ISLAMIC UNIVERSITY OF TECHNOLOGY

UNDER GRADUATE THESIS

A Cellular Automata Model for Object Monitoring in Mobile Wireless Sensor Network in Hexagonal Grid

---

Authors:

Shahidullah Kaiser Dipu(134420)

Md. Rahat Hussein Khan(134426)

Supervisor:

Md. Sakhawat Hossen

Assistant Professor, Department of CSE

Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE in fulfilment of the requirements for the Degree of B.Sc Engineering in CSE.

Academic Year: 2016-17.

November, 2017

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and investigation carried out by Shahidullah Kaiser Dipu and Md. Rahat Hussein Khan under the supervision of Md. Sakhawat Hossen in the Department of Computer Science and Engineering (CSE), IUT, Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Shahidullah Kaiser Dipu
Student ID - 134420

Md. Rahat Hussein Khan
Student ID – 134426

Supervisor:

Md. Sakhawat Hossen
Assistant Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Abstract

B.Sc in Computer Science and Engineering

A Cellular Automata Model for Object Monitoring in Mobile Wireless Sensor Network in Hexagonal Grid By Shahidullah Kaiser (134420), Md. Rahat Hussein Khan (134426)

Mobile Wireless Sensor Networks (MWSN) is an ad-hoc network that comprises of a large number of sensors. The sensors usually have limited sensing and communication capabilities. We present a CA model that efficiently monitor a moving object in distributed mobile wireless sensor network. CA is a biologically inspired discrete model. Although there are works that have used CA for developing Object Monitoring in Distributed Mobile Wireless Sensor Network, they mostly used square grids which is not a good representation of the actual MWSN model. We focus on simulating the mobile wireless sensor networks for monitoring a moving object considering energy-efficiency. In a general case, a number of objects need to be monitored by the sensors and the objects can be static or mobile. Monitoring mobile objects is considerably harder than monitoring static objects. Mobile sensor networks can be useful in monitoring animal behavior. In this case, animals can move randomly within a network and the mobile sensors can also move to monitor their behavior and other aspects.

Acknowledgements

In full gratitude, we would like to acknowledge the following individuals who encouraged, inspired, supported, assisted, and sacrificed themselves to help our pursuit of the successful thesis work.

From our academy, we would like to thanks Md. Sakhawat Hossen, Assistant Professor, CSE, IUT for the continuous support of our thesis work and related research, for his patience, motivation, and immense knowledge. His guidance helped us in all the time of research and writing of this thesis.

We would also like to thanks Ahnaf Munir, Lecturer, CSE, IUT for his insightful comments and encouragement, but also for the hard question which incented us to widen our research from various perspectives.

Last but not the least, we would like to thank our families for supporting us spiritually throughout the writing of this thesis and our life in general.

With Regards,
Shahidullah Kaiser Dipu (134420)
Md. Rahat Hussein Khan (134426)

Table of Contents

LIST OF FIGURES

LIST OF TABLE

Chapter 1

Introduction

1.1 Motivation

Wireless sensor networks (WSNs) play important roles in a wide range of applications such as industrial control and monitoring, security, military sensing. A wireless sensor usually consists of a processor, memory unit and power supply, and is therefore limited in its processing power, memory and transmission range. A sensor can sense, measure and gather information from the environment. It can also communicate with other sensors, but only within its limited range. A WSN has a limited lifespan since sensor batteries offer finite energy reserves.

One of the important optimization problems in an MWSN is to maximize the coverage of the network while maintaining connectivity. The sensors in a MWSN have movement capabilities. They are deployed in a certain area and they need to move around to increase the coverage of the network while maintaining connectivity among themselves.

The movement of a sensor requires significant amount of energy. Since the sensors have limited power this movement should be restricted as much as possible in order to save energy for sensing and communication. So a balance must be maintained among coverage, connectivity and movement.

We proposed a cellular automaton based algorithm for the mobile object monitoring problem in a mobile wireless sensor network where both the objects and the sensors can move and our main goal is to monitor the objects as long as possible.

A CA consists of a regular grid of cells, each of which is in one of a finite number of states. Each cell has a state and a set of neighboring cells called *neighbors* around itself. The state of each cell changes depending on the states of neighboring cells. Many researchers have focused on applications of cellular automata to address phenomena caused by local interactions.

The measurement for evaluating the proposed model is the percentage of correctly monitored objects in a given region of interest. In Section 2 related works are described. In section 3 we outline the basic concepts of cellular automata.

1.2 Thesis Contribution

The contribution of this thesis is as follows:

1. Object Monitoring using CA has been done in Square Grid [1] by checking the number of objects and sensors available in the neighborhood. But the coverage is not sufficient as all the mobile dispersion algorithms that uses CA were developed for square grid. The square grid is not a good representation of the actual sensors as compared to hexagonal grid. So, we implemented Object Monitoring algorithm in hexagonal grid.

2. Motion Planning Algorithm using Hexagonal grid has been implemented in [2]. Coverage has been increased using different Movement Algorithm. We implement both Motion planning algorithm and Object Monitoring algorithm in Hexagonal grid and compare the result with Monitoring algorithm. Monitoring Algorithm showed better result than Motion planning algorithm.

1.3 Thesis Outline

The paper is organized as follows: Chapter 2 provides a short overview of MWSN and CA; Chapter 3 discusses the state of the art in the field; Chapter 4 discusses Monitoring algorithms; Chapter 5 shows the results of Monitoring algorithms; and Chapter 6 concludes the paper and presents future work.

