grid were considered as grid points. The coverage is measured as the total number of such grid points covered by all the sensors divided by the total number of grid points in the mission area i.e.

$$\text{Coverage} = \frac{\sum_{j=1}^{n} Cj}{Total\ number\ of\ gird\ points} \text{x}100\% \quad (3)$$

Where C_j = number of grid points covered by sensor j.

3.3. Evaluation of effectiveness of a solution

There are two commonly used detection models for calculating the area coverage in wireless sensor networks namely the binary model and the probabilistic model. The binary model only cares about the detection range of the sensor and the Euclidian distance between the sensor and the point under consideration. It assumes that if the point to be sensed is within a sensor's sensing radius then the sensor will with absolute certainty be able to sense the point. The binary model is expressed in the following manner:

Consider there are m sensors in the deployment scheme and each having a radius r , if a sensor S_i is positioned at a point, P(a,b) and the Euclidian distance between the sensor S_i and a point X be denoted by $D(S_i,X)$.The binary model is mathematically expressed using equation(4)

$$C_{ab} = \begin{cases} 1, & D(Si, X) < r \\ 0, & otherwise \end{cases} \quad (4)$$

The probabilistic model, on the other hand, states that the probability of detection of two points within the sensing radius of a sensor S is not always the same. The probabilistic model introduces a notion of uncertainty in the detection of a point if the Euclidian distance between the sensor and the point is greater than a distance denoted by $r_{eff.}$. The model is expressed using the equation (5)

$$C_{ab} = \begin{cases} 0 & if\ r + reff \leq D(Si, X) \\ e^{\frac{-\gamma 1 \alpha 1^{\beta 1}}{\alpha 2^{\beta 2}} + \gamma 2} & if\ r - reff < D(Si, X) < r + reff \\ 1 & if\ D(Si, X) \leq r - reff \end{cases} \quad (5)$$

Here γ_1, β_1, β_2 are measuring parameters, $\alpha 1 = r_{eff} - r + D(S_i, X)$ and $\alpha_2 = r_{eff} + r - D(S_i, X)$ and γ_2 is the disturbing factor, r_{eff} is the detection uncertainty range.

In our work, we have adopted the probabilistic model of detection. If an area X_{ovp} is covered by k_{ovp} sensors then a measure of the coverage is given by equation (6)

$$C_{ab}(X_{ovp}) = 1-\prod_{Si \in Sovp}(1 - Cab(Si)) \qquad (6)$$

The fitness of a particular coverage can be calculated with respect to a coverage threshold (C_{thres}) as in equation (7)

$$C_{ab}(X_{ovp}) >= C_{thres} \qquad (7)$$

3.4. Dynamic Node Deployment in WSNs using Artificial Bee Colony Algorithm

Ozturk et al [15] in their work demonstrated how node deployment problem can be solved using ABC algorithm, their results showed that ABC outperformed PSO by generating deployments that had better area coverage than the ones generated by PSO in much fewer iterations.

Pseudo code for dynamic node deployment using ABC algorithm

1. Initialize the Control Parameters of the ABC Algorithm: initialize the number of food sources, number of bees = colony size/2, limit etc. and randomly generate the positions of the sensor nodes,
2. Set Iteration = 0
   a) Initialize the food source positions
   b) Evaluate the nectar amount (fitness) of food sources
3. Repeat Until (cycle is not equal to MAXCYCLE)
   for each employee bee do
   a) Search in the neighbourhood of the food source for new solutions using the equation.

$$X_{ij} = Y_{ij} + \varphi * (Y_{kj} - Y_{ij})$$

   where j is the position of a randomly selected sensor, φ is a random number, where $\varphi \in [-1,1]$
   b) Check if X_{ij} is within the bounds of the monitoring area
   c) Evaluate the fitness of the new food source using equation (6)

d) Make a greedy selection between old solution and the new solution

4. Compute the probability $Prob_i$ of the solution using equation

$$Prob_i = \frac{0.9*fitnessi}{fitnessbest}+0.1$$

5. For each onlooker bee do

a) Generate a random number rand ϵ (0, 1) and select a food source depending on the value of $Prob_i$ and rand

b) Look for new food source in the neighbourhood using the equation (1)

c) Check if the new solution is within the bounds of the area

d) Evaluate the fitness of the new food source using equation (6)

e) Make a greedy selection between old solution and the new solution

6. Iterate through the trial array of food sources

if (the trials for a food source is greater than max trials)

a) Replace the food source with a randomly generated food source

b) Set trial = 0

7.Memorize the best solution found so far

8.cycle = cycle +1

3.5. Proposed Techniques for Dynamic Node Deployment in WSNs

3.5.1 ABC algorithm with crossover and opposition based selection for dynamic node deployment (modified ABC I)

Researchers through their works have demonstrated that the optimization capability of the conventional ABC algorithm can be considerably improved by incorporating various techniques, operations akin to other swarm/evolutionary optimization algorithms. These modifications, in general, outperform the performance of ABC when applied to domain specific problems or standard benchmark functions [8] [9] [16] [20] [21].

In the first proposed technique (this algorithm is referred to as modified ABC I from here on in this text), two new phases in addition to those of the conventional ABC have been incorporated to enhance the performance of the algorithm. The two additional phases are the crossover phase and the opposition based greedy selection.

The crossover phase contains a multipoint crossover operation (pictorially depicted in Figure 3.1) akin to the GA. In GA the crossover operation is used for the guided search phase which is responsible for the convergence of the algorithm on an optimal solution or a suboptimal one.

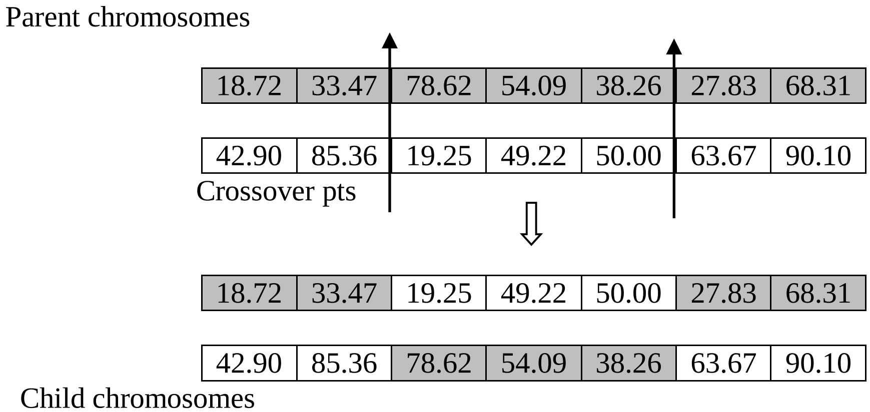


Figure 3.1: Two point crossover

The crossover operation aims at finding new solutions that inherit the characteristics of their parent solutions as a result the new solution exhibits similar traits as the parent solutions, this property of the crossover operation causes a good solution to converge towards the optimal solution. In our work, the genotype for the crossover operation is value encoded string which contains the position of the sensors in a deployment sequence. A typical genotype is represented as 1 x 2*M matrix where M is the number of sensors. The first index of the matrix stores the x coordinate of the first sensor, the second index stores its y coordinate and so on for the rest of the sensors.

In our work, for the crossover operation a population of size equal half of number of food sources is generated using tournament selection. For generating the population the food sources were sorted in decreasing order in accordance to their fitness value and three random food sources were chosen amongst them, a three-way comparison of their fitness values was made and the best food source amongst the three was included in the population. Once the population was created we choose a fixed amount of food sources around the center of the sorted food sources for replacement. A number of food sources to be selected is the product of crossover probability and the number of food sources.

