Improvised Greedy Algorithm of Sensors Scheduling for Target Coverage in Wireless Sensor Networks

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

by

Saurabh Kumar

(Roll No: 213CS1145)

under the guidance of

Dr. Bibhudatta Sahoo

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Odisha, India
June, 2015.

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled " ***Improvised Greedy Algorithm of Sensors Scheduling for Target Coverage in Wireless Sensor Networks*** " submitted by ***Saurabh Kumar*** is a record of an original research work carried out by him under our supervision and guidance in partial fulfilment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering, National Institute of Technology, Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Dr. Bibhudatta Sahoo
Assistant Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 28 - 05 - 2015

Abstract

Wireless Sensor Networks (WSNs) have many fields of application, including industrial, environmental, military, health and home domains. Monitoring a given zone is one of the main goals of this technology. This consists in deploying sensor nodes in order to detect any event occurring in the zone of interest considered and report this event to the sink. The monitoring task can vary depending on the application domain concerned. In the industrial domain, the fast and easy deployment of wireless sensor nodes allows a better monitoring of the area of interest in temporary work sites. This deployment must be able to cope with obstacles and be energy efficient in order to maximize the network lifetime. If the deployment is made after a disaster, it will operate in an unfriendly environment that is discovered dynamically. The lifetime maximization in sensors network with target coverage can be explained by these statements: How to find the maximum number of sets from all sensors such that each set can cover all the target at any particular instant of time, and then schedule those sets to be active and sleep, so that this arrangement can maximize the lifetime of the network. In this research we have discussed a greedy algorithm that produce maximum number of disjoint sets of the sensors, such that each sensor set is a set-cover.

Contents

List of Figures 6

1 Introduction 8

1.1 Introduction: Wireless Sensor Networks . . . 8
1.2 Background of Sensor Network Technology . . . 9
1.3 Components . . . 11
1.4 Applications . . . 13
1.5 Wireless Sensor Networks Architecture . . . 14
1.5.1 Layered Architecture . . . 14
1.5.2 Cluster Architecture . . . 15
1.6 Wireless Sensor Device . . . 15
1.7 Issues in Wireless Sensor Networks . . . 17
1.8 Research Topics in WSNs . . . 22
1.9 Motivation . . . 24
1.10 Area of Work . . . 25
1.11 Organization of Thesis . . . 26

List of Acronyms 8

2 Various Methods and Literature Survey 27

2.1 Terms and Definitions . . . 27
2.2 Example: Consider a set of Targets and Sensors . . . 29
2.3 Literature Survey . . . 31
2.3.1 Coverage Problem . . . 31
2.3.2 Connectivity . . . 37
2.3.3 Deployment algorithms . . . 39
2.3.4 Observation of Deployment Algorithms . . . 53
2.3.5 Point/Targets coverage and connectivity algorithms . . . 54
2.4 Various Algorithms for Lifetime Maximization . . . 57

2.5 Approaches to Complete Target Coverage and Maximizes Lifetime of the Network 59
2.6 Our Approach 60

3 **Greedy Method for Set-Cover Generation** **63**
3.1 Introduction to Greedy Algorithms 63
3.1.1 Characteristics and Features of Problems solved by Greedy Algorithms 64
3.1.2 Structure Greedy Algorithm 64
3.2 Types of Greedy Algorithms for Set Cover generation 65
3.3 Simple Greedy Set-cover Generation 65
3.3.1 Introduction 65
3.3.2 Assumptions 65
3.3.3 Set-Cover Generation 66
3.3.4 Greedy-Sensors Set Covers Generation 66
3.3.5 Performance Evaluation 67
3.4 Greedy MSSC (Maximum Sensor Set-Cover Generation) 69
3.4.1 Introduction 69
3.4.2 Performance Evaluation 70

4 **Improvised Greedy Method for Set-Cover Generation** **73**
4.1 Introduction 73
4.2 Improvised Greedy MDSC(Maximum Disjoint Set Covers) Generation Algorithm 75
4.3 Performance Evaluation 76

5 **Conclusions and Future Work** **79**

List of Figures

1.1 Components of WSNs 12
1.2 Layered Architecture of WSNs 14
1.3 Cluster Architecture of WSNs 15
1.4 Wireless Sensor Device 16
1.5 Communication medium in WSNs 18
1.6 Operating System for WSN Device 19
1.7 Deployment and Localization of WSN Device 20
1.8 Communication Protocols 21

2.1 Example of targets and sensors 29
2.2 Full Coverage 34
2.3 Partial Coverage 35
2.4 Barrier Coverage 37
2.5 Connectivity 38
2.6 Initial Disconnected Topology 42
2.7 Forces Based Strategy 47
2.8 Triangular Lattice 51
2.9 Square Grid 53
2.10 Deployment Algorithms 54
2.11 Target coverage using Annular Ferries 55
2.12 Target Coverage using Grid 56

3.1 Sensing Range = 30mtr. 67
3.2 Sensing Range = 40mtr. 68
3.3 Sensing Range = 50mtr. 68
3.4 Sensing Range = 30 71
3.5 Sensing Range = 40 71
3.6 Sensing Range = 50 72

4.1 Example 74
4.2 Sensing Range =30 77
4.3 Sensing Range =40 77
4.4 Sensing Range =50 78

Chapter 1

Introduction

1.1 Introduction: Wireless Sensor Networks

Wireless Sensor Networks is an arrangement of some specified components that can sense, compute and communicate to each other, in such a manner that can extract and observe the information from the outside environment, then environment can be a biological system, physical environment or an IT framework. Due to the recent and current research in the micro electro-mechanical system (MEMS) technology, the communication in wireless system, electronics (digital and analog), performance improvement in cost and power of the wireless sensor devices which are tiny in size and reliable in nature have become possible. By these researches the capabilities and powers of the small wireless sensor nods are being increasing that includes the long sensing range, better communication range. The collective effort of these type of sensors makes WSNs possible in real world. Wireless Sensor Networks have a large range of the applications. In current time wireless sensor networks are being use everywhere. Due to the applications and convince of these WSNs we are getting into habit of these networks and in the current days these networks are being part of our life. Research efforts done recently have made it possible to make a tailored network system according to the application requirements in real time, e.g. specialized WSNs for the specified requirements like military application, home application, bio-information system etc. To make these networks possible in real time system it requires efficient and extremely effective protocols for communication and synchronization. A wireless sensor networks contains a huge number of wireless sensors. These wireless sensor nodes are deployed into a physical environment. A wireless sensor node have the capability of capturing the outside environment that is covering it. To find an efficient, effective and reliable performance, the deployment and sensor node selection must be done by keeping characteristics of physical environment in mind. Wireless sensor nodes observe the data from outside world, but these nodes do not send the data in its original raw form. Instead of sending raw data these sensor nodes performs the initial pre-processing by easy computation on this raw data and extract the useful data from raw data, and then sends the useful data only that makes more challenging and difficult for the communication and deployment protocols. Every sensor node have its special intrinsic properties e.g. power consumption, sensing range etc., these properties of each sensor node have different values that include more challenges for communication and deployment protocols. The protocols of communication and application of Wireless Sensor Networks are specially tailored to provide better performance in terms of energy efficiency. Due to the size limitation each sensor node have limited power. So in most of the wireless sensor networks design of most protocols focus on as less as possible energy consumption.

The deployment of wireless sensor nodes in one of the most crucial issue in development of the protocols of the Wireless Sensor Network. The deployment strategy determines the position of the wireless sensor nodes in the deployed network. It is not necessary that the deployment should be predefined. Deployment can be random or according to any algorithm. If the deployment is random then it requires the special self-organizing communication protocols. The sensing range of the sensors is low, so it makes the dense deployment in the nature. Hence the communication use multi-hop communication protocol, that makes it more challenging for routing, communication and overhead problems.

1.2 Background of Sensor Network Technology

Wireless Sensor Networks consider as an emerging domain of deeply networked system of low power wireless motes/wireless sensors/wireless nodes with a small amount of CPU (Processor and OS) and memory and large featured networks for high resolution sensing of environment. The field is now growing with new technologies. Sensor Networking is a multidisciplinary area which involves many technologies such as Radio Networking, Signal Processing, Artificial intelligence system, Communication platform media, Power management, Database management, Deployment issues Algorithms, Platform Technologies (Hardware and Software), and Sensing technologies. WSNs employ connection oriented random access channel sharing and transmission techniques that are now incorporated in IEEE802 family of standard. Sensors are typically deployed in a high density manner and in large quantities. A WSNs consists of densely distributed nodes that support sensing and connectivity, sensors are logically linked by self-organization. WSNs typically transmit info to collecting aggregate some or all of information.

The sensors in the WSNs have various capabilities, powers, functions and powers. As the advance technology is taking place the new fundamentals and power arising. So this field in now becoming push the latest techniques and find remove myriad. Various low cost wireless sensor networks are trying to manage for lots of application like commerce, physical extraction, healthcare, bio-information system. Sensor networking includes various disciplines, e.g. networking and radio communication, architecture, routing, signal processing, optimization of resources, algorithms for various operations, hardware improvising etc.

Typically, a wireless sensor node (or simply sensor node) consists of sensing, computing, communication, actuation, and power components. These components are integrated on a single or multiple boards, and packaged in a few cubic inches. With state-of-the-art, low-power circuit and networking technologies, A WSN usually consists of tens to thousands of such nodes that communicate through wireless channels for information sharing and cooperative processing. WSNs can be deployed on a global scale for environmental monitoring and habitat study, over a battle field for military surveillance and reconnaissance, in emergent environments for search and rescue, in factories for condition based maintenance, in buildings for infrastructure health monitoring, in homes to realize smart homes, or even in bodies for patient monitoring. Sensors use the magnetic and electric field, radio wave frequency, infrared sensors, optical sensors, lasers-radar sensors technology to sense the nearby environment. Now a sensor can be defined as low cost device holding various low- power consuming, chipper sensing elements.

In WSNs all the sensor nodes are connected to a particular node that connects these nodes to other or to the upper level architecture. This node is called as the “Sink Node”. Each node observe the data from the outside environment and pass it to the sink node. This sink node is directly connected to the base station. So the sink node sends the data to Base Station. Mostly the deployment of sensor take place in a dense manner where large number of sensors are takes placement in the deployment. The WSNs includes various sensors that performs various operation like sensing, computing, connecting, processing, etc. Most of the time these sensors use self-organising strategy. Sensors span deployment is of two types based on the size of sensors in deployment. 1) Nanoscopic to microscopic Scale. Usually 1 – 100 nano-meter in diameter 2) Microscopic to Macroscopic scale. Usually 100 – 1000 nano-meter in diameter

Timeline

- 1970's: Wired sensors connected to central location
- 1980's: Distributed wired sensor networks
- 1993: LWIM project at UCLA
- 1999-2003: DARPA SensIT project: UC Berkeley, USC, Cornell etc.
- 2001: Intel Research Lab at Berkeley focused on WSN
- 2002: NSF Center for Embedded Networked Sensing
- 2001-2002: Emergence of sensor networks industry; startup companies including Sensoria, Crossbow, Ember Corp, SensiCast plus established ones: Intel, Bosch, Motorola, General Electric, Samsung.
- 2003-2004: IEEE 802.15.4 standard, Zigbee Alliance.

1.3 Components

A Wireless Sensor Network is a combination of various components that join and work together and form a real time WSNs. There are four basic components in a network:

- **Assembly of distributed or localized sensors**

  A wireless sensor network is a collection of huge number of wireless sensor devices. These sensors are deployed into the environment to observe the outside environment. The deployment of the sensors can be random or according any deployment strategy. These sensors need to sends the data or need various communication among them. To perform the communication in a particular manner we need to arrange these sensor in a particular manner that can be distributed or localized based on the application requirement and the resources availability.