Chapter 2

Background

2.1 Mobile Wireless Sensor Network (MWSN)

Mobile Wireless Mobile Networks (MWSNs) are distributed ad-hoc networks of nodes that can sense, actuate, compute and communicate with each other using point-to-point, multi-hop communication. This kind of network may consist of hundreds to thousands of sensor nodes that have the capability of sensing, processing and communicating using a wireless medium. Nodes in such networks include mobile sensors, robots, and even humans. Such systems combine the most advanced concepts in perception, communication and control to create computational systems capable of large-scale interaction with the environment, extending the individual capabilities of each network component to encompass a much wider area and range of data.

Due to the absence of any networking infrastructure the nodes must cooperate to accomplish communication, global control and distributed information aggregation. MWSNs aims to collect data and sometimes control an environment.

The establishment of a mobile wireless sensor network is illustrated in Figure 2.1. Initially, the sensor nodes are deployed over an area of interest as illustrated in Figures 2.1(a) and 2.1(b). The node deployment can be done, for example, by dropping a large number of sensor nodes from an airplane in a certain area, or placing them in this area by hand or using a robot. They are able to discover their locations (see Figure 2.1 (c)) and organize themselves as a wireless network (see Figure 2.1 (d)). A MWSN must be able to operate under very dynamic conditions.

Once the network is formed and the sensor nodes are operating, most sensor nodes will be able to sustain a steady state of operation, where energy reservoirs will be nearly full, and they will be able to support all the sensing, processing, and communication tasks required. In this mode, sensor nodes will constitute a multi-hop network. The sensor nodes begin to establish routes by which information passed to one or more sink nodes.

Sink nodes are typical sensor nodes that usually differ from other types of sensor nodes in the following aspects: they have more energy, longer radio range and do not perform sensing. When there are structural differences among the nodes we say that the network is heterogeneous. In this work, we just address the homogeneous networks (that is, we consider that all nodes have the same structure and properties).

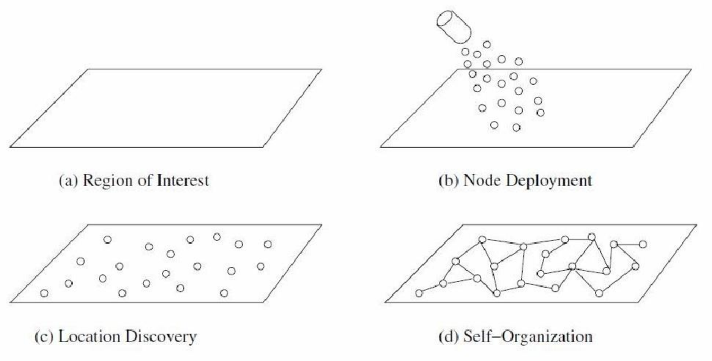

FIGURE 2.1: Creation of a MWSN

2.2 Cellular Automata (CA)

A CA is a biologically inspired model that has been widely used to model different physical systems. There is an advantage of the cellular automaton model.The CA algorithms basically rely only on a small amount of local information.These local information can easily be stored in individual nodes. A cellular automaton consists of a regular grid of cells.Each cell of the grid is in one of a finite number of states.The grid can be in any finite number of dimensions.For each cell, a set of cells are called its neighborhood is defined relative to the specified cell.For each Cell, an initial state (time t = 0) is selected by assigning a state. A new generation is created increasing t by 1 (t+1), according to some fixed rule and it determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood. Typically, the rule for updating the state of cells is the same for each cell and does not change over time, and is applied to the whole grid simultaneously.

2.2.1 Formal Definition

CA can be defined as a 4-tuple (L,S,N,f) where:

- L is the regular grid. The elements that compose the grid are called cells
- S is the finite set of possible states of the cells
- N is the finite set of neighborhood of the cell.
- f is the transition rule of the automaton.

So, ∀ c ∈ N and ∀ r ∈ L : r+c ∈ L and f : S^n →S is a transition function[5].
At time t, each cell of L is assigned a state of S. The state of a cell c L , at time t+1, is determined by the transition function f depending on the current state of c and the states of cells in the neighborhood of c at time t. In case of a square grid, if all the adjacent cells are considered as neighbors, then it is called a Moore neighborhood. We consider the neighbors for the hexagonal grid in a similar manner. The neighbors at radius 1 and radius 2 are shown for a square grid and a hexagonal grid in Figure 1(a) and 1(b) respectively.In a typical CA, all the cells synchronously verify the states of their neighbors and change their states accordingly.

The four components are discussed briefly below:

Grid

The cells in the cellular automata reside on grids. This grid can be of various types. They can be in any finite number of dimensions. The shape can me rectangular, triangular, hexagonal etc. The grid can have either finite or infinite size.