For each of the selected food source two parents were selected arbitrarily y from the population and after generating two random crossover points we crossed the two parents to produce two new solutions (food sources).The selected food source is then replaced by the best food source among the two newly created ones (offsprings generated by crossover) and itself. If the selected food sources is replaced by one of the newly created food source then set the trial count for that food source to 0.

Another possible improvement in the ABC algorithm is how we try to replace the current solution by some better solutions towards some optimal solutions. Karaboga in his paper proposed the use of a randomly generated food source for replacing a current food source. Researchers studying the ways of improving the performance of ABC algorithm suggested that in any optimization problem while generating random solutions for exploration and exploitation we should also try out the opposite solution [30]. The basic underpinning of this argument is that the probability of an opposite solution being fitter than a randomly selected solution is about 50%. The opposite solution of a particular solution is generated using the equation (8).

$$A_{ij}' = UB + LB - A_{ij} \tag{8}$$

here A_{ij}' represents a newly formulated solution from an older solution represented by A_{ij}, UB denotes the upper bound of the solution space i.e. UB represents the upper bound on the value attained by the dimension A_{ij} and similarly LB denotes the lower bound on the value of dimension A_{ij}. Once an opposite solution is generated, it is compared with the randomly generated solution and the fitter amongst the two in chosen for further exploration.

The generation of opposition based learning solutions can be carried out at different stages of the execution of the algorithm the most common is during the initialization phase and the solution replacement phase.

In our work, the opposition based learning phase is carried out immediately after the crossover phase. Once we had replaced all the selected food sources in the crossover phase we generated the opposite solution for each of the food sources generating twice the number of original food sources (assuming there were F number of food sources to start with) in the process. The opposite food sources were generated using equation (8). With respect to this problem, the parameter a_{ij} represented the current location of the sensor node and the UB and LB denotes the upper bound and lower bound of the position coordinates of the sensor A_{ij} respectively. We then applied a greedy selection where we sorted the 2F food sources in the decreasing order of their fitness values and selected the top F food sources. Empirically it was found that the algorithm performed best when this operation is carried out around 30% of the times.

Pseudo code for dynamic node deployment using modified ABC I algorithm

1. Initialize the Control Parameters of the ABC Algorithm: initialize the number of food sources, number of bees = colony size/2, limit etc., randomly generate the positions of the sensor nodes, set the crossover_probability = 20%.
2. Set Iteration = 0

    Initialize the food source positions

    Evaluate the nectar amount (fitness) of food sources
3. Repeat Until (cycle is not equal to MAXCYCLE)

    for each employee bee do

    a) Search the neighbourhood of the food source for new solutions using the equation.

    $$X_{ij} = Y_{ij} + \varphi * (Y_{kj} - Y_{ij})$$

    where j is the position of a randomly selected sensor, φ is a random number, where $\varphi \in [-1,1]$

    b) Check if X_{ij} is within the bounds of the monitoring area

    c) Evaluate the fitness of the new food source using equation (6)

    d) Make a greedy selection between old solution and the new solution

4. Compute the probability $Prob_i$ of the solution using equation

    $$Prob_i = \frac{0.9 * fitnessi}{fitnessbest} + 0.1$$
5. For each onlooker bee do

    a) Generate a random number $r \in (0, 1)$ and select a food source depending on the value of $Prob_i$ and r

    b) Look for new food source in the neighbourhood using the equation (1)

c) Check if the new solution is within the bounds of the area

d) Evaluate the fitness of the new food source using equation (6)

e) Make a greedy selection between old solution and the new solution

6. Select a fixed number of food sources (N) for replacement using the formula

N = crossover_probability * number of food sources

a) Generate the mating pool for crossover operation using tournament selection

b) For each selected food source do

i) Make a random selection of two food sources that will serve as parent genotype

ii) Generate two random crossover points and swap genes as in Figure 3.1

iii) Make a three way greedy selection amongst the selected food source and the two newly created food sources as a candidate for replacing the selected food source

7. Generate a uniformly distributed random number r, where $r \in (0,1]$.

If $r < 0.30$ do

a) Generate opposite food sources for each food source using equation (8)

b) Sort the 2F food sources in decreasing order of their fitness values

c) Select the top F food sources

8. Iterate through the trial array of food sources

if (the trials for a food source is greater than max trials)

a) Replace the food source with a randomly generated food source

b) Set trial = 0

9. Memorize the best solution found so far

10.cycle = cycle +1

3.5.2 Artificial Bee Colony algorithm with crossover and modified local search for dynamic node deployment (modified ABC II)

A potential area of improvement in the ABC is in its local search. The ABC algorithm uses the equation (1) for exploring around an existing solution. Both the employee bee as well as the onlooker bee use the same equation for finding new solutions. Researchers have studied the local search characteristic of the algorithm and have suggested multiple approaches for improving the exploration capability of the algorithm, their study exposed two areas where the local search falls short, namely the random selection of food sources around an old food source for exploration and the way it balances exploration and exploitation.

One approach suggested for overcoming the first shortcoming of the ABC is to modify the local search so that randomly searching around an old solution the search for new solutions be guided along the global best solution found till that instant. This approach derives its roots from another popular swarm intelligence meta-heuristic called particle swarm optimization. In PSO while the swarm of particles is exploring the solution space, each particle periodically recalibrates its movement as guided by two parameters namely pBest and gBest. The particle factors in its own past best position and the position of the fittest particle in the swarm to adjust its velocity and position in the search space. The PSO algorithm converges at a very rapid rate in comparison to other contemporary meta-heuristics like the GA, ACO and even ABC. This high convergence rate of PSO is attributed to the fact that each particle always tries to move towards the best solution found so far.

Taking a leaf out of the book, researchers have modified the local search of the conventional ABC algorithm to take into consideration the current global best solution and try to search for new solution in its direction rather than randomly looking around the solution under consideration. The local search guided by global best is expressed by equation (9)

$$X_{ij} = Y_{ij} + \varphi * (gBest_j - Y_{ij}) \qquad (9)$$

The results of their experiments using the modified local search in the canonical ABC algorithm showed that the amendment made in the local search enhanced the optimizing capability of the algorithm. In our work, we have used another variant of global best guided local search expressed equation (10).

$$X_{ij} = gBest_j + \varphi * (Y_{kj} - Y_{ij}) \qquad (10)$$

Where, $gBest_j$ represents the j^{th} parameter of the global best solution found so far, Y_{kj} and Y_{ij} represent the current solution under consideration and a randomly generated solution around the current solution respectively and φ is a random number ranging from -1 to 1.

The second shortcoming of the ABC is in the way it balances exploration and exploitation phase. The balance between exploration and exploitation is maintained by the employee bees and the scout bees. The studies done in the past pointed out the fact that the local search of ABC favoured exploration to exploitation, which generally results in valuable domain knowledge gathered by the swarm of bees being thrown away and this in turn affects the convergence rate of the algorithm which becomes evident when ABC is applied for optimizing large sized complex problems. The solution to this problem is the use of variable length step size.

In our work, we incorporated a memetic based local search for handling the exploration and exploitation. Memetic search internally uses the golden section search to determine the step size. The golden section search (GSS) cuts down the solution space in accordance to the fitness of the solutions by updating two parameters namely ‘a’ and ‘b’. The GSS works as follows:

Pseudo code of the GSS algorithm

```
While (abs(a-b) < ϵ)
        F_S1 = b – ( b - a ) * Ψ;
        F_S2 = a + ( b – a ) * Ψ;
        If ( Obj(F_S1) < Obj(F_S2) )
                b = F_S2;
        Else
                a = F_S1;
```

The parameter Ψ, is a constant value and is called the golden ratio. The value of Ψ is 0.618. Obj (F_{S1}) is the fitness of the solution generated by the objective function, Obj () with step size F_{S1}. Obj (F_{S2}) is the fitness of the solution generated by the objective function, Obj () with step size F_{S2}. ϵ is a constant which defines the ending point of the algorithm, the value of ϵ is 0.01.