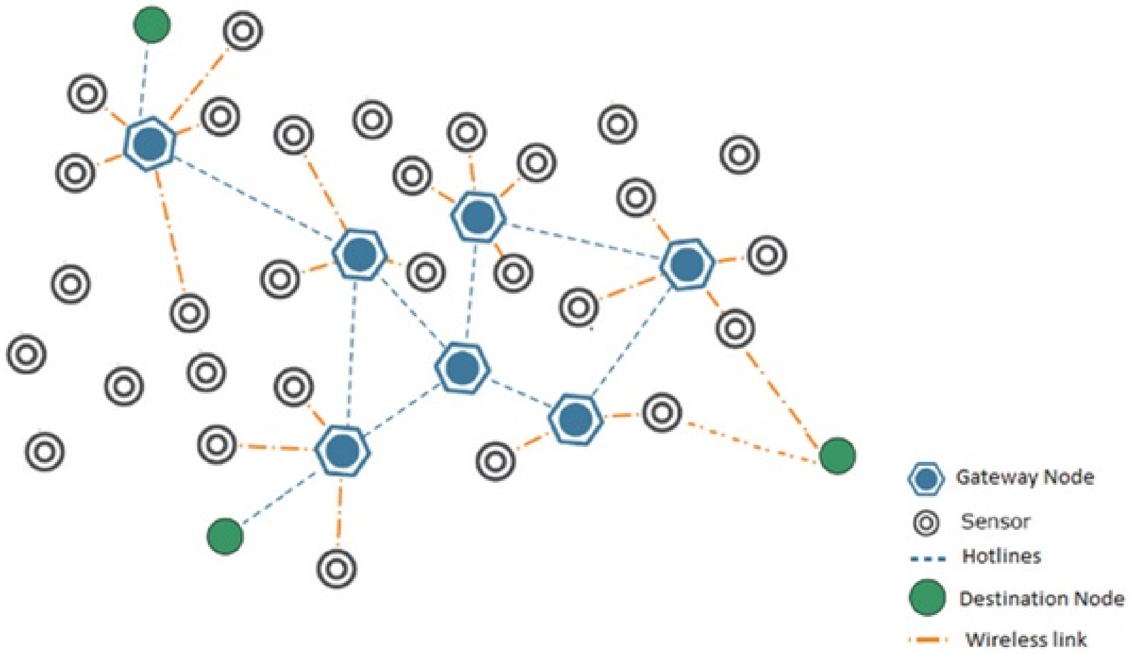


Figure 1.1: Components of WSNs

- **An interconnecting network**

  As it is known that a WSN is a collection of large number of sensor nodes, these nodes takes information from physical world and pass it to the system. Here the question arise “How??..” How these nodes pass data and communicate to each other? All the sensor nodes are connected to each other with a wireless connection. Or the nodes are connected together with radio network. As like the sensing range the communication range of each sensor is also very low. So in order to connect all the nodes to each other and to make communication possible between every pair of sensor node it uses an interconnecting network. An interconnection network connects all the individual sensor nodes together that makes these nodes enable to communicate each other. Interconnecting network provide the routing path and other communication path for data transfer or data retrieving. As the size of the network is very large so it provides the multi-hop communication.

- **A central point of information clustering**

  Due to the size restriction the memory of the sensor device is very low. It cannot store more data. And it is very difficult and challenging to access the data from all the sensor one by one. It creates many problem like data duplicity, time taking process and other challenges. So we need a central point to store data. This point store all the information sensed by all the nearby sensors. All sensors sensed data from the environment and send the data to the central point. This single node makes easy to access all the data at a time. This central data centre stores the routing information and all other information of the network as well.

- **A set of computing resources at a central point to handle data correlation status query**
  To access the data from the central data centre we need specified computing resources such as specified tool for data extraction, special querying command set and other tools to refine the data. These tools provides only required data for the query and remove all other irrelevant data.

1.4 Applications

Wireless Sensor Networks have a wide range of the applications. Now a days WSNs are being use almost everywhere in the real life environment. Initially sensor networks used for applications of the high level. In starting sensor networks have used for typical applications like radiation leakage, or the radioactive sensing, or nuclear/thermal threaten detection, weapon sensors for the fighter planes and ships, habit sensing, crucial physical worlds applications, biomedical/bio-information system. Now recently the focus is diverting more towards the chemical and biological sensors for the nation security aspects. And now a days the applications are being develop for the local and small applications like home or small offices and other such kind of applications. Its main focus is on making performance batter and improving quality of networks.

Existing and current Wireless Sensor Networks application includes

- Military application
- Environment monitoring
- Physical security
- Traffic surveillance
- Video surveillance

- Process control
- Inventory management
- Weather surveillance
- Workshop monitoring
- Home Application and surveillance
- Health Application and surveillance etc.

1.5 Wireless Sensor Networks Architecture

A WSNs architecture is based on its scalability, power consumption, coverage, and connectivity. Basically there are two types of the WSNs architecture
1) Layer Architecture
2) Cluster Architecture

1.5.1 Layered Architecture

In this architecture we use a single powerful node as sink or a base station. All the sensors in the nearby layer range connected to the node have the same hop-count to the sink node.

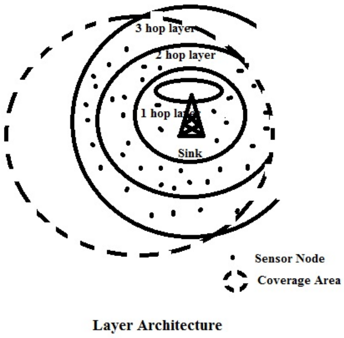


Figure 1.2: Layered Architecture of WSNs

1.5.2 Cluster Architecture

In this architecture the sensor nodes are arranged in the cluster manner. Each cluster of a sensor have a single sink node also called as Cluster head. Sensors of the cluster are directly connected to the sink node. And again some of the sink node become combine together and make a cluster. And each node in that cluster is connected to the Bases Station.

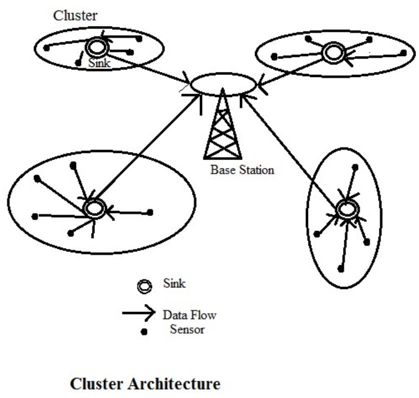


Figure 1.3: Cluster Architecture of WSNs

1.6 Wireless Sensor Device

Consider the fig shown below...

- **Processor:** It is an embedded device. All the computation tasks on the wireless sensor including the communication with other sensor and data transfer control are done by the processor. The embedded processors are significantly constrained with respect to computational power. Due to these type of limitations of embedded processors, wireless sensor devices typically runs on specialized component based embedded operating system, e.g. Tiny O.S.

- **Memory:** It is the place which use to store the sensed and other communication data. It uses as the primary memory. It contain both parts random access memory and read only memory. Random-Access memory use for storing data temporally like sensed data routing table neighbour information etc. Read-Only memory use

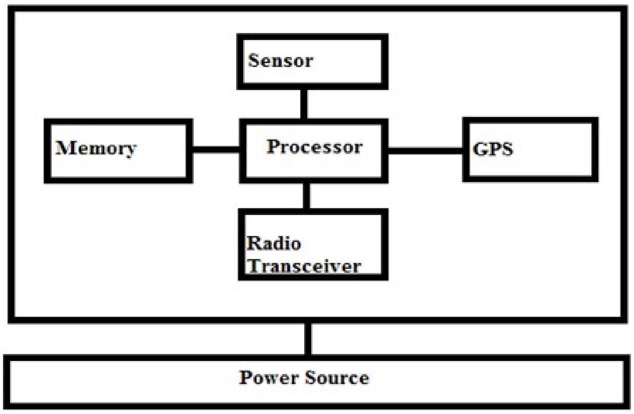


Figure 1.4: Wireless Sensor Device

for storing the Operating system details programs details and the other local details of the sensor node.

- **Radio Transceiver:** This part is use for the communication with other devices. As we know that the communication medium is wireless so we use radio signals for communication among the sensors and sink node. Wireless sensor device contains the short range and low rate radio device. The bandwidth is of around 10-100Kbps, and the range is of around 100 meters. The communication by radio signals is one of the most power consuming operations in wireless sensor networks, so device must designed with the energy efficient sleep and wake up modes.

- **Sensor:** It is the part that collect the data from the environment or outside world. One sensor senses only one specific type of data (for which it is designed, e.g. temperature sensors, light sensors, humidity sensors, pressure sensors, accelerometers, magnetometers, chemical sensors, acoustic sensors, or even low-resolution imagers.). Different applications may require different type of data, so it designs on the base of the requirement. Some application may require more than one data, hence one wireless sensor device can contain more than one sensor.

- **Global Positioning System:** Many Wireless Sensor Networks application demand for the location of the sensors in the networks. This part provide the current location of the sensor. This characteristic mostly uses in the mobile wireless sensor networks, because in such networks the locations of the sensors may change time to time. The position information can easily defined for the static and house use sensor network, while for the outdoor and mobile networks we can use the satellite based GPS device.

- **Power Source:** Part that provide energy to the all other part for functioning properly. This may use the fixed energy or the renewal energy. For the indoor networks and the other small networks we can use the battery power, while for the networks where the human intervention is not possible we can use the renewal energy sources like solar energy etc.

1.7 Issues in Wireless Sensor Networks

In past few years a great interest has grown in wireless sensor networks. There are various issues in wireless sensor networks.

- **1. Hardware** We have define the structure of the sensor node above. It contains the sensing, processing, memory, positioning, transmission and power elements. As we know that mostly sensor networks are deployed in such type of areas where human intervention is not possible, so we have to we design the sensor node with hardware that have a long life, maximum possible sensing range and power. A sensor network contains hundreds of node so the hardware should not be expensive.

- **2. Communication Medium** As we know that the communication medium is wireless. So we use radio signals for communication. These signals should be less power consuming. Radio signals with the long range consumes the high power, so to reduce the power usage we should use the radio signals with moderate range or with low range. It makes the sensors deployment dense.

- **3. Operating System** Operating system for any device controls and operate all the activity of the device. Here we have some major activities like storage sensing data, transmitting sensing data, memory management, and synchronization with other devices etc. We know that all of these activities are the more power consuming, and battery power is limited. So operating system should be power effective. It should contain the power saving qualities like sleep-mode and others.

- **4. Deployment and Localization of WSNs** Deployment refers to the es-

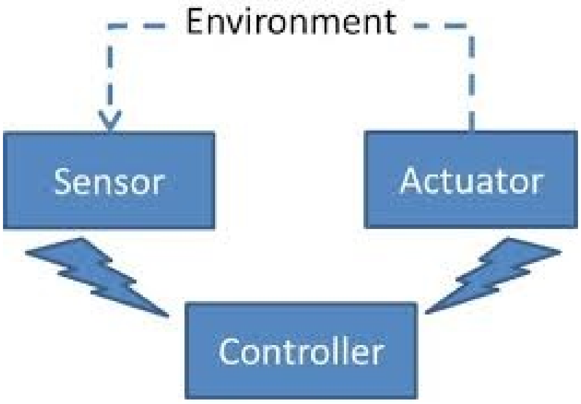


Figure 1.5: Communication medium in WSNs

tablishment of wireless sensor networks in the real world system. It means the placement of the sensor in the desired area or at the desired places. The deployment should be in such a manner that cover all the area or all the targets for the maximum possible time.

The deployment should be based on the application requirement. The deployment can be of two types i) Dense Deployment and ii) Sparse Deployment. Dense deployment uses in application like Military operations, it provides the highly coverage and connectivity, but it can lead to expensive network. Sparse Deployment can use in like domestic applications.
Localization refers to assigning the location to the sensors. In many applications the sensor nodes are static, once deployment is completed then the location of sensor never change. So we have to find a perfect location for each sensor that make improves the performance of networks. The localization of sensors must be like that it provide maximum lifetime and better coverage with least number of the sensors.

