A

Dissertation

On

Base Station Positioning For Maximum Lifetime in Wireless Sensor Network

Submitted in Partial Fulfillment of Requirement

For the Award of the Degree of

Master of Technology

in

Computer Science and Engineering

by

Tushar Saini

University Roll No. 2K13/CSE/25

Under Esteemed Guidance of

Vinod Kumar

Associate Professor, Computer Engineering Department, DTU

2013-2015

Delhi Technological University

Delhi-110042, India

ABSTRACT

Wireless sensor networks consists of large number of resource constrained sensor nodes which performs the functionality of data gathering and transfer their data to the base station for analysis to gain meaningful information from it. In recent years, wireless sensor network has found its applications in varied areas like border surveillance, hazard management, security, weather monitoring etc. With such critical and real time applications it is very important to make wireless sensor network reliable and robust. One of the open problems in wireless sensor network is optimal positioning of the base station

In this dissertation, an approach to solve the base station positioning problem in wireless sensor networks using Artificial Bee Colony (ABC) optimization technique is presented. Along with conventional ABC approach, six modified ABC approaches are presented in this work. The first modification incorporates modified local search mechanism in place of conventional ABC local search. The second modification uses a special phase apart from conventional ABC phases, which selects the worst 10% food sources and replaces them at the end of every cycle. Third and fourth medications use memetic search technique based on global section search. Fifth modification uses a special phase which calculates the opposites of the current food sources based on the opposition based learning technique and find the best out of the lot.

Sixth modification incorporates a modified local search as it was in first modification and a special phase which incorporates opposition based learning technique to find the best solution. All of these modified algorithms are simulated and results are compared with the conventional ABC algorithm, which was outperformed by all six modifications in finding a location for the base station for maximizing lifetime of the network under same constraints.

Keywords: Artificial Bee Colony, network lifetime, wireless sensor network, base station positioning.

ACKNOWLEDGMENT

I would like to express my deep sense of respect and gratitude to my project supervisor Mr. Vinod Kumar for providing the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

I am grateful to Prof. O.P.Verma, HOD, Computer Engineering Department, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Tushar Saini

Roll No. 2K13/CSE/25

M.Tech (Computer Science & Engineering)

Department of Computer Engineering

Delhi Technological University

Delhi-110042

CERTIFICATE

This is to certify that the project report entitled **"Base Station Positioning for maximum lifetime in wireless sensor network"** is a bonafide record of work carried out by **Tushar Saini (2K13/CSE/25)** under my guidance and supervision, during the academic session 2013-2015 in partial fulfillment of the requirement for the degree of Master of Technology in Computer Science & Engineering from Delhi Technological University, Delhi.

Vinod Kumar
Associate professor
(Project Guide)
Department of Computer Engineering
Delhi Technological University
Delhi-110042

Table of Contents

List of Figures

List of Abbreviations

WSN	Wireless Sensor Network
SN	Sensor Node
BS	Base Station
CH	Cluster Head
NP	Non Polynomial
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
GABC	Gbest guided Artificial bee Colony
MSABC	Memetic Search based Artificial Bee Colony
MLSABC	Memetic Local Search based Artificial Bee Colony
OABC	Opposition based Artificial Bee Colony
HABC	Hybrid Artificial Bee Colony
MEC	Minimum Enclosing Circle

CHAPTER 1

INTRODUCTION

Wireless sensor network is a wireless interconnection of sensor units which are generally used to monitor and observe environmental conditions. Wireless sensor network is a subfield of ad-hoc network. Although it is a subfield of ad-hoc network but still we cannot use techniques and mechanisms used for managing ad-hoc network for WSN because of the following reasons:

- Numbers of nodes in a WSN are far more greater than number of nodes in an ad-hoc network.
- Nodes in WSN works in resource constrained environment, whereas in ad-hoc network nodes, there no such constraints.

Because of the above reasons WSN behaves differently than ad-hoc network. The energy constraint on WSN nodes is of utmost importance which is studied intensively to make WSN robust, dependable and usable in real life scenarios. Sensor nodes work with limited amount of energy provided by a small DC source incorporated in the sensor node unit which cannot be replaced after the network is deployed. Along with energy other important constraints of WSN are:

- Limited energy.
- Limited sensing range.
- Limited transmitting range.
- Limited processing capability of sensor node.
- Limited memory capacity of sensor node.

The sensing range of a sensor node represents the range of area around the sensor which can be observed or monitored by it while the transmitting range represents the range of area around the sensor till where the sensor node can send data collected by it. These two constraints are important because sensing and transmitting are the two most expensive operations performed by a sensor node. Limited processing and memory capacity comes into light when the sensor nodes preprocess the data before sending it. By preprocessing, data can be compressed and the amount data to be transmitted can be reduced which in result reduces the energy consumption. We can easily identify the three main operations of a sensor node in WSN as:

- Sensing
- Processing
- Communication

Sensing includes gathering the data about the parameters in question in the sensing range of the sensor. These parameters could anything like temperature, humidity, etc depending on the motive and design of the WSN. Processing includes the mechanism of managing the data gathered in the sensing operation and manipulating it. Communication includes the transmission of data to other nodes in the network to gather some useful information from the data gathered. Communication includes routing of data packets which itself is a typical problem in standard communication network. So, routing in constrained environment is much more difficult problem to face.

1.1 Architecture of WSN

The scalability and performance of a WSN highly depends on architecture of the network. We generally design network according the application area for which we are designing the network. The design chosen depends on the performance measures and to which extent we want to scale the network. The architecture of a WSN can be of multiple types, but the two mail architectures intensively studied are as follows:

- Flat
- Hierarchical or Two tier

1.1.1 Flat Architecture

In flat architecture as shown in figure below, we have multiple sensor nodes (SN) are deployed in the sensing area to gather useful data. Along with multiple sensor nodes we have a single base station (BS) to which every sensor node sends its gathered data to. In flat architecture, if the sensor node is placed far away form base station and not reachable in its transmitting range then sensor node sends its data using multi-hop transmission in which sensor node sends data to another sensor node in its transmission range, and then that intermediate sensor node sends the data received to the base station. A general structure of flat architecture of WSN is shown below in figure.

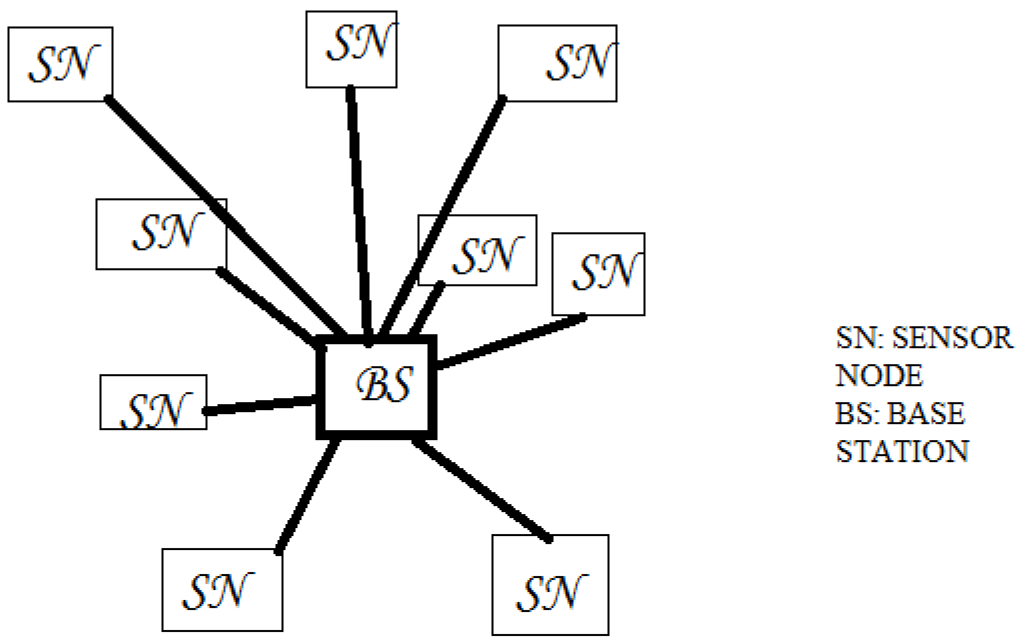


Fig.1.1 Flat Architecture of WSN

1.1.2 Hierarchical Architecture

In hierarchical architecture multiple sensor nodes are deployed as shown in figure below to gather useful data in their sensing range. As shown in figure 2, in hierarchical architecture multiple clusters are created and each cluster have a cluster head which gathers the data from every sensor in its cluster and then sent it to the base station itself. The cluster head selection procedure is typical one in WSN. There are variety of techniques used for selecting cluster head (CH), one of them which is readily used is to select a sensor node within that cluster as cluster head for specific period of time and after that select another node in the network as cluster head for the next time period and relieve the previous one from the duties of cluster head. In this way, the energy consumption of all the sensors in the cluster is managed to make sure that a single designated cluster head does not die (run out of energy) considerably before other nodes in cluster.

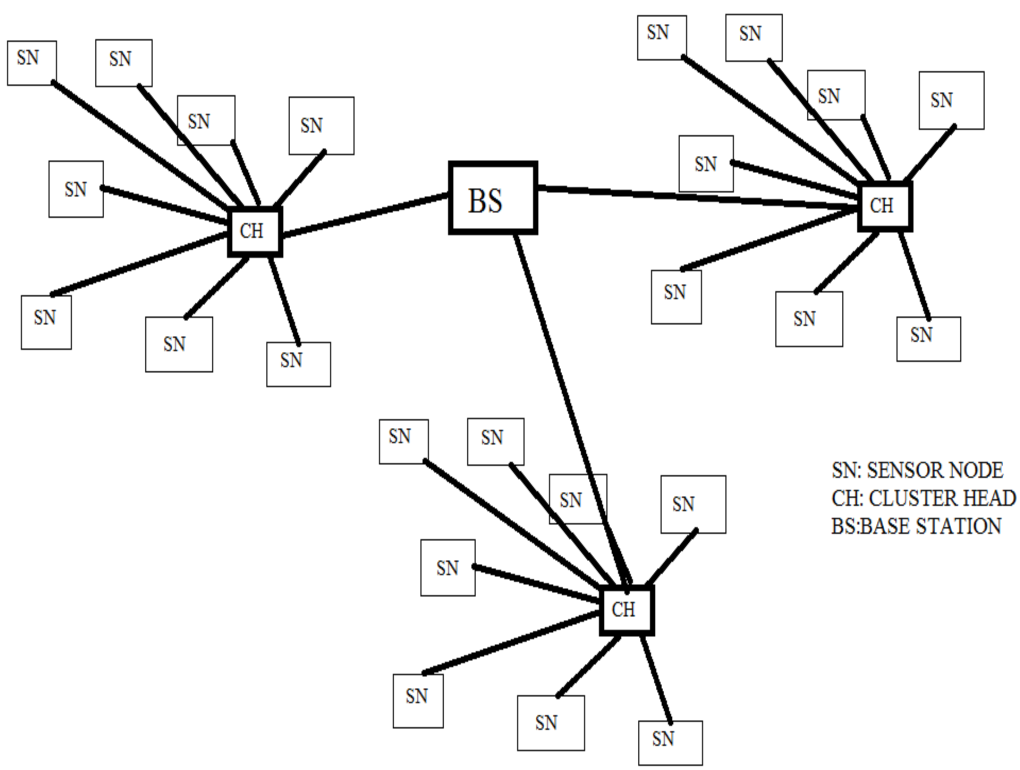


Fig.1.2 Hierarchical Architecture of WSN

In hierarchical architecture too we can have multi-hop transmission. When a cluster head is far away from base station such that the base station is not in cluster head's transmission range then the cluster head can transfer the data gathered by it from all the sensor nodes in its cluster to cluster head of another cluster that can transmit the data received to the base station.

1.2 Components of WSN

The components of WSN enable the wireless connectivity within the network, connecting multiple sensor nodes to the base station and cluster heads. These are the main components of a WSN:

- Sensor Node
- Base Station

1.2.1 Sensor Node

A sensor node consists of few sensors and a processing unit. They are used to gather useful data around them in their sensing range and to send it to the base station. These are the following five main components of a sensor node:

- Controller
- Memory
- Transceiver
- power source
- Sensors

Controller is the brain of the sensor node. It controls all the operations and functioning of a sensor node. Memory unit is used to store the data gathered by the sensors before sending it to the base station or any intermediate node (or relay node). Transceiver is a component which is used for communication purposes. It is used for receiving data from other nodes as well as sending data to other nodes. Power source is the component which provides energy to all the components of sensor node. Power source uses a lithium-ion battery for storing energy. Sensors are the hardware device which senses the area surrounding the sensor for physical parameters like temperature, pressure, humidity, or any other problem specific parameter. These are connected to analog to digital converter so that the analog data gathered by them can be converted to the digital form which can be processed and manipulated by the controller and other components of sensor node.

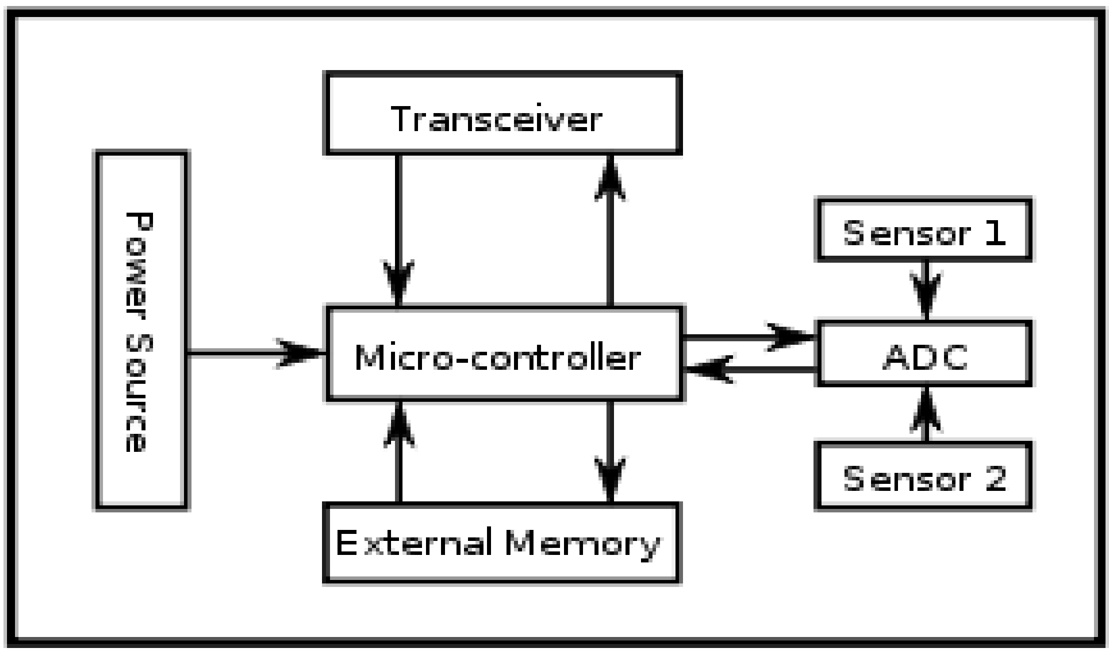


Fig.1.3 Architecture of Sensor Node

As shown above in the figure above the inside structure of a sensor node which consists of all the five components as discussed above. Micro-controller is the brain of the sensor node as we can se all the other four components (sensors via Analog to Digital Converter) are connected to it. Along with the functionality of sensing environment, sensor node can act as a relay node in multi-hop architecture where sensor node just receives the data and forward it to base station.

1.2.2 Base Station

Base station is the component of WSN responsible gathering data from the sensor nodes. It is also called sink node in some notations. Along with gathering data it also posses the functionality of processing the data received an analyzing for capturing meaningful information from the data. On contrary to sensor nodes, base stations do have much better resources in respect of processing power, memory and energy. In some applications in fact base stations are connected to direct power supply rather than being battery powered if the base stations are static (i.e. not mobile). They also perform the functionality of sending the processed data collected by sensor nodes to the remote server in some applications like border surveillance or if the network is established in some remote locations.

Base stations are also important component of WSN in terms of energy utilization and reliability of the network. Since every sensor node in the network sends gathered data to the base station and transmission of data wirelessly is the most energy consuming operation of a node, if we can somehow place the base station in position where energy required by all the nodes for transmission of data is minimum, then we can save a lot of energy and out network can last longer.

Along with energy utilization base stations are also important for WSN because every sensor node sends its data to the base station and if base station is malfunctioning or down then the whole network is down, which is not the case when a sensor node is down since we can still collect data from other live sensor nodes.

1.3 Motivation

Base station positioning problem in wireless sensor network is a very important problem because it affects the functionality of the whole sensor network. With the extensive use of wireless sensor network in real world now in form of different applications it is very important to work on the problems of WSN and make it more usable and reliable. WSN is used in some critical application areas like border surveillance where every aspect of the network is very critical because even a short glitch in network functionality can result in tremendous after effects.

Optimal base station positioning problem is a well known NP-complete problem [1]. Since it cannot be solved deterministically in polynomial time, we try to find the near optimal solution for it using different approximation techniques. The fact that we cannot solve it deterministically in polynomial time, makes us look for approximate solution for this problem since it is a readily occurring problem in WSN.

There are different population based optimization techniques applied on this problem like particle swarm optimization (PSO) [2] and genetic algorithm (GA) [3]. In recent time, a new swarm based optimization technique called the Artificial Bee Colony (ABC) proposed by Dervis Karaboga and Bahriye Akay [4]. They compared ABC on different benchmark functions in their comparative study and showed that for most benchmark functions, ABC performs better as compared to other well known population based techniques like PSO and GA.

1.4 Research Objective

As explained in the previous section of motivation for this work, following are the research objective of this work:

- To find the optimal position of the base station of the sensor network deployed. This is the basic core of our work
- In order to find the optimal location of the base station for the given sensor network, our criteria is the energy consumption by each sensor.
- We measure the energy left with each sensor for a possible location of base station and find the minimum energy left for all sensors which represents the objective function value for that position of base station.