The memetic search introduces a new parameter called the “perturbation rate” denoted by pRate. This parameter controls the modification of the i^{th} dimension of the newly generated solution. The working of the memetic search algorithm is shown in the figure below. The term X_{kj} in the figure denotes the j^{th} dimension of an arbitrarily selected k^{th} food source.

Pseudo code for memetic search

```
Do
    1.Initialize FS1 and FS2
        a) FS1 = b – (b - a) * Ψ.
        b) FS2 = a + (b – a) * Ψ.
    2. Generate two new solutions Yn1 and Yn2 using FS1 and FS2 respectively.
    3. If the solution Yn1 is better than the solution Yn2
        a)  Assign the value of FS2 to the variable b.
        b)  Compare the best solution so far denoted by Ybst with Yn1 and update Ybst
            accordingly.
    4.Else if Yn2 is better than Yn1
        a)  Assign the value of FS1 to the variable a.
        b)  Compare the best solution so far denoted by Ybst with Yn1 and update Ybst
            accordingly.
  If abs (a-b) < ϵ, Repeat.
     a)  For each dimension in the solution do
     b)  Generate a random number r ϵ [0,1]
     c)  If r > pRate
            Generate a new solution using the following equation
                    Ynew = Ybst + F * ( Ybst - Ykj );
     d)  else use the best solution found so far as the new solution Ynew
return Ynew
```

The above explained observations suggested that a hybrid local search that incorporated all the characteristics of the three above mentioned local searches together could possibly find better solutions. This intuition leads us to incorporate the hybrid local search in both the employee bee phase as well as in the onlooker bee phase. To summarize, the proposed hybrid local search contains the following computations.

Pseudo code for hybrid local search

For a given food source Y_{ij}, generate new solutions namely,

U_{ij} using the local search of conventional ABC algorithm using equation (1)

V_{ij} using the guided best search using equation (10)

W_{ij} using memetic search

Make a greedy selection between U_{ij}, V_{ij}, W_{ij} based on their fitness value.

$$X_{ij} = \max(\text{fitness}(U_{ij}), \text{fitness}(V_{ij}), \text{fitness}(W_{ij}))$$

To ensure that a maximum area of the solution space is searched by the algorithm within the stipulated timeframe which is expressed in terms of the parameter "MaxCycles" that is predetermined during the initialization of the algorithm's control parameters we introduced a crossover phase for the same reasons as mentioned in section 3.4.1. This improved version of the Artificial Bee Colony algorithm is referred as modified ABC II, from here on in this text.

Pseudo code for dynamic node deployment using modified ABC II

1. Initialize the Control Parameters of the ABC Algorithm: initialize the number of food sources, number of bees = colony size/2, limit etc., randomly generate the positions of the sensor nodes, perturbation rate = 0.4, ϵ = 0.01, set the crossover_probability = 20%.
2. Set Iteration = 0

   Initialize the food source positions.

   Evaluate the nectar amount (fitness) of food sources.
3. Repeat Until (cycle is not equal to MAXCYCLE)

   for each employee bee do

   a) Search the neighbourhood of the food source for new solutions using hybrid local search

   b) Check if X_{ij} is within the bounds of the monitoring area.

   c) Evaluate the fitness of the new food source using equation (6).

   d) Make a greedy selection between old solution and the new solution.
4. Compute the probability $Prob_i$ of the solution using equation

$$Prob_i = \frac{0.9 * fitnessi}{fitnessbest} + 0.1$$

5. For each onlooker bee do

   a) Generate a random number r ϵ (0, 1) and select a food source depending on the value of $Prob_i$ and r.

   b) Look for new food source in the neighbourhood using hybrid local search.

   c) Check if the new solution is within the bounds of the area.

   d) Evaluate the fitness of the new food source using equation (6).

   e) Make a greedy selection between old solution and the new solution.

6. Select a fixed number of food sources (N) for replacement using the formula

   N = crossover_ probability * number of food sources.

   a) Generate the mating pool for crossover operation using tournament selection.

   b) For each selected food source do

      i) Make a random selection of two food sources that will serve as parent genotype.

      ii) Generate two random crossover points and swap genes as in Figure 3.1.

      iii) Make a three way greedy selection amongst the selected food source and the two newly created food sources as a candidate for replacing the selected food source.

7. Iterate through the trial array of food sources

   if (the trials for a food source is greater than max trials)

   a) Replace the food source with a randomly generated food source.

   b) Set trial = 0.

8. Memorize the best solution found so far.

9. cycle = cycle +1.

This chapter covers a detailed analysis of the performance of the proposed algorithms with the conventional artificial bee colony algorithm. There are generally two ways of analyzing the performance of an algorithm, first is the theoretical analysis in this method the working of the algorithm is evaluated using mathematical tools and assertions are corroborated with the help of theorems with valid proofs and corollaries. In the theoretical analysis, the emphasis is on the key operations of the algorithm and the number of times these operations are performed. Another way of analyzing an algorithm's performance is by carrying out experiments, and observing the performance of the algorithm in a simulated test environment where various parameters are varied and the results are tabulated for later comparison.

We have used simulation as the tool for analysis of our two proposed algorithms, the rest of the chapter is organized as follows. Section 4.1 covers the details about the simulation setup, the environment as well as the assumptions made. Section 4.3 covers the simulation results of the modified ABC I algorithm under three different setups. Section 4.4 covers the simulation of the modified ABC II algorithm and the performance is measured using the same yardstick as used for the evaluation of the first proposed algorithm.

4.1 Experimental Setup

In the simulation, a sensor network with 45 mobile sensors has been used. The sensing radius is 7m, the detection error r_{eff} is half of the sensing radius, the monitoring region is a square area of 10000m^2, the value of the parameters of the probabilistic model are as follows: $\gamma1=1$, $\gamma2=0$, $\beta1=1$, $\beta2=0.5$. The crossover probability is 0.2 and the tournament population size is 20. The colony size is 80 and the number of food sources is 40. The grid based model that we used for calculating the coverage, split the surveillance area into equal sized square grids of 5x5 units each. The scenario of a random deployment of sensors is run 30 times with each run having 1000 cycles. The simulations were run on a PC with 1.80 GHz Intel i5 processor and 4 GB RAM. The source code was developed in C++ using Visual Studio 2010 IDE.

For comparing the performance of our proposed algorithm with that of standard ABC, in each iteration we generated random food sources that served as input for both the algorithms. Both algorithms worked on the same input set and produced results in the form of a 1x2S matrix, where ‘S’ is the number of sensors as shown in figure 2.

1	2	3	4			2S-1	2S
X_1	Y_1	X_2	Y_2	…	…	X_S	Y_S

Figure 4.1

The solution matrix stores the X and Y coordinates of each sensor, denoting their positions in the best deployment sequence.

4.2 Performance Metric

The metric used for evaluation of the proposed algorithms and the ABC algorithm is the parameter called “Maxcycles”. This parameter denotes the stipulated number of trials (referred to as cycles here after) that the algorithm carries out for finding better solutions starting from a randomly generated solution that was fed to it as an input.

The next sections of the chapter contain the results of the experiments carried out for comparing the performance of both the proposed algorithms with that of conventional ABC. Each simulation was executed for 30 iterations (mentioned as “runs” here after), each run internally comprised of multiple cycles, the upper bound of which is specified by the “MaxCycle” parameter.