- **5. Synchronization** Synchronization is a crucial issue in the WSNs. We know that when we deploy a sensor network, many points are being covered by more than one sensors. So a single point is being covered by more than one sensor simultaneously. In this case there is great possibility of data duplication that can cause the

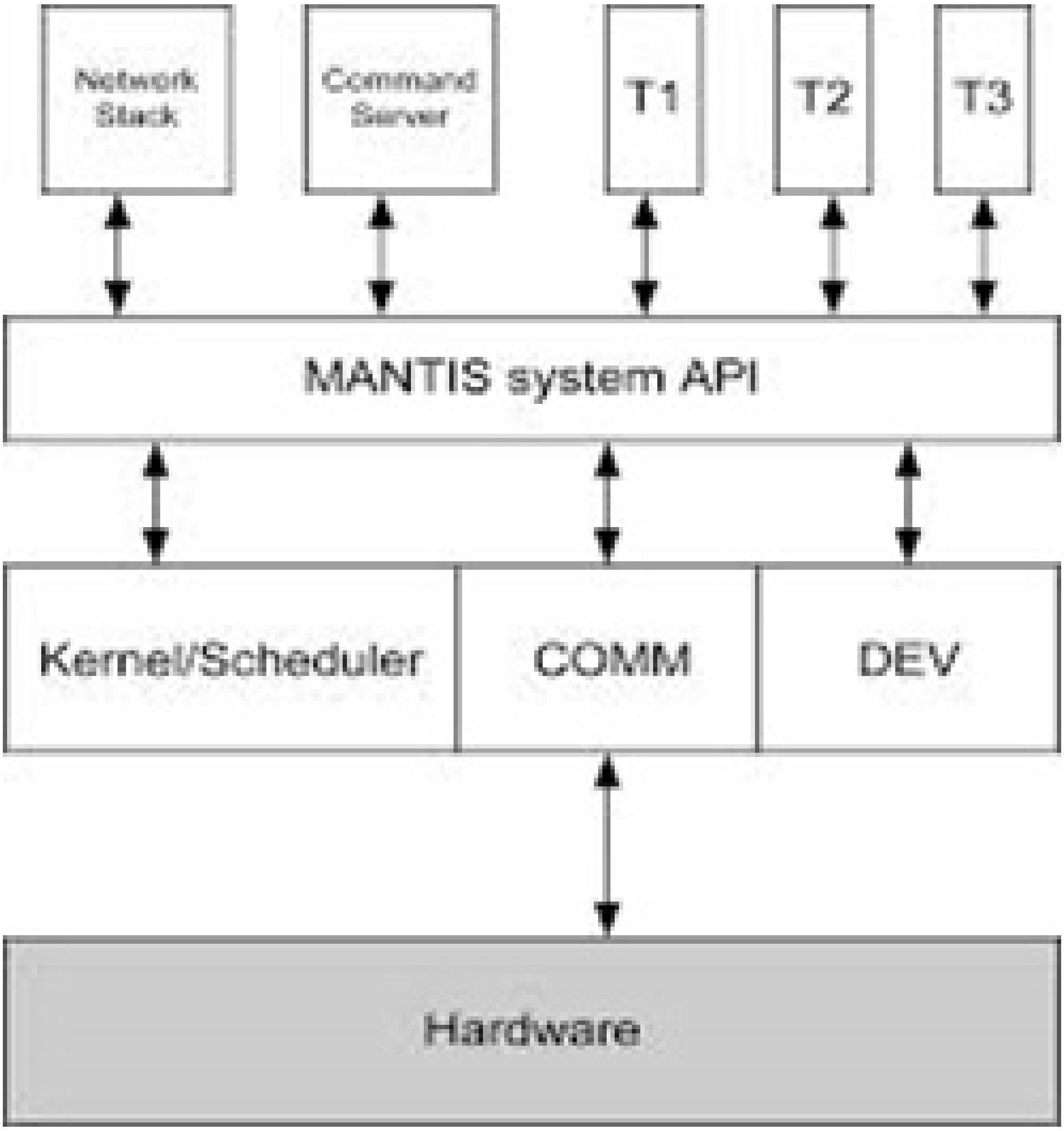


Figure 1.6: Operating System for WSN Device

more power uses more bandwidth uses and, i.e. wastage of resources and our resources are limited here. To stop the data duplication and resource wastage we need a clock synchronization between sensors, targets and sink node. Synchronization provides the better performance with better accuracy and best resource utilization.

- **6. Data Aggregation and Dissemination** The prime objective of the sensor network is the collect analyse and transmit the data from the outside environment. Data Aggregation and Dissemination refers to the collection and of sensed data and then transmission of the sensed data. When the degree of coverage of a single point

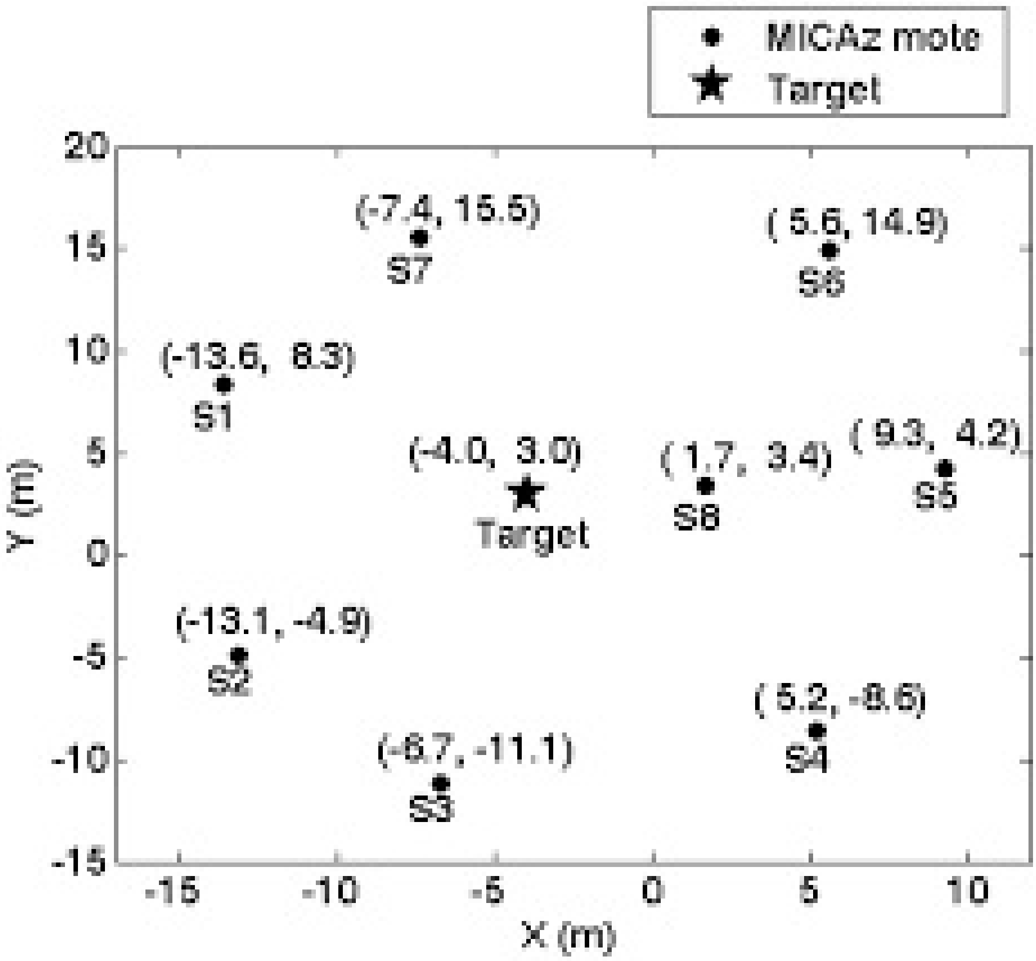


Figure 1.7: Deployment and Localization of WSN Device

is more than one. So it is possible the problem of data duplication at the sink node by transmitting the same data by more than one sensors. This problem leads to the requirement of the high level process of collected data that can generate the good quality highly useful data. So the process of removing redundant content and generating the good quality data by analyse, operate and process the raw data is called as the Data Aggregation.

Data Dissemination is the process of collecting the required data by querying to the neighbour nodes, or by broadcasting the query into the network. In this process the query can be done by any kind of node to any other kind of the node.

- **7. Communication Protocols** ***a. Medium Access Schema***, most of the energy uses is done in the communication, and MAC layer/protocol directly connected to the communication medium uses. So the MAC protocol should be designed in such a manner that it take low power. The MAC protocol should be designed with the sleeping property. And it must remove the collision of the packets and reduce the overhead with the packet to reduce the bandwidth utilization.

  b. Network Layer Protocol, Now a days most of the sensor networks are

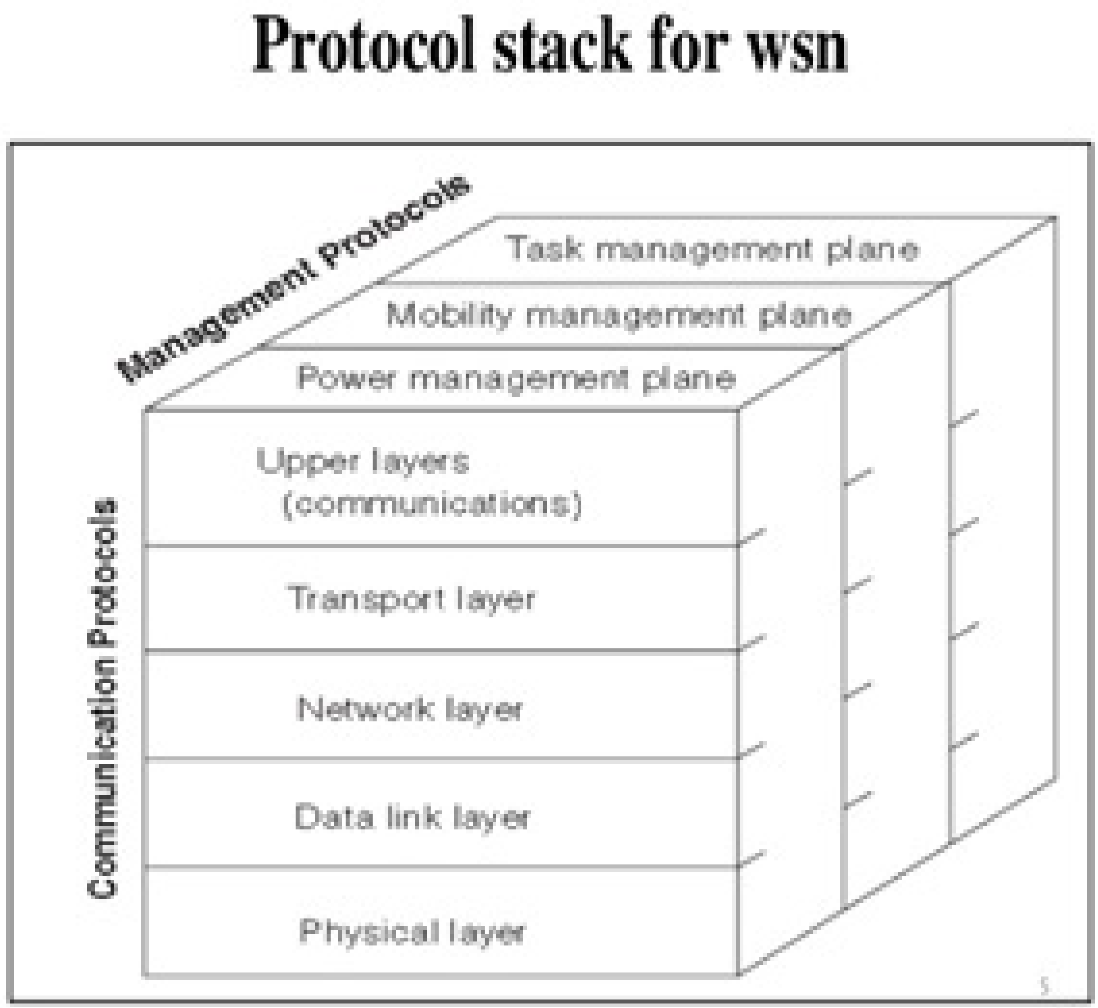


Figure 1.8: Communication Protocols

designed according to the application requirements. And most of the applications are crucial applications, and in WSNs our resources are limited, so to reduce the communication power utilization we need the best routing algorithms that provide the better performance with lesser energy requirement. Routing is one of the most crucial issue for a WSNs.

c. Transport Layer Protocol, This layer is responsible for the process to process delivery/communication. The transport layer divides the data packet into several small packets that lead to the congestion in the network. Due to many reasons like collisions, packet lost, long route etc. the data packet don't reach to the destination, then process of resending takes place, that increase the usage of the bandwidth.

- **8. Architecture** The architecture of the network depends on the application. Architecture of a network defines the rule for the deployment of the network, communication protocols, data transmission and other operations. The main objective of the architecture is to co-ordinate the various activities in synchronize and parallel manner with better usage of resource. The architecture should be scalable to add new nodes or to leave the nodes because many times the sensor nodes goes down/stop working due to battery power limitation. It must contain the better protocols to improve performance and reduce power consumption.

- **9. Security** Security is one of the most crucial issue for the WSNs. Many of the WNSs applications are of the military operations. For a secure network we need the integrity security from the initial phase of the deployment. We can use the confidential operations, authentications and time dependent security mechanism.

1.8 Research Topics in WSNs

We have discussed various issues in WSNs. All of them are the research topics for the WSNs. But when we further study we find some crucial topics for research are currently in the fashion. . . .

- **Energy Efficient Design:** Most of the wireless sensor networks takes place where human intervention is not possible and we also know that the power is one of the crucial constrains for the network. So we need an architecture that require low power to work.