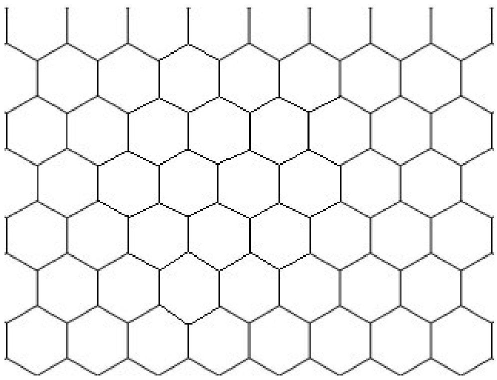

FIGURE 2.2: Hexagonal Grid

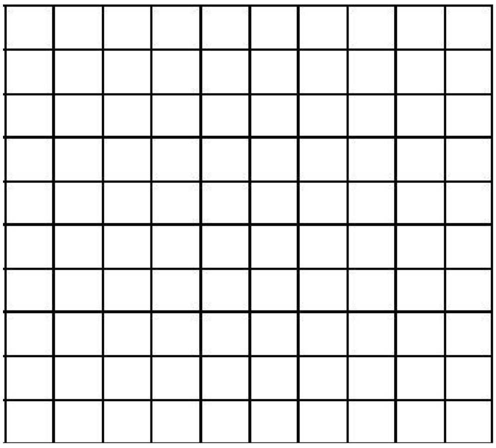

FIGURE 2.3: Square Grid

Generally, a grid is called regular if it is a coverage of d-dimensional space, that is, the elements of the grid (cells) together complete the d-dimensional space entirely, and the translation of the grid in d independent directions results in the grid itself.

State

A finite set of states is associated with any cellular automata. The cells can be in any of these states. The simplest set of states are the on and off states or 0 and 1. In every time step each of the cell changes or retains its state by consulting the rules that have been defined for that particular automata. The rules usually consider the state of the cell and also of those that are its neighbors (discussed below) in the previous time step.

Transition Function

The transition function is responsible for changing the states of the cells. The state of a cell C, at time $t + 1$, is determined via the transition function δ depending on the current state of c and the states of cells in the neighborhood of c at time t.

Neighbourhood

The neighbors of a cell are the cells that are considered to be adjacent to it. The number of adjacent cells that are considered neighbor depends on that particular cellular automata. In case of a square grid if all the adjacent cells (maximum 8) of a cell are considered as the neighbors, then this is called a Moore neighborhood [9].

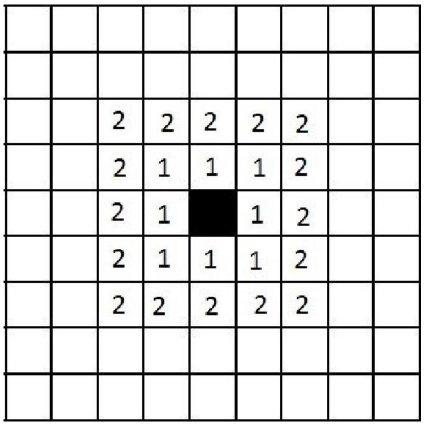

FIGURE 2.4: Neighbors of the central cell in square grid

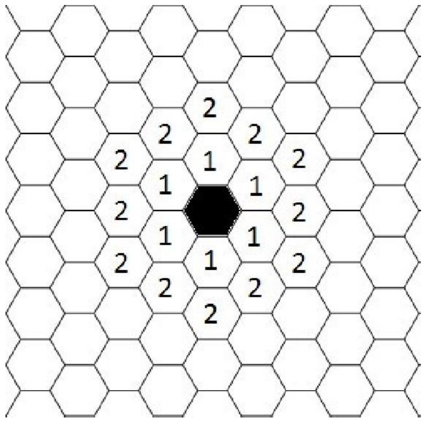

FIGURE 2.5: Neighbors of the central cell in hexagonal grid

In figures 2.4 and 2.5 the neighbors of the central black cell are shown. The number within the cells denote the distance of the neighbors from the central
Cell

2.3 Range of a Sensor

Each sensor in a MWSN has two kind of range associated with it; the communication range and the sensing range.

The communication range of a sensor is the area surrounding the sensor within which the sensor can communicate with other sensors. So if any sensor is located within the communication range of a sensor then the two sensors can communicate with each other. We represent communication range by R_c.

The communication range of a sensor is the area surrounding the sensor within which it can sense the environment. So a sensor can sense the environmental area that falls within its sensing range. We represent communication range by R_s.

Chapter 3

Literature Review

Our work is based on the Salimur Choudhury Cellular Automaton Based Localized Algorithms for Mobile Sensor Networks [1]. Motion Planning Algorithm using Hexagonal grid has been implemented in [2].

A set of mobile sensors ($m_1, m_2, ..., m_n$) are deployed in a two dimensional sensor network. A number of mobile objects ($o_1, o, ..., o_k$) are also deployed in the network. Typically we consider situations where $k < n$. The objects can move randomly within the network which has a bounded area and the mobile sensors try to move accordingly to monitor them as long as possible. We consider the movement speed of the mobile objects to be higher than the speed of the mobile sensors, because otherwise, the problem becomes trivial. We consider that the mobile sensors can move one cell in one time unit from their current cell using the algorithm. They can move diagonally too whereas the mobile objects choose any location within a radius-2 neighborhood in random and can move there. So the speed of the mobile objects can be twice compared to the mobile sensors. Initially all the sensors and objects are deployed densely within the network where each object is monitored by at least one of the sensors. However, as soon as the objects start moving they are not guaranteed to be monitored if the sensors do not move accordingly. The sensors use the following CA based movement algorithm in order to maximize the number of objects monitored and the length of the time they are monitored.

Neighborhood: A sensor monitors objects that are within its sensing radius. A sensor communicates with other sensors within its communication radius and uses the information obtained to determine the movement direction.

