World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 5.924

A MATHEMATICAL MODELLING FOR EFFECTIVE MOBILE AGENT ITINERARY USING FARTHEST NODE FIRST, NEXT NEARER NODE ALGORITHM

Prapulla S. B.[1*], Shrishah S.[2], Shobha G.[3], Thanuja T. C.[4] and Prakash R.[5]

[1,2,3,5]RV College of Engineering, Bengaluru.
[4]VTU Extension Centre, Muddenahalli, Chikkaballapur.

Article Received on 21/10/2021 **Article Revised on 11/11/2021** **Article Accepted on 01/12/2021**

***Corresponding Author**
Prapulla S. B.
RV College of Engineering, Bengaluru.
DOI: https://doi.org/10.17605/OSF.IO/8DPJ3

ABSTRACT

Due to the rising exploitation of Wireless Sensor Networks (WSN) for boosting all major traditional application domains and enabling brand new application domains, applications based on WSNs have gained lot of attention. Design techniques, and frameworks have been built and made accessible to aid high-level development of WSN applications. Despite the abundance of ideas, more study should be focused on energy efficient techniques as WSNs are built with limited bandwidth and inconsistent connections. The Sensor Nodes(SN) in a WSN are densely packed, resulting in a large number of SN across the sensing zone that can perceive a geographic spot. To reduce sensor data collection, transmission costs and extend sensor battery life, Mobile Agents (MA) and its itinerary planning algorithm are proposed for collecting sensor data. Many approaches have been offered to address the issue of MA itinerary, but majority of these methodologies ignore the cut nodes. This study presents a dynamic itinerary planning using algorithm Farthest Node First, Next Nearer Node, which considers signal strength rather than physical distance. Cut nodes are not included in the itinerary since they do not emit electromagnetic radiation. Clusters arise when SNs are more and the cluster head selection is computed mathematically. Simulation was used to assess the mobile agent's migration from base station to sensor nodes and back to the base station. SN were simulated as points in a two-dimensional coordinate space. The outcomes of simulations of static-static, static-dynamic, and dynamic-dynamic scenarios are discussed.

1. INTRODUCTION

Wireless Sensor Networks(WSN) refers to **a set** of spatially dispersed and dedicated sensors for observing and tracking the physical conditions of the environment and consolidating the collected data at one location.[11] Each node is capable of remote sensing, processing, and communicating. These sensors can gather information about the physical environment for monitoring purpose. Mobile Agent (MA) - based WSNs present an interesting option in contrast to the client-server architecture. Rather than sending the information accumulated by every sensor to the sink as in client-server architecture, Mobile Agents are software entities that are used to communicate and transfer information between the various nodes in a WSN. They share the same hardware and software as of the network. Multi Mobile Agents are used to parallelize the activity of data gathering, executing codes and transferring information in a WSN. Mobile agent paradigm is an emerging and exciting paradigm for mobile computing applications. There are several inefficiencies associated with traditional client-server applications in terms of latency, bandwidth, vulnerability to network disconnection, mobility etc. Mobile agents help the client-server applications to be more autonomous. They also enable client-server architectures to be flexible applications with non-permanent connections by adding mobility to code, intelligence and improved network lifetime.

Mobile Agent technology can be used in multiple places like e-commerce, health care monitoring, communication and personal assistance and so on. To make the mobile agent operation efficient, it is important to consider the itinerary of the mobile agents in a network. The use of mobile agents is a viable option since it can reduce network traffic and latency. The itinerary of an agent is the path that the agent follows in the network. This path is with respect to the order in which it visits nodes, starting from the origin or the base station. There are three main types of itinerary. Static itinerary is the one in which the path is computed before dispatch of agents from the base station and the agent visits only the nodes marked in its path. Dynamic itinerary involves an agent roaming in a network with the agent having enough intelligence to visit nodes that require service. Hybrid itineraries involve a combination of both dynamic and static itinerary to serve the purpose of movement of agents in the network. Finding the ideal itinerary has been discovered to be NP-hard and is still a research topic.