- **Coverage:** Coverage in a sensor network refers that how well each point in the network is being cover by any number of sensors. Coverage of a point means that the number of sensors that are covering to the point at same time. Coverage is a parameters for the quality of service of the network. If all the points in the area are being cover simultaneously then it calls as fully covered. There are two type of coverage 1) Area Coverage 2) Target Coverage. Area Coverage refers to how well a particular area is being cover by a sensor network. In Target coverage we have some targets with fix locations, we have to design a network that cover all the targets all the time.

- **Connectivity:** Connectivity is also equally important to the coverage. It refers that weather all the sensor are in the communication range of each other. We can represent the connectivity of a network using a graph. Consider complete network as a graph where sensor represents as a node and if any sensor is connected to other sensor then there exists an edge between them. Create the complete the graph and if the graph is connected the network is also connected i.e. there is a communication path available between any two node of the sensor network.

- **Routing:** Routing refers to finding the best path for the packet/data transmission from node to sink or from node to other node. In WSNs we use radio signals for the communication, and our resources are limited. And the usage of the communication medium is directly depends on the routing protocols being use by the network. So we have to fine the cost effective routing protocol that reduce the bandwidth wastage increase the performance.

- **Data Aggregation:** The prime objective of the sensor network is the collect analyse and transmit the data from the outside environment. Data Aggregation and Dissemination refers to the collection and of sensed data and then transmission of the sensed data. When the degree of coverage of a single point is more than one. So it is possible the problem of data duplication at the sink node by transmitting the same data by more than one sensors. This problem leads to the requirement of the high level process of collected data that can generate the good quality highly useful data. So the process of removing redundant content and generating the good quality data by analyse, operate and process the raw data is called as the Data Aggregation. Data Dissemination is the process of collecting the required data by querying to the neighbour nodes, or by broadcasting the query into the network. In this process the query can be done by any kind of node to any other kind of the node.

- **Communication Protocols:** ***a. Medium Access Schema***, most of the energy uses is done in the communication, and MAC layer/protocol directly connected to the communication medium uses. So the MAC protocol should be designed in such a manner that it take low power. The MAC protocol should be designed with the sleeping property. And it must remove the collision of the packets and reduce the overhead with the packet to reduce the bandwidth utilization.

b. Network Layer Protocol, Now a days most of the sensor networks are designed according to the application requirements. And most of the applications are crucial applications, and in WSNs our resources are limited, so to reduce the communication power utilization we need the best routing algorithms that provide the better performance with lesser energy requirement. Routing is one of the most crucial issue for a WSNs.

c. Transport Layer Protocol, This layer is responsible for the process to process delivery/communication. The transport layer divides the data packet into several small packets that lead to the congestion in the network. Due to many reasons like collisions, packet lost, long route etc. the data packet don't reach to the destination, then process of resending takes place, that increase the usage of the bandwidth.

1.9 Motivation

In the current world system Wireless Sensor Networks has a very large number of applications. Now a days sensor networks are being use almost everywhere. Many application of sensor networks take place where human intervention is not possible. The placement of sensors is one of the crucial part of the sensor network. It is an optimization problem to find the optimal place for sensors in sensor network deployment. Due to the physical constraints of the sensors and application constraints it is impossible to replace the energy resource of the sensors, e.g. Military Applications, Applications on pole of earth. There is only one time placement of sensors in WSNs. So the network lifetime maximization is a very important part in WSNs deployment. To maximize the life time we divide the all sensors into various sets and then activate those sets one by one. It is also an **optimization problem** divide the sensors into various sets. This motivates me to work towards maximizing the network lifetime by finding the maximum set covers.

1.10 Area of Work

Coverage is an important issue in the Wireless Sensor Networks. It determines how well an area or the targets are being covered by a sensor network, i.e. it gives a measure to define the quality of performance of the sensor network. Different measure captures the different aspects of performance. In wireless Sensor Networks there are two types of coverage 1) Target Coverage and 2) Area Coverage. In Area Coverage we have to deploy sensors in such a manner that covers to the entire given area all the time of the sensor network life. And in Target Coverage, we have some targets (locations of the targets are known already) and we have to deploy the sensors in such a manner that covers all the targets all the time of the sensor network life.
Target coverage in wireless sensor network is the task of covering all the targets spread in an area with the given sensors all the time. We know the locations of the sensors. In Target Coverage Problem, the fixed number of targets are continuously observed by a number of sensor nodes. Possibly, each target is monitored by at least one sensor node. There are a specific number of targets which are to be covered by a set of sensor nodes. After getting deployed, the sensor nodes start the task of monitoring the said targets. Since sensor nodes are provided with only some limited resources and can't withstand extreme environmental conditions, they are deployed in large number much more than actual requirements. A sensor covers all the targets which lies within its sensing range.

Objective: My objective in this research is to maximize the lifetime of the wireless sensor network with covering all the targets all the time.
To achieve this objective I am going to divide the sensors into more than one sets in such a manner that each set can cover all the targets at a time.

Target Coverage in WSNs:

- Consider we have 'n' sensors as $S_1, S_2, S_3, \ldots, S_n$; with the sensing range $R_1, R_2, R_3, \ldots, R_n$;
- And 'm' targets as $T_1, T_2, T_3, \ldots, T_m$.
- The deployment of sensors must be like this that the Euclidian Distance$(D(S_i, T_j))$ between target T_j and sensor S_i must be in sensing range of S_j, i.e.

- A sensor S_i covers the Target T_j if T_j lies within the sensing range of the S_i, $D(S_i, T_j) <= R_i$
- Where i belongs to 1,2,3,.....,n
- And j belongs to 1,2,3,.....,m
- A sensor cover S_c is a set of sensors that jointly cover all the targets. And we have to maximize the life time of the S_c.

All the targets that need to cover are spread into a particular plane have predefined and fixed location. We have some sensors that are placed into that plane. We need to arrange all the sensors in such a manner that can cover all the targets all the time or continuously without any fail.

The sensor sense and transmit the data is processed through a sink node. When the sensor is in the working state for the coverage operation it extract data from outside environment, perform computing and generates data message at a particular rate. This data generator sensor calls as source node/sensor. Then source node sends the generated data message to sink node via wireless communication medium. Communication may requires multiple-hop-communication. When a sensor needs to be in working condition but do not performing coverage operation call as relay node/sensor. Relay node can use for communication between other nodes. When a sensor does not need to be in working condition it goes into power saving state or the sleep state.

1.11 Organization of Thesis

The rest of Thesis is organized as follow:

Chapter 2:Various Methods and Literature Survey
Chapter 3:Greedy Method
Chapter 4:Improvised Greedy Method
Chapter 5:Conclusions and Future Work

Chapter 2

Various Methods and Literature Survey

2.1 Terms and Definitions

- **Area Coverage:** Area coverage refers to the covering a fixed geographical area by using a set of sensors, i.e. a set of sensors is deployed in a manner that can monitor a given geographical area continuously. The arrangement of sensors cover all the points in that particular region all the time.

- **Target Coverage:** Target coverage refers to covering some predefined targets are spread in a plane using a set of sensors, i.e. a set of sensors is deployed in a manner that can monitor all the targets continuously.

- **Sensing Range:** Each sensor can sense the data within a certain range that depends on the physical characteristics of the sensor. The distance within a sensor can sense the data is called as the Sensing Range of the sensor.

- **Set Cover:** Consider we have to cover a set of targets which are spread in a given plane using sensors. To complete this task we will have to use some arrangement of sensors that can cover all the targets. Now consider we have an arrangement of sensor which is deployed in that plane to cover those targets. In all the sensors, a set or subset of sensor that can cover all the targets at a time is called as set cover. A set cover can contain any number of the sensors, but it can't be NULL. Consider we have n sensors to cover all the targets there are (2n – 1) set covers are possible.

- **Minimal Set Cover:** An arrangement of minimum number of sensors that can cover all the targets/complete area at a time is called as the Minimal set cover. A set cover with the minimum number of sensors is called as the Minimal Set Cover. In a minimal set cover each sensor cover at least one such target which is being cover by that particular sensor only.

- **Communication Range:** A sensor sense the data from environment using its sensing range, and then it sends the data to the sink node. To send the data to sink node sensor use wireless medium, which can work till a certain distance. The range in which a sensors can communicate/transfer the data to each other is called as communication range. It depends on the physical characteristics of the sensors.

- **Degree of Coverage:** Consider we have some targets spread in a plane, and an arrangement of the sensor that is monitoring to the targets. Now each targets is being cover by at least one or more than one sensors. The number of sensors which in monitoring to a particular target/point a particular time simultaneously is called as the degree of coverage of that target/point.

- **K-Coverage:** Consider we have some targets spread in a plane, and an arrangement of the sensor that is monitoring to the targets. Now if all the targets are being covered by at least k sensors then it is called as k-coverage. Or an arrangement of sensors for targets such that each target have at least 'k' degree of coverage.

- **Set Cover Lifetime:** The lifetime of a set cover refers to the duration for which a set cover can monitor all the targets continuously. The lifetime of the set cover depends on the lifetime of the sensors. If it is a minimal set cover then the minimum life time of any sensor among all the sensor in set cover is equals to the life time of the set cover because when sensor stops working then property of covering all the targets becomes violate.

- **Network Life Time:** The time duration for which an arrangement/network of sensor can cover all the targets continuously is called as the Network Life time. To cover all the targets if we have more number of sensors than requirement to cover all the targets at a time, we can divide the sensors into various sets and deploy them that can activate one by one, then the sum of lifetime of all the set cover is called as the network life time.

2.2 Example: Consider a set of Targets and Sensors

In our project we are focusing on target coverage so we'll discuss all the definitions in terms of the target coverage in wireless sensor networks.

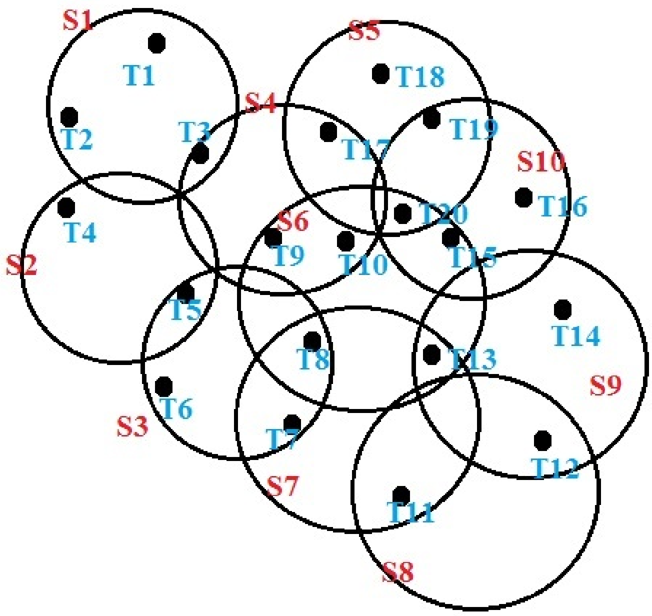


Figure 2.1: Example of targets and sensors

- **Notations in fig.:** Each black point refers to a target, there are twenty targets with numbers $T_1, T_2, T_3, \ldots, T_{20}$.
  Each Circle refers to the sensing range of a sensor, the sensors are placed at the centre of the circle, Here we have 10 sensors with numbers $S_1, S_2, S_3, \ldots, S_{10}$. If a target is inside of a circle then that target is being monitored by that particular sensor.

- **Sensor Characteristics:** In this model we are using all the sensors are of homogeneous type. All sensors having same sensing range and same communication range. We know that we care only about target coverage so we'll not discuss the communication among the sensors.