State Representation: The state of a cell gives the number of mobile sensors located in that cell and the locations and positions of the objects monitored by sensors in the cell. Thus, when the sensing radius is one, the state of a cell *C,* is a 8-tuple ($l, o_1, . . . , o_7$), where l is the number of sensors in cell C and o_i , i = 1, . . . ,7 give the numbers of objects in each of the cells with radius one from cell C. Note that if the state would simply store the number of objects within sensing radius from C, when a sensor m communicates with two different cells $C1$ and $C2$ with overlapping sensing radii, the sensor *m* would have no way of knowing how many of the objects within the sensing radii of *C*1 and *C*2, respectively, are "in common". For this reason the state of a cell needs to store the precise location of each object within its sensing radius.

Movement Rules: The movement rule for each of the sensors is described below for the X-axis. Similar rules are applicable for the Y-axis. This algorithm has two phases. In the first phase, a sensor computes its future location and in the second phase it decides whether it should move to that location or not.

First Phase:

1. For the X-axis, each sensor counts the number of mobile sensors in the positive and negative directions within the $\boldsymbol{R_c}$ radius. It also counts the number of objects which is the sum of the number of objects the sensor finds in its sensing radius and the number of mobile objects its neighbors (those that are within its communication range) find in their respective sensing radii. Note that, an object can be sensed by multiple sensors but it is counted only once.

2. If the sensor does not find any object in one of these two directions then it moves towards that direction that has at least one object.

3. If both directions have no object then the sensor applies the following rules:

   a. It moves towards that direction that has fewer sensors.
   b. If there is no mobile sensor in both directions or the directions have the same number of sensors then it does not move.

   In this case, even though a sensor does not see any object in its communication radius but it wants to move in the hope that in the future it can detect and monitor the objects. Moreover, it does not need to stay in the current position as there are other sensors already there to monitor the shared region.

4. If a sensor finds that both directions have at least one object then it calculates the following ratio for the positive X direction.

$$\boldsymbol{Ratio_{xPOS}} = \boldsymbol{n_{xpos}/k_{xpos}}$$

Where
$\boldsymbol{n_{xpos}}$= total number of neighboring sensors the sensor within R_c

$\boldsymbol{k_{xpos}}$=total number of objects sensed by the sensor and its neighbors in the positive X-direction.

The sensor decides to move toward that direction that has the smaller ratio.

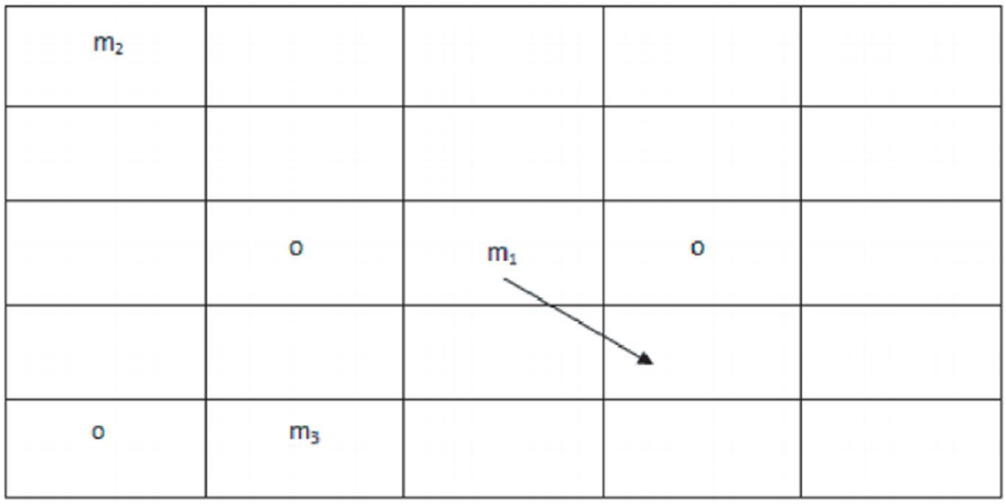

Fig 2.6 : An Example of Object Monitoring Problem with R_c =2 and R_s=1

Second Phase: Now the sensor moves to the location it has already chosen above if the location is empty. If the location is not empty then the sensor does not need to go there as there is already at least one sensor to monitor the objects of its R_s neighborhood. However, this rule does not totally eliminate the chance of having multiple sensors in a cell.

If any cell has more than one sensor at a given time then only one sensor applies the above mentioned algorithm and moves if it satisfies the rules. If the sensor does not satisfy any rule that allows it to move, then the sensor randomly chooses any location from its radius 1 neighborhood and move to that location. Other sensors do not move at that time period.

Chapter 4

Proposed Method

The CA model used in [1] uses a square grid for their algorithm. Our goal is to develop a similar Object monitoring algorithm that can be applied in hexagonal grids. We intend to achieve this goal through the following steps:

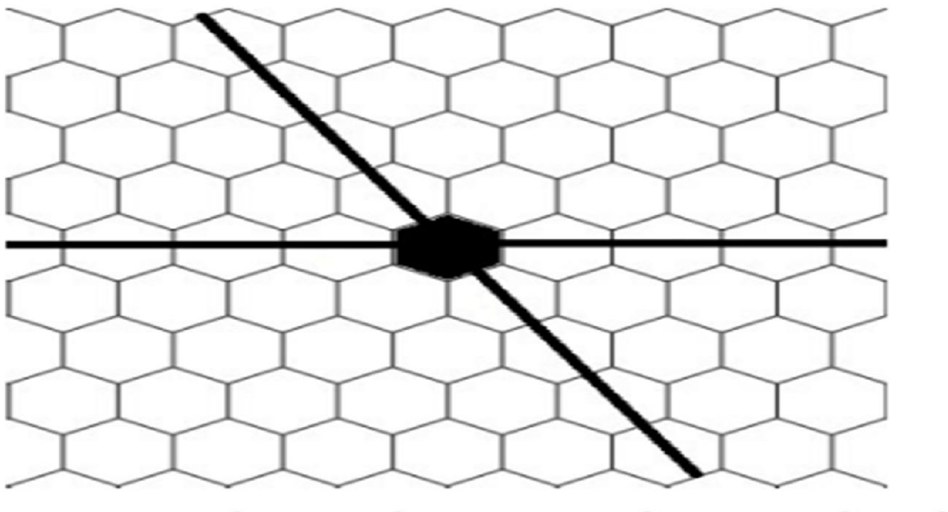

Fig : 2.7 The x and y axes in a hexagonal grid

The axis in hexagonal grid is different from square grid. The algorithm is implemented in hexagonal grid follow all the movement rules like Square Grid. But there are few variation in selecting the directions. Both sensor and Object will be move in different direction.

i. **Object Monitoring and Movement Rules**: The movement rule for each of the sensors is described below for the *X* axis. Similar rules are applicable for the *Y* axis. This algorithm has two phases. In the first phase, a sensor computes its future location. and in the second phase it decides whether it should move to that location or not.

First Phase:

- For the *X* axis, each sensor counts the number of mobile sensors in the positive and negative directions within the R_c radius. It also counts the number of objects which is the sum of the number of objects the sensor finds in its sensing radius and the number of mobile objects its neighbors (those that are within its communication range) find in their respective sensing radii. Note that, an object can be sensed by multiple sensors but it is counted only once.
- If the sensor does not find any object in one of these two directions then it moves towards that direction that has at least one object. If both directions have no object then the sensor applies the following rules.
- It moves towards that direction that has fewer sensors.
- If there is no mobile sensor in both directions or the directions have the same number of sensors then it does not move. In this case, even though a sensor does not see any object in its communication radius but it wants to move in the hope that in the future it can detect and monitor the objects. Moreover, it does not need to stay in the current position as there are other sensors already there to monitor the shared region.

- If a sensor finds that both directions have at least one object then it Calculates the following ratio for the positive *X* direction.

$$Ratio_{xPOS} = n_{xpos}/k_{xpos}$$

Where

n_{xpos} is the total number of neighboring sensors the sensor sees within distance R_c in the positive *x*-direction and k_{xpos} is the total number of objects sensed by the sensor and its neighbors in the positive *x*-direction.

- It calculates $ratio_{xneg}$ similarly for the negative *X* direction.
- Now the sensor decides to move toward that direction that has the smaller ratio. The idea is to move towards that direction where the objects have a smaller number of sensors on average to monitor them.
- If the ratios are equal then the sensor does not move.

Second Phase:

Now the sensor moves to the location it has already chosen above if the location is empty. If the location is not empty, then the sensor does not need to go there as there is already at least one sensor to monitor the objects of its R_s neighborhood. However, this rule does not totally eliminate the chance of having multiple sensors in a cell.

If any cell has more than one sensor at a given time, then only one sensor applies the above mentioned algorithm and moves if it satisfies the rules. If the sensor does not satisfy any rule that allows it to move, then the sensor randomly chooses any location from its radius 1 neighborhood and move to that location. Other sensors do not move at that time period.

Chapter 5
Simulation environment and results

5.1 Simulator

To test our CA-based algorithms we developed our own simulator. The codes of the simulator were written on Python. The simulator determines the position of each of the deployed nodes and presents a graphical representation of the network for each time step. It also calculates the performance metrics which allows us to compare the performance of the different algorithms.

5.2 Simulation Environment

The simulations use different grid sizes, different numbers of mobile sensors, and different numbers of mobile objects. We consider $R_c = 3$ or $R_c = 2$. In all the cases, the sensing radius $Rs = 1$.

We show the comparison between Rc=2 and Rc=3 for Object Monitor Algorithm. We also compare our result with dispersion algorithm. For this comparison we consider Rc=3.

Initially we deploy different numbers of mobile sensors and the objects in a hexagonal grid. Then we also deploy Object randomly. The object direction random which is based on probability.

For each sensor and object, we randomly pick a location from this block where it is deployed. Note that, at any given time, a cell can have more than one sensor or more than one object. We also specify that in the initial deployment each cell cannot have more than three sensors and each cell can have a maximum of five objects. However, there is no such bound after the deployment.

Fig: 5.1 shows initial deployment and Fig: 5.2 shows the scenario after 50 time step of the sensors. Here the Red dot are sensors and Blue dot are Objects.