- So, our objective in this work is to maximize the minimum energy left with all sensor.
- We try to optimize our objective function using Artificial Bee Colony (ABC) and try to improve it via different modification over basic ABC.

1.5 Report Organization

We start this dissertation report with the introduction in chapter 1 describing wireless sensor network (WSN). In chapter 2, a detailed background of the base station positioning problem along with literature review is given. Chapter 3 explains about the proposed algorithms for finding optimized solution. Chapter 3 explains different modifications used to optimize the solution. Chapter 4 consists of performance evaluation and results of the proposed algorithms. We conclude about the word done and discuss the future prospect of work in this field in chapter 5.

CHAPTER 2

BACKGROUND AND RELATED WORK

In recent times, with increased use of wireless sensor network (WSN) in varied real life application areas, a lot of research is being done to optimize WSN and its various functionalities to make more dependable. By dependability we mean the responsiveness, lifetime, availability, security, etc. Since WSN finds its major application area as border surveillance, disaster management, and security surveillance which are highly critical applications areas, it is very important to make our WSN more dependable since no human assistance will be present in those areas. To improve the overall performance of WSN various researchers have worked on various functional and non functional requirements. A lot of work is done on different layers of communication protocol for WSN, routing of data packets, data aggregation, collision avoidance, data authentication and encryption, load balancing, query optimization, etc. with minimization of energy utilization as prime concern.

Along with above areas, a lot of work is also done to find the optimal position of base station in a WSN. The motivation for this research area is if we can place the base station at a optimal position with respect to all the sensor nodes in the network, then all the communication between sensor nodes and base station will in result be optimized. And since communication is the main energy consuming activity in sensor networks, placing the base station at best possible location will decrease the energy consumption which will result in increased network lifetime and dependability.

Base station positioning problem can be classified according to environment of WSN as follows [5]:

- Static base station positioning
- Dynamic base station positioning

2.1 Static Base Station Positioning

Considerable amount of work is done in the area of static base station positioning in recent times. Work of different researchers differs because of the assumptions they make for their network model, energy model, performance matrices, etc. We can classify static base station positioning problem according to the number of base stations to be positioned as follows:

- Single base station positioning
- Multiple base station positioning

2.1.1 Single Base Station Positioning

In this variant of static base station positioning problem our objective is to place a single base station for the whole sensor network. Approach to solve this problem differs according to the network structure assumed. If network architecture is flat, and sensor nodes are homogeneous i.e all sensor nodes have same amount of initial energy and other parameters as well, then it becomes a multi-hop data transfer problem which is a typical problem because in this, some nodes will act as relay node (node which receives data and transfers it further without sensing data in its own range) also.

The complexity of problem can be reduced if we assume that each sensor node can transmit data directly to the base station in a single-hop transfer mode. Since the base station problem is NP-complete, various approximation techniques are given by researchers. Along with approximation techniques, some computational geometry solutions are also given [6]. In this technique, we try to find the smallest circle which encloses all the sensor nodes of the network and then place the base station at the centre of that circle. The above technique is shown in the figure below.

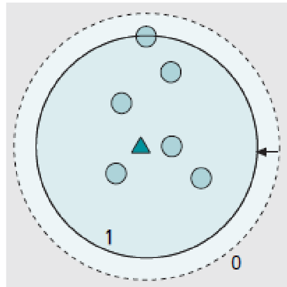


Fig.2.1 Base station positioning using computational geometry

As shown in the figure above, initially the dotted circle (0) is the solution containing all the sensor nodes represented by small circles, and after processing we were able to find a smaller circle (1) for which we place the base station at the centre shown as triangle.

2.1.2 Multiple Base Station Positioning

Multi base station positioning problem is much more difficult than single base station positioning problem. The problem can be defined as finding the optimal number of base stations to be placed along with their optimized location. If we somehow know the number of base station to be placed then its complexity reduces and the problem converges to single base station positioning problem.

Multiple base station positioning problem can be approached in one of the following ways:

- Approximation algorithms
- Integer programming

To solve this problem using approximation algorithms for increased data rate and network lifetime following techniques are used:

- Greedy algorithm
- Local search algorithm

In the greedy algorithm, the base station position is restricted to the location of sensors, and then the base stations are individually placed in an arbitrary order for increased data rate and network lifetime. In local search strategy, we start with some random solutions and then perform local search around those solutions to find the optimal solution.

In integer linear programming, the data routing load among the sensors is evenly distributed to minimize the maximum energy consumption individual sensors for minimizing the total communication cost.

2.2 Dynamic Base Station Positioning

All the techniques described in previous section considered only static network i.e. the sensor nodes does not move after the deployment of the network. So, before deploying the network we can calculate the optimal location for the base station and then place the base station along with the sensor nodes for which the optimal location is found out using the static base station positioning techniques.

But if we use the same static base station positioning techniques for dynamic network, where sensor nodes can move in the environment where they are placed, then in this scenario, when the sensor node will move, the optimal location of base station might change and the previous optimal location of base station might become obsolete. Therefore, dynamically repositioning the base station after particular amount of period according to the movement of the sensor node of the network can increase the performance of the sensor network.

Repositioning of base station according to network changes is very difficult to incorporate because for this we have continuously sense the network and its changes. But if we do incorporate dynamic repositioning of base station according to the network, then we can defiantly increase the following parameters of the network:

- Network lifetime
- Data delivery delay

If we do not reposition the base station then the network lifetime might decrease because the nodes which will be closed to base station will die faster as compared to nodes which are further away form the base station. Along with network lifetime, data delivery delay will also get hit due to the fact that data from nodes far away will take more time to get transmitted to the base station as compared to the nodes which are closer to base station.

Repositioning of base station is also required because of one more reason. When a network is established in a hostile environment like disaster management, if the base station is somehow gotten closer to the sensor nodes which are gathering information like fire, flood, or any other natural disaster, then the base station might get caught itself in the natural disaster.

2.3 Literature Review

In recent times, a lot of researchers have worked on this field of base station positioning problem because of the growing importance of wireless sensor networks. To make wireless sensor network more dependable and secure, researchers are working on lot of different aspects of a wireless sensor network. One important area of wireless sensor network that has come up in recent times is base station positioning. Researchers have tried varied techniques to solve different versions of this problem under different assumptions like population based algorithms, combinatorial geometric based solutions, neural network based solutions, etc. Rest of this section will include a brief description of various techniques used for base station positioning.

Authors Ivan Vilovic, Niksa Burum, and Zvonimir Sipus in [7] proposed a technique for solving base station positioning problem using a population based optimization technique called Ant Colony Optimization (ACO). ACO was proposed by Marco Dorigo and Gianni di Caro in [8], where they explained the functioning of ACO which is inherently used for discrete optimization problems. To make it applicable to continuous base station positioning problem, authors in [7] proposed to make pheromone matrix comprising of all the initial location of ants and their corresponding pheromone value. Ants then choose the path to be traversed by selecting a path using the pheromone probability equations of the ACO. This technique was compared by authors with PSO and GA to which the results showed that ACO takes more cycles as compared to others to converge to optimal solution.

Authors Yunyue Lin, QishiWu, Xiaoshan Cai, Xiaojiang Du, and Ki-Hyeon Kwon in [9] discussed strategies to place base station in two variants of the problems based on communication i.e. single-hop and multi-hop. For single-hop, authors proposed minimum enclosing circle (MEC) in which, first we randomly choose k sensor nodes as base stations out of total n. after selecting base stations, we cluster rest of the n-k nodes in clusters closest to them. After making clusters, we reposition the base station of that cluster to the centre of the MEC of that cluster. We continue this process till there are changes in previous step. For multi-hop, first select the base station position randomly as it was case in the previous single-hop technique. After choosing the base stations, we cluster rest of the sensor nodes to cluster by finding minimum energy path for every sensor node and then placing that node to the cluster where energy consumption is minimum.

Authors Andrej Bogdanov Elitza Maneva, Samantha Riesenfeld in [10] considerd the base station positioning problem with main concern on power consumption of the sensor network. They proposed two techniques for solving the base station positioning problem i.e. greedy algorithm and local search algorithm. The greedy algorithm is a deterministic algorithm which picks up base station locations one by one while keeping the previous base stations fixed to improve the rate of network. In local search algorithm, a set of initial solutions is randomly selected and then using local search technique, better solution in near vicinity of selected solution is searched. The number of iterations that the local search algorithm runs is predefined and result is published after those number of iterations are completed. Authors published that the greedy algorithm performs better as compared to local search algorithm by finding a solution closer to optimal solution.

Authors Tzung-Pei Hong, Guo-Neng Shiu in [2] proposed a technique for positioning multiple base stations in two tier wireless sensor network architecture using Particle Swarm Optimization (PSO). Constraint used by authors for positioning multiple base stations is power consumption. Authors in this paper assumed a heterogeneous two tier network in which each sensor node may own different initial energy and data rate. PSO is a population based optimization technique proposed by James Kennedy and Russell Eberhart in [11] for continuous optimization problems. Authors used PSO for solving base station positioning problem by selecting randomly a set of particles which represents the possible solution. In every iteration of the algorithm, position of particles is updated using the velocity associated with every particle. At the end of every cycle, velocity of every particle is updated using PSO equations and a global best solution is remembered.
After completion of all iterations, this global best solution represents out solution. In result, the PSO technique works much faster and provides much better solution than the exhaustive search technique.

Authors P.D. Hossein Zadeh, C. Schlegel, M.H. MacGregor in [12] proposed a distributed technique for finding optimal location of the base station in dynamic wireless sensor network. In this work, authors provided power aware and network traffic aware positioning scheme for positioning of base station. Two techniques given in this paper are proposed based on weighted linear least squares and weighted non-linear least squares depending on the path loss exponent value which is a major characteristic in wireless communication.

Authors Rajiv Kr. Tripathi, Sateeshkrishna Dhuli2, Y. N. Singh, Nishchal K. Verma in [13] analyzed the effect of weighing factor on weighted centroid optimal base station positioning algorithm. The authors concluded that varying th weighing factor (alpha) do varies the energy consumption in wireless sensor networks,

In [14] Dorottya Vass , Attila Vidács proposed a technique for placing base station in a static wireless sensor network. In this work, base station is assumed to be dynamic i.e. movable and rest of the sensor nodes are static. So when one of the sensor node decides to transmit data, it can then send a request message to base station which move in the direction of the sensor node to an optimal position to gather data form the sensor node. They introduced three different mechanisms to optimize base station location. First, minimizing the average energy consumption. Second, minimizing the maximum energy consumption and third, minimizing the relative energy consumption.

In [15] Yong Seouk Choi, Kyung Soo Kim, Nam Kim proposed a evolution based optimization technique solution for solving base station positioning problem. The optimization technique used by authors to solve base station positioning is Genetic Algorithm (GA). GA was proposed by J.H.Holland and K.A.de Jong. In this work, author describe the representation of solution as location of base station along with number of base station to be positioned. Roulette wheel method is used for selection operation and the fitness function is multi-objective with coverage and power both as prime concern for optimization in this work.

In [16] Soo Kim, Jeong-Gil Ko, Jongwon Yoon, Heejo Lee presented a multi objective approach for solving base station positioning problem. In this, authors devised four different metrics as follows:

- Coverage of sensor nodes
- Fault tolerance of sensor network
- Energy consumption
- Congestion in network

Coverage of sensor nodes means number of nodes that can send data to the base station via single-hop or multi-hop. Fault tolerance means average number of nodes live after the failure of base station. Energy consumption converges to the distance between sensor nodes and the nearest base station. Congestion in network means average delay in network due to the traffic congestion. The authors used greedy search technique to implement these metrics.

In [17] Shashidhar Rao Gandham, Milind Dawande, Ravi Prakash, S. Venkatesan, proposed a model in which lifetime is spitted into equal periods of time and ebergy consumption is evaluated after every round. The authors used integer linear programming to solve find the base station position.

In [18] Sabbir Mahmud, Hui Wu, Jingling Xue solved the problem of placing multiple base station in wireless sensor network. The technique used by then includes dividing all the sensor nodes in k clusters with each cluster having a cluster head. Now, the problem is reduced to placing a base station optimally between these k cluster heads which will communicate with the base station.

In [19] M.Amac GUVENSAN, Z.Cihan TAYSI, A.Gokhan YAVUZ devised a new algorithm for placing multiple base stations in wireless sensor called k-means local+. Authors provided results in which they showed 45% better network lifetime than standard k-means algorithm. In k-means local+ algorithm first, the location of base station is found using standard k-means algorithm and then the base station is moved closer to their one-hop neighbor nodes with greater traffic load. Both of these algorithms i.e. k-means local+ and standard k-means are routing dependent. By routing dependent we mean, they evaluate the routing path also along with the position of the base station.

In [20] Anurag Singh Tomar, Gaurav Kumar Tak proposed a unique solution to black hole attack on a wireless sensor network using base station positioning. A wireless sensor network is under black hole attack when an intruder tries to capture some of the sensor nodes reprogram them to not send sensed data back to base station and even collect different type of data than it was previously programmed initially at the time of its deployment in the wireless sensor network. The solution proposed by author for black hole attack is to position multiple base station in the wireless sensor network using genetic algorithms (GA). In this work author published the results of black hole attack protection on the basis of data delivery rate of the proposed technique increased significantly but by positioning multiple base stations, transmission of same data multiple times increase the energy consumption of the network significantly.

In [21] Yi Shi and Y. Thomas Hou presented two approximation algorithms for solving base station positioning problem. Authors considered flat sensor network architecture with possibility of multi-hop communication. In first algorithm, authors converted their continuous cost function to discrete form and then the continuous search space to finite number of sub areas. In second technique, authors exploited the cost property of each sub area and used this property to find the optimal position of the base station.

CHAPTER 3

PROPOSED WORK

3.1 Problem Statement

Let us consider ‘n’ sensor nodes in a sensor network where E(i) and D(i) represents the initial energy and data rate for a sensor node, where i= 1,2,3…n. our objective is to find a location of the base station for which the ‘lifetime’ of the sensor network is maximized.

Lifetime of a wireless sensor network represents amount of time the network is alive after its initial deployment i.e. all the sensor nodes of the network are performing all the required operations successfully. The moment any one of the ‘n’ sensor nodes of the network dies, network is considered as non-functional.

To calculate the lifetime of wireless sensor network, we calculate the lifetime of each individual sensor for a particular location of base station and then find minimum of all of them, which will tell us the time when the first sensor node of the network will die if we place the base station at that particular location.

So our objective converges to maximizing the lifetime of the network which in turn is the minimum lifetime of all the sensor nodes in the network. Assumptions taken into account for the following work are as follows:

- Heterogeneous system i.e. initial energy and data rate of sensors may vary from node to node.
- Flat network architecture i.e. all the sensor nodes in the network are at the same level in network hierarchy.
- Single-hop communication i.e. every sensor node can transmit its data to the base station in single hop.

- Transmission range i.e. irrespective of the location of sensor node(i) and base station, the base station is always in transmission range of sensor node(i) for i= 1,2,3….n.

3.2 Energy Model

The energy model used for this work is represented in the equations below which was adopted by [22] to calculate the energy consumption per unit time.

$$p(r,d) = r(\alpha_1 + \alpha_2 d^n) \qquad (1)$$

where, r represents data rate transmission rate

d represents Euclidean distance

α_1 represents distance independent parameter

α_2 represents distance dependent parameter

n represents path loss exponent

p(r,d) represents the energy consumed per unit time for node with 'r' data transmission rate and 'd' Euclidean distance between sender and receiver. [23] Studied that for same transmission rate i.e. homogeneous system, our problem boils down to finding the minimum enclosing circle containing all the sensor node of the wireless sensor network.

We are considering heterogeneous system, therefore, initial energy and data rate for i^{th} sensor node is represented as:

- $e_i(0)$ represents initial energy of i^{th} sensor node.
- r_i represents data transmission rate of i^{th} sensor node.

The lifetime of i^{th} sensor node for the j^{th} location of the base station is calculated as follows [2]:

$$lifetime_{ij} = e_i(0) / r_i (\alpha_{i1} + \alpha_{i2} d_{ij}^{\,n}) \qquad (2)$$

In order to find out the lifetime of wireless sensor network we have to find the minimum value of lifetime of all sensor nodes as follows:

$$lifetime_{wsn} = Min \{ lifetime_{ij} \} \quad (3)$$

Where i represents i^{th} sensor node for i = 1,2,3….m.

As discussed in previous sections of this report, our objective function can be represented as follows:

$$Fitness = Max \{ lifetime_{wsn} \} \quad (4)$$

3.3 ABC Approach

As discussed in the chapter 2 of this report, base station positioning problem is NP-complete irrespective of whether it is a single base station problem or multiple base station problem. But with the emerging importance of wireless sensor networks in real world applications, we somehow have to solve this problem to a satisfactory level. To achieve this, we propose Artificial Bee Colony (ABC) approach. ABC was proposed by Dervis Karaboga and Bahriye Akay [4]. ABC is a population based optimization technique inherently used for optimizing continues functions. As our objective function is also continues, this technique is a perfect fit for our problem.

In [4] authors also described the applicability of ABC by comparing it with other readily optimization techniques like GA and PSO over standard mathematical benchmark functions. The comparison showed that for most of the benchmark functions ABC performs substantially better than GA and PSO. Although this comparison does not proves that ABC is better than PSO or GA for every problem, but it is a very good metric to start with.

ABC is an optimization technique based on the natural behavior of bees in the process of gathering nectar from food sources. According to the behavior of bees, they are classified as follows:

- employee bee
- onlooker bee
- scout bee

Each bee performs a specific task in nature. In ABC algorithm, we imitate all these tasks in a computerized way. Employee starts the process of collecting nectars from different food sources. They bring back the nectars to the hive and exchange information about the food source they visited by a particular dance called waggle dance. Physical interpretation of waggle dance is, more a bee dances higher the quality of the food source visited by it. After seeing the waggle dance, onlooker bee decides the food source they are going to visit and this process continues until a food source is dead. When a food source is dead, it is abandoned, and scout bee starts searching for a new food source to replace the abandoned one.