The goal of networking is to maximise the lifetime of the network and at the same time reduce the time required to gather information from the nodes in the network. The lifetime of the network is usually defined as the first node in a network that goes down. In a traditional multi-hop network, it is usually the nodes close to the sink which go down the earliest since nearly all traffic flows through them. The use of mobile agents is done to manage this depletion of energy and prolong the lifetime of the network.

Traversing through holes/ cut or cut nodes in the network consumes time as well as energy. This problem is implicitly taken care here. The nodes which do not emit electromagnetic waves are assumed to be cut or they are referred to as cut nodes.

Cloning of the agent is implemented for energy efficiency and to avoid long traversal paths by the agent.

Mathematical modelling is performed for node placement, setting the direction towards the farthest and nearest nodes based on signal strength and mobile agent migration in synchronization with the sensor sensing using superposition principle.

Considering the wide applications of Wireless Sensor Networks refers to **a set** of spatially dispersed and dedicated sensors for observing and tracking the physical conditions of the environment and consolidating the collected data at one location.

Different data accumulation processes require various quantities of energy for data processing. The determination of data collection strategies does not just rely on the applications yet additionally on the overall energy conservation. As quite a few sensor nodes frequently recognize common occurrences, there might be **redundant** data, so most of the applications install more sensors than required to accomplish good QoS.

2. Literature Survey

The authors of[1] have statistically proved that the energy consumption is minimized in WSN by transferring data using multi-hops over PDA’s. The lifetime of the network is defined as the first sensor to go down in the network.

The authors of[2] mention that Multi-agent Itinerary Planning (MIP) is of nearly the same complexity as Single-agent Itinerary Planning (SIP) but it is more flexible in different network scales in comparison to SIP.

The authors of[3] have proved that energy consumption is significantly minimized using the MIP. The authors analysed the background of data gathering using MA-based WSN models. They proved that energy consumption is significantly minimized using the MIP (Multi-agent Itinerary Planning). Also, unlike SIP (Single-agent Itinerary Planning) MIP doesn't suffer from problems such as task duration and MA packet size. Some drawbacks of the MIP approach are discussed. Many MIP algorithms such as CLMIP, GA-MIP, GIGM_MIP, and directional angle-based MIP are analysed.

The authors of[4] have suggested the use of a linking agent and data agent to optimize energy consumption in WSN. It uses the Farthest Node First, Nearest Node Next Algorithm to improve efficiency.

The authors of[5] have proposed an access control mechanism for communication between the agent and node using public and private keys.

The authors of[6] discuss some of the common itinerary planning algorithms like LCF, GCF and variants of the same. They also list down the various network level assumptions and compare the performance of the different itineraries. MIP algorithm showed better performance in comparison to SIP with respect to network lifetime.

The authors of[7] have discussed that the basis of KTA (Karagül, Tokat and Aydemir) algorithm is Newton's law of mass gravity. In the KTA Algorithm, a distance matrix is constructed with respect to customer demand. The solution weight matrix is computed using the centre of gravity method. The least costly route out of all the routes is chosen. Van Breedam's heuristic algorithm uses Chain Crossing (SC), Chain Changer (SE), Chain Displacement (SR) and Chain Linkage (SM). It requires an initial solution which is derived from the KTA algorithm and then the various operations are performed. In Savings Algorithm, if the two nodes are not connected to each other, the vehicle leaving the depot will go back and forth to each of the nodes whereas if they are connected then it will go to the first point and then to the second point. It's inferred that Savings algorithm outperforms KTA algorithm and Van Breedam's heuristic algorithm.

3. Proposed Scheme

The sensors (motes) have a memory unit and a processing unit. The sensors are capable of processing the data. A mobile agent is a composition of computer software and data which is able to migrate (move) from one computer to another autonomously and continue its execution on the destination computer. The base station generates electromagnetic waves. The nature of waves considered is sinusoidal over time. Sensor nodes as well as the base station emit electromagnetic waves. Base station and origin are used interchangeably.