For a fair evaluation, both the algorithms were fed the same deployment sequence as input. Each algorithm then worked on the input for a stipulated number of times as dictated by the “MaxCycle” parameter, and finally came up with a best deployment sequence found so far. The quality of both the outputs were then evaluated. The initial deployment sequence that served as input for both the algorithms was randomly generated in each run.

4.3 Performance Evaluation – modified ABC I

4.3.1 Execution of ABC and modified ABC I with MaxCycle set at 1000.

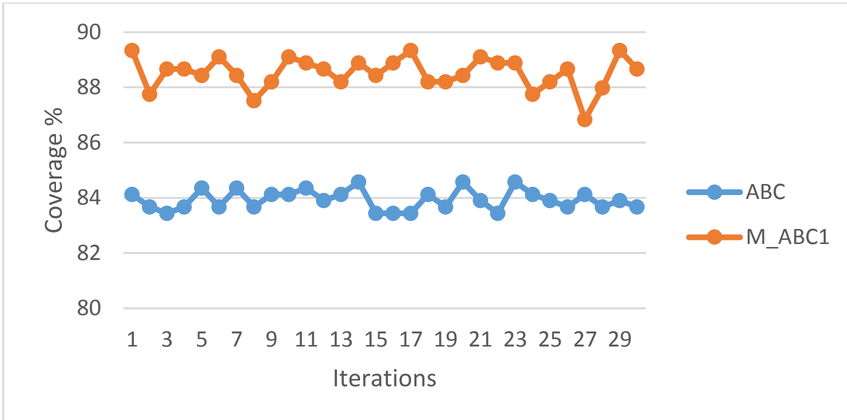


Figure 4.2

Figure 4.2 shows the execution of conventional ABC with that of modified ABC I (referred to as M_ABC1 in the legend of all figures of section 4.3 from here on). The value of the parameter "MaxCycle" for this experiment was set to 1000.The execution of ABC is marked by the line in blue where as that of modified ABC I is marked by the line in orange. The graph shows that the proposed algorithm outperforms ABC in every run. The mean coverage of ABC for this experimental setup was 83.61% whereas as that of modified ABC I was 88.41%.

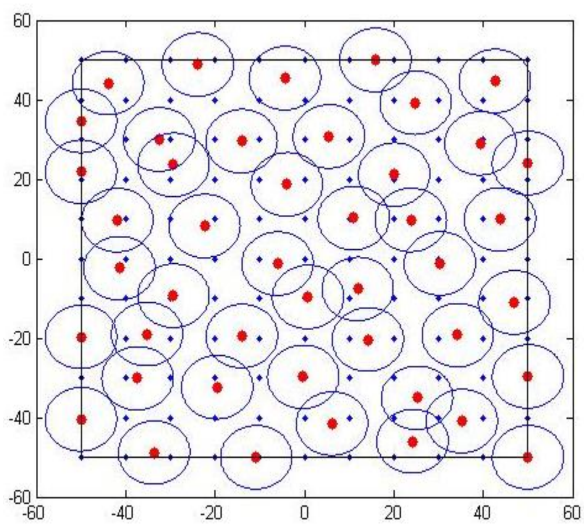


Figure 4.3

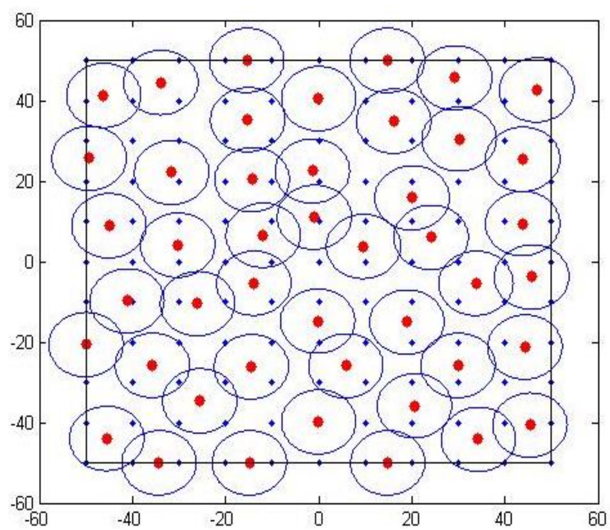


Figure 4.4

Figure 4.3 shows the best deployment sequence generated by ABC and Figure 4.4 shows the best deployment sequence generated by modified ABC I. In the figure, the points in blue colour denote the grid points whereas the points in red colour denote the position of a sensor whose sensing range is denoted by the circle in blue colour. The solution provided by ABC achieved a coverage of 85.26%, whereas the solution found by the modified ABC I algorithm had a coverage of 89.12%.

4.3.2 Execution of ABC and modified ABC I with MaxCycle set at 1500

The setup for this experiment was same as that in section 4.3.1 with the exception that for this test the value of the parameter "MaxCycle" was set to 1500.

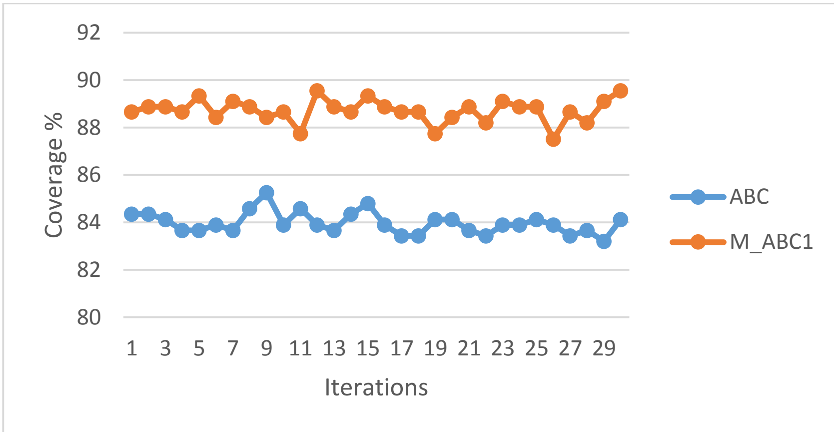


Figure 4.5

Figure 4.5 shows the execution of conventional ABC with that of modified ABC I. The execution of ABC is marked by the line in blue where as that of modified ABC I is marked by the line in orange. It can be seen that modified ABC I gets the better of ABC algorithm. The mean coverage of ABC for this experimental setup was 83.92% whereas as that of modified ABC I was 88.52%.

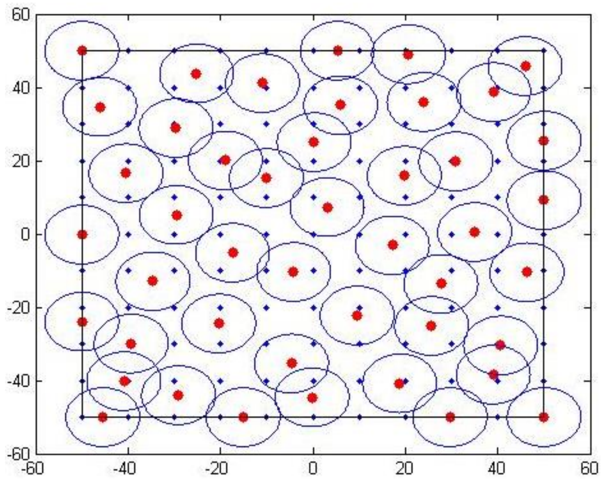


Figure 4.6

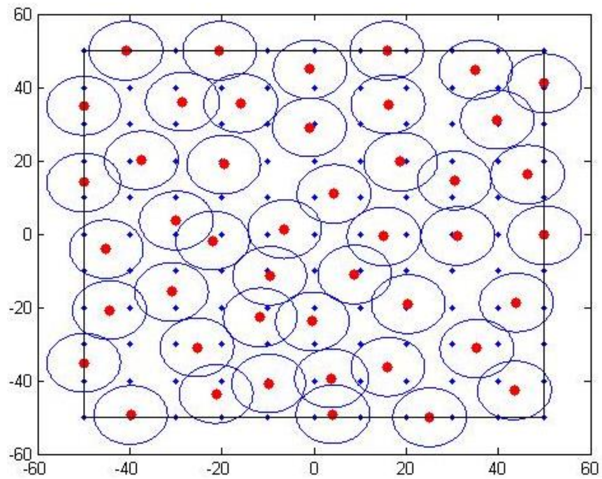


Figure 4.7

Figure 4.6 shows the best deployment sequence generated by ABC and Figure 4.7 shows the best deployment sequence generated by modified ABC I. The solution provided by ABC achieved a coverage of 84.58%, whereas the solution found by the modified ABC I algorithm had a coverage of 89.34%.