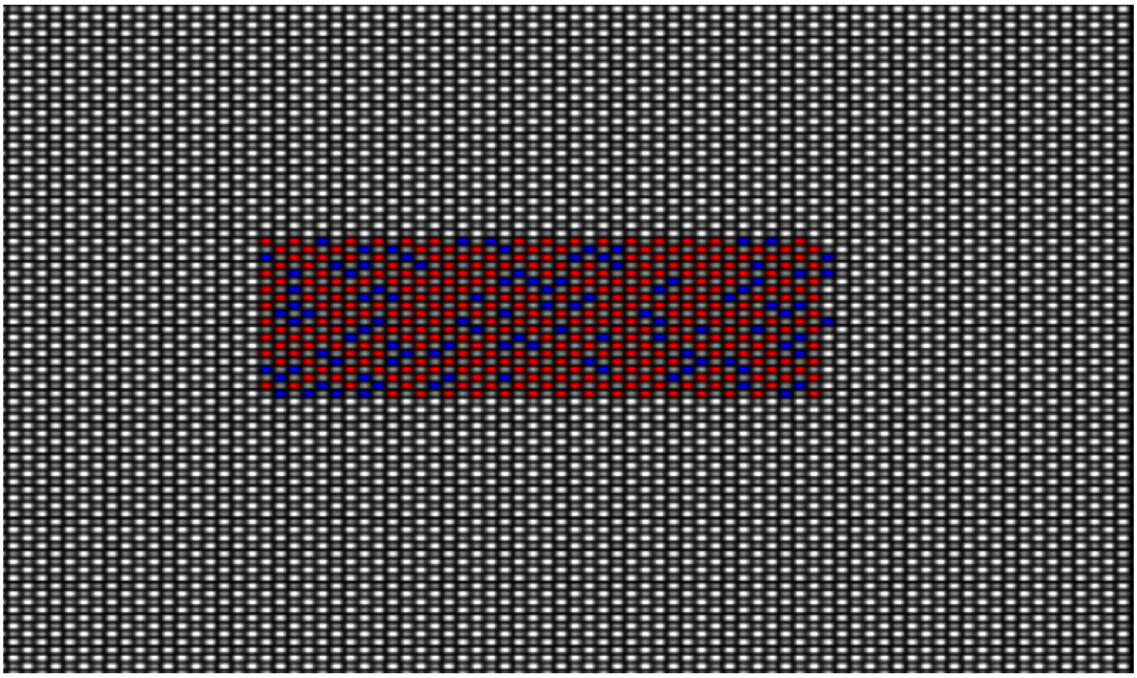

Fig 5.1: Initial Deployment of Sensor and Object

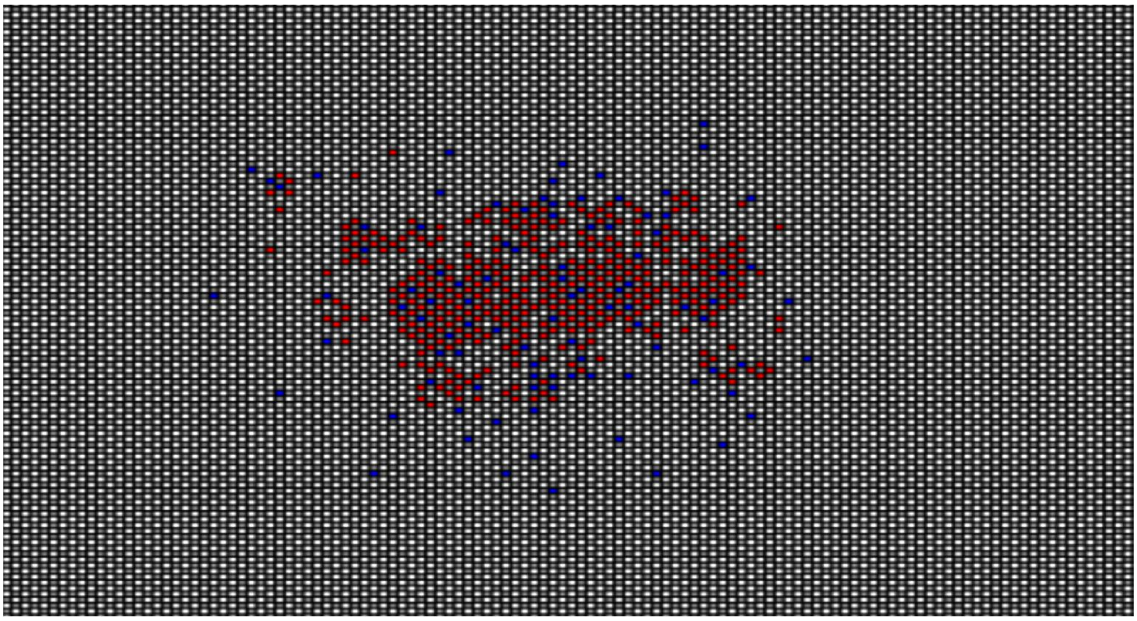

Fig 5.2: After 50 Time Steps Sensor and Object

An object is considered to be monitored if it is in the sensing radius of at least one mobile sensor. Initially, typically, all the mobile objects (100%) are monitored by the mobile sensors. However, when the objects start moving this percentage gets lower but after a certain period of time (time depending on the simulation parameters, for example, grid size and the number of mobile sensors) this number does not change much.

Please note that designing a local algorithm that provides 100% monitoring is quite difficult to achieve, and when the grid size is large compared to the number of sensors achieving monitoring of all the objects is obviously impossible. We are interested to get a competitive percentage that can be applicable in different real life applications.

5.3 Performance Metrics

1) Calculating the Percentage of Monitored Objects with respect to time in hexagonal environment.
2) Compare the Result with Dispersion Algorithm.

5.4 Performance Evaluation

The graphs in Figure 5.3 and 5.4 below showing the time step vs monitored object in Hexagonal grid. Here we considered Rs=1, 100 objects, 400 sensors and a grid size of 150×150.

This result in Fig: 5.3 shows that the monitored algorithm have consistency of 40%. It doesn't go below 40% .As the object speed and direction is random, some object may go outside the Boundary and they bare difficult to monitor. In that case, we stopped the simulation if any sensor goes near grid size and hit the boundary.