- **Set Cover:** As we have already discussed that if 'n' sensors are being use to cover some target then there are $2^n - 1$ set covers are possible. Here we have 10 sensors to cover all 20 targets, so here $2^{10} - 1 = 1023$ set covers are possible. If we consider the placement of sensors according to our image then here we have 9 set covers which are as follows

I. $S_1, S_2, S_3, S_4, S_5, S_7, S_9, S_{10}$
II. $S_1, S_2, S_3, S_4, S_5, S_8, S_9, S_{10}$
III. $S_1, S_2, S_3, S_4, S_5, S_7, S_8, S_9, S_{10}$
IV. $S_1, S_2, S_3, S_5, S_6, S_7, S_9, S_{10}$
V. $S_1, S_2, S_3, S_5, S_6, S_8, S_9, S_{10}$
VI. $S_1, S_2, S_3, S_5, S_6, S_7, S_8, S_9, S_{10}$
VII. $S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_9, S_{10}$
VIII. $S_1, S_2, S_3, S_4, S_5, S_6, S_8, S_9, S_{10}$
IX. $S1_, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}$

- **Minimal Set Cover:** As we know that the minimal sensor cover is the set cover with the minimum number of sensors. We have found all the set cover for the above arrangement. Now we have to find the minimal set among all the set covers.
  We have found there are nine set covers from the length of 8 to 10. Here we have 4 sets are of minimal length that is 8, those are as follows,
  I. $S_1, S_2, S_3, S_4, S_5, S_7, S_9, S_{10}$
  II. $S_1, S_2, S_3, S_4, S_5, S_8, S_9, S_{10}$
  III. $S_1, S_2, S_3, S_5, S_6, S_7, S_9, S_{10}$
  IV. $S_1, S_2, S_3, S_5, S_6, S_8, S_9, S_{10}$

- **Degree Of Coverage:** In the above arrangement we see that the targets are being covered by one to three sensors. Those are as follows
  - Targets with the degree of coverage 1 are $T_1, T_2, T_4, T_6, T_{14}, T_{16}, T_{18}$
  - Targets with the degree of coverage 2 are $T_3, T_5, T_7, T_9, T_{10}, T_{11}, T_{12}, T_{15}, T_{17}, T_{19}$
  - Targets with the degree of coverage 3 are T_8, T_{13}, T_{20}

  It is a 1- coverage network because the minimum degree of coverage for any target is 1.

2.3 Literature Survey

2.3.1 Coverage Problem

Here in [16] Deying Li described that many kind of problems are there in Wireless Sensor Networks regarding the coverage and connectivity. With the term coverage many kind of aspects are there like area coverage, target coverage, and barrier coverage. The coverage should have done according to the application requirements. Coverage may be completely or partially dependent towards the application. When there is a complete coverage it means every point can be measure by at least one or more sensors.

The term connectivity refers how well the nodes of a network are connected to each other. There are many kind of operation and activities in the WSNs for what the sensor nodes in the WSNs need the communication among them. There are some special sensors node that must be connected to other all the times like sink node. The sensor nodes sense a huge amount of data and due to the memory restriction problem these nodes have to send the data to the central database. To improve the performance and reduce the delay in data reaching WSNs require a complete good and highly connected network. So the term connectivity is also have equal importance in the WSNs. The performance of a network is highly dependent on the connectivity of network. The connectivity must be robust and fault tolerated.

All the WSNs use for the data observation from the outside environment. Most of the WSNs are used for the crucial applications. So here the data is most important. And the quality of the data must be best. And the quality of the gathered data depends on the coverage quality, delay, security, time etc. So the quality of the data is directly dependent on the strength of coverage of the network. The coverage is directly dependent on the deployment of the sensors. Deployment refers to the localization or placing of the sensors devices to the environment. Deployment must be like this that all the desired targets or the complete area must be covered at a time. Most of the WSNs applications are very crucial where human intervention is not possible. So it requires a very effective deployment method that takes lesser sensors and give more coverage. Mostly for the first time the Random deployment takes place. All the sensors deployed randomly onto the plane where targets need to be monitor. This placement can be done by the any flying machine. Sometimes this type of deployment fails to give proper coverage and connectivity. It is possible that some particular region in the area is getting very high level of coverage and

some region is very poorly covered. This deployment can neither guarantee for the proper coverage nor for proper connectivity. So in order to get better coverage and connectivity and to increase lifetime of the network as well we need more intelligent algorithms. Now when human intervention is not possible for deployment so we need a self-driven system. Self-driven system also called as self-organization. In the self-organizing system the sensor nodes are mobile units as well. These sensors find their current location using GPS system and move towards their perfect location according to the given algorithm. And then the self-organization algorithm executes that arrange these sensors to the correct place for better performance.
"An area is said to be covered if and only if each location of this area is within the sensing range of at least one active sensor node."

Types of Coverage:

The coverage in a Wireless Sensor Networks the coverage can divided into various types according to the application requirements. Here, we discuss three types of coverage problems:
1) *Area coverage*
2) *Point coverage (Target Coverage)*
3) *Barrier coverage*

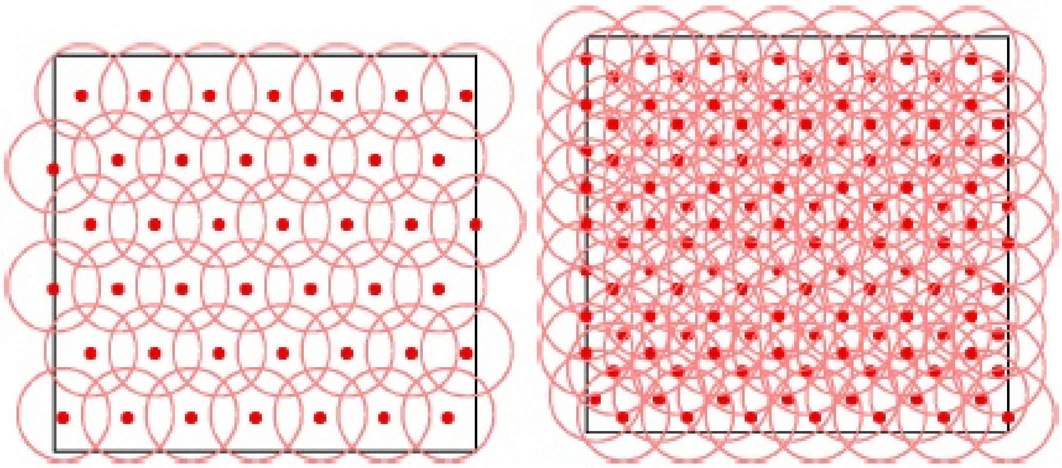
1. Area coverage: When the term area coverage occurs it refers to cover a given area. There are many application where we need to monitor a particular area like the military applications where it has to cover the complete battle field. Here in the area coverage problem, it aims to monitor the complete area. According to the application requirements, the area coverage can be further divided into the various parts. When the application have all the resources and need to cover entire area then it calls Full Area Coverage. Sometimes either the requirement is to monitor a small part of area or the resources are limited then it focus on monitoring a part of the entire area is calls as Partial Area Coverage.

- **a. Full Area coverage:**
  Most of the crucial applications like battlefield observation requires full area coverage. Applications like this, every particular location inside the given area must be covered with at-least one sensor node. When the deployment of sensor is like this

that all the particular locations are being monitor by at-least one sensor then it calls as 1-coverage or if all particular locations are being monitor by at-least k sensors then the such arrangement calls as k-coverage. Deployment of sensors into a large area with consideration of full area coverage and complete/full network connectivity leads to be expensive network. While, the full area coverage along with full connectivity assures the best monitoring.
Now here in following discussion we'll discuss the detail 1-coverage considers as simple-coverage, the k-coverage considers as multiple-coverage, according to the degree of monitoring/coverage, robustness desired by the application.

– **Simple coverage:** In Wireless Sensors Networks, many times it requires to provide full area coverage of entire area with while keeping in mind to deploy as least as possible number of sensor nodes. It can be possible if every sensor is monitoring the separate region/area, i.e. each region in the entire area is being cover by at-least number of sensors or each region is being cover by one sensor. Many researches try to find the deployment of minimum number of the sensors while ensuring complete area coverage and providing complete connectivity. There are various algorithms for this work such as the triangular lattice deployment provides full coverage, connectivity and uniform deployment using the minimum number of sensor nodes.

– **Multiple-coverage:** Most of the crucial applications require higher degree of the coverage. Or in many applications where human intervention is not even possible it need to ensure co cover every particular region to ensure the fault tolerance system or getting better data. Multiple-coverage can be considered as higher level of simple-coverage. Multiple-coverage is denoted using k-coverage. Multiple-coverage is according to the application for example mobility tracking, monitoring in high security areas, distributed detection, and military intelligence in a battlefield. As it is known that the single sensor node failure can leads into the loss of data or corruption of important data. So the simple-coverage cannot assure good performance for these kind of applications. These kind of applications demands very highly-accurate information that can assure a fault tolerated system and gives good decisions. As the k-



a Simple coverage. b Multiple coverage.

Figure 2.2: Full Coverage

coverage provides that every region must be covered by at-least k sensors so it provides the fault tolerance from k-1 sensors failure.

- **b. Partial Coverage:** There are various applications of the WSNs where the coverage of entire area is not required. These kind of applications requires to monitor some special/ particular regions. So in these kind of applications it is wastage of resources if it covers complete region. So here the deployment is in the manner that the set of sensors cover just the required area. This kind of coverage of covering a part of entire area is calls as Partial coverage.

  Partial coverage can explained with the following example consider there is a set of sensors which is covering at least ‘x’ percent of complete given area and is defined with x-coverage where 0 ¡ x ¡ 1. Normally environment monitoring applications of the WSNs demands the partial coverage. For example we can consider the application of temperature sensing, we can get to know the temperature of the entire region by getting the temperature of around 75 percent of area.

2. Target/Point coverage:

There are several applications of the Wireless sensor networks where the coverage of complete area is not necessary. In these kind of applications it is being sufficient to monitor only some particular points. These specific points are also calls as the target points or

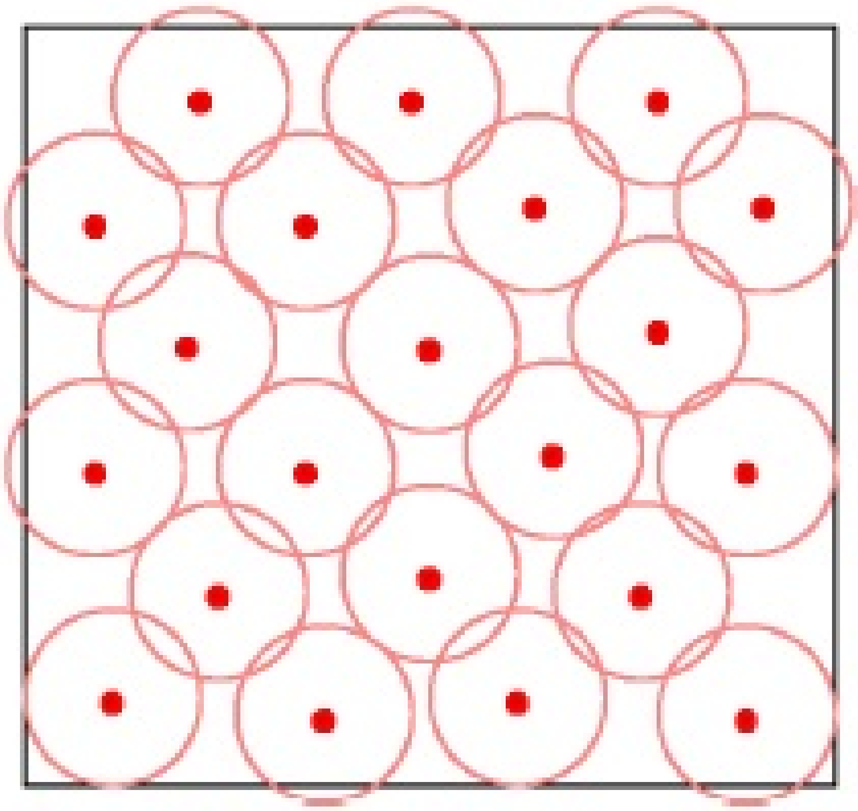


Figure 2.3: Partial Coverage

targets. These type of applications demand that every particular point/ target is being covered by at least one sensor node. Consequently, covering only these specific targets will improve the performance of sensing and reduce the cost of the networks. When all the sensors are being use to cover/monitor some specific targets it ensures the better coverage and better performance of network resources. Since it requires to cover specific targets so the number of required sensors decrease that leads to the deployment cost decrease. Examples of specific target monitoring, include monitoring of enemy troops and bases, capturing the real-time video material of possibly mobile targets. In such applications, mobile flying sensors can be deployed to monitor a target.
According to application the targets can be of two types.

- **Fixed Points/Targets** There are some applications where we need to capture static targets. The location of these targets are predefined and always remains same for lifetime. The property of being static in nature makes it simple to cover all the targets easily because the location is fix and we need to deploy sensors according to the known location.

- **Mobile Points/Targets** A target calls a mobile if the location of this point is being change time to time, means it is dynamic in nature and moves from one location to other location. So to cover these kinds of targets we need some intelligent system. Here we will discuss two methods to cover these kind of mobile targets. First is using static sensor. In this deployment we first find the track of target movement and record it. Then the deployment is in such manner that when target moves from the area of one sensor then it must enters into the area of other sensor. Second method is by using mobile sensors. Here in this method initially it finds the track of the targets and then according to that track it deploys the mobile sensor that move according to the target movement and monitor it.