4.3.3 Execution of ABC and modified ABC I with MaxCycle set at 2000

The setup for this experiment was same as that in section 4.3.1 with the exception that for this test the value of the parameter "MaxCycle" was set to 2000.

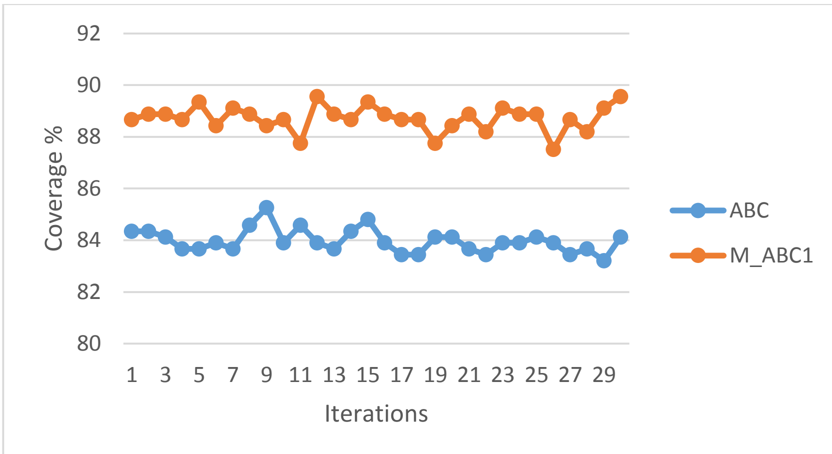


Figure 4.8

Figure 4.8 shows the execution of conventional ABC with that of modified ABC I. The execution of ABC is marked by the line in blue where as that of modified ABC I is marked by the line in orange. It can be seen that the modified ABC I outperforms the ABC algorithm in this experiment. The mean coverage of ABC for this experimental setup was 83.97% whereas as that of modified ABC I was 88.71%.

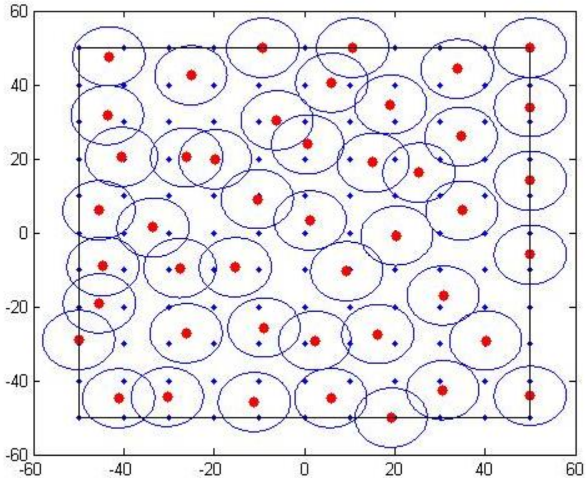


Figure 4.9

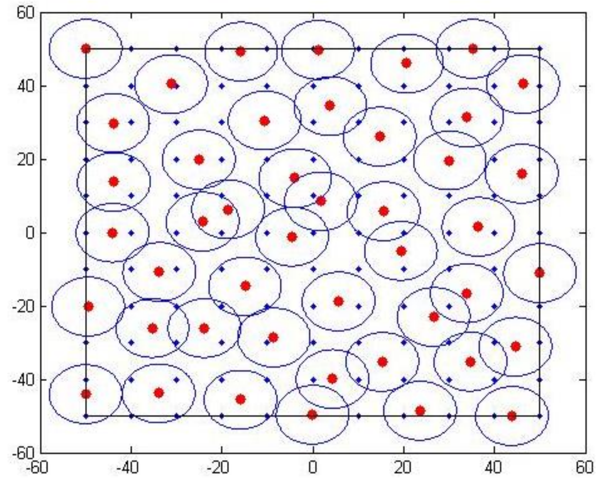


Figure 4.10

Figure 4.9 shows the best deployment sequence generated by ABC and Figure 4.10 shows the best deployment sequence generated by modified ABC I. The solution provided by ABC achieved a coverage of 85.26%, whereas the solution found by the modified ABC I algorithm had a coverage of 89.66%.

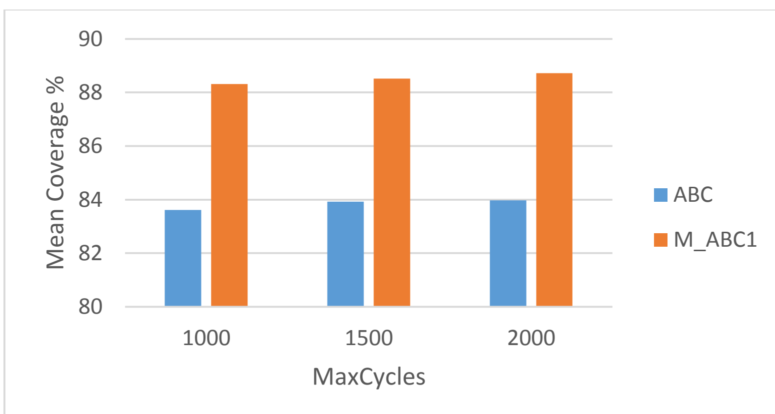


Figure 4.11

Figure 4.11 shows a comparison of the mean of the percentage area covered over 30 runs by both ABC and modified ABC I algorithm, with the value of MaxCycles as 1000, 1500 and 2000 respectively. It can be seen that the mean area covered increases as the MaxCycles increases.

4.4 Performance Evaluation – modified ABC II

4.4.1 Execution of ABC and modified ABC II with MaxCycle set at 1000.

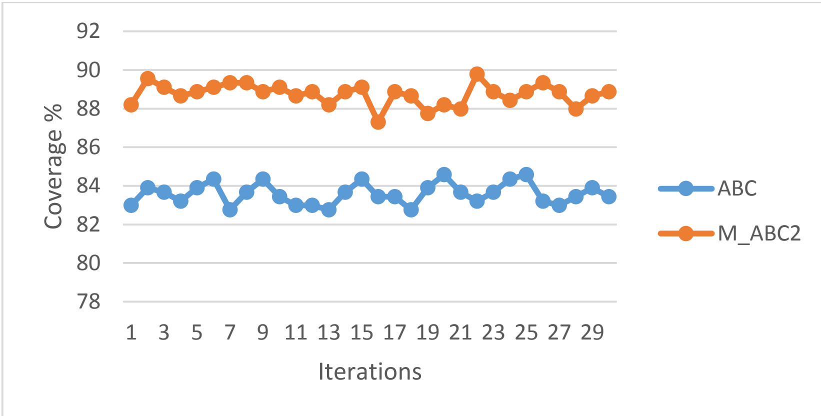


Figure 4.12

Figure 4.12 shows the execution of conventional ABC with that of modified ABC II (referred to as M_ABC2 in the legend of all figures in section 4.4 from here on). The value of the parameter "MaxCycle" for this experiment was set to 1000.The execution of ABC is marked by the line in blue where as that of modified ABC II is marked by the line in orange. The graph shows that the proposed algorithm outperforms ABC in every run. The mean coverage of ABC for this experimental setup was 83.58% whereas as that of modified ABC II was 88.84%.