4. Design

Mobile agent migration[9] has significant contribution to the energy efficiency of the Wireless sensor network. Single agent itinerary has some drawbacks like inflation, delay, low throughput, etc. Thus, Multi mobile agents are used. In this paper, multiple agents are used. The migration strategy of a mobile agent will fall into any one of the following categories. (Venkatesan et al 2011). Static Itinerary with Static Order (SISO) Static Itinerary with Dynamic Order (SIDO) Dynamic Itinerary with Dynamic Order (DIDO) mobile agents are deployed in the network without any predefined travel plan. That is, the code designed for the mobile agent does not include any itinerary details. SIDO is not accessible with versatile alternatives. Essentially, agents are versatile in nature if the nodes to be visited are static; the versatile nature is not effectively used. Hence DIDO is more relevant.

Here two cases are considered, static and dynamic. Under the static case, static itinerary is considered. Under the dynamic case, both sensors as well as itinerary are dynamic. The mathematical model of network is divided into concentric circles as shown in Figure1. Figure 1 depicts the way the sensors are deployed and in each circle the number of nodes is increased by two. The angle between the nodes is.

$$\frac{360}{No.\ of\ nodes}$$

And the region is further divided into four quadrants/zones. Four quadrants are assumed due to simplicity. Later this can be extended to any number of zones depending on the Region of Interest. The number of mobile agents is proportional to the number of quadrants. In each quadrant initially, an agent migrates from the Base Station (BS) to the sensor nodes. In case of branching, the agent clones and the cloned agent collects the data from the sensor nodes.

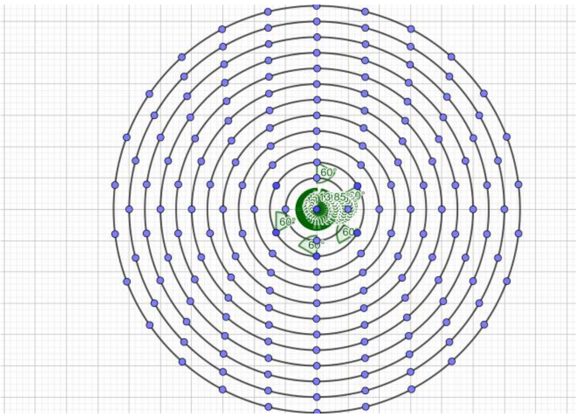

Figure 1: Node placement.

Mobile agents migrate from base station to sensor nodes via cluster heads to collect the data and migrate back to cluster head (Ferry node) and finally to BS.

It is known that the signal strength is inversely proportional to the square of the distance between them. The clusters are formed in the network using the X means algorithm.

Mobile agent migration

Mobile agents migrate from base station to sensor nodes via cluster heads to collect the data and migrate back to cluster head (Ferry node).

Ia. Since the assumption of waves is sinusoidal, the wave function generated by base station is

$$f(X,t) = \frac{A sin \omega_1 t}{X^2 + 1}$$

A is peak signal strength, ω_1 is angular frequency of wave, t is the time synchronizationand X is the distance of wave front from the origin(BS).

Ib. Sensors sensing information - is nothing but noise. Consider the wave function of sensor nodes as

$$g(x,t) = \frac{B sin(\omega_2 t + \emptyset)}{x^2 + 1}$$

B is the peak signal strength (amplitude) at sensor node,ω_2-angular frequency of waves at sensors, t is the time synchronization and x distance of sensor node wave front from its origin in local coordinate.

II. Setting the direction towards the sensor node to find the farthest node

a. Base station assumptions

BS is located at the origin of global coordinate

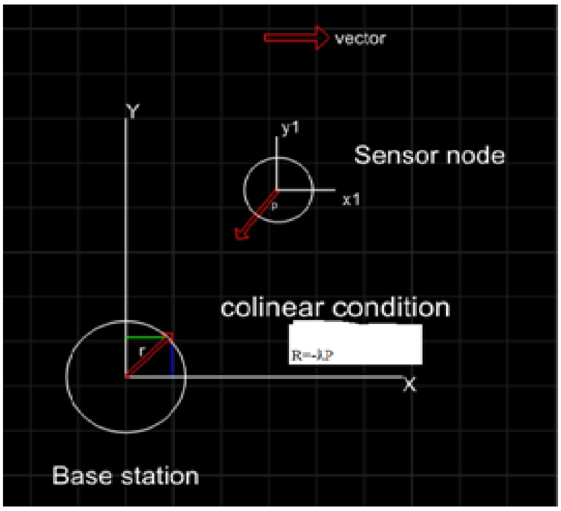

Figure 2: Collinear condition.