To imitate the above functionality in a computer program, we divide it in three phases namely employee bee phase, onlooker bee phase, and scout bee phase. In employee bee phase, each employee bee goes to the different food sources and perform a local search around the selected food source and evaluate the objective function value of the nearby solution. If the nearby solution is better than the previous one then we replace the previous solution by the new solution and store the objective function value of the new food source and also initialize the trials count of that food source to 0. But if the previous solution is better than the new solution than we increment the trials count of that food source by 1.

On the basis of the objective function values, onlooker bee finds probability of selection of each food source and uses it to select food source to visit. The onlooker bee also performs local search around the selected food source and tries to find a better solution. If it is successful in finding a better food source around previous one, then it replaces it and set the number of trials of that food source to 0, otherwise increment the trials of that food source by 1.

After onlooker bee, in scout bee phase, we check the trials count of every food source and find the food source with maximum trials value. If the maximum trial value found is greater than the predefined ‘limit’ then, scout bee abandons that food source and replace it with new one and initialize its trials count to 0.

After the scout bee phase, we find out the best food source among all according to the objective function value i.e. fitness value. We repeat about process ‘maxcycle’ times and returns the best food source found after maxcycle times. The above process represented in pseudo code form as follows:

```
Initialization();
find_best_food_source();
While(maxcycle)
{
        employee_bee();
        onlooker_bee();
        scout_bee();
        find_best_food_source();
        maxcycle--;
}
return best_food_source;
```

As shown in the pseudo code above, before employee bee phase for the first time we initialize the food sources. The number of food sources to be employed is fixed by the authors as follows:

$$\text{Colony size} = \text{Number of employee bees} + \text{Number of onlooker bees} \qquad (5)$$

And along with this, the number of scout bees is fixed as 1 by author. Although number of scout bees can be changed according to the problem domain.

The local search performed by employee bee and onlooker bees follows the following equation:

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \qquad (6)$$

where, x_{ij} represents the jth parameter of ith food source

x_{kj} represents the jth parameter of kth food source

ϕ_{ij} represents a random number in range [-1,1]

v_{ij} represents jth parameter of the newly formulated food source

the parameter 'limit' of ABC is a very important factor. The value of limit defines the exploration and exploitation capabilities of ABC. If we keep a high value of limit, then more exploitation will happen and algorithm might get caught up local optimal solution. If we keep limit value low, then more exploration will happen and algorithm might not be able to exploit the food sources completely. So, in order to find a optimal value of limit, we use the following formula:

$$\text{limit} = (\text{ colony size } / 2) * D \qquad (7)$$

where, D is the number of parameters in the problem.

Now, in our problem to initialize the food sources, we are randomly generating food sources using a random number generator in the specified range. The number of parameters in our problem of base station positioning problem is 2 since we are working in a 2-dimensional area only two coordinates of the location of base station is our parameters. After generating initial food sources and employee bee phase, onlooker bees calculate the probability of selection of food source by the following formulae:

$$probability_i = ((0.9 * fitness_i) / best\_fitness) + 0.1 \tag{8}$$

where, best_fitness represent fitness of the best solution.

3.4 Gbest Guided ABC Approach

It is well known that, exploration and exploitation are the two main features of a population based optimization technique. For best performance, both exploitation and exploration should be balanced perfectly. The standard ABC algorithm is good in exploration but poor in exploitation. To improve the exploitation feature of our basic ABC algorithm, [23] introduced a PSO inspired local search mechanism on standard benchmark functions. Authors Guopu Zhu, Sam Kwong observed that the new modified ABC algorithm with PSO inspired exploitation mechanisms performed better than the basic version of ABC.

The local search equation of ABC algorithm is shown in equation (6) in pervious section of this report. As shown in the equation, to generate a new solution, we randomly select a food source and subtract it from the previous food source and multiply it with a number in range of [-1,1] generated randomly. Hence, the newly generated solution is has a good exploration feature but poor in exploitation since the probability of selecting a food source which is good is equal to the probability of selecting a food source which is poor. To improve the local search, we need to exploit the currently best solution more by incorporating it in local search equation. To improve the exploitation mechanism of ABC algorithm, we attach another component to this local search equation as shown below:

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) + \Psi_{ij} (y_j - x_{ij}) \tag{9}$$

In the above equation, Ψ_{ij} is a random number which in the range [0,C]

y_j is the j^{th} parameter of the global best solution.

For Ψ_{ij} , a random number in the range of [0,C], C is non-negative constant. A standard value for C found experimentally is 1.5. What the new third component of our local search equation does is improve the exploitation of the algorithm by also considering the current global best solution in procedure of finding the new solution so that we can exploit the current best solution.

The y_j parameter of the local search equation is called the gbest parameter. We incorporate this modified local search equation in place of previous basic ABC local search equation in employee bee phase and onlooker bee phase. To employ this mechanism, we need to know the global best parameters before both employee bee and onlooker bee phases. The above process represented in pseudo code form as follows:

```
Initialization();
find_best_food_source();
While(maxcycle)
{
        employee_bee();
        find_best_food_source();
        onlooker_bee();
        scout_bee();
        find_best_food_source();
        maxcycle--;
}
return best_food_source;
```

As discussed earlier, we need to know the best solution before both employee bee phase and onlooker bee phase. So to incorporate this change, we added another call to function finding best food source so far before onlooker phase also as shown above. Along with this, both employee bee and onlooker bee used the modified local search equation (9) for generating new solutions.

3.5 JA-ABC Approach

The basic ABC algorithm suffers from premature convergence and slow convergence problem. By premature, we mean that algorithm do get trapped in local optimal readily. To rectify this problem of ABC, a lot of work has been done. In [24], authors Noorazliza Sulaiman, Junita Mohamad-Saleh, Abdul Ghani Abro proosed a technique called JA-ABC to rectify the premature convergence and slow convergence problem of ABC.

In JA-ABC , instead of trying to find a better global best solution by improving the best solution, we try to increase the objective function values of poor solutions, in order to achieve a population of food sources in which all the food sources are good. To achieve this, after every cycle of ABC operations, we replace the worst 10% food sources of the population by the following equation:

$$Z_{ij} = Y_j + \phi_{ij} (X_{pj} - X_{kj}) \qquad (10)$$

Where, Y_j is the j^{th} parameter of global best solution.

ϕ_{ij} is a random number in range of [-1,1]

X_{pj} and X_{kj} are j^{th} parameters of the p^{th} and k^{th} food sources respectively, which are chosen randomly from all the food sources. Another constraint as far as selection of p and k is concerned is, that p, k and i should be not be equal to each other. Here, i represent the i^{th} food source which is chosen to be replaced by equation (10).

The concept behind JA-ABC is, if we mutate poor solution with better solutions around the global best solution, then our total population of food sources will consists of only good food sources. And when we randomly select a food source in the employee bee phase and onlooker bee phase for local search, the randomly selected food source will be a good one only, resulting in a better local search. The replacement of poor solutions by solutions generated by equation (10) is done after the onlooker bee phase and before the scout bee phase in every cycle of the basic ABC algorithm.

3.6 Memetic Search Based ABC Approach

To optimize the search results of ABC algorithm, we need to maintain a fine balance between exploration and exploitation features of ABC algorithm. To manage this, authors Jagdish Chand Bansal, Harish Sharma, K. V. Arya, Atulya Nagar in [25] introduced a approach in which we include another local search phase in ABC algorithm in the form of memetic search phase. This modification was tested on standard benchmark functions and on comparison with other standard techniques, results showed that the modified approach performed better.

We incorporated this approach in our base station positioning problem by including a memetic search phase after the scout bee phase in the basic ABC algorithm. Via this local search mechanism, we generate new solutions and compare them with the current best solution that we know. If the new solution is found better than the current best, only then it replaced the current best solution, otherwise ignored. The pseudo code for memetic search based ABC approach is as follows:

```
Initialization();
find_best_food_source();
While(maxcycle)
{
        employee_bee();
        find_best_food_source();
```

```
        onlooker_bee();
        scout_bee();
        find_best_food_source();
        memetic_search_phase();
        find_best_food_source();
        maxcycle--;
}
return best_food_source;
```

The memetic search phase in called after the scout bee phase as shown in pseudo code above. The memetic search phase uses golden selection search (GSS) technique to find the step size. The mechanism of GSS is explained in the pseudo code below for a minimization function f(x) in the range of [a,b].

```
while(termination condition)
{
        F1 = b – ( b - a ) * Ψ;
        F2 = a + ( b – a ) * Ψ;
        If( f(F1) < f(F2) )
                b = F2;
        Else
                a = F1;
}
```

Here, Ψ is the golden ratio i.e. 0.618.

$f(F_1)$ is the objective function value with F_1 as the step size.

$f(F_2)$ is the objective function value with F_2 as the step size.

The above GSS mechanism reduced the search space for step size by updating a or b according to the objective function values generated by them. The memetic search phase (MSP) uses the above GSS mechanism as explained in the pseudo code below for minimization of function f(x) in range [a,b].

```
while( |a-b| < ε )
{
    F1 = b – ( b - a ) * Ψ;
    F2 = a + ( b – a ) * Ψ;
    Generate two new solutions Xnew1 and Xnew2 using F1 and F2 respectively.
    If( f(Xnew1) < f(Xnew2) )
    {
        b = F2;
        if (f(Xnew1) < f(Xbest) )
            Xbest = Xnew1;
    }
    else
        a = F1;
        if (f(Xnew2) < f(Xbest) )
            Xbest = Xnew2;
}
```

Here, ε determines the termination of the local search. To generate new solutions X_{new1} and X_{new2} following mechanism is used for a particular value of F.

```
for (i = 1 to D)
{
    if ( random_number[0,1] > pr )
        Xnew = Xbest + F * ( Xbest - Xkj );
    Else
        Xnew = Xbest;
}
return Xnew
```

Here, pr is the perturbation rate determining the probability of modification of the i^{th} parameter of the new solution and X_{kj} represents the j^{th} parameter of the randomly selected k^{th} food source. Along with above mechanisms used in standalone memetic search phase (MSP), the local search equation in employee bee phase and onlooker bee phase is also modified. Instead of the basic equation of local equation, this approach uses the gbest guided approach as shown in equation (9) in the previous section.

3.7 Memetic Local Search Based ABC Approach

In the previous section, we provided a approach in which a standalone local search phase based on memetic search was incorporated. To counter the large step size problem (i.e. due to the large step size we might skip the global optimal solution), authors Sandeep Kumar, Vivek Kumar Sharma, Rajani Kumari in [26] proposed a technique in which we incorporate the memetic search technique in the local search phases of the basic ABC algorithm i.e. employee bee phase and onlooker bee phase.

To implement memetic search in local search phases of ABC i.e. employee bee phase and onlooker bee phase, we have to run the following memetic search phase and generate F_1 and F_2. For both F_1 and F_2 we generate two new solutions and we also generate a new solution using the standard ABC algorithm local search equation. After generating all three solutions, we perform the greedy selection between all three newly generated solutions and the previous food source which is under contention and select the best one.

Both the local search phases i.e. employee bee phase and onlooker bee phase used the following approach to perform local search in improved modified memetic search based ABC approach.

Generate F_1 and F_2 as follows:

while(termination condition)

{

$F_1 = b – (b - a) * \Psi;$

```
        F2 = a + ( b – a ) * Ψ;
        If( f(F1) < f(F2) )
                b = F2;
        Else
                a = F1;
}
```

To generate the objective function values $f(F_1)$ and $f(F_2)$, we uses the following procedure.

```
for (i = 1 to D)
{
        if ( random_number[0,1] > pr )
                Xnew = Xbest + F * ( Xbest - Xkj );
        Else
                Xnew = Xbest;
}
return Xnew
```

Here, X_{kj} represents the jth parameter of the randomly selected kth food source. The final values of F_1 and F_2 are remembered for which the termination condition failed. These two values are used as step size for generating two new solution using the basic local search equation of ABC as shown in equation (6) as follows.

$X_{new1} = X_{ij} + F_1 (X_{ij} - X_{kj})$

$X_{new2} = X_{ij} + F_2 (X_{ij} - X_{kj})$

$X_{new3} = X_{ij} + \phi_{ij} (X_{ij} - X_{kj})$

After generating all three new solutions, we perform the greedy selection between all three of them and the ith food source which is selected by employee bee in employee bee phase or onlooker bee in onlooker bee phase. This mechanism is repeated for every employee bee and onlooker bee in the employee bee phase and onlooker bee phase respectively.

Authors also suggested that instead of using $\Psi = 0.618$ (i.e. golden ratio) as fixed, we should randomly generate value of Ψ in the range of [0.55,0.65] to make sure that the global optimal solution is not skipped by local search mechanisms by keeping the Ψ fixed.

3.8 Opposition Based ABC Approach

Opposition based learning (OBL) is an optimization technique in which we try to find the optimal solution to our problem by estimating the opposite solutions simultaneously with random solutions. OBL was initially proposed in [27] by Tizhoosh. OBL improves the exploration feature of the basic ABC algorithm by considering random as well as their opposite fodd sources for computation. The basic idea of opposition based leaning is represented by following equation.

Let x belongs to [a,b] where a and b are real numbers. Then opposite of x represented by x' is calculated as:

$$x' = a + b - x \tag{11}$$

Here, a and b are the lower and upper bound of our search space. We implemented OBL in three different forms in ABC algorithm. Firstly, in the initialization phase, we generate 'n' random food sources. With these food sources, we generate their opposite food sources also and calculate the objective function value for all of the 2n solutions consisting of 'n' random and 'n' opposite solutions. From these '2n' solutions, we find out the best 'n' food sources and start our ABC algorithm with these 'n' food sources.

After the initialization step, the employee bee and onlooker bee phase works similarly as it was in basic ABC algorithm. The second use of OBL in our modified ABC comes in scout phase, where instead of generating a random new food source for the abandoned food source, we generate the opposite food source of the abandoned one.

After the scout bee phase, we implemented a special OBL phase in which we generate the opposites of all the food sources at that point of time so that we have now ‘2n’ food sources. Now from these ‘2n’ food sources we select the best ‘n’ on the base of objective function value and continue our search with these ‘n’ food sources. This special OBL phase is constrained to a factor called jumping rate. In every cycle of ABC algorithm, we generate a random number and compare it with jumping rate. If the generated random number is less than the jumping rate only then we perform the special OBL phase otherwise not.

The pseudo code representing the above described mechanism is shown below.

```
Initialization_with_OBL();
find_best_food_source();
While(maxcycle)
{
        employee_bee();
        find_best_food_source();
        onlooker_bee();
        scout_bee_OBL();
        if( random_number[0,1] < jumping rate )
                OBL_phase();
        find_best_food_source();
        maxcycle--;
}
return best_food_source;
```

As shown in the pseudo code above, the OBL phase is executed only when the random number generated is less than jumping rate. The physical interpretation of jumping rate is, OBL phase will be executed ‘jumping rate’ times. This technique is introduced by

Mohammed El-Abd in [28]. In the OBL phase, when we generate opposite food sources of the current food sources, we assign them a trial value of 0. This mechanism as shown by authors works better on benchmark functions as compared to other standard optimization techniques.

There is another variant of OBL i.e. Generalized OBL (GOBL) in which the equation used to generate the opposite of x belonging to [a,b] is:

$$x' = (a + b) * r - x \tag{12}$$

Here, r is a random number generated in the range of [0,1]. Although this variant Performs better on benchmark functions but when applied on base station positioning problem it performs poorly as compared to simple OBL using equation. To implement the OBL phase, we sort all the ‘n’ food sources and their ‘n’ opposites and then select the top ‘n’ food sources to continue further in the ABC algorithm.

3.9 Hybrid ABC Approach

All the approaches that we have provided prior to this focused on only one of the two main features of ABC algorithm i.e. either exploration or exploitation. In this section, we propose a new algorithm based on ABC namely HABC (Hybrid ABC) which focuses on both the features of ABC i.e. exploration and exploitation which makes it more robust than basic ABC.

For the exploitation feature, we have decide to include gbest guided local search equation (9) for performing local search in the employee bee phase and the onlooker bee phase. This gbest guided equation is showed to be performing better than the standard ABC local search equation (6) on benchmark functions. The intuition to incorporate this equation is to exploit the current best food sources more and more so that all the food sources move towards the best food source.

For the exploration feature, we have decided to include opposition based learning technique in our algorithm. What opposition based learning does is, as explained in section 3.7, it explores more options of food sources in the form of evaluating the opposite location of our current food sources which makes our algorithm explore more options rather than just sticking to the our own food sources or just relying on the exploitation feature to find the optimal solution to our problem. This HABC algorithm is not applied anywhere before and it is being applied for the first time in this work for finding the optimal position of the base station. The pseudo code for our proposed HABC algorithm is shown below.

```
Initialization_with_OBL();
find_best_food_source();
While(maxcycle)
{
        employee_bee_with_equation(9)();
        find_best_food_source();
        onlooker_bee_with_equation(9)();
        scout_bee_OBL();
        if( random_number[0,1] < jumping rate )
                OBL_phase();
        find_best_food_source();
        maxcycle--;
}
return best_food_source;
```

CHAPTER 4

RESULTS AND ANALYSIS

4.1 Experimental Setup

In this chapter, we provide the performance evaluation of our proposed work to compare it with the basic ABC algorithm. The proposed OABC (Opposition based ABC), GABC(gbest guided ABC), MSABC(memetic search ABC), MLSABC(memetic local search ABC), JA-ABC and HABC(Hybrid ABC) are implemented in C language using CodeBlocks IDE. These implementations are run on a system with Intel core2 duo 2.53 GHz processor, 4gb RAM, and windows 7 as the operating system.

This chapter consists of 6 sub-sections; in which each proposed algorithm is compared with the basic ABC algorithm to analyze is performance. Each sub-section consists of performace evaluation of a proposed algorithm with reaspect to basic ABC algorithm on a parameter called MaxCycle.