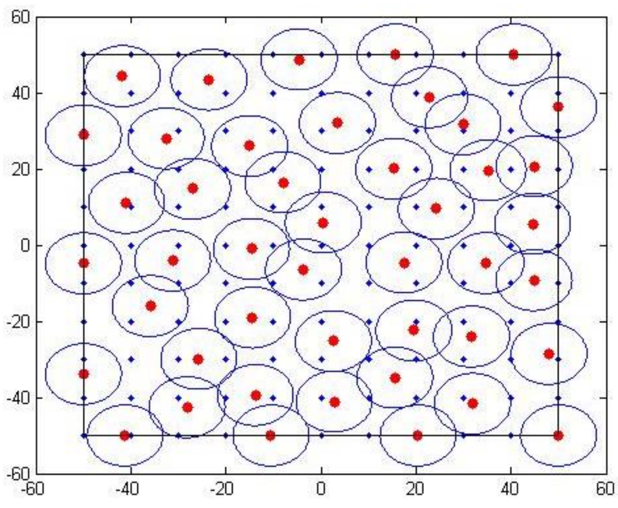


Figure 4.13

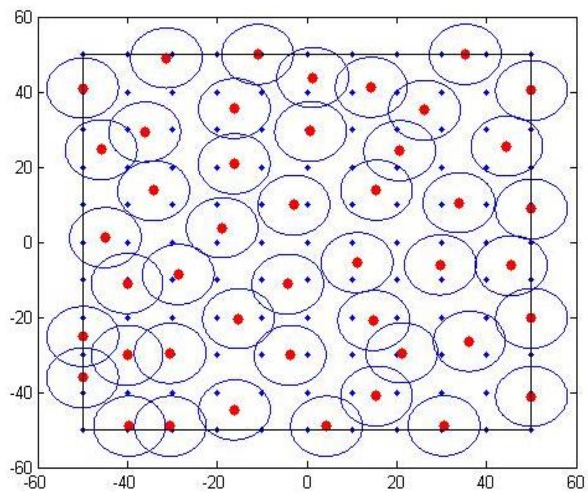


Figure 4.14

Figure 4.13 shows the best deployment sequence generated by ABC and Figure 4.14 shows the best deployment sequence generated by modified ABC II. The solution provided by ABC achieved a coverage of 84.58%, whereas the solution found by the modified ABC II algorithm had a coverage of 89.79%.

4.4.2 Execution of ABC and modified ABC II with MaxCycle set at 1500

The setup for this experiment was same as that in section 4.4.1 with the exception that for this test the value of the parameter "MaxCycle" was set to 1500.

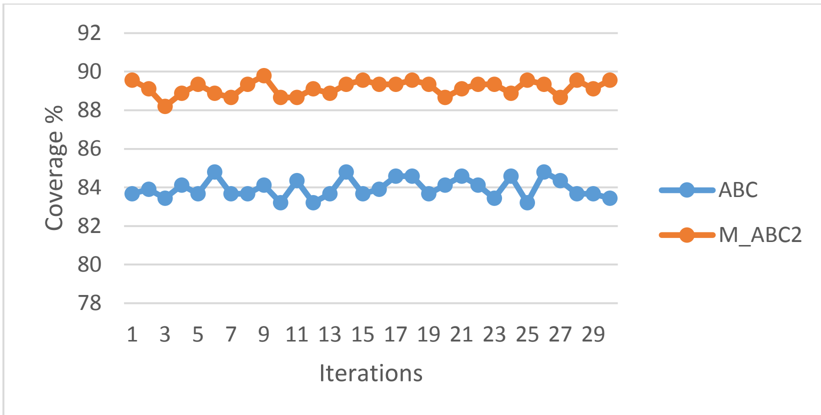


Figure 4.15

Figure 4.15 shows the execution of conventional ABC with that of modified ABC II. The execution of ABC is marked by the line in blue where as that of modified ABC II is marked by the line in orange. It can be seen that modified ABC II gets the better of ABC algorithm. The mean coverage of ABC for this experimental setup was 83.95% whereas as that of modified ABC II was 89.25%.

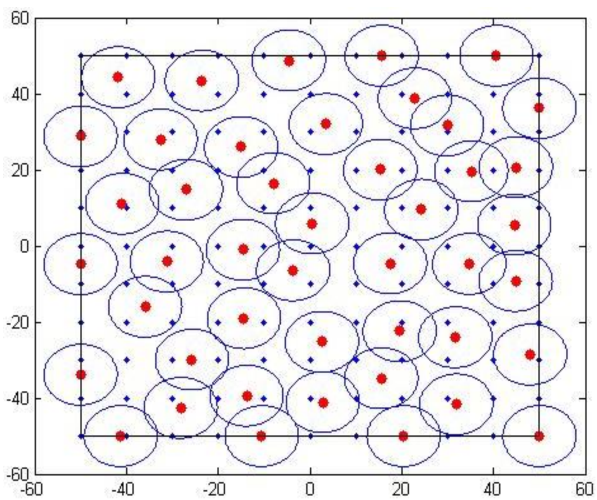


Figure 4.16

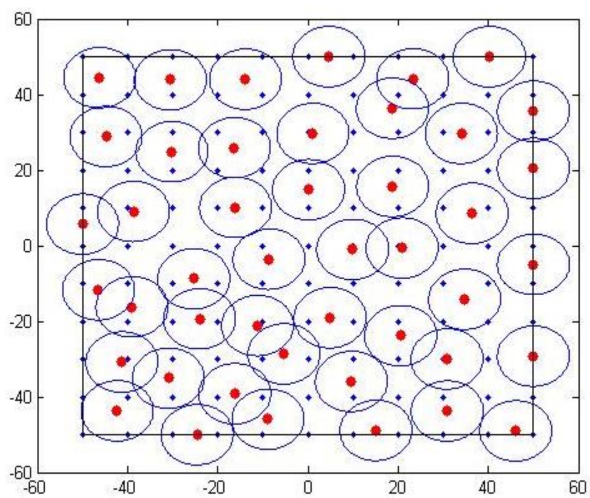


Figure 4.17

Figure 4.16 shows the best deployment sequence generated by ABC and Figure 4.17 shows the best deployment sequence generated by modified ABC II. The solution provided by ABC achieved a coverage of 84.80%, whereas the solution found by the modified ABC II algorithm had a coverage of 90.02 %.

4.4.3 Execution of ABC and modified ABC II with MaxCycle set at 2000

The setup for this experiment was same as that in section 4.4.1 with the exception that for this test the value of the parameter "MaxCycle" was set to 2000.