Mobile agent channelization using superposition principle is implemented so that the agent migration to sensor's sensing is synchronized.

Let us consider that at time t_1, the wave front location shall be at distance X.

The wave function of Base station shall be

$$f(X,t) = \frac{Asin\omega_1 t}{X^2 + 1}$$

The wave function of sensor node shall be

$$g(x,t) = \frac{Bsin(\omega_1 t + \emptyset)}{x^2 + 1}$$

Let us assume that sensor node is located somewhere at distance R where R is the sum of X and x.

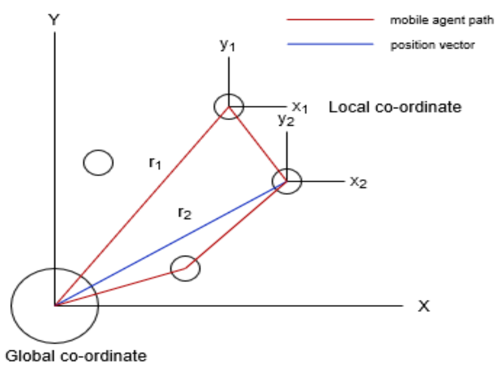

Figure 3: Location of sensor nodes.

Selection of node criteria

The function assigned to base station is

$$f(X,t) = \frac{Asin\omega_1 t}{X^2 + 1}$$

and to sensor node is.

$$g(x,t) = \frac{Bsin(\omega_1 t + \emptyset)}{x^2 + 1}$$

where ω_1 and ω_2 are angular frequencies which are assumed to be constant. Consider at time t, neglecting the sinusoidal wave nature over the time.Let f(X) be defined as $f(X) = \frac{A}{X^2+1}$ and sensor node g(x) be defined as$g(x) = \frac{B}{x^2+1}$, where A and B are the maximum signal strength of base station and sensor nodes respectively.

Selecting the farthest node by using the following.

$$[H(R)]_{x=0} = [f(X)]_{max} + [g(x)]_{min}$$

which is based on superposition principle. The process of migration of mobile agent is done using the following condition.

$$f(X) = g(x) \qquad \ldots (1)$$

This will help in computing the distance between base station and farthest node

The point where the condition given in (1) occurs as following.

i. From base station.

$$X_1 = \sqrt{\frac{A}{f(X)} - 1}$$

ii. From sensor node.

$$x_1 = \sqrt{\frac{B}{g(x)} - 1}$$

The maximum distance is computed as follows.

$$[R_1]_{max} = X_1 + x_1$$

which is the distance between the base station and sensor node.

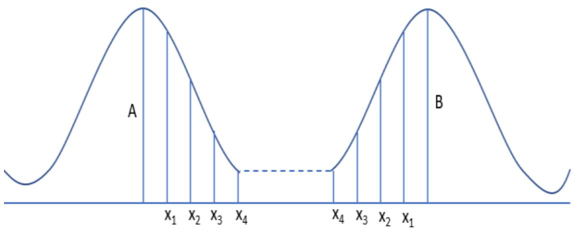

Figure 4: Superposition of wave signals form base station.

The next nearest node with respect to the farthest node is selected using the following condition. Let sensor node 1 and sensor node 2 be defined as.

$$P(x) = \frac{B_1}{{x_1}^2 + 1}$$

and

$$Q(x) = \frac{B_2}{{x_2}^2 + 1}$$

respectively.

Computation of the next nearest node is as follows.

$[h(x)]_{\ x=0} = [P(x)]_{max} + [Q(x)]_{max}]$

which is based on the superposition principle.

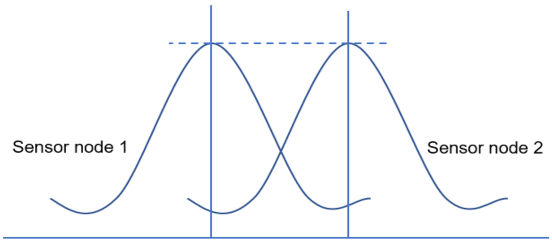

Figure 5: Superposition of wave signals from sensor nodes.