3. Barrier coverage:

There are various crucial applications, where sensors does not required to monitor inside occurring events for a particular area considered, while it needs to detect intruders which are trying attempt to penetrate in that particular area. For the example of this application we can consider the Zoo tracking system for the animal cage, international boundary tracking system, Forest surveillance for illegal hunting of animals, movement detection, surveillance nearby a chemical factory to detect the spread of lethal chemicals, and on both sides of a gas pipeline to detect potential sabotage.
A barrier coverage that gives surety that each movement occurring across the barrier of sensors must be detected, can be defined as the perfect design for coverage for these kind of applications. On the base of the coverage we can divide it into two parts.
a. Full barrier coverage
b. Partial barrier coverage.

- **Full barrier coverage:** A barrier is fully covered if every location of this barrier is covered by at least one sensor node as it is shown in Figure.

- **Partial barrier coverage:** In some case the resources are limited for example consider the case of insufficient number of sensors, while it needs to find the complete coverage. As it is impossible to find the complete coverage with less number of sensors. So the deployment strategy use the mobile sensor nodes. So sensors try

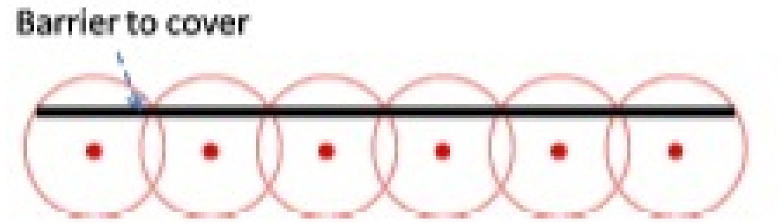


Figure 2.4: Barrier Coverage

to cover area by moving a try to provide the performance better than a threshold value.

2.3.2 Connectivity

Two sensor nodes are said to be connected if and only if they can communicate directly (one-hop connectivity) or indirectly (multi-hop connectivity). In WSNs, the network is considered to be connected if there is at least one path between the sink and each sensor node in the considered area.

In order to get monitoring of an area only coverage is not enough to provide information without reaching monitored information to sink node. Coverage only means capturing the required/ given area or target completely only. But what is the use of a complete coverage if system is unable to deliver the capture information and the monitored information does not reach to the base station. For that it require a communication connection and network that joins all the nodes to each other and enables all the sensors to communicate each other and provide data transformation facility between these nodes. This communication network that guarantees to provide connection and data transfer from source to sink and other nodes calls as connectivity.

The connectivity cab be divided in to two types:

1. Complete connectivity.
2. Irregular connectivity.

- **Complete connectivity:** As connectivity is essential to guarantee the transfer of information, it cannot be neglected and should have the same degree of importance as coverage. Thus, to efficiently monitor a given area, many applications require not only full coverage but also full connectivity in order to collect information and report it. As we saw in the previous section dealing with full coverage, full network connectivity can also be either simple (1-connectivity) or multiple (k-connectivity). In addition, full connectivity can be maintained during the deployment procedure or it can be provided only when sensors have been deployed in the area. In the following, we use connectivity to represent full connectivity.
  1. Simple/Multiple connectivity: Full connectivity is said to be simple if there is a single path from any sensor node to the sink. Full connectivity is termed multiple if there are multiple disjoint paths between any sensor node and the sink.
  2. Preserved connectivity: Considering only initial sensor deployments where all the nodes are connected to each other and to the sink, this connectivity is maintained during the deployment procedure. This means that at any time during the deployment, there is a path connecting every sensor node to the sink.
  3. Connectivity at the end of the algorithm: During the deployment process connectivity can be lost. However, at the end of its execution, the deployment algorithm should guarantee full connectivity.

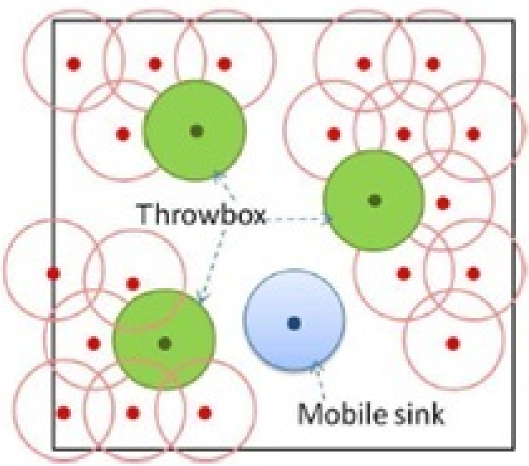


Figure 2.5: Connectivity

- **Irregular connectivity:** In some applications, it is not necessary to ensure full connectivity in the area considered. It is sufficient to guarantee intermittent connectivity by using a mobile sink that moves and collects information from disconnected nodes. There are two types of intermittent connectivity: the first one uses only one or several mobile sinks and the second uses a mobile sink and multiple throw boxes (Cluster heads).

1. Isolated nodes: When the radio range is less than the sensing range, full coverage can be achieved but without maintaining connectivity between neighbouring nodes. Consequently, these nodes will be isolated. One solution to collect the detected information from isolated nodes is to use one or several mobile sinks. One or several nodes are in charge of visiting any sensor node that is not connected to the sink.

2. Connected components: In any connected component, all sensor nodes of this component are connected to each other. However, they are disconnected from nodes in another connected component and they can also be disconnected from the sink. To take advantage of the connectivity within a connected component, a throw box, illustrated in Figure by green nodes, can be assigned to each connected component. A throw box has the task of collecting the information of each node belonging to its component. Then, a mobile sink (blue node in Figure) will not collect information from each node in the network but just take information from throw boxes. One or several nodes are in charge of visiting the throw box of each connected component.

2.3.3 Deployment algorithms

From the above discussion it is concluded that in order to get good performance and better coverage with better connectivity, the sensors deployment must be very good. The deployment must use less number of the sensors and it should provide higher degree of coverage.

To provide better deployment it is necessary to find and analyse the criteria for deployment algorithms.

- **Criteria Analysis:** In this section, we analyse the different factors which have a positive or negative impact on the deployment. We discuss the common assumptions and models found in the literature before focusing on the relationship between the sensing range, r, and the communication range, R, which highly impact the behaviour of the deployment algorithm. Moreover, we define performance criteria for evaluation purposes. We end this section by highlighting the salient features of representative deployment algorithms.

– **Factors impacting the deployment:** Several factors impact the deployment provided and determine how satisfactory the application is. They concern:
  The assumptions and models used concerning r the sensing range and R the communication range. Such assumptions and models are discussed in the next section. The discrepancy between these oversimplified models and reality may explain why the results obtained are not those which might be expected. The values of r and R determine the minimum number of sensors needed to fully cover the entity monitored (i.e. area, barrier or PoI). The deployment algorithms that use exactly this number are said to be optimal. Depending on the relationship between r and R, detailed in Section 3.3, some algorithms either work or not. Others are valid whatever the relationship between r and R, but are not, however, optimal in all cases.
  The number of sensor nodes available for the deployment and the dimensions of the entity monitored will determine whether this number is sufficient to fully cover the entity monitored. It is usually assumed that this entity has a regular shape (e.g. rectangle, disk, etc). However, the reality is often more complex with irregular borders.
  The sensor nodes' ability to move is a determining factor. If sensor nodes are unable to move, the only possible deployment is an assisted one, where a mobile robot for example is in charge of placing the static sensor nodes at their final location. Otherwise, self-deployment is done, where each sensor node is autonomous and able to move. Notice that in such a case, the sensor nodes' movement will consume more energy than communication during the deployment.
  The initial topology may require some extensions to the deployment algorithm. For instance, if the initial topology comprises several disconnected components and a centralized deployment algorithm is used, a mobile robot should be used to collect the initial positions of the nodes needed by the centralized deployment algorithm to compute the final positions of these nodes

and this information should be disseminated to them. If on the other hand, a distributed deployment algorithm is chosen, this algorithm should include a neighbourhood discovery phase as well as a spreading phase to allow sensor nodes to quickly discover other connected components.

The energy of sensor nodes is difficult or impossible to renew, and this fact is of great importance. In the deployment phase, the main reason for energy consumption is the movement of the nodes, whereas in the data gathering phase it is communication between the nodes. In both phases, energy-efficient techniques must be used.

The presence of obstacles makes the deployment more complex: no sensor node should be placed within an obstacle. Hence, the obstacles must be detected and a strategy must be used by the deployment algorithm to get around the obstacles. Furthermore, if the shape of the entity monitored is complex with irregular borders, some extensions to the deployment algorithm will be needed.

The quality of the data gathering required by the application may lead to a uniform and regular deployment. Such a deployment provides smaller data gathering delays, a better time and space consistency of the data gathered, which leads to a more accurate snapshot of the measures taken.

The positioning system may introduce some inaccuracy in the position of the nodes; such a positioning error is very common with GPS. To meet the application requirements, the deployment algorithm should not accumulate the positioning errors during the deployment.

– **Common assumptions and models:** The common assumptions and models found in the literature concern: Communication:

  A unit disk graph model is generally adopted, where any two nodes whose Euclidean distance from each other is less than or equal to the communication range R, have a communication link: they are able to communicate in both directions. This binary model is, however, too simple and does not match the real world. Some authors have introduced more complex models where the probability of success falls less abruptly when the distance increases up to R [25].

A consequence of the unit disk graph model is that any wireless link is assumed to be symmetric. This assumption is not always true in the real world. A frequent assumption is that all sensor nodes have the same communication range. Sensor nodes may differ in their age, their manufacturer, and their communication capacity. Hence some sensor nodes may have a higher transmission range than others.
The initial topology considered in centralized deployment algorithms is usually connected with the sink. This may not be the case in the real world. In distributed deployment algorithms, the initial topology is generally random, as it facilitates the spreading of nodes, leading to shorter convergence delays. For instance Figure a depicts an initial topology where some sensor nodes are unable to communicate with the sink. In addition, Figure b depicts another initial topology where all the sensor nodes are grouped at an entry point but unable to communicate with the sink.

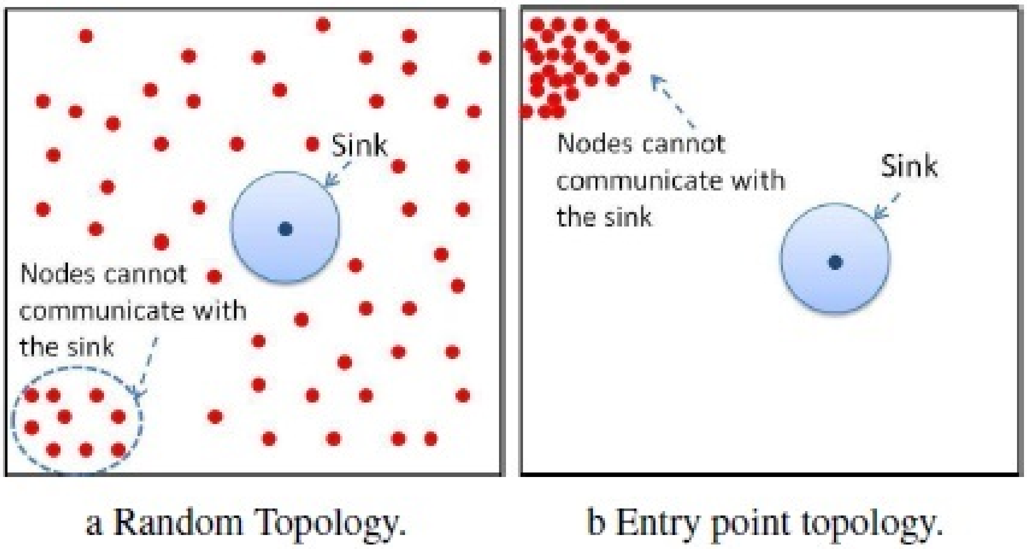


Figure 2.6: Initial Disconnected Topology

Sensing: A unit disk graph model is used to model the sensing of a sensor node. Any event occurring within the disk of radius the sensing range r, cantered at the sensor node is detected. This assumption is too optimistic in the presence of obstacles, for instance. The homogeneity of sensors (i.e. the same sensing model with the same sensing range) is generally assumed. This may not be the case in the real world.

The presence of obstacles: Most authors assume that the entity to monitor is flat and nodes can move freely without obstacles. Such an assumption cannot be made for rescue applications after a disaster, for instance. In the next section, we study the relationship between r and R in more detail.