All the algorithms are run 30 times, with different MaxCycle values, and the mean of all the 30 runs is taken to finally conclude the performance of that particular algorithm. To find the base station of a wireless sensor network, we need a sensor deployment and to generate that, we used a random function to initially generate the positions of the sensors for which we have to find the base station. The static deployment of the sensor nodes is shown in the figure below.

Every dot in the figure shown below represents a sensor node. The solution space taken for experimental result is [0,100] for both the coordinates i.e. x and y coordinate and there are total of 50 sensors. All the algorithms are compared on this static deployment only.

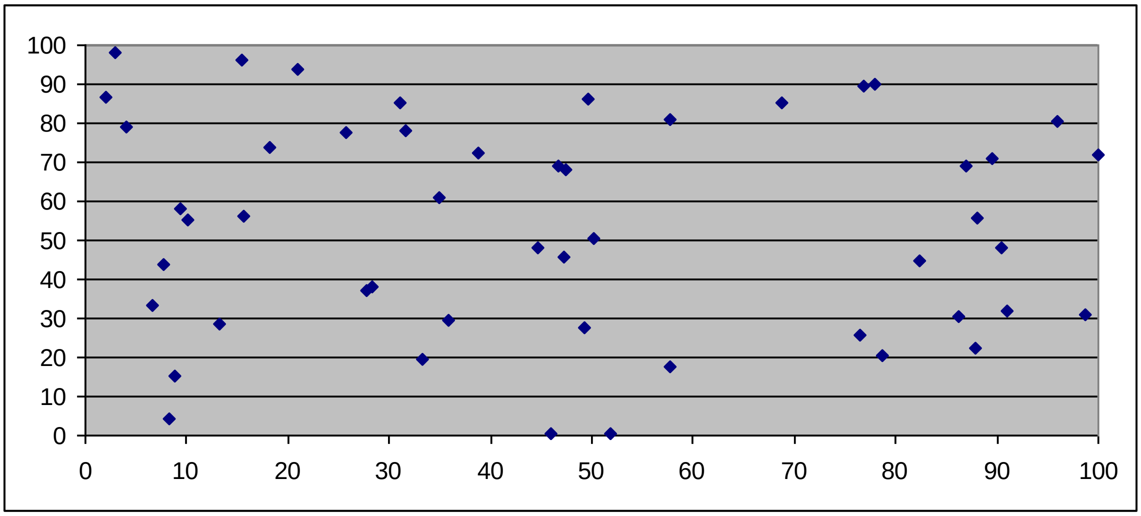


Fig.4.1 Deployment of Sensor Nodes

Along with the static deployment, initial energy and data rate of the static sensors is also generated using random number generator. The range for initial energy is 10000000 to 99999999 and range for data rate is 1 to 10. the relation between data rate and initial energy is maintained so that, if the data rate of sensor $node_i$ is greater than data rate of sensor $node_j$ then, initial energy allocated to sensor $node_i$ will also be greater than the initial energy allocated to sensor $node_j$.

The colony size taken for all the proposed algorithms are 10 i.e. 5 employee bees, 5 onlooker bees and 1 scout bee and the limit for ABC is set to 10 according to the equation (7). The parameters α_1 representing distance independent parameter, α_2 representing the distance dependent parameter in lifetime evaluation equation (2) are taken as 0 and 1. Along with this, the path loss exponent 'n' in equation (2) is taken as 2 for all computations. The jumping rate for opposition based ABC is set to 0.3 and in memetic search based ABC the values of ε and pr are set to 0.01 and 0.4. The performance of all the proposed algorithms are represented for the following values of MaxCycle i.e. 1000, 1500, 2000, 2500, and 3000 as shown in the figures in subsequent sections. For any value of MaxCycle we took value after every 50 cycles and took mean of it for every run.

4.2 Performance Evaluation-GABC

In this section, we present the performance evaluation of GABC (Gbest guided ABC) and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

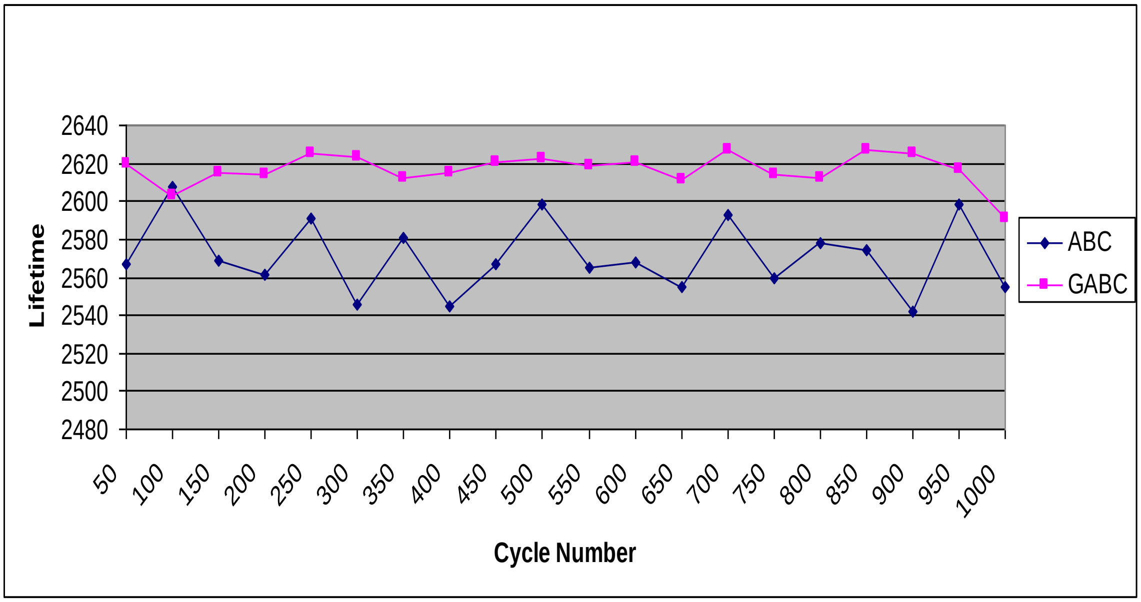


Fig.4.2 ABC Vs GABC for MaxCycle 1000.

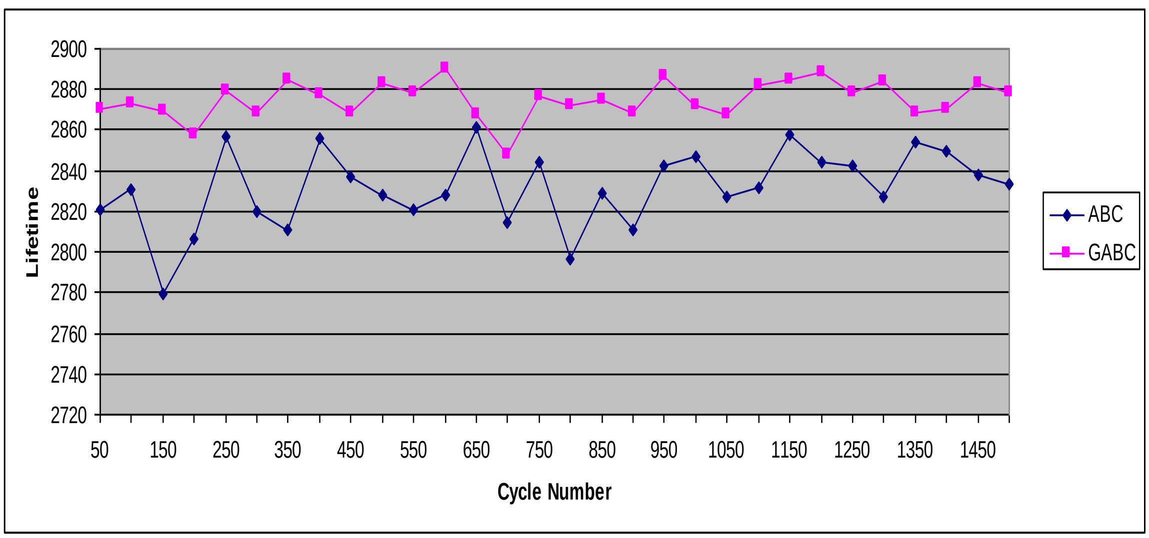


Fig.4.3 ABC Vs GABC for MaxCycle 1500

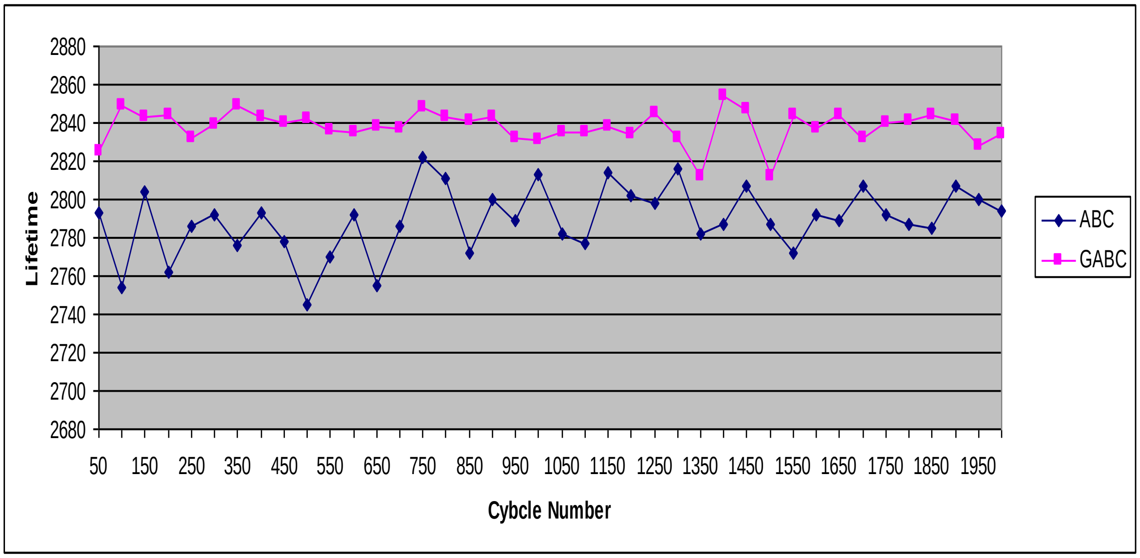


Fig.4.4 ABC Vs GABC for MaxCycle 2000.

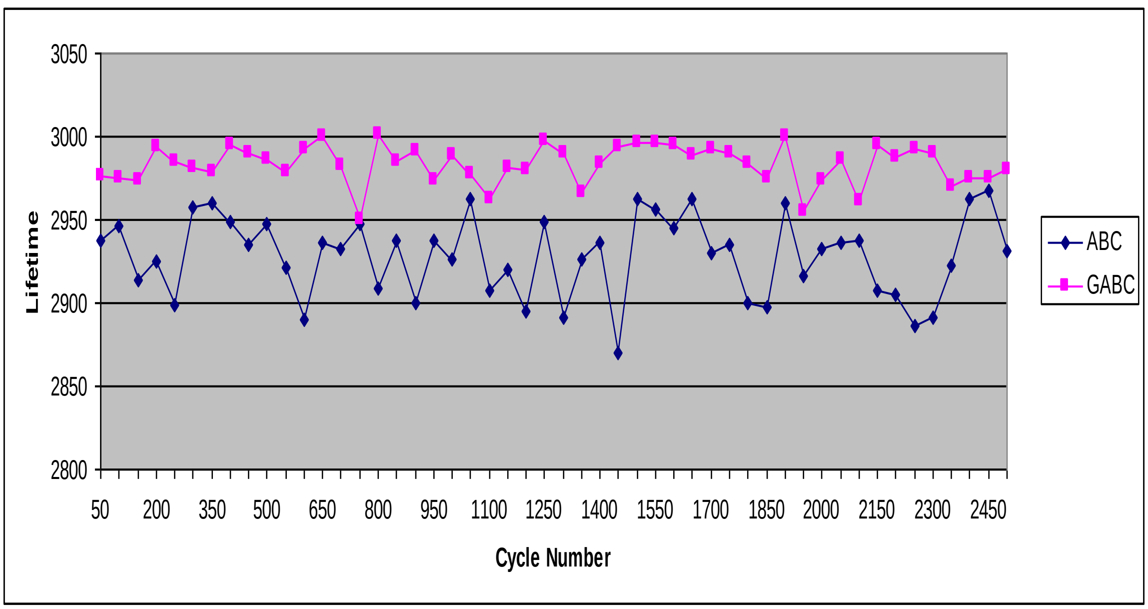


Fig.4.5 ABC Vs GABC for MaxCycle 2500.

As shown in the figures 4.2 to 4.6, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the Maxcycle value, GABC always perform better than ABC.

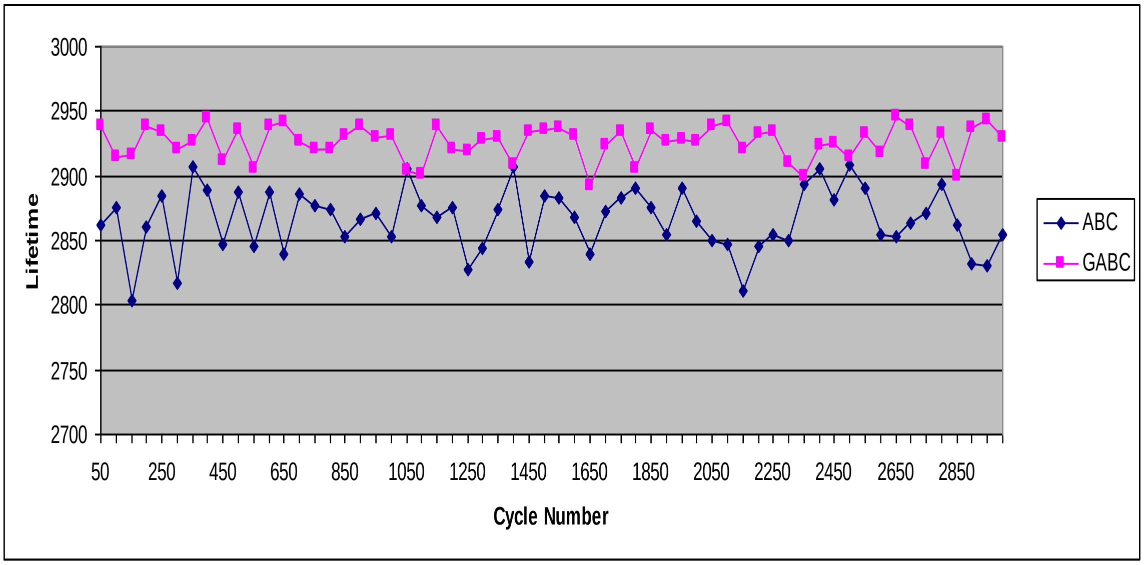


Fig.4.6 ABC Vs GABC for MaxCycle 3000

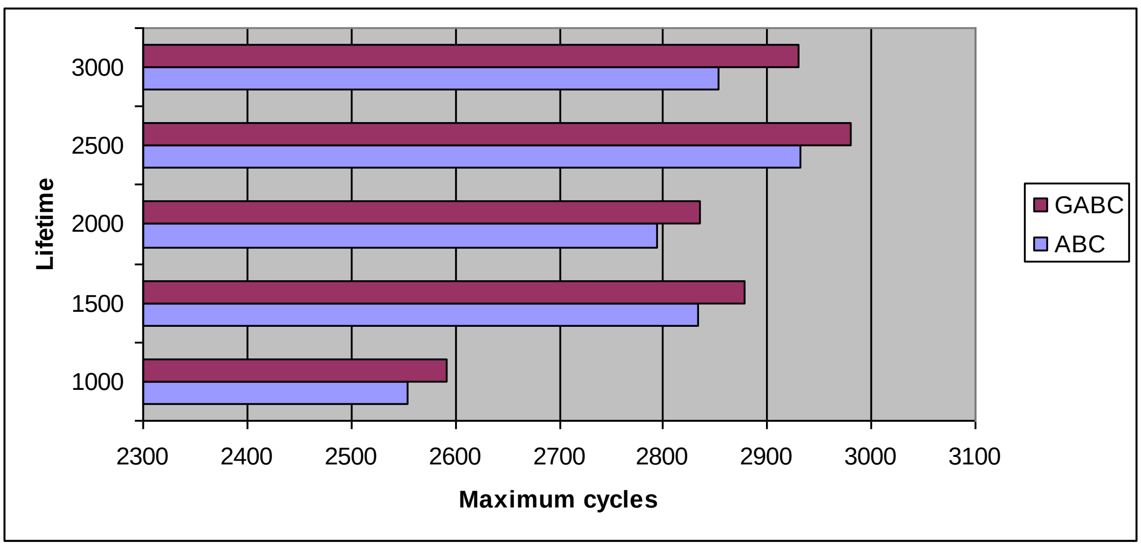


Fig.4.7 Mean Values of ABC and GABC

Fig.4.7 represents the mean value of lifetime returned by both ABC and GABC for different values of MaxCycles, and it is clearly visible that GABC performs better than ABC. The mean difference between ABC and GABC is 1.74% on average according to values in fig.4.7.