5. Implementation

Consider a two dimensional electro-magnetically sensitive space with sensor nodes or sensor clusters distributed uniformly and randomly.

These sensor nodes or cluster heads act as nodes or points to which the mobile agent has to travel in an optimal path, taking many parameters such as data, the length of the overall path, etc. into consideration.

Here an average of forty sensor nodes is considered, in the overall two-dimensional area and thus an average of ten nodes per quadrant with centre at the origin(BS), for simulation. The simulation tool used is Turtle package which is a two-dimensional mapping and drawing tool in Python.

The itinerary and the methods discussed in the results section was implemented for ten nodes and the same can be extended to the other three quadrants with negative and positive values of x and y coordinates accordingly and thus cover all forty nodes.

The three methodologies implemented and for which the graphical results are shown are.

a. An implementation of static-static case using matplotlib is carried out, where the sensors are static; here the mobile agent moves from origin (base station) to the farthest position and then it moves to the next farthest position with respect to the origin.

In this way the farthest point which is considered to emit the **weakest radiation** is reached first and the next one is reached next and so on. Here each distance is considered from origin hence it can be called as static-static case.

Here the simulation is done by simulating the sensor nodes as points taken randomly but which can be substituted by specific points and checking the distance from the origin.

The mobile agent is simulated to move from origin to the farthest point first and then the nearest point from the last reached point thus making this algorithm Farthest Node First Nearest Node Next Algorithm **(Figure 6)**.

b. The problem with the static-static approach is that the length that the mobile agent would have to traverse in order to visit and collect data from all the sensor nodes in the two-dimensional area is very high. In each quadrant one mobile agent migrates from the base station to the sensor nodes.

The mobile agent first goes to the farthest point from the origin and then the second farthest and so on based on the distance from the origin. This makes the mobile agent criss-cross its own path many a times and hence not only increases the length of the path it has to travel but also has high energy consumption. This may also be a problem in the actual implementation as higher the distance the mobile agent travels, more is the memory required to store the data collected from many nodes **(Figure 7)**.

To optimize this path, consider the concept of multiple mobile agents[8] and cloning of agents.

Cloning[10] is the duplication of a mobile agent. This is done when it has to travel to a node which makes its path across its previous paths thus increasing the path length.

In this method, consider the distance from the previously visited node after travelling to the farthest node first. The line joining the origin and the farthest point is taken as the base or reference line and every other point is compared to this line with respect to the slope it makes with the x axis, and two sets of points are created as points to the left and right of the reference line. **(Figure 8)**. The mobile agent from the origin choses one path either to the right or to the left. This agent is cloned at the farthest point to create a new agent with all similar characteristics other than the memory which will be empty, and this new cloned agent goes to the other path. Both these agents cover the points on both sides of the line joining the origin and the farthest point and returns to the origin (base station).

This method is applied only to a set of ten random points. However, this can be extended to many points, making the cloning a recursive functionality with cloning happening at all possibilities of intersection of paths. This can also be extended to the other three quadrants.
c. The third methodology is a simulation of the case where the sensors are not static but moving in a direction with a velocity.

The sensor nodes may move in opposite directions towards each other, opposite direction away from each other or in the same directions.

The figures show the simulation of these cases **(Figures 10, 11 and 12).**

6. RESULTS

The results as depicted by graphs are shown in the following pictures.
Figure 6 depicts static-static approach in which the distances are considered from the origin.
Figure 7 depicts static-static approach in which the distances are considered from the previously visited sensor node.

Figure 8 depicts the cloning of agents which helps to save energy and optimize the length of the path travelled.

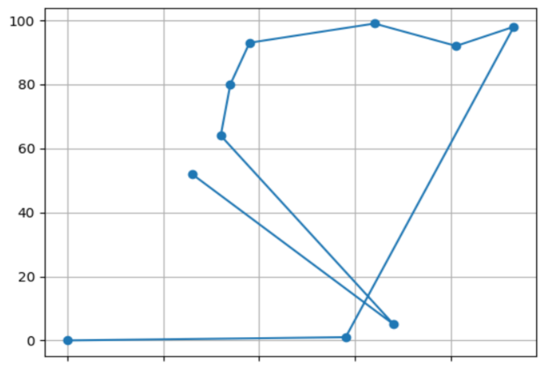

Figure 6: Static -Static approach.