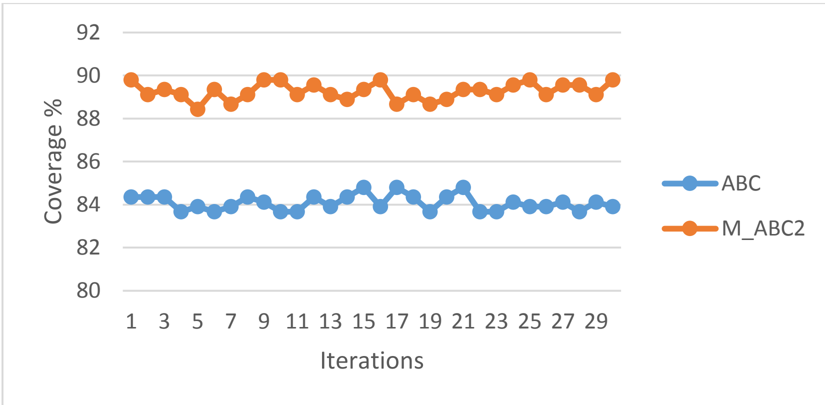
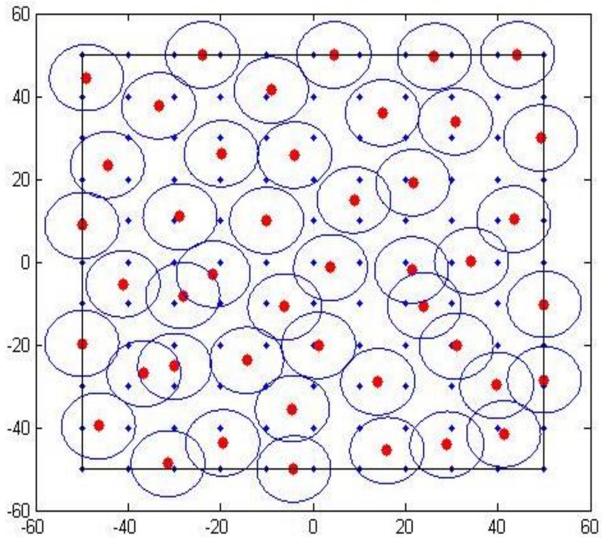


Figure 4.18

Figure 4.18 shows the execution of conventional ABC with that of modified ABC II. The execution of ABC is marked by the line in blue where as that of modified ABC II is marked by the line in orange. It can be seen that the modified ABC II outperforms the ABC algorithm in this experiment. The mean coverage of ABC for this experimental setup was 83.95% whereas as that of modified ABC II was 89.25%.



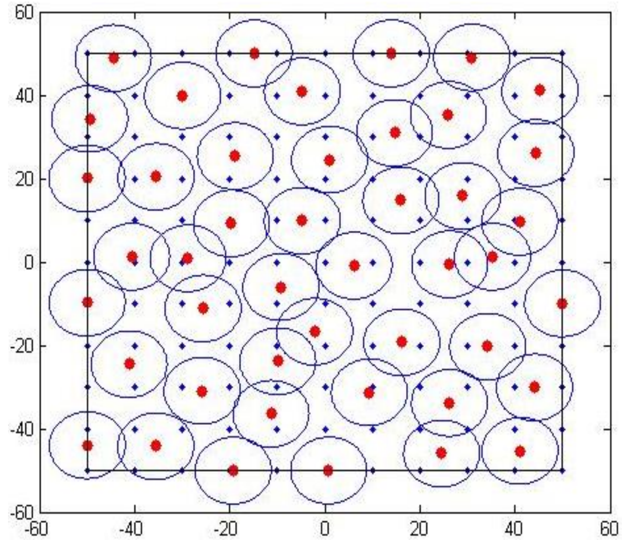


Figure 4.19 **Figure 4.20**

Figure 4.19 shows the best deployment sequence generated by ABC and Figure 4.20 shows the best deployment sequence generated by modified ABC II. The solution provided by ABC achieved a coverage of 84.80% whereas the solution found by the modified ABC II algorithm had a coverage of 90.24%.

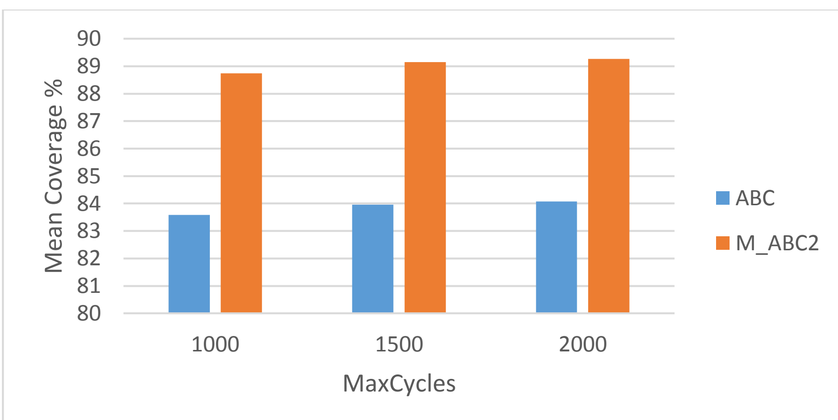


Figure 4.21

Figure 4.21 shows a comparison of the mean of the percentage area covered over 30 runs by both ABC and modified ABC II algorithm, with the value of "MaxCycles" as 1000, 1500 and 2000 respectively. It is evident from the graph that the mean area covered increases as the MaxCycles increases.

Chapter 5 Conclusion and Future Scope of Work

We devoted our work to the problem of node deployment in wireless sensor networks with the objective of maximizing the coverage area and concerted our efforts in the direction of finding better solutions by drawing inspiration from the past work carried out by researchers pertinent to the problem at hand as well as from those seminal works on improving the performance of artificial bee colony algorithm in general, to come up with alternate solutions that could help us modify the existing artificial bee algorithm framework by incorporating changes in existing operations as well as introducing new ones so as to enhance the potency of the algorithm for dealing with the problem.

In our work, we proposed two different algorithms built on the outlines of the conventional artificial bee colony algorithm for solving the node deployment problem. The results of both the proposed algorithms when compared with those of the conventional artificial bee colony algorithm executed under similar simulation environment show that both the proposed algorithms outperform the artificial bee colony algorithm in tackling the problem. Multiple metrics was used to gauge the performance of both the algorithms the details of which are covered in chapter 4. We were able to improve the exploration capability of ABC algorithm (in our first proposed algorithm) by introducing a crossover operation analogous to that of GA and the use of opposition based knowledge of the found solutions and greedily selecting the best solution out of the newly generated ones and the existing solutions. We also improved upon the haphazard nature of ABC algorithm's local search (in our second proposed algorithm) that favours exploration to exploitation by introducing multiple local search techniques namely the guided best local search which directs the search towards best solution found so far and the memetic search based local search which internally uses GSS for varying the step size of the local search thereby balancing the exploration and exploitation characteristics of the local search.

Though the proposed algorithms do outperform the conventional ABC algorithm in solving the node deployment problem, the room of improvement in our proposed work still remains wide open. There do exist some alternate approaches that could be used in conjunction of our proposed changes or even in a standalone fashion. Possible modification could be the use of a levy distribution, Poisson distribution etc. to determine the variable step size for enhancing exploitation of the domain knowledge accumulated by the swarm. Another possible scope of optimizing the proposed algorithms could be in the way we generate the initial population, as of now we generate a random deployment of sensors which gives a very uneven distribution with sensors at times placed very close to each other or on top of each other, the algorithm then spreads the sensors to cover more and more area, it is possible that with a uniform initial distribution would assist the underlying meta-heuristic to find deployments with better area coverage. For uniform initial deployment geometrical solutions such as Voronoi diagram with Delaunay triangulation can be studied or ideas can be borrowed from other meta-heuristics like those based on Van Der Waal forces etc. can be used.

Though the second proposed algorithm currently uses a hybrid local search involving the haphazard local search of the ABC, the guided best search, and the memetic search, researchers have disclosed other hybrid local search methods like the hybrid harmony search (HS) with ABC and ABC with Simulated Annealing (SA), these hybrid local searches overcome the limitations of the local search of the ABC algorithm thereby enhancing the optimization capability of the ABC algorithm, such hybrid local searches can be studied for improving the performance of the proposed algorithms further.