4.3 Performance Evaluation-JA-ABC

In this section, we present the performance evaluation of JA-ABC and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

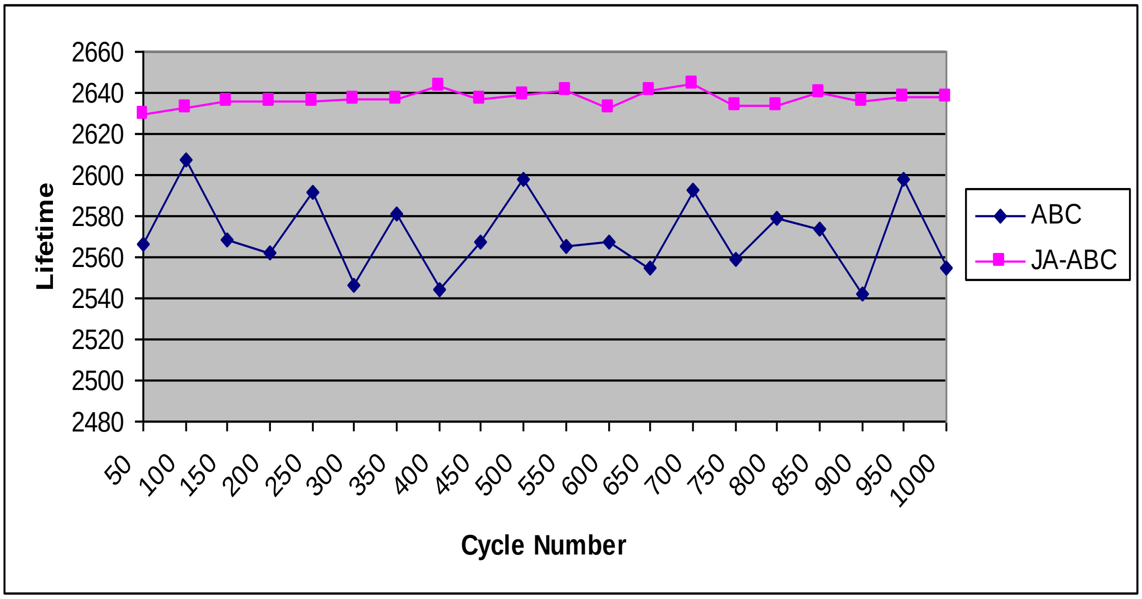


Fig.4.8 ABC Vs JA-ABC for MaxCycle 1000.

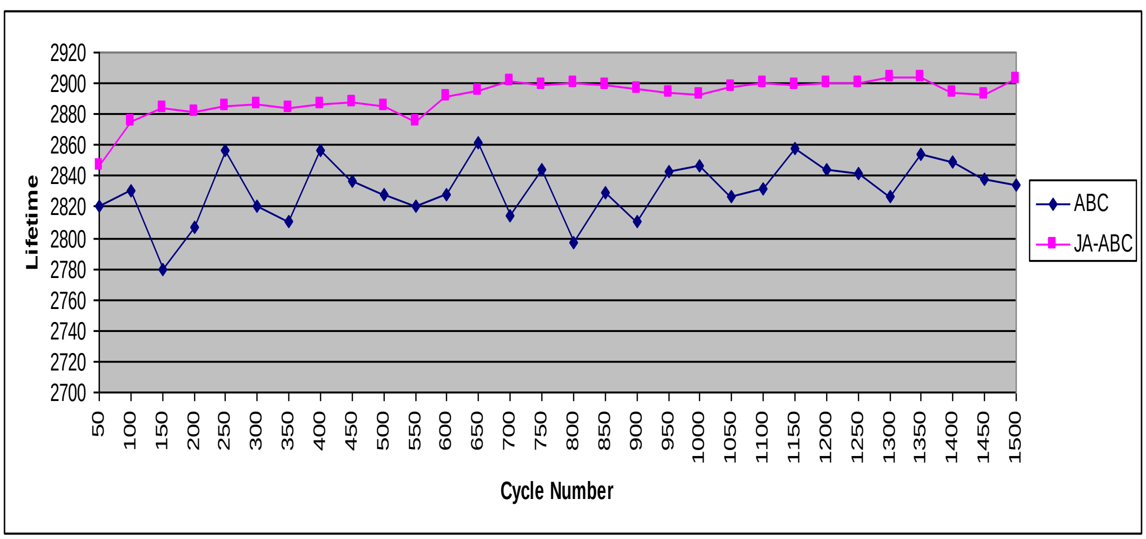


Fig.4.9 ABC Vs JA-ABC for MaxCycle 1500.

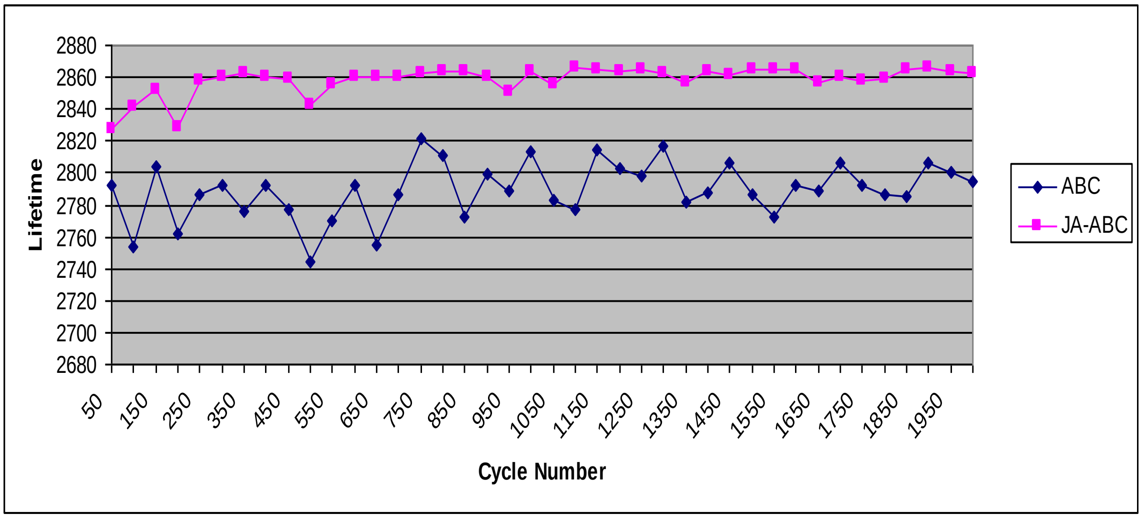


Fig.4.10 ABC Vs JA-ABC for MaxCycle 2000.

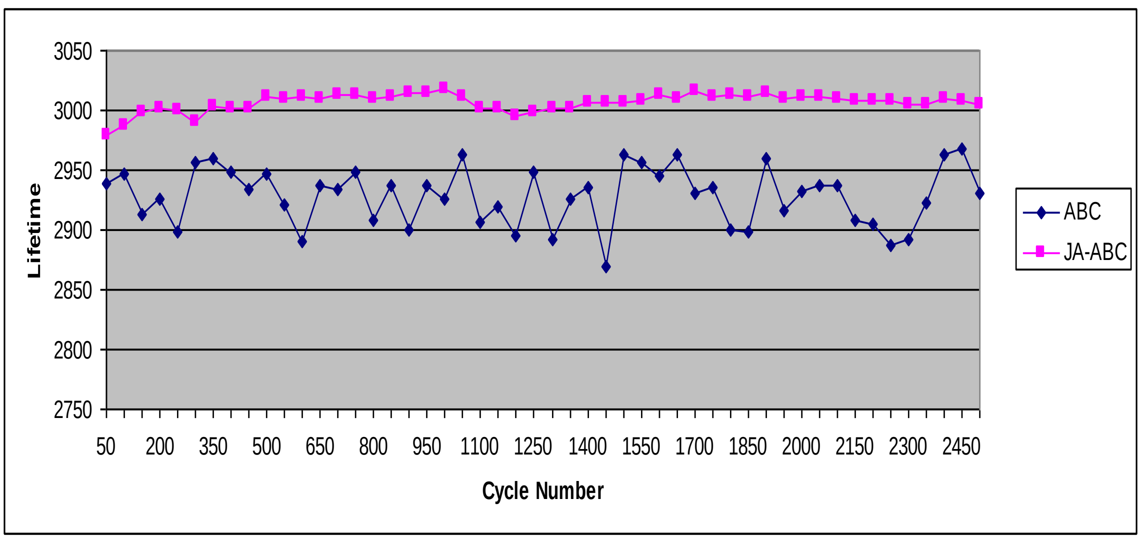


Fig.4.11 ABC Vs JA-ABC for MaxCycle 2500.

As shown in the figures 4.8 to 4.12, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the MaxCycle value, JA-ABC always perform better than ABC.

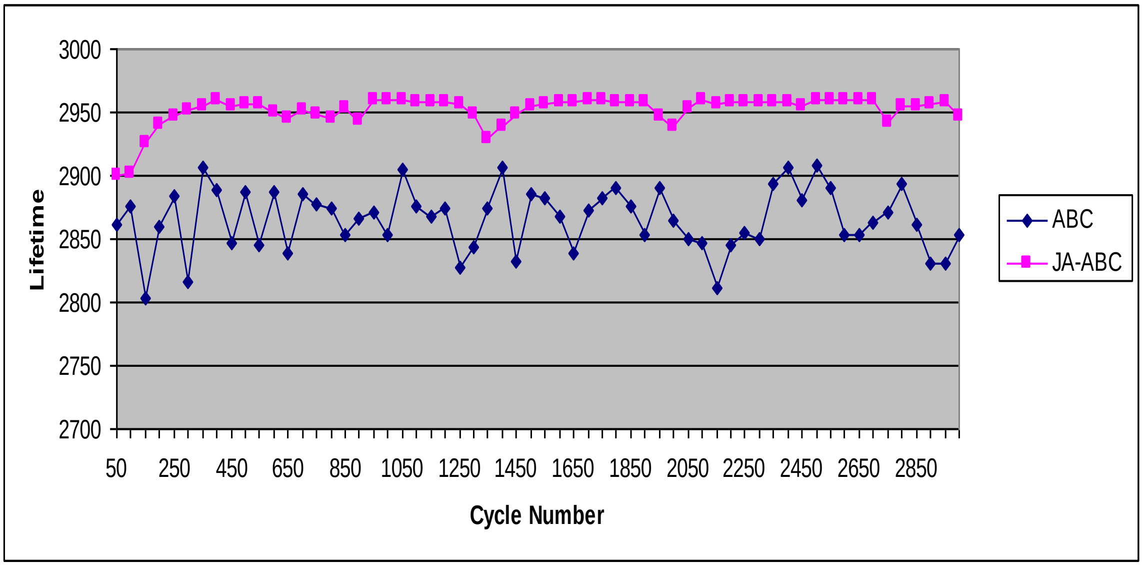


Fig.4.12 ABC Vs JA-ABC for MaxCycle 3000.

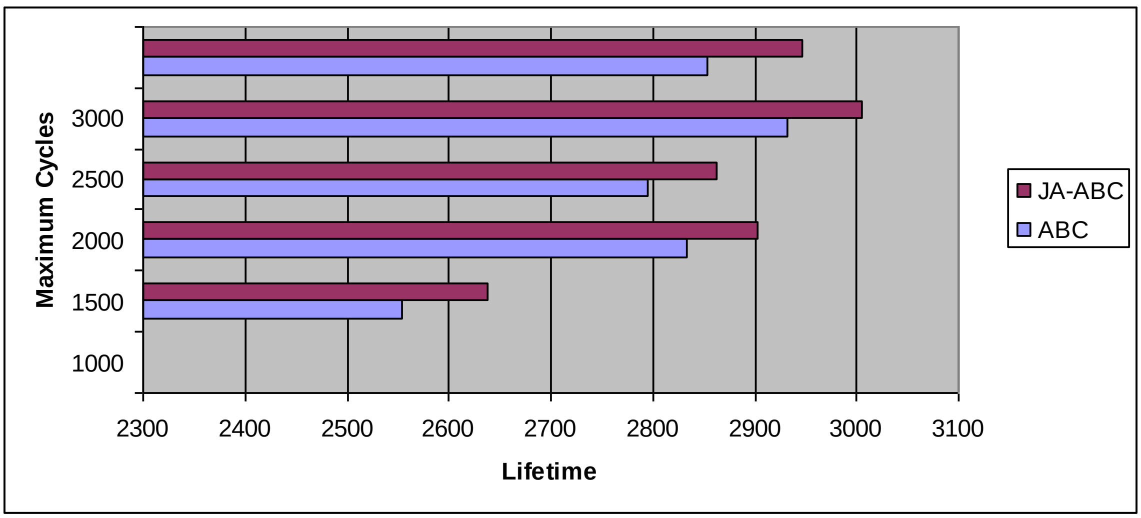


Fig.4.13 Mean Values of ABC and JA-ABC

Fig.4.13 represents the mean value of lifetime returned by both ABC and JA-ABC for different values of MaxCycles, and it is clearly visible that JA-ABC performs better than ABC. The mean difference between ABC and JA-ABC is 2.7% on average according to values in fig.4.13.

4.4 Performance Evaluation-MSABC

In this section, we present the performance evaluation of MSABC (Memetic Seach based ABC) and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

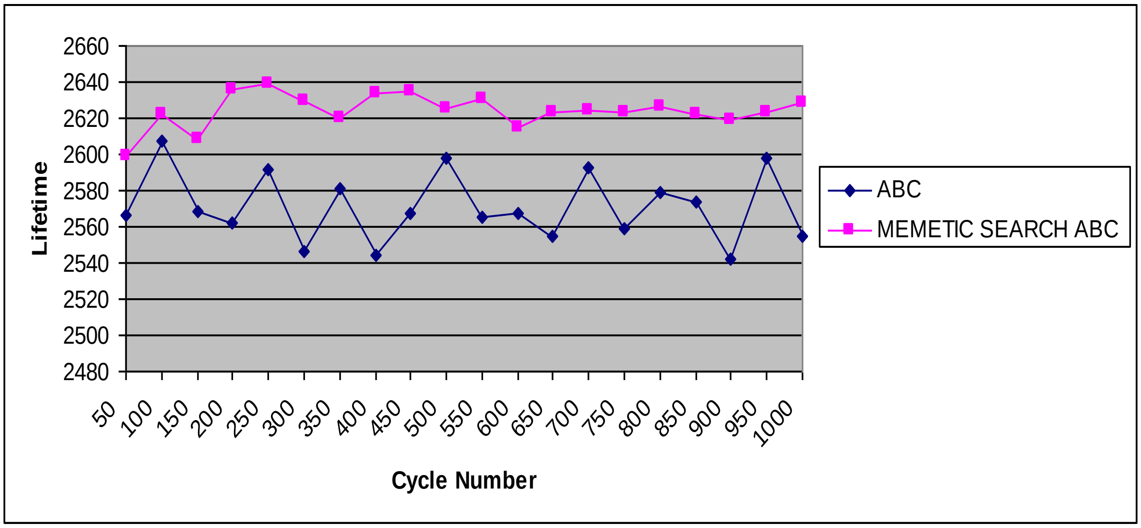


Fig.4.14 ABC Vs MSABC for MaxCycle 1000.

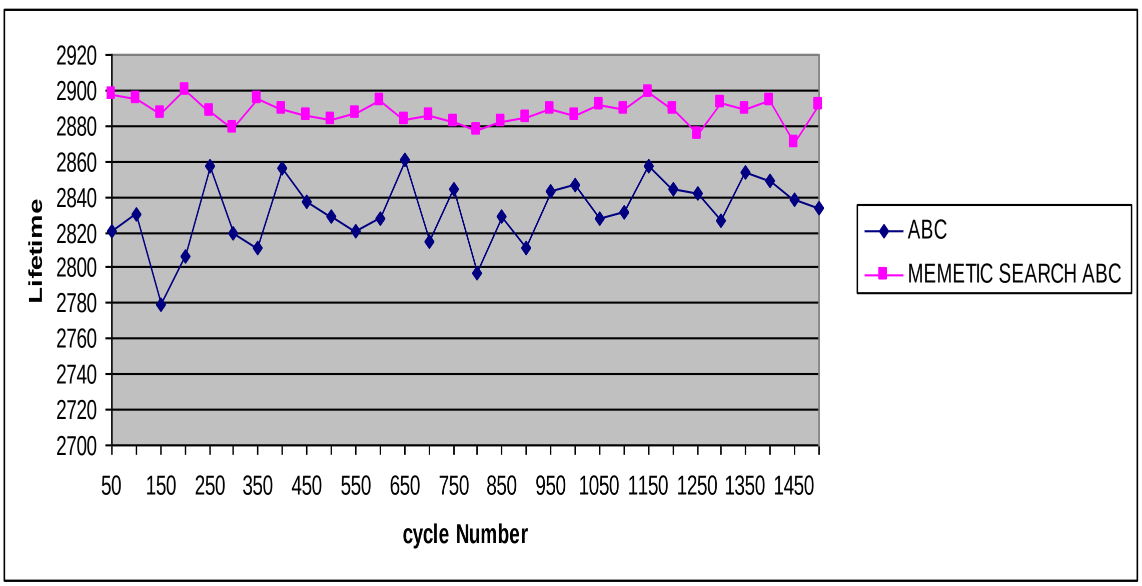


Fig.4.15 ABC Vs MSABC for MaxCycle 1500.

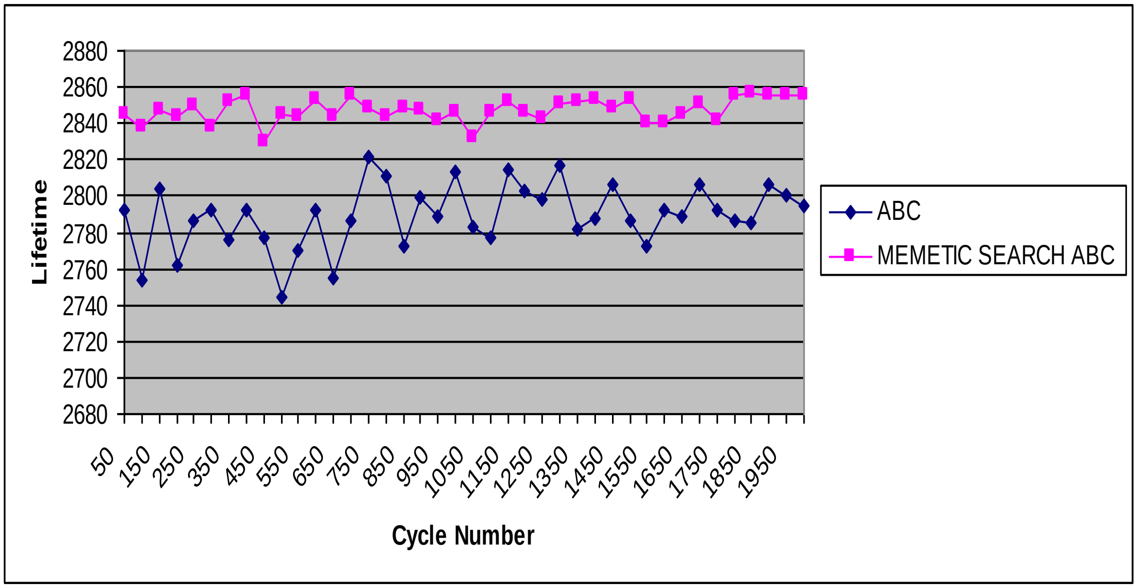


Fig.4.16 ABC Vs MSABC for MaxCycle 2000.