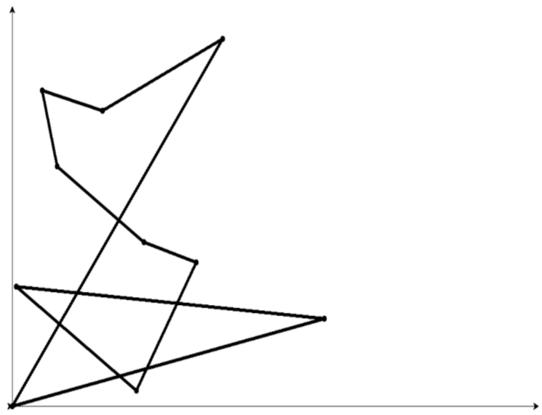

Figure 7: Static-Static approach (2).

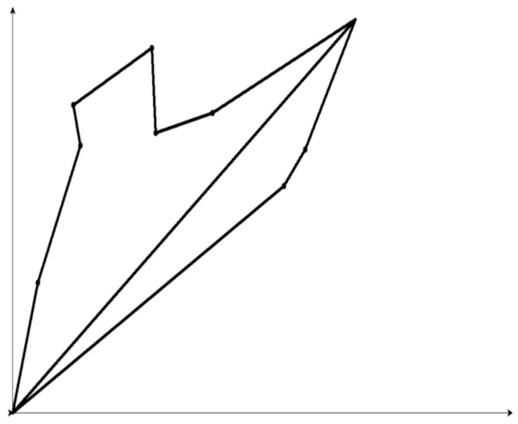

Figure 8: Cloning of Agent.

```
Sref ← Slope(farthest point from origin);
X← set of random x co-ordinates;
Y ← set of random y co-ordinates;
X1,Y1 ← points with (slope>Sref);
X2,Y2 ← points with (slope < Sref);
X11,Y11 ←nearest(X1,Y1);        // returns nearest points from the previous point in X1,Y1
X22,Y22 ←nearest(X2,Y2);        // returns nearest points from the previous point in X2,Y2
// start plotting from origin to farthest point
 Move(X11,Y11);                 // starts plotting from farthest point the points X11,Y11 and
                                   returns to origin.
Move(X22,Y22);                     // starts plotting from farthest point the points X22,Y22
and
                                   returns to origin.
```

Figure 9: Pseudo code for cloning of Mobile Agent.

Figure 9: This pseudo code will execute the cloning of the mobile agent as depicted in Figure 8.

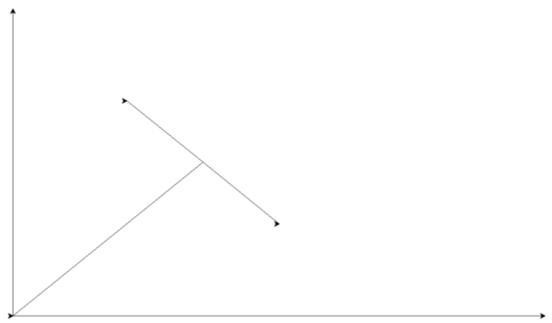

Figure 10: Velocities in opposite direction away from each other.

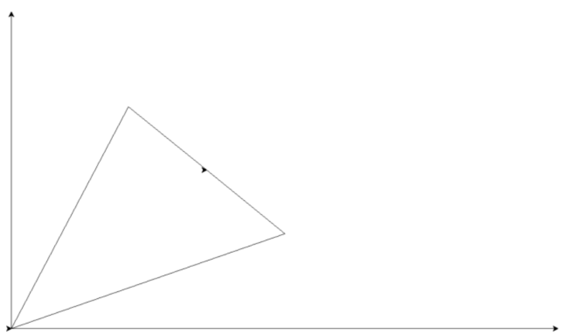

Figure 11: Velocities in opposite direction towards each other.