– **Relationship between coverage and connectivity:** Some deployment algorithms only work when a given relationship exists between the radio range R and the sensing range r. For instance, if R ¡= 2r, it is sufficient to ensure full coverage, and connectivity will be provided as a consequence. In the following, we study the different cases considered in the literature. Furthermore, we recall some results concerning optimal deployments based on regular patterns.

Sensor deployment algorithms based on the relationship between R and r:

∗ **Case $R \geq 2r$:** Full coverage implies connectivity in [31] and [26] anthers prove that when $R \geq 2r$ the full coverage of a convex area implies full network connectivity. This result is extended to k-coverage and k-connectivity. Then, using this assumption, it is sufficient to ensure full coverage, and connectivity will be a consequence.

∗ **Case $R \geq \sqrt{3}r$:** Full coverage implies connectivity, In [2] it is proved that when $R \geq \sqrt{3}r$, ensuring full coverage implies full connectivity. Moreover, the number of sensors needed is optimal, when the triangular lattice is used as a deployment pattern. For instance, one authors propose a deployment algorithm where each sensor node should be placed in a vertex of an equilateral triangle of edge $\geq \sqrt{3}r$.

∗ **Case $R = r$:** An optimal deployment algorithm is proposed to ensure full coverage and 1-connectivity when R = r. In this algorithm, sensor nodes are deployed along a horizontal line, each two neighbouring nodes are at a distance of r. Adjacent lines are at a distance of $(\frac{\sqrt{3}}{2} + 1)r$. In such a deployment, full coverage is ensured but only sensor nodes located in the same line are connected. That is why the authors propose adding a sensor node between each two adjacent lines in order to connect them, such that these nodes form a vertical line.Then 1-connectivity is ensured.

The optimality of this deployment in terms of the number of sensor nodes was proved in [2].

* **Case $R < \sqrt{3}r$:** When $R < \sqrt{3}r$, full coverage does not imply network connectivity. Network connectivity is necessary to report information and it is an important part of the monitoring task. Then, ensuring connectivity while maximizing the area coverage becomes the goal of the deployment algorithm. The deployment algorithm proposed in [12] which deploys sensor nodes in horizontal lines and connects these lines by placing sensor nodes between two adjacent lines, is generalized in [2] as illustrated in Figure. In addition, this deployment is optimal when the distance between neighbouring sensor nodes in the same line R and the distance between two adjacent lines is $r + \sqrt{r2 - \frac{r2}{4}}$.
* **Case arbitrary R and r:** In [23] author proposed an algorithm that aims at preserving network connectivity while maximizing area coverage. Starting with an initial deployment where all sensor nodes are connected to the sink, a virtual force algorithm is applied in order to redeploy sensor nodes in the area considered. As the sensing and radio ranges do not meet the assumption $R \geq \sqrt{3}r$, when sensor nodes move to their new positions they check whether they are still connected to the sink. If they are not, they move towards the sink until connectivity is established. This algorithm preserves full network connectivity during the deployment process and tries to maximize the area coverage with any given values of R and r. In [27] author propose a deployment algorithm that aims at ensuring full coverage and full network connectivity of an area containing obstacles of different shapes. The authors propose dividing the area into two different types of region: small regions or large regions which can contain boundaries and obstacles. As there are no assumptions concerning R and r, in the small regions (like a belt), sensors are deployed along the bisectors of this region and are separated by $r_m in$ = minR, r. In the large region, sensor nodes are deployed in rows. The distances which separate sensor nodes and rows are determined according to the values of R and r.

- **Optimal number of sensor nodes for regular deployment patterns:**

Sensor nodes can be deployed in a regular pattern. This pattern can be a triangular lattice, a square grid, a hexagonal grid or a rhomboid grid. In [24] the author has specified that for each pattern a condition that ensures coverage of the area and guarantees network connectivity as a consequence.

- **If** $R \geq r$ and the hexagonal grid pattern is used, then full area coverage is ensured and the network is connected.
- **If R$\geq$ 2r** and the square grid or rhomboid pattern is used, then full area coverage is ensured and the network is connected.
- **If** $R \geq \sqrt{3}r$ and the triangular lattice pattern is used, then full area coverage is ensured and the network is connected. The triangular lattice is the optimal deployment pattern to ensure full area coverage and guarantee network connectivity.
- **Criteria for performance evaluation:** Each pattern may fit some application requirements. The question is then how to evaluate and select the best one. Different evaluation criteria have been introduced:
- **Coverage:** (e.g. area, barrier, point of interest) is the main criteria to evaluate the efficiency of the algorithm. Usually, coverage is computed as follows: the area to cover is divided virtually into LxW grid units. A grid unit is considered to be covered if and only if its cantered point is covered by at least one sensor node. The coverage rate is computed as the percentage of grid units covered.
- **Connectivity** is also important. The type of connectivity (e.g. full or intermittent) is application dependent. For some applications, maintaining full connectivity is required in order to report any detected event immediately to the sink. Other applications with fewer constraints require intermittent connectivity: usually a data mule.
- **Convergence and stability:** convergence is evaluated by the convergence time defined as the time needed to achieve the required coverage and connectivity. In distributed deployment algorithms, the convergence may be difficult to reach because of node oscillations. Hence, the stability of the deployment is an important criterion to detect the completion of the deployment.
- **Energy and distance travelled:** During the deployment, the main cause

of energy consumption is the mobility of the nodes. That is why the total distance travelled by the nodes must be measured, as this measure reflects the energy consumed. Obviously, minimizing the total distance travelled leads to savings in energy. Notice that the convergence and stability performance has a strong impact on the distance travelled and the energy consumed. Once the deployment has been carried out and the nodes are stationary, the data gathering takes place. The main cause of energy consumption in this phase is communication. To maximize network lifetime, node activity scheduling can be used to make nodes sleep when they are not needed for the data gathering.

- **Communication overhead** comes from the control messages exchanged between the nodes to organize the deployment and the data gathering. In the case of contention based medium access, collisions imply retransmission and increase the overall bandwidth and energy consumption. The aim is to reduce this overhead.
- **Uniformity, regularity and optimality of the deployment:** if the space consistency of measures taken is expected, a uniform deployment is needed: all the nodes (except the border ones) should have the same number of neighbours. Similarly, if the measures should be taken at equidistant positions, a uniform and regular deployment is needed. Usually, such a deployment reproduces the same geometric pattern (e.g. triangle, hexagon, square, etc). Depending on the relationship between r and R, some patterns are optimal. This optimality is useful because it requires the smallest number of sensor nodes to meet the application requirements. A uniform and regular deployment is also mandatory when the application requires time and space consistency of the data gathered.

• **Area coverage and connectivity algorithms**

- **Full coverage:** Many deployment algorithms aim to ensure full coverage of the area considered. These algorithms are classified into three strategies. We distinguish the forces-based strategy, the grid based strategy and the computational geometry-based strategy.
  * **Forces-based strategy:** The forces-based strategy is known by its simple deployment principle. This principle is based on virtual forces that

can be attractive, repulsive or null. In this strategy, a sensor node should maintain a fixed threshold distance called D^{th} with its 1-hop neighbours. Then, if the distance separating two neighbouring nodes is greater than D^{th}, an attractive force is exerted, whereas if this distance is less than D^{th}, a repulsive force is exerted. Otherwise, the force is null since the distance separating neighbouring sensor nodes is equal to D^{th}, the required distance. This principle is illustrated in Figure,

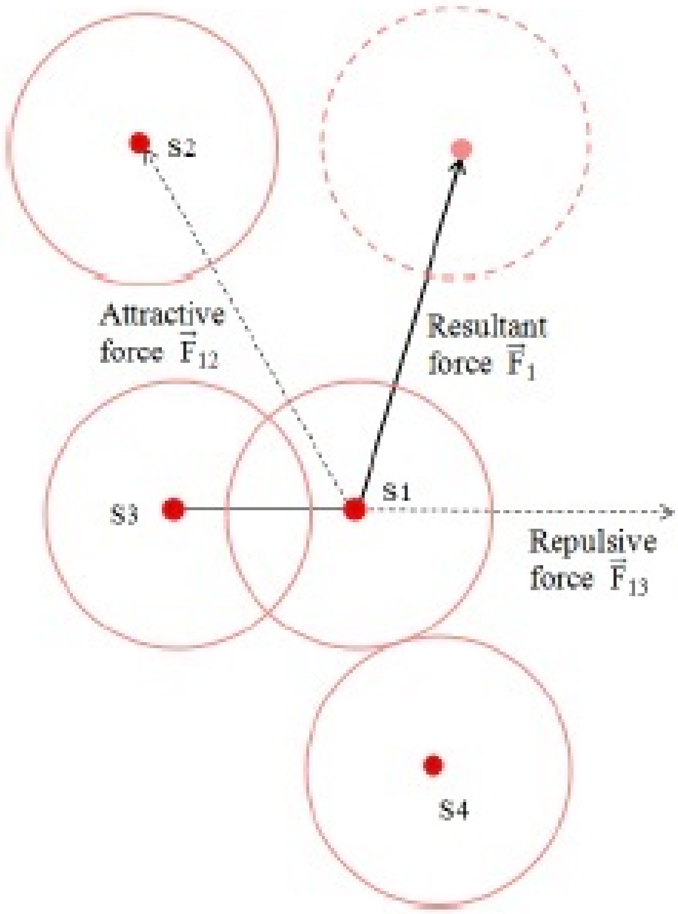


Figure 2.7: Forces Based Strategy

Where F_{ij} denotes the force exerted by sensor node j on sensor node i.

The **Virtual Forces Algorithm (VFA)** proposed in [33] as a centralized redeployment algorithm to enhance an initial random deployment. In the initial deployment, any sensor node is able to communicate with the sink in a one-hop or multi-hop manner. Then, the sink computes the appropriate new position of each sensor node based on the coverage requirements and using the virtual forces mechanism. In this work, obstacles exert a repulsive force and an area of preferential coverage exerts an attractive force on sensor nodes. During the execution of the virtual forces algorithm, sensor nodes do not change their positions. It is only when they receive their final positions from the sink that they move directly to them. VFA is a centralized algorithm that offers a good coverage rate of the area considered while maintaining network connectivity. However, a central entity must know the initial positions of all sensor nodes, compute their final positions and disseminate the positions to all sensor nodes. This principle is problematic when network connectivity is not initially ensured. Furthermore, when the network is very dense, this algorithm has a poor performance due to the gathering of the initial positions of sensor nodes. To cope with the scalability problem, distributed versions of VFA are proposed in [17] the literature. For instance, the extended virtual forces-based approach copes with two drawbacks of the virtual forces algorithm: the connectivity maintenance and nodes stacking problems (i.e. two or more sensor nodes occupy the same position). The connectivity maintenance problem occurs when the communication range is low, $R/r < 2.5$. Thus, the authors propose adding an orientation force which is exerted only if the node has fewer than 6 neighbours. This force aims to keep the angle formed by one node and its two neighbours equal to $\pi/3$ in order to provide a reliable connectivity and eliminate coverage holes. Notice that these authors observe a stacking problem, where several nodes are located in almost the same position. This is because the coefficient of the attractive forces is not well tuned. As a solution, the authors propose an exponential force model to adjust the distance between a node and its distant neighbours. However, the threshold value of $R/r = 2.5$ is not explained and the maintained connectivity is not proved in the paper. Furthermore, the additional orientation force may induce node oscillations.

The IVFA (Improved Virtual Force Algorithm) , and EVFA (Exponential Virtual Force Algorithm) proposed in [6] are two distributed deployment algorithms. EVFA aims at speeding up convergence because forces increase exponentially with the distance between sensors. IVFA limits the scope of virtual forces: only nodes in radio range of a given node exert virtual forces on it. Furthermore, the stacking problem is solved by using a very small attractive force with regard to the repulsive force. IVFA converges to a steady state faster than the basic virtual forces algorithm, and defines a maximum movement in each iteration to reduce useless moves and save energy.