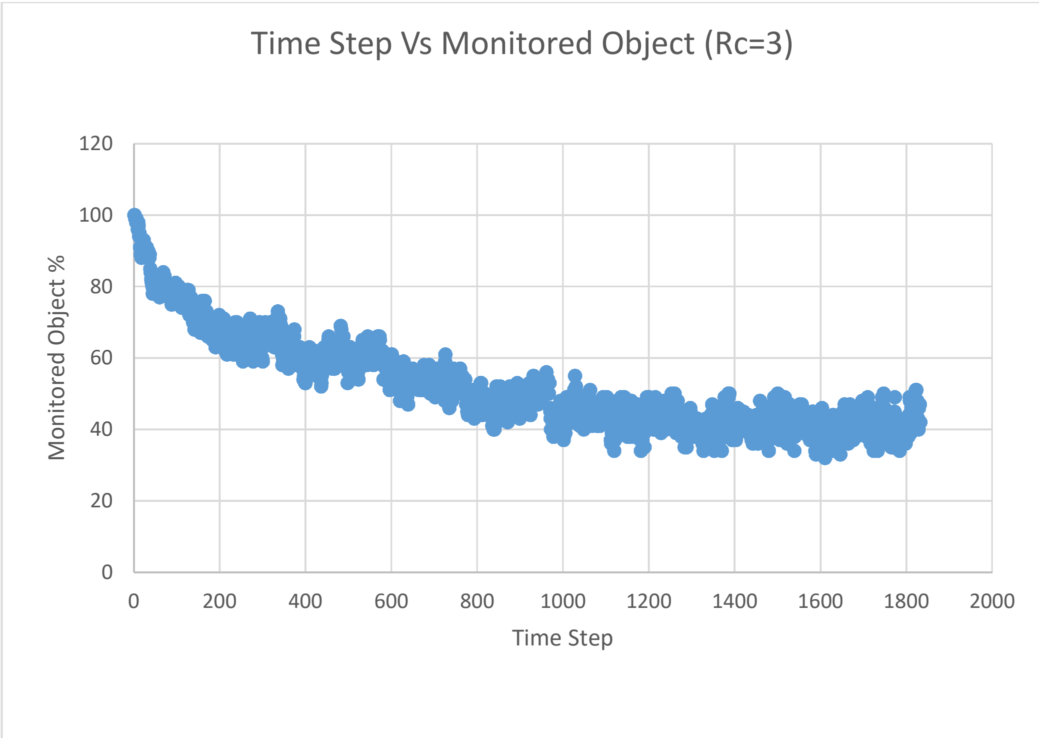

Fig 5.3: Time Step Vs Monitored Object when Rc=3

We also consider the Monitor Algorithm for Rc=2. We run the Simulation for 650 time steps .In that case, less objects were monitored as the number of neighbor decreases.

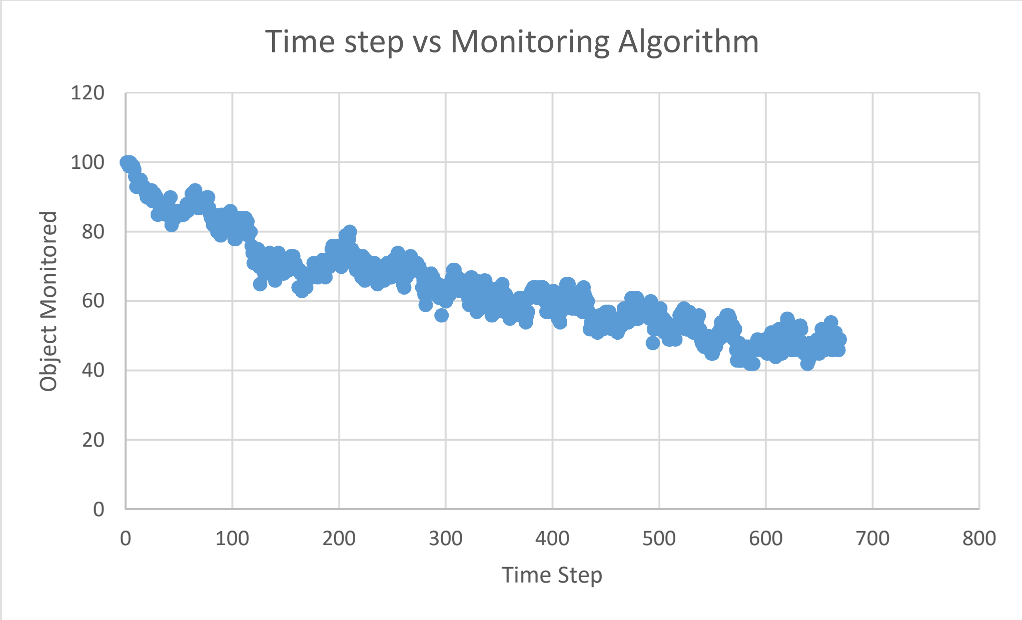

Fig 5.4: Time Step Vs Monitored Object when Rc=2

Now we evaluate Monitor Algorithm by comparing between Rc=2 (Series-1) and Rc=3(Series-2) using Grid Size=100 .Fig: 5.5 and 5.6 show this comparison is given below.