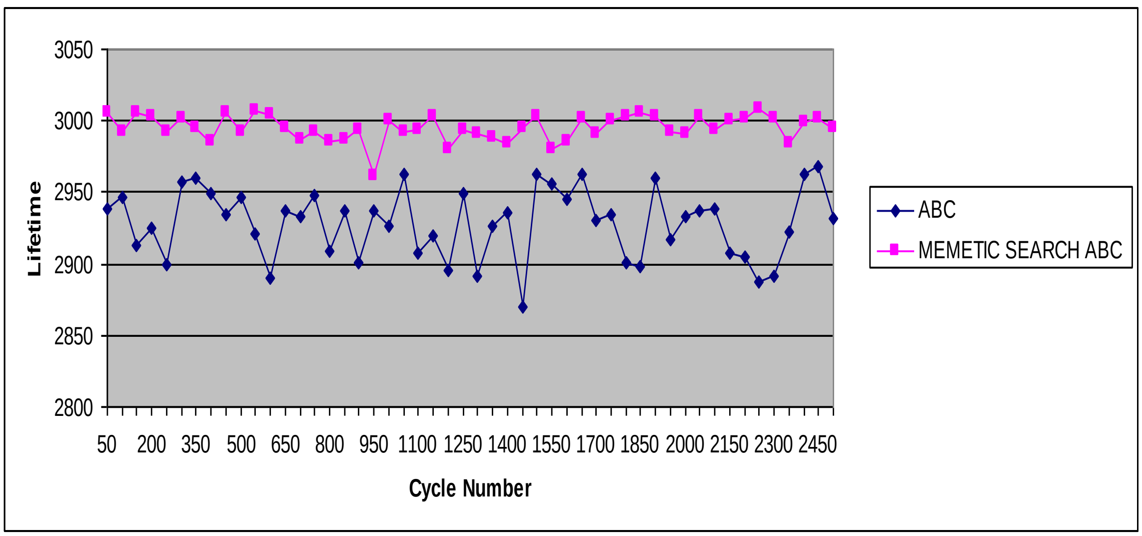


Fig.4.17 ABC Vs MSABC for MaxCycle 2500.

As shown in the figures 4.14 to 4.18, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the Maxcycle value, MSABC always perform better than ABC.

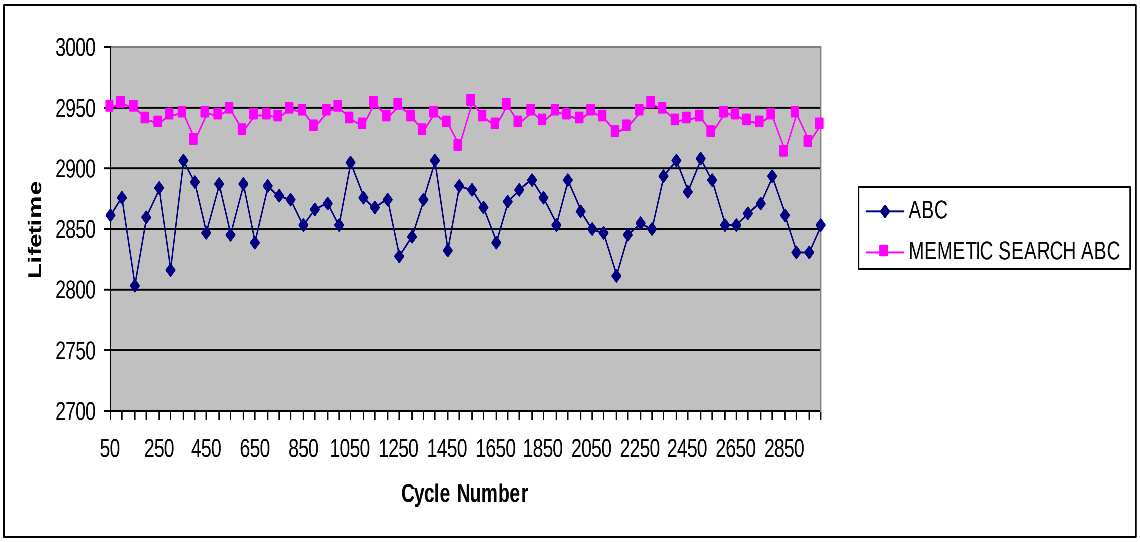


Fig.4.18 ABC Vs MSABC for MaxCycle 3000.

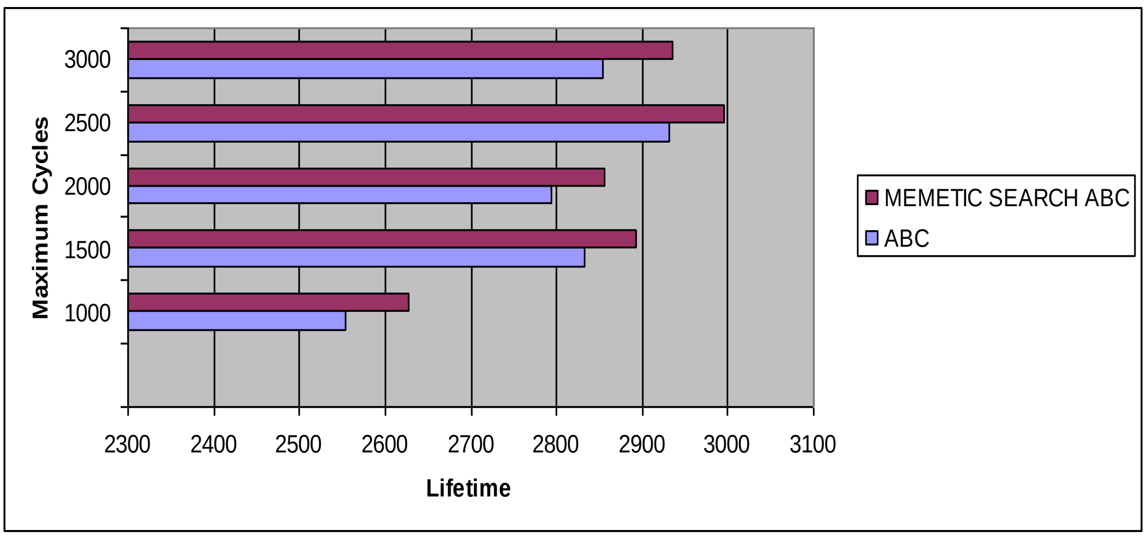


Fig.4.19 Mean Values of ABC and MSABC

Fig.4.19 represents the mean value of lifetime returned by both ABC and MSABC for different values of MaxCycles, and it is clearly visible that MSABC performs better than ABC. The mean difference between ABC and MSABC is 2.47% on average according to values in fig.4.19.

4.5 Performance Evaluation-MLSABC

In this section, we present the performance evaluation of MLSABC (Memetic Local Seach based ABC) and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

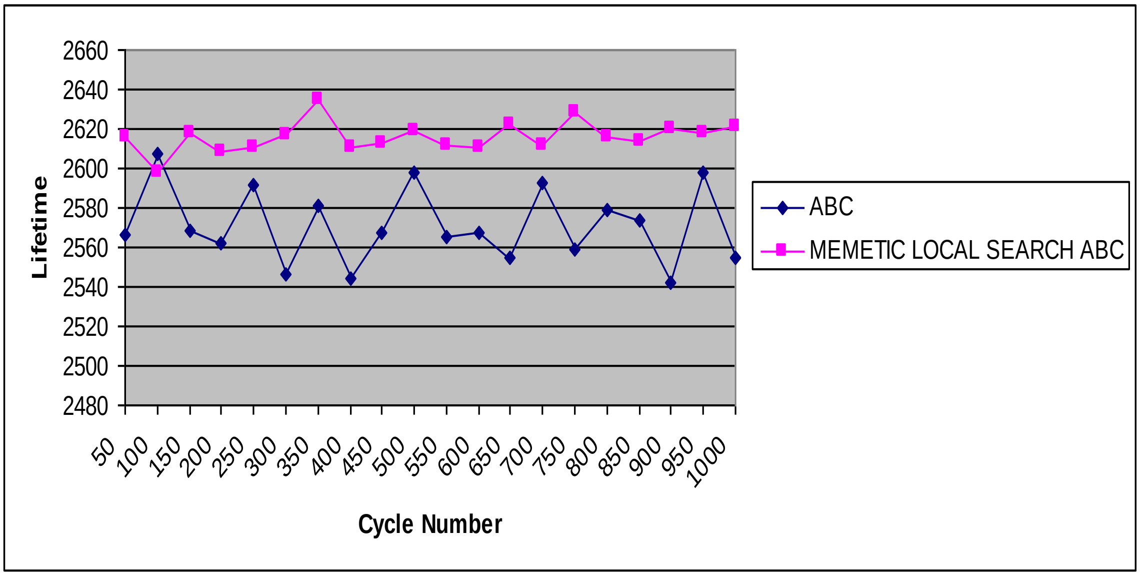


Fig.4.20 ABC Vs MLSABC for MaxCycle 1000.

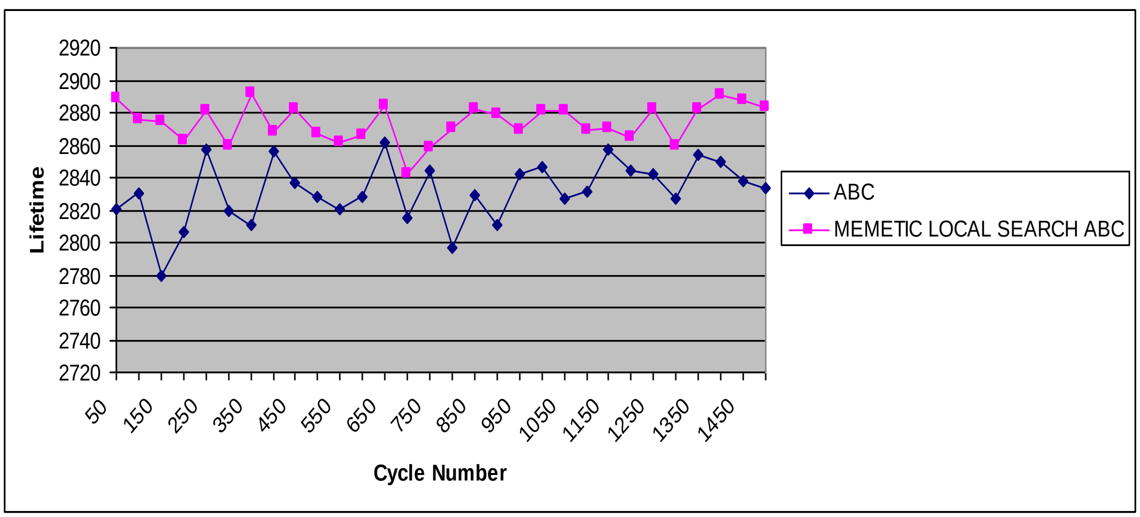


Fig.4.21 ABC Vs MLSABC for MaxCycle 1500.

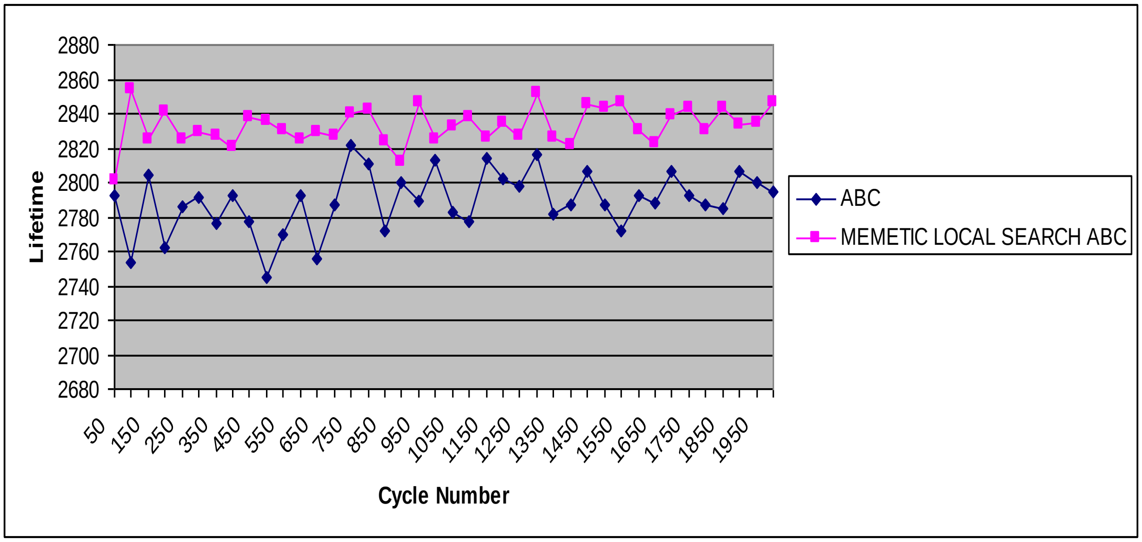


Fig.4.22 ABC Vs MLSABC for MaxCycle 2000.

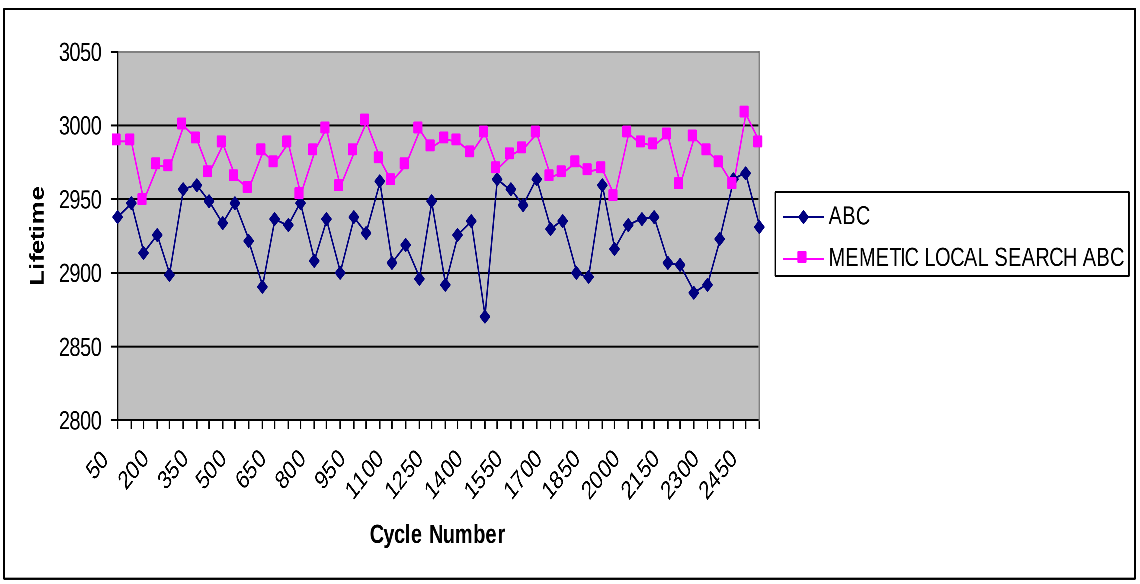


Fig.4.23 ABC Vs MLSABC for MaxCycle 2500.

As shown in the figures 4.20 to 4.24, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the Maxcycle value, MLSABC always perform better than ABC.

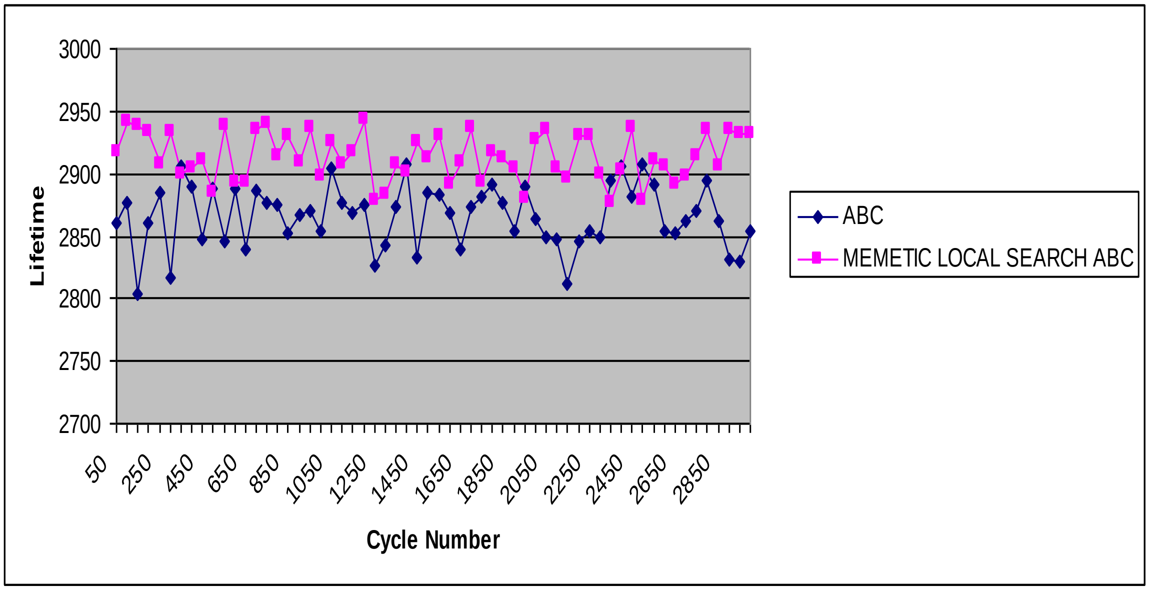


Fig.4.24 ABC Vs MLSABC for MaxCycle 3000.