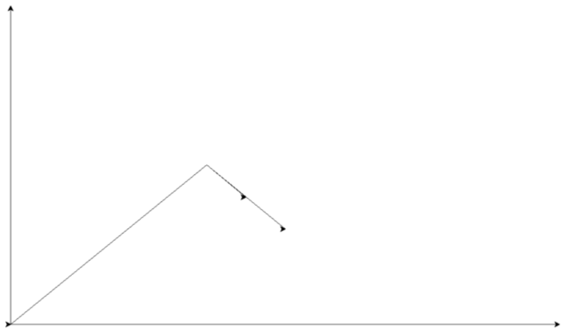

Figure 12: Velocities in the same direction.

7. Inference

From the above graphs, it can be inferred that cloning is not only efficient in solving the criss-cross problem but also reduces the energy consumed due to long path traversal.

8. CONCLUSION

Mobile agent technology plays an important role in WSN. The itinerary of mobile agent significantly contributes to the energy efficiency. Here a novel algorithm is proposed and the mathematical modelling for node placement to agent migration and synchronization. Order of Agent visit to the nodes is a significant contribution towards achieving energy efficiency. Cloning of Agent further adds to the energy efficiency by reducing the agent traversing long paths and also solves any criss-cross problems. Results show that the methods and techniques proposed here work efficiently.

Future work: Develop more realistic solutions and analyze the mathematical model using simulation tools and also to validate the algorithm.

9. REFERENCES

1. Maryam El Azhari, Nadya El Moussaid, Ahmed Toumanari, and Rachid Latif, "Equalized energy consumption in wireless body area networks for a prolonged network lifetime," Hindawi, Wireless Communications and Mobile Computing Volume 2017, Article ID 4157858 IbnZohr University, Agadir, Morocco.
2. Lingaraj, K., and Aradhana, D., “A survey on mobile agent itinerary planning in wireless sensor networks,” International Journal of Computer & Communication Technology ISSN (PRINT): 0975 - 7449, Volume-3, Issue-6, 7, 8, 2012; pp.51-56.
3. Huthiafa, Q., Qadori, Zuriati A., Zulkarnain, ZurinaMohdHanapi, and Shamala Subramaniam, “Multi-mobile agent itinerary planning algorithms for data gathering in

wireless sensor networks: A review paper," International Journal of Distributed Sensor Networks, 2017; Vol. 13(2).

4. Prapulla S B, Jayanth Chandra, Madhwesh B Mudakavi, Shobha G, Thanuja T C, "Multi Mobile Agent itinerary planning using Farthest Node First Nearest Node Next (FNFNNN) technique", 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore.
5. Prapulla S B, Trisha Venkatesh, Jayanth Chandra, Sindhu B Dinesh, Vindhya Nagaraj, "Mobile Agent Data Security using Triple Data Encryption Standard", International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 22 (2017) pp. 11824- 11829.
6. Ioannis E. Venetis, GrammatiPantziou, DamianosGavalas, Charalampos Konstantopoulos "Benchmarking Mobile Agent Itinerary Planning Algorithms for Data Aggregation on WSNs".
7. Erhan Baran, Route Determination for Capacitated Vehicle Routing Problem with Two Different Hybrid Heuristic Algorithm, International Journal of Engineering Science and applications, Baranet al, June 2018; 2(2).
8. M Chen, W Cai, S Gonzalez, VCM Leung, in Ad Hoc Networks, ed. by J Zheng, D Simplot-Ryl, and VCM Leung. Balanced Itinerary Planning for Multiple Mobile Agents in Wireless Sensor Networks, (SpringerBerlin, 2010); pp. 416–28.
9. Onn Shehory, Katia Sycara, Prasad Chalasani, and Somesh Jha Carnegie Mellon University "Agent Cloning: An Approach to gent Mobility and Resource Allocation" IEEE Communications Magazine, July 1998; PP 58-67.
10. El Fissaoui, Mohamed Beni-hssane, AbderrahimSaadi, Mostafa,"Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks", EURASIP Journal on Wireless Communications and Networking, 2018; Volume 1: PP 1687-1499.
11. Geetika Dhand and S.S.Tyagi "Data aggregation techniques in WSN: Survey", in 2nd International Conference on Intelligent Computing, Communication & Convergence (ICCC-2016) by Interscience Institute of Management and Technology Bhubaneswar, Odisha, India.