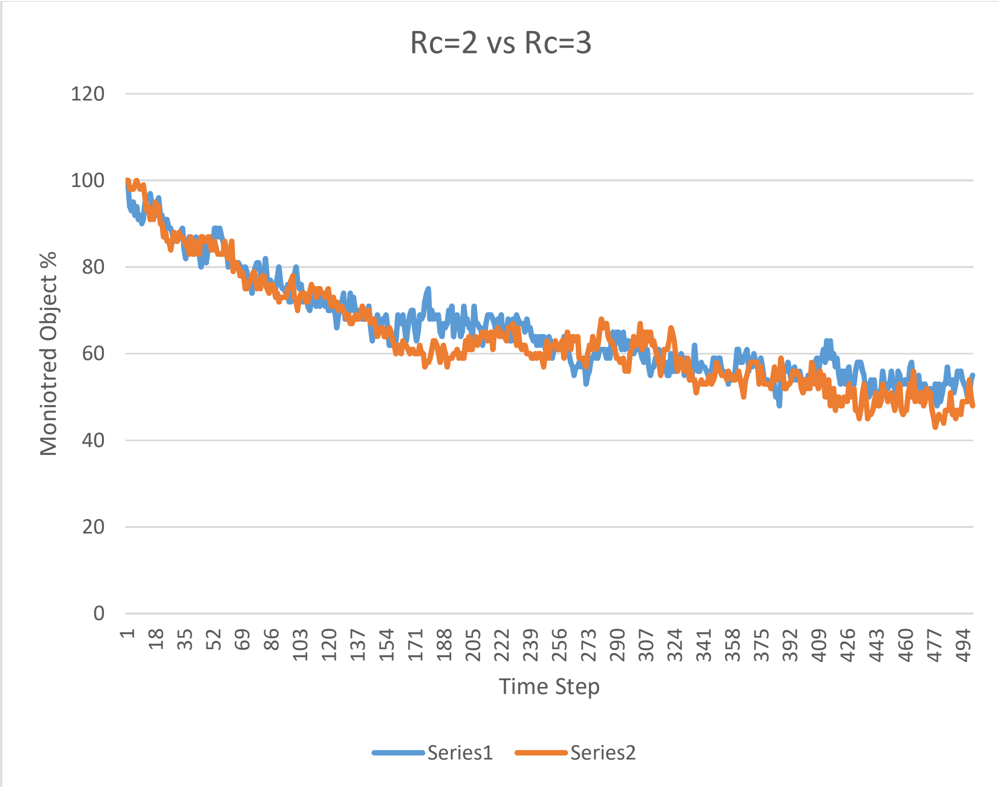

Fig 5.5: Rc=2(Series1) and Rc=3(series2) of Monitoring Algorithm

In table 5.1 we show comparison between different Grid Size, Rc, Number of sensors. In table 5.2 we will show the result between Monitor Algorithm and Motion Planning Algorithm in Hexagonal.

For the average, we pick the data between time step 200 and 300. This is where the Algorithm Shows consistency.

TABLE 5.1: Results for Monitor Algorithm using different parameter

Grid Size	Number of Sensors	Monitoring Algorithm (Rc=2)	Monitoring Algorithm (Rc=3)
100x100	200	58%	62%
100x100	300	63%	66%
125x125	300	58%	65%
125x125	400	63%	70%
150x 150	300	53%	59%
150x 150	400	63%	69%

Table 5.2 Show the comparison result between Monitor algorithm and Dispersion Algorithm

Grid Size	Number of Sensors	Monitoring Algorithm (Rc=3)	Dispersion Agorithm (Rc=3)
100x100	200	62%	20%
100x100	300	66%	22%
125x125	300	65%	20%
125x125	400	70%	21%
150x 150	300	59%	16%
150x 150	400	69%	19%

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We modified a cellular automaton based algorithm for the mobile object monitoring problem in a mobile wireless sensor network on a hexagonal grid, where both the objects and the sensors can move and our main goal is to monitor the objects as long as possible.

The simulation results indicated that our proposed algorithm worked well in practice and could monitor a good number of mobile objects constantly.

6.2 Future Work

In future work we plan to modify the algorithm in order to increase the efficiency. Moreover, we will consider the energy constraint while monitoring the object. Sleep-Awake schedule will be implemented in order to increase the lifetime of the network. Other optimization problems of mobile wireless sensor networks, such as, for example, tracking the locations of the mobile objects, gathering sensors in different geometric shapes, and so on will also be considered.

Bibliography

[1] SALIMUR CHOUDHURY, KAI SALOMAA, SELIM G. AKL .Cellular Automaton Based Localized Algorithms for Mobile Sensor Networks, *Int. Journ. of Unconventional Computing*

[2] Salimur Choudhury, Sakhawat Hossen, Muhammad Mahbub Alam, Ahnaf Munir, Shihab Uzzaman. Localized Motion Planning algorithm for Mobile Wireless Sensor Networks, *Int. Journ. of Unconventional Computing*

[3] S. Choudhury, S. G. Akl, and K. Salomaa. Energy efficient cellular automaton based algorithms for mobile wireless sensor networks. In *Proceedings of the 2012 IEEE Wireless Communications and Networking Conference*, WCNC'12, pages 2341–2346, 2012.

[4] S. Choudhury, K. Salomaa, and S. G. Akl. A cellular automaton model for wireless sensor networks. *Journal of Celluar Automata*, 7(3):223–241, 2012. (jca)

[5] S. Wolfram. A new kind of science. Wolfram Media, Champaign, Ill, 2002.