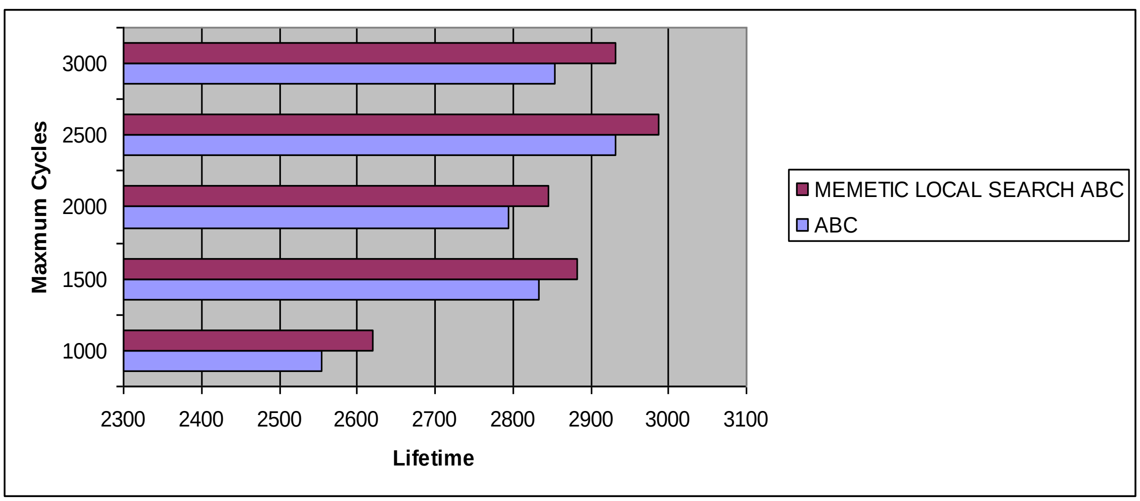


Fig.4.25 Mean Values of ABC and MLSABC

Fig.4.25 represents the mean value of lifetime returned by both ABC and MLSABC for different values of MaxCycles, and it is clearly visible that MLSABC performs better than ABC. The mean difference between ABC and MLSABC is 1.92% on average according to values in fig.4.25.

4.6 Performance Evaluation-OABC

In this section, we present the performance evaluation of OABC (Opposition based ABC) and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

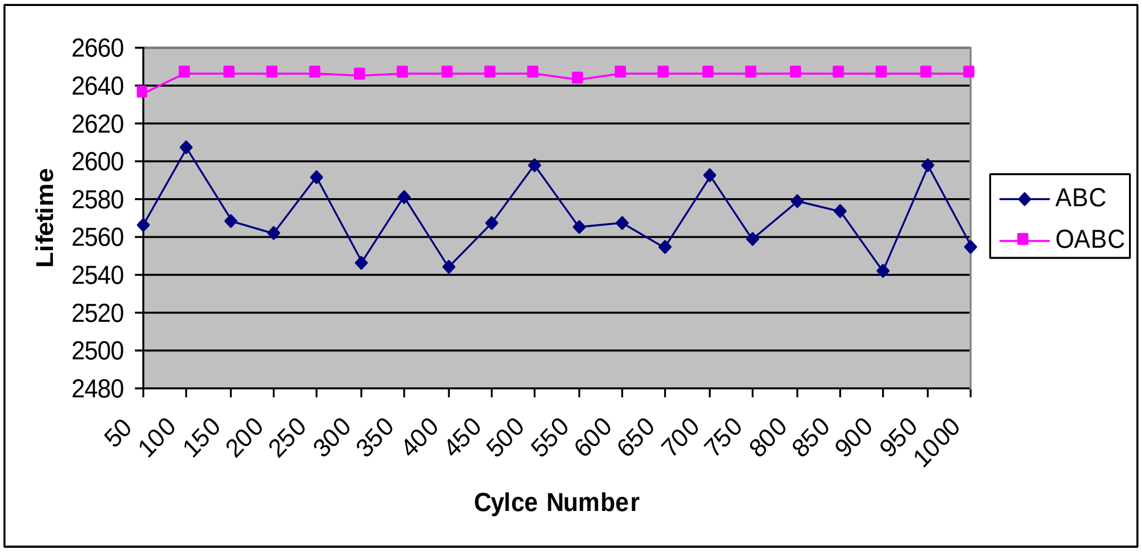


Fig.4.26 ABC Vs OABC for MaxCycle 1000.

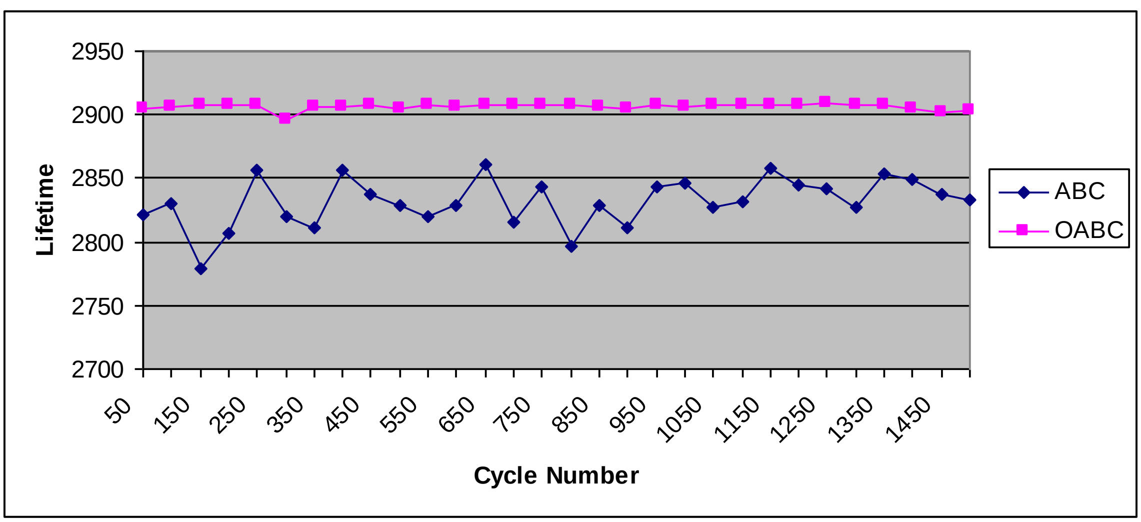


Fig.4.27 ABC Vs OABC for MaxCycle 1500.

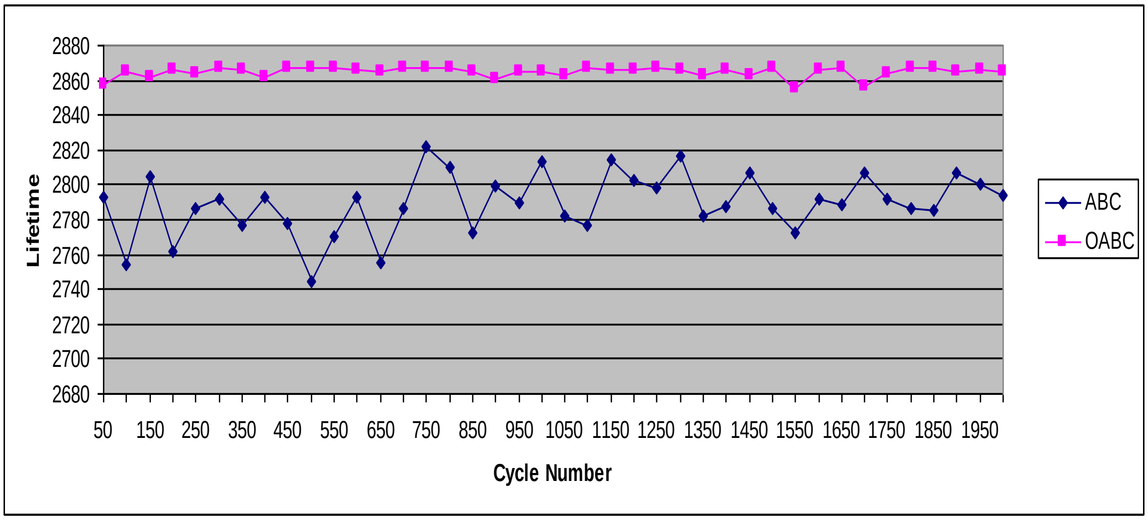


Fig.4.28 ABC Vs OABC for MaxCycle 2000.

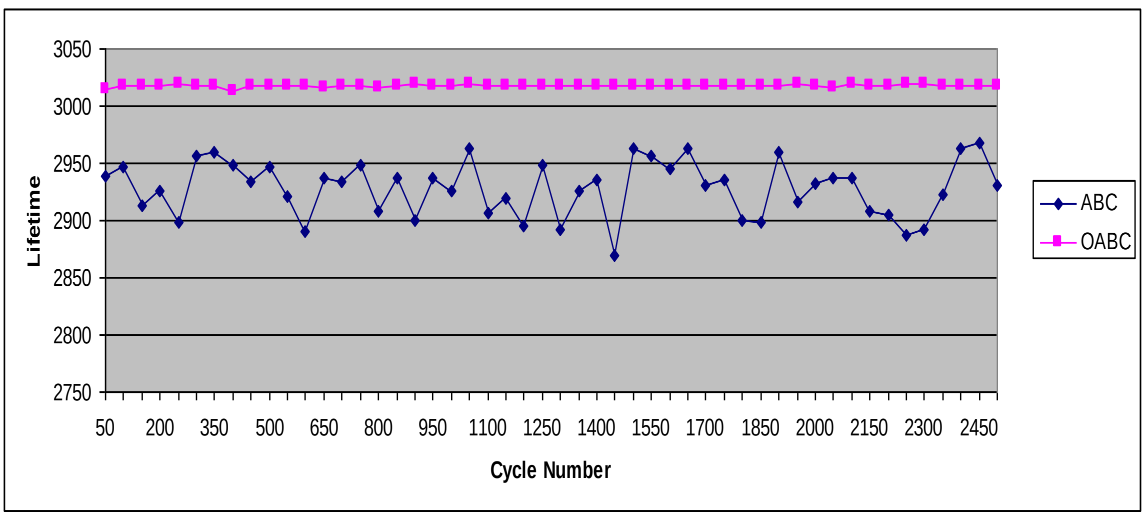


Fig.4.29 ABC Vs OABC for MaxCycle 2500.

As shown in the figures 4.26 to 4.30, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the Maxcycle value, OABC always perform better than ABC.

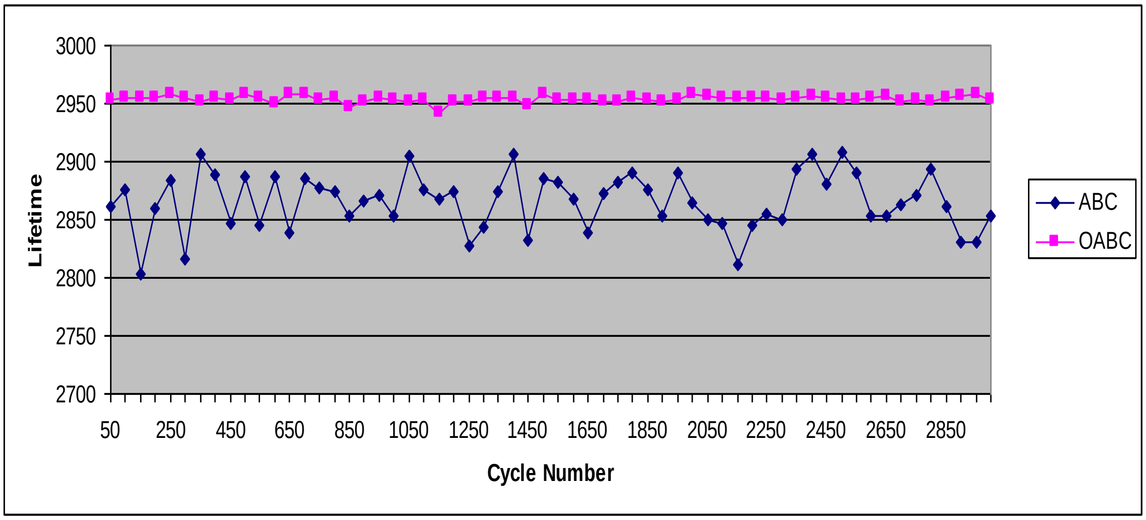


Fig.4.30 ABC Vs OABC for MaxCycle 3000.

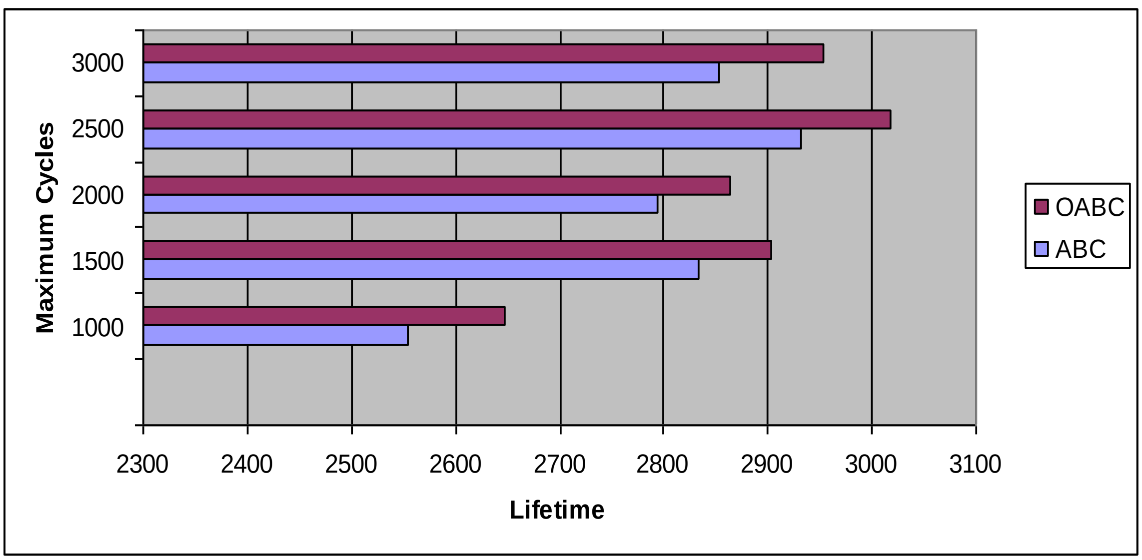


Fig.4.31 Mean Values of ABC and OABC

Fig.4.31 represents the mean value of lifetime returned by both ABC and OABC for different values of MaxCycles, and it is clearly visible that OABC performs better than ABC. The mean difference between ABC and OABC is 2.97% on average according to values in fig.4.31.

4.7 Performance Evaluation-HABC

In this section, we present the performance evaluation of HABC (Hybrid ABC) and compare it with basic ABC algorithm. The results are shown for different value of our performance metric i.e. MaxCycle as follows:

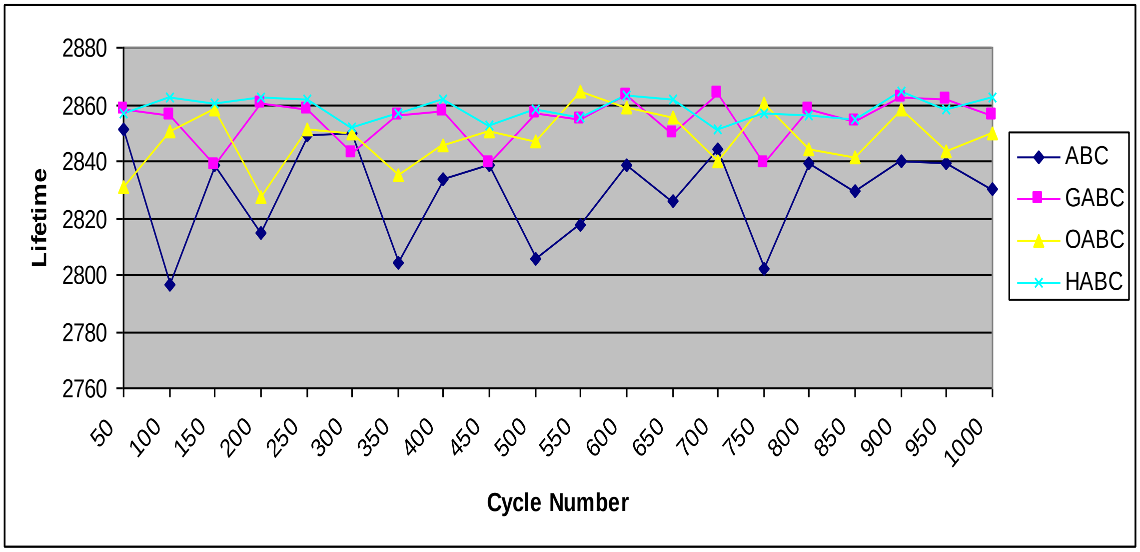


Fig.4.32 ABC Vs HABC Vs GABC Vs OABC for MaxCycle 1000.