In our work, the problem of holes in the coverage has not been addressed by either of the algorithms. The holes arise due to the fact that we have controlled the amount of overlapping allowed, we could use lower value of coverage threshold parameter to allow more overlapping but we do need to significantly increase the number of sensors as each sensor will now sense lesser area due to the overlapping, the other way of covering the holes with the use of variable radius sensors, literature is today available on multiple ways of covering holes using myriad of different techniques, such techniques can be incorporated into our proposed algorithms for increasing coverage even further.

Last but not the least, in our work we have solely focused on increasing the area coverage, our algorithm could easily be extended for optimizing other characteristics of a sensor network like the lifetime of the network, minimize the number of sensors required to cover the area, minimize the communication cost etc. i.e. a multi objective fitness function can be used.

The results corroborate our claims on the efficiency of both the proposed algorithms and show that the modifications carried out leads to enhancement of optimization capability of ABC as we are able to cover more area using the modified ABC algorithms in comparison to the conventional ABC algorithm. Therefore, this approach can be used for designing WSN for enhancing area coverage or used as the base of other algorithms that deal with other complex issues of WSN like clustering, routing etc.

References

[1] D Karaboga, B Akay,"A comparative study of Artificial Bee Colony algorithm", Applied Mathematics and Computation 214 (2009) 108–132.

[2] K Balasubramani, "A Comprehensive review of Artificial Bee Colony Algorithm", International Journal of Computers & Technology, Volume 5, No. 1, May -June, 2013, ISSN 2277-3061.

[3] I. Maleki, S. R. Khaze, M. M. Tabrizi, A. Bagherinia "A new approach for area coverage problem in wireless sensor networks with hybrid particle swarm optimization and differential evolution algorithms", International Journal of Mobile Network Communications & Telematics (IJMNCT) Vol. 3, No.6, December 2013.

[4] X. Yu, N. Liu, W. Huang, X. Qian, T. Zhang," A Node Deployment Algorithm Based on Van Der Waals Force in Wireless Sensor Networks", Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume 2013, Article ID 505710.

[5] H. Zhang, C. Liu," A Review on Node Deployment of Wireless Sensor Network**",** IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 3, November 2012 ISSN (Online): 1694-0814.

[6] C. Zhu, C. Zheng, L. Shu, G. Han," A survey on coverage and connectivity issues in wireless sensor networks", Journal of Network and Computer Applications 35 (2012) 619–632,Elsevier Publication.

[7] S. Li, Z.Qin, L.Shan, R. Zhang, X. Yang "A survey on target tracking in well-deployed wireless sensor networks", Journal of software, vol. 9, no. 5, May 2014, IEEE.

[8] W. Yu, J. Zhang, W. Chen, "Adaptive Artificial Bee Colony Optimization**",** *GECCO'13*, July 6–10, 2013, Amsterdam, Netherlands. Copyright © 2013 ACM.

[9] X. Kong, S. Liu, Z. Wang" An Improved Artificial Bee Colony Algorithm and Its Application", International Journal of Signal Processing, Image Processing and Pattern Recognition Vol.6, No.6 (2013), pp.259-274.

[10] X. Liu, D. He "Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks", Journal of Network and Computer Applications 39 (2014) 310–318.

[11] Md. Alam, Md. Islam, K. Murase, "Artificial Bee Colony Algorithm with Adaptive Explorations and Exploitations: A Novel Approach for Continuous Optimization", International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 8– No.2, January 2015.

[12] G. Fan, S. Jin "Coverage Problem in Wireless Sensor Network: A Survey", Journal of Networks, Vol. 5, No. 9, September 2010, 2010 Academy publisher.

[13] X. Wang , J.J. Ma , S. Wang, D.W. Bi "Distributed Particle Swarm Optimization and Simulated Annealing for Energy-efficient Coverage in Wireless Sensor Networks", Sensors 2007, 7, 628-648, ISSN 1424-8220.

[14] G.Wang , L. Guo ,H. Duan ,L. Liu ,H. Wang "Dynamic Deployment of Wireless Sensor Networks by Biogeography Based Optimization Algorithm", Journal of Sensor and Actuator Networks, ISSN 2224-2708, 2012.

[15] C. Ozturk, D. Karaboga, B. Gorkemli, "Probabilistic Dynamic Deployment of Wireless Sensor Networks by Artificial Bee Colony Algorithm", Sensors 2011, 11, 6056-6065, ISSN 1424-8220**.**

[16] E. Mezura-Montes, R.E. Velez-Koeppel "Elitist Artificial Bee Colony for Constrained Real-Parameter Optimization**",** CEC'2010**,** IEEE.

[17] J. Wang, S. Medidi "Energy Efficient Coverage with Variable Sensing Radii in Wireless Sensor Networks", National Science Foundation grants CNS-0454416 and ECS-0646215.

[18] O. Banimelhem, M. Mowafi, W. Aljoby, "Genetic Algorithm Based Node Deployment in Hybrid Wireless Sensor Networks", Communications and Network, 2013, 5, 273-279, (http://www.scirp.org/journal/cn).

[19] S Babaie, S. S. Pirahesh, "Hole Detection for Increasing Coverage in Wireless Sensor Network Using Triangular Structure", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 2, January 2012, ISSN (Online): 1694-0814.

[20] B.Wu, C. Qian, W. Ni, S. Fan "Hybrid harmony search and artificial bee colony algorithm for global optimization problems", Computers and Mathematics with Applications 64 (2012) 2621–2634.

[21] P. Shi, S. Jia,"A Hybrid Artificial Bee Colony Algorithm Combined with Simulated Annealing Algorithm for Traveling Salesman Problem", 2013 International Conference on Information Science and Cloud Computing Companion.

[22] T. K. Sharma, M. Pant, V.P.Singh, "Improved Local Search in Artificial Bee Colony using Golden Section Search".

[23] Z. Chen, S. Li, W. Yue, "Memetic Algorithm-Based Multi-Objective Coverage Optimization for Wireless Sensor Networks", Sensors 2014, 14, 20500-20518, ISSN 1424-8220.

[24] S. M. Jameii, S.Jameii, "Multi-objective energy efficient optimization algorithm for coverage control in wireless sensor networks", International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Vol.3, No.4, August 2013.

[25] J. Jia, J. Chen, G. Chang, Y. Wen, J. Song, "Multi-objective optimization for coverage control in wireless sensor network with adjustable sensing radius", Computers and Mathematics with Applications 57 (2009) 1767–1775.

[26] S. M. Jameii, K. Faez, M. Dehghan, "Multiobjective Optimization for Topology and Coverage Control in Wireless Sensor Networks", Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks, Volume 2015, Article ID 363815.

[27] J. Chen, H. Qian, "Node Deployment Algorithm Based on Viscous Fluid Model for Wireless Sensor Networks", Hindawi Publishing Corporation, The Scientific World Journal, Volume 2014, Article ID 350789.

[28] S. S. Dhillon, K. Chakrabarty, "Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks", 2003 IEEE.

[29] F. Nematy, N. Rahmani "Using Voronoi Diagram and Genetic Algorithm to Deploy Nodes in Wireless Sensor Network", The Proceeding of International Conference on Soft

Computing and Software Engineering 2013, The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], San Francisco, CA, U.S.A., March 2013**,** The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3**.**

[30] Mohammed El-Abd, "Generalized Opposition-Based Artificial Bee Colony Algorithm", WCCI 2012 IEEE World Congress on Computational Intelligence June, 10 15, 2012 - Brisbane, Australia