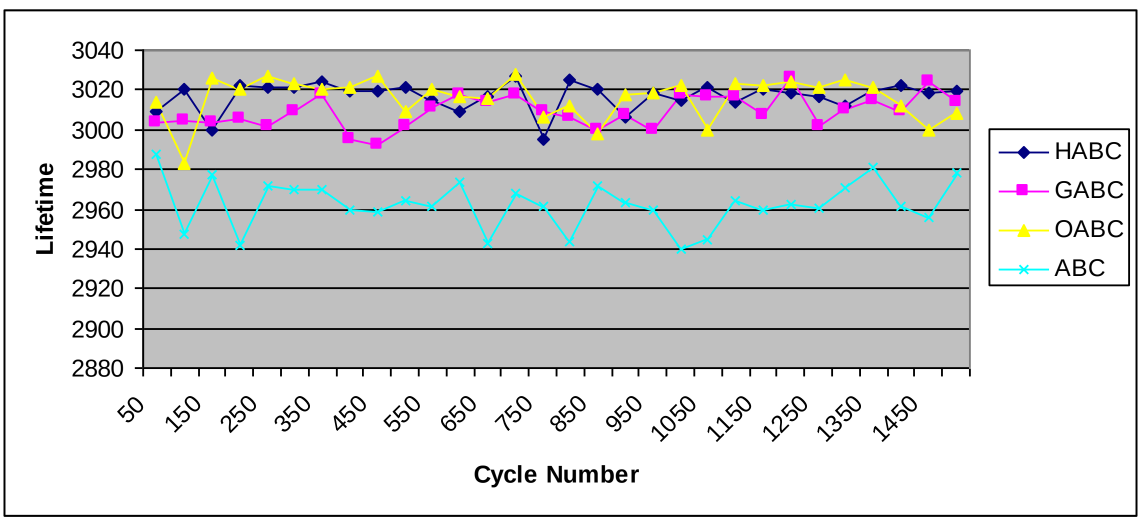


Fig.4.33 ABC Vs HABC Vs GABC Vs OABC for MaxCycle 1500.

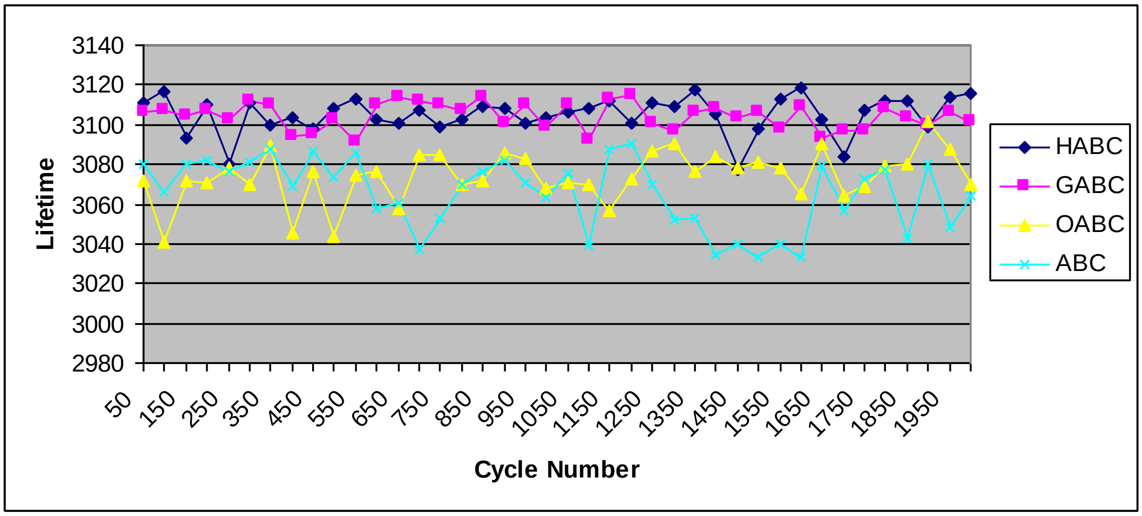
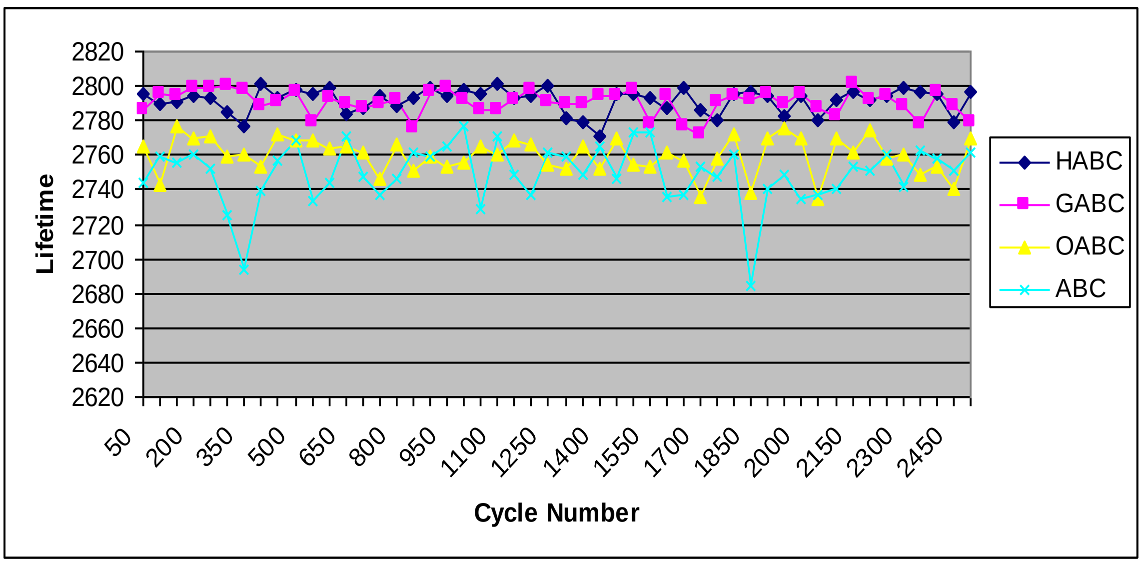
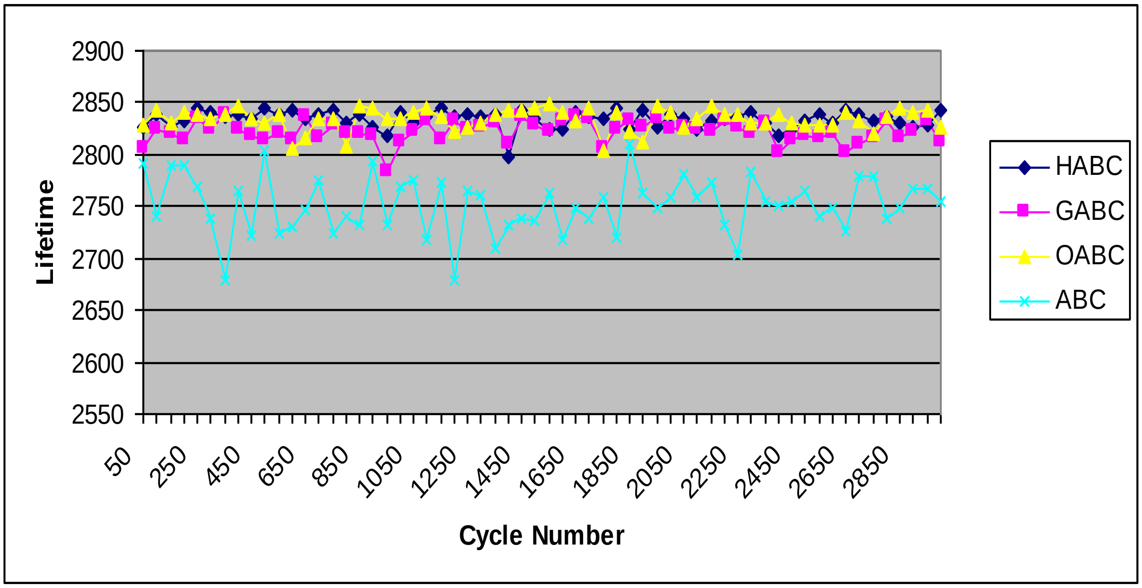


Fig.4.34 ABC Vs HABC Vs GABC Vs OABC for MaxCycle 2000.



4.35 ABC Vs HABC Vs GABC Vs OABC for MaxCycle 2500.

As shown in the figures 4.32 to 4.36, we took the value of lifetime after every 50 cycles for every run and took mean of it which represented in these figures which explains that irrespective of the Maxcycle value, HABC always perform better than ABC and OABC but most of the times similar to GABC.



4.36 ABC Vs HABC Vs GABC Vs OABC for MaxCycle 3000.

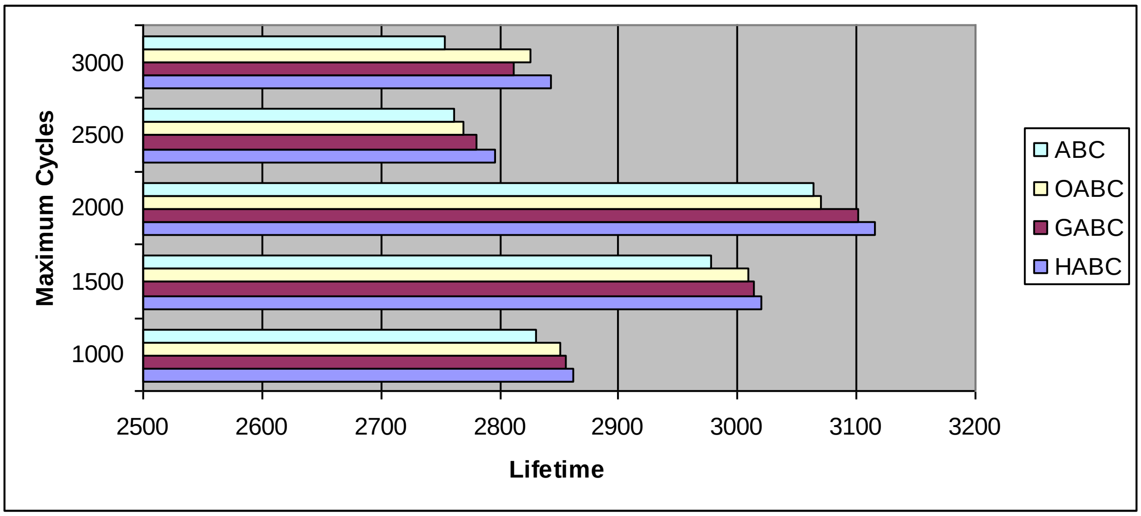


Fig.4.37 Mean Values of ABC, OABC, GABC & HABC.

Fig.4.37 represents the mean value of lifetime returned by both ABC and HABC for different values of MaxCycles, and it is clearly visible that HABC performs better than ABC. The mean difference between ABC and HABC is 3.201% on average according to values in fig.4.37.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

We dedicated our work to base station positioning problem in wireless sensor network for maximizing the network lifetime. We used ABC algorithm for solving the base station positioning problem which consists of two main features i.e. exploitation and exploration. For finding out the optimal results we need to find the perfect balance between these two features. We tried to find the balance via different modified algorithms discussed in the previous chapters. As shown in the previous results chapter, modified algorithms were successful in finding better solutions to our base station positioning problem. HABC provide us the best result in form of 3.2% increase in network lifetime with respect to conventional ABC algorithm. Apart from HABC, OABC also increased the network lifetime by approximately 2.97%. Thus the proposed approaches are observed to be more efficient than the conventional ABC approach and successfully address the objective that we undertook as part of this work.

Though the proposed approaches performed better than the conventional ABC algorithm, there is room for improvement in base station positioning problem field as well as ABC algorithm. We took the assumption for the network model of our problem that every sensor node can transmit its data to the base station in a single hop irrespective of the location of base station and its own location too. We could use a network model in which there is possibility of multi hop transmission. We can incorporate the multi hop network by managing the intermediate nodes before calculating the lifetime of sensor node for particular location of base station.

Another viable future scope of work is incorporating levy distribution for finding the step size used in the local search equation of the ABC algorithm. Levy distribution will increase the exploitation of our search by generating variable step size. Another modification that can be used in local search phase of ABC algorithm is chaotic search technique in which we generate a chaotic sequence which is used in the local search in the form of multiplication component of the search equation.

Apart from levy distribution and chaotic search, we could use a modification on ABC called as adaptive ABC. In adaptive ABC, we analyze the performance of our food sources after every cycle and on the basis of the quality of food sources we decide the number of employee bees and number of onlooker bees for the next cycle. In this way, we can analyze the performance of our algorithm at runtime and tweak it according to the requirement for exploration or exploitation.

Another scope of work in this field is to incorporate dynamic network. By dynamic network we mean that, the sensor nodes can move in the network and as they move, the previously found base station position may become sub-optimal. As discussed above, all our modifications outperforms conventional ABC by increasing the network lifetime, so this approach can be used in finding the optimal base station for the wireless sensor network or can be used as initial step for solving more complex problems like routing, or clustering.

REFERENCES

[1] A. Efrat, S. Har-Peled, J.S.B. Mitchell, “Approximation algorithms for two optimal location problems in sensor networks”, Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems (Broadnets’05), Boston, Massachusetts , October 2005.

[2] Tzung-Pei Hong, Guo-Neng Shiu, “Allocating multiple base stations under general power consumption by the particle swarm optimization”, proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007).

[3] Amir Mollanejad, Leili Mohammad Khanli, Mohammad Zeynali, “DBSR:Dynamic base station repositioning using genetic algorithm in wireless sensor network”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 2, March 2010.

[4] Dervis Karaboga, Bahriye Akay, “A comparative study of artificial bee colony algorithm”, Applied Mathematics and Computation 214 (2009) 108–132.

[5] A. Akkaya, M. Younis, W. Youssef, "Positioning of Base Stations in Wireless Sensor Networks", IEEE Communications Magazine, Vol. 45, No.4, pp. 96-102, April 2007.

[6] J. Pan , Y. T. Hou , L. Cai , Y. Shi and S. X. Shen "Locating base-stations for video sensor networks", Proc. IEEE Vehicular Technology Conf., 2003

[7] Ivan Vilovic, Niksa Burum, Zvonimir Sipus, “Ant colony approach in optimization of base station position”, Proceedings of the European Conference on Antennas and Propagation (EuCAP '09), pp. 2882–2886, Berlin, Germany, March 2009.

[8] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic, New Ideas in Optimization, McGraw-Hill, London, 1999, pp. 11–32.

[9] Yunyue Lin, QishiWu, Xiaoshan Cai, Xiaojiang Du, and Ki-Hyeon Kwon, "On Deployment of Multiple Base Stations for Energy-Efficient Communication in Wireless Sensor Networks" Hindawi Publishing Corporation International Journal of Distributed Sensor Networks Volume 2010, Article ID 563156, 8 pages doi:10.1155/2010/563156.

[10] A. Bogdanov E. Maneva, S. Riesenfeld, Power-aware base station positioning for sensor networks, in: Proceedings of the 23rd International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), Hong Kong, March 2004.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization." Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995 (in press).

[12] P.D. Hossein Zadeh, C. Schlegel, and M.H. MacGregor, "Distributed optimal dynamic base station positioning in wireless sensor networks", Computer Networks, 56(1):34-49, January 2012.

[13] Rajiv Kr. Tripathi, Sateeshkrishna Dhuli, Y. N. Singh, Nishchal K. Verma, "Analysis of Weights for Optimal Positioning of Base Station in a Wireless Sensor Network" Communications (NCC), 2014 Twentieth National Conference on, 1-4

[14] Dorottya Vass , Attila Vidács, Positioning mobile base station to prolong wireless sensor network lifetime, Proceedings of the 2005 ACM conference on Emerging network experiment and technology, October 24-27, 2005, Toulouse, France

[15] Choi, Y. S., Kim, K. S., Kim, N., 2007, in IFIP International Federation for Information Processing, Volume 245, Personal Wireless Communications, eds. Simak, B., Bestak, R., Kozowska, E., (Boston: Springer), pp. 217-229.

[16] Soo Kim, Jeong-Gil Ko, Jongwon Yoon, Heejo Lee, “Multiple-Objective Metric for Placing Multiple Base Stations in Wireless Sensor Networks”, In Proceedings of the International Symposium on Wireless Pervasive Computing, pages 627-631, February 2007.

[17] S.R. Gandham, M. Dawande, R. Prakash, S. Venkatesan, Energy efficient schemes for wireless sensor networks with multiple mobile base stations, in: Proc. IEEE Globecom 2003, San Francisco, CA December 1–5, vol. 1, 2003, pp. 377–381.

[18] Mahmud, S., Wu, H., Xue, J.: Efficient Energy Balancing Aware Multiple Base Station Deployment for WSNs. In: Proceedings of the 8th European Conference on Wireless Sensor Networks, Bonn, Germany, pp. 179–194 (2011).

[19] M. A. Guvensan, Z. C. Taysi, and A. G. Yavuz, "Multiple base station positioning for sensor applications," in Proceedings of the European Computing Conference, ser. Lecture Notes in Electrical Engineering, N. Mastorakis, V. Mladenov, and V. T. Kontargyri, Eds. Springer US, 2009, vol. 27, pp. 619-629.

[20] Anurag Singh Tomar, Gaurav Kumar Tak, “Optimized Positioning Of Multiple Base Station for Black Hole Attack”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 8, August 2014.

[21] Y. Shi, Y.T. Hou, Approximation algorithm for base station placement in wireless sensor networks, in: Proceedings of the Fourth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON ’07), June 2007, pp. 512–519.

[22] J. Pan, Y. Hou, L. Cai, Y. Shi and X. Shen, "Topology Control for Wireless Sensor Networks,"*Proc. of Ninth ACM Mobicom*, pp. 286-299, 2003.

[23] Guopu Zhu, Sam Kwong, “Gbest-guided artificial bee colony algorithm for numerical function optimization” Applied Mathematics and Computation 217 (2010) 3166–3173.

[24] Noorazliza Sulaiman, Junita Mohamad-Saleh, Abdul Ghani Abro, “A Modified Artificial Bee Colony (JA-ABC) Optimization Algorithm”, Proceedings of the 2013 International Conference on Applied Mathematics and Computational Methods in Engineering.

[25] Jagdish Chand Bansal, Harish Sharma, K. V. Arya, Atulya Nagar, “Memetic search in artificial bee colony algorithm” Soft Comput (2013) 17:1911–1928 DOI 10.1007/s00500-013-1032-8.

[26] Sandeep Kumar, Vivek Kumar Sharma, Rajani Kumari, “An Improved Memetic Search in Artificial Bee Colony Algorithm”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 1237-1247.

[27] Tizhoosh, H.R, “Opposition-based learning:A new scheme for machine intelligence.” ,In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, 28– 30 November, Vienna, Austria, pp.695–701.

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