DVFA, proposed in [13] is another example of the distributed algorithm that uses the virtual forces to spread sensor nodes until the entire area is covered. The main drawback of this algorithm is node oscillations. To deal with this problem, the authors of DVFA limit the distance sensor nodes move to a certain threshold. In this way, energy consumption is reduced during the deployment which provides a fast convergence to a coverage rate close to 100 percent. DVFA is also used in [20] to cope with obstacles of different shapes. By using the virtual forces principle and a method to avoid the obstacles, full area coverage is ensured even when an obstacle has a confined shape. Usually, the virtual forces strategy is used to ensure full area coverage as the attractive and repulsive forces spread sensor nodes over the whole area and consequently achieve a high coverage rate rapidly. Furthermore, this strategy is used in [23] with the goal of preserving network connectivity. This deployment algorithm, called CPVF, Connectivity-Preserved Virtual Force, is used to monitor an unknown area with an arbitrary ratio R/r. To achieve that, a sink periodically broadcasts a message to neighbouring sensors which in turn flood the message to all connecting nodes. A sensor node is considered to be disconnected from the network if it does not receive the flooding message. Then, it moves toward the sink in order to reconnect. This algorithm induces a high overhead in terms of messages broadcast in the network to check the connectivity of the nodes with the sink. This paper also proposes a floor-based scheme to improve the global network coverage by reducing overlapping. This scheme is based on the division of the area into equidistant floors (distant of 2r) and encourages sensors to stay in the floor lines. Sensor nodes are added in a column between floor lines to ensure connectivity. Although this work aims at preserving network connectivity when the ratio R/r is arbitrary, it requires a high number of sensor nodes, because the inter-floor distance is fixed to 2r for any value of R and r.

* **Grid-based Strategy:** The grid-based strategy provides a deterministic deployment where the position of the sensor nodes is fixed according to a special grid pattern such as a triangular lattice, a square grid or a hexagonal grid (see Figures). Then, the area is divided into virtual cells and depending on the deployment algorithm used, sensor nodes are located either in cell vertices or at the cell centre.

  The grid deployment is also a regular deployment pattern as all the generated grid cells have the same shape and size. The regular deployment pattern is studied in [15] order to provide multiple coverage (p-coverage) and multiple connectivity (q-connectivity) using the triangular lattice, square or hexagonal pattern. The value of p and q are provided by adjusting the distance separating sensor nodes and limiting the ratio R/r. A comparative study of regular pattern performance in terms of the number of nodes required is also provided to achieve 1, 3 and 5-coverage and q-connectivity. With the ratio $R/r \geq \sqrt{3}$, the triangular lattice is better than the square grid, which is better than the hexagonal grid. However, with the value of $R/r < \sqrt{3}$, the triangular lattice becomes the worst. Multiple coverage and connectivity with regard to the regular deployment pattern studied in [30]. The authors propose the optimal deployment patterns to ensure full coverage and q-connectivity while $q \leq 6$ for certain values of R/r. They consider the hexagonal deployment pattern as a universal basic pattern that can generate all optimal patterns. Then, they present different forms derived from the hexagonal pattern by changing the edge length and the angle between adjacent edges. When the applications require time and space consistency of the measures taken by sensor nodes regularly distributed in the area, the regular deployment pattern can be a good solution to provide a high level of coverage and connectivity with a minimum number of sensor nodes. In the following we present some research studies proposing a regular deployment pattern based on a triangular lattice and a square grid.

- **Triangular grid:** In [1] it has proved that the triangular lattice shown in Figure b offers the smallest overlapping area and requires the smallest number of sensor nodes. When the triangular lattice is used as a deployment pattern, each sensor node occupies a hexagonal cell. However, the deployment is not considered to be a hexagonal deployment since a sensor node is at the centre of a hexagon and neighbouring sensors form a triangular pattern. For instance, the author in [28] has already propose a deployment algorithm called HGSDA that deploys sensor nodes in a triangular lattice. This deployment starts by dividing the area into small hexagonal cells and each cell centre corresponds to a sensor position. Although the cells are hexagonal, sensor nodes are deployed in a triangular lattice since the distance between two neighbours is $\sqrt{3}r$ and there is a sensor node at the cell center. HGSDA identifies redundant sensor nodes in order to place them in empty hexagonal cells. Since the size of a hexagonal cell is computed according to sensor sensing range and the area size, full coverage is achieved using the smallest number of sensor nodes.

  This algorithm is carried out by a sink. Then, all the sensor nodes

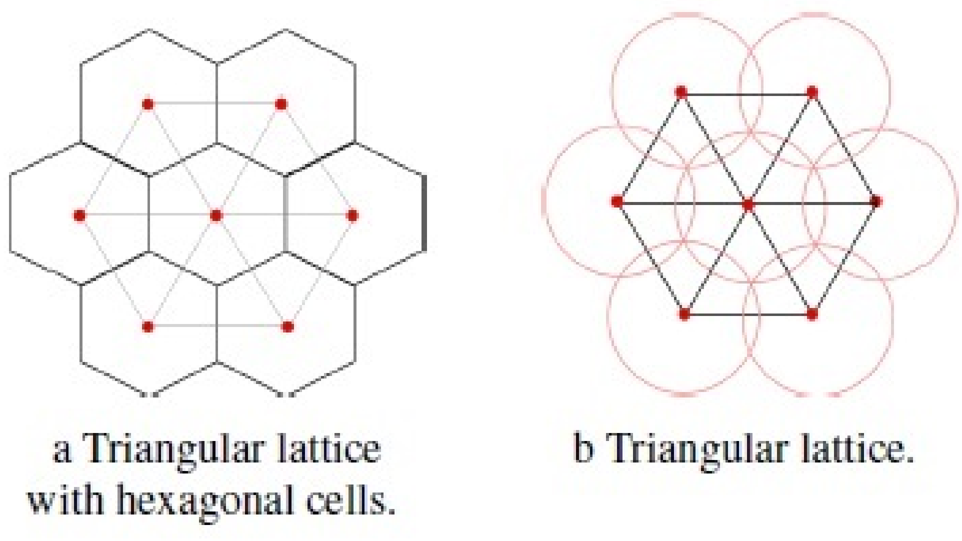


Figure 2.8: Triangular Lattice

receive their final position from the sink and move to it. HGSDA is a centralized algorithm that ensures full coverage using the minimum number of sensor nodes while ensuring simple connectivity with the sink in the final deployment. This centralized algorithm can only be used if connectivity with the sink is ensured in the initial deployment.The same deployment pattern is presented in [3] , but in a distributed version. However, at the beginning of the deployment, the area is not yet divided into hexagonal cells. An initiative sensor node starts by snapping itself at the center of the first hexagonal cell and selects six sensor nodes in its vicinity to snap them in the adjacent hexagonal cells. The selected sensor nodes move to their cells and in turn select other sensor nodes to occupy their adjacent cells. Then, hexagonal cells are built progressively in a distributed way: the hexagonal side length is equal to the sensing range. Since the sensor occupies the center of the cell, the triangular lattice is used as the deployment pattern.

- **Square grid:** The square grid strategy is used in [21] where the area monitored is divided into square cells, as shown in Figure below. Each cell represents the maximum square size that is covered by one sensor node. Each sensor node occupies a cell centre to cover the corresponding square cell. If an empty cell exists, neighbouring sensor nodes should decide to which one will move to cover it, such that if new empty cells appear, they will be around the sink. Redundant nodes should move toward the sink in order to cover empty cells that can occur along the path to the sink.

  A grid-based approach is also used for robot-assisted sensor deployment. As an example in [29] a robot places sensor nodes at the vertices of a square cell. Then, each deployed sensor node colours itself white if it is adjacent to an empty cell and black otherwise. Neighbouring sensor nodes exchange hello messages to inform each other about white nodes (empty cells) and maintain a back pointer corresponding to the nearest empty cell along the backward path of the robot. Then, the robot backtracks this back pointer to drop sensor

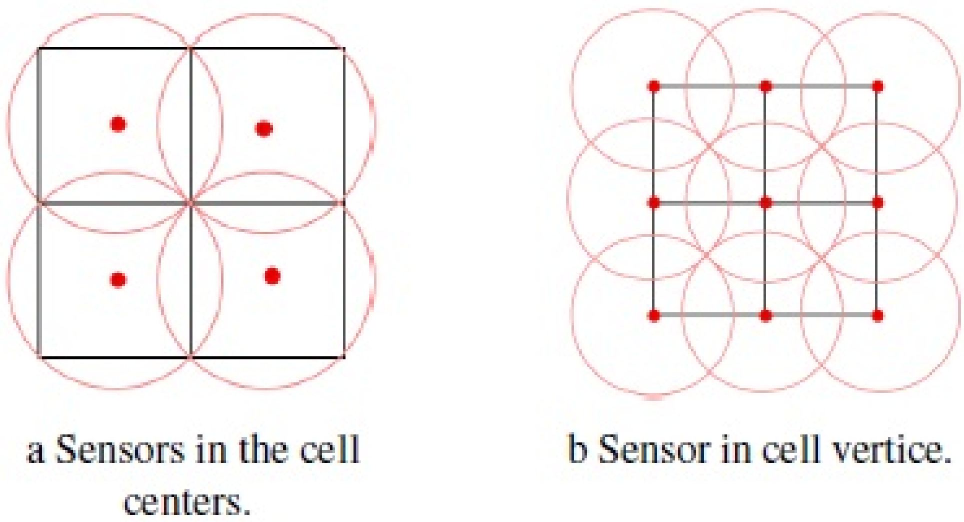


Figure 2.9: Square Grid

node in the empty cell. This algorithm guarantees full coverage in a failure free environment using a mobile robot in a square grid. It is assumed that the robot carries enough sensors to heal any coverage hole (i.e. empty cell) that is detected. Such strategies are used when the sensor nodes are static, and a mobile robot is used to ensure coverage by repairing any coverage hole detected by the sensor nodes. The new problem is that of detecting coverage holes and optimizing the robot movements.

2.3.4 Observation of Deployment Algorithms

In the [10] author has discussed all kind of coverages and various deployment algorithms for sensors deployment in Wireless sensor networks for coverage problem. The different sensors deployment algorithms have different assumptions with specific goals that depends on the application specification requirements with the nature of the WSNs. Many of these algorithms are applicable to only static sensor networks, while some other methods works for mobile sensor networks also.

Algorithm	Network and Goals	Advantages	Disadvantages	Performance
Imprecise Detections Algorithm (IDA)	• Static; • minimize the number of sensors	• Allows modelling of obstacles and preferential areas	• sensor detections assumed to be Independent	• random deployment in case of obstacles
Potential Field Algorithm (PFA)	• Mobile; • redeploy mobile nodes from an initial configuration to maximize coverage;	• Centralized algo not required; • Self-deployment from initial deployment; • hence Scalable	• One time computation and sensor location determination;	• Poorer in performance than tiled networks.
Virtual Force Algorithm (VFA)	• Mobile; • redeploy mobile nodes from an initial random placement to enhance coverage;	• One time computation and sensor location determination;	• No route plan for repositioning of nodes; • efficiency depends on the force parameters.	• O(nmk) time complexity for a n*m grid with k sensors deployed.
Distributed Self-Spreading Algorithm (DSSA)	• Mobile; • spread nodes fro an initial random deployment to maximize coverage and maintain	• Distributed self deployment	• Every node should know its own location.	• Outperforms simulated annealing in terms of uniformity, deployment time, and the mean distance travelled by the nodes

Figure 2.10: Deployment Algorithms

2.3.5 Point/Targets coverage and connectivity algorithms

The other type of coverage is given by the coverage of Points of Interest (PoI). Examples of applications include the detection of some static or moving, using the smallest number of sensors. We distinguish between static Targetss and dynamic Targetss.

- **Static Targets:** In [9] authors are interested in the deployment of mobile sensors to cover predefined Targetss, while preserving connectivity with the sink. The sink has the task of disseminating information about the PoI locations to the sensors as well as collecting the information reported from the sensors about the events happening at the Target. The basic idea of this deployment algorithm for Target coverage is as follows: initially all the sensors are within radio range of the sink. All the sensors run the same algorithm but the motion decision is taken individually by each sensor node. The sensors move toward one predefined targetnt that could be the Target or the barycenter of the Targetss. Then they form straight lines between the Target and the sink. The distance the sensors move is bounded in order to maintain connectivity. Finally a sensor stops moving, when it covers the Target (i.e. the Target is in the sensing range of the sensor). The strategy of this deployment algorithm minimizes the number of sensors used to maintain connectivity by using the RNG graph (Relative Neighbourhood Graph).

  If multiple Targetss exist in the area considered, two approaches can be adopted:

• Random Target deployment: the sensor chooses one of the Target at random;

• Barycenter Target deployment: Every sensor calculates the barycenter of all the Targetss and the sink to cover it. Then any sensor chooses a Target at random and covers it.

In [8] there is a distributed deployment scheme is proposed where mobile sensors nodes move following concentric circular paths (ferries with annular trajectories) that cover static Targetss (See Figure). The goal of this work is to ensure Target coverage and that events are reported to the sink. This sink is located at the barycenter. Two neighboring circular paths are at a distance of R. The authors assume that $R \geq 2r$ and mobile sensors have no global knowledge of the Targetss in the area considered. This work combines three aspects which are: Target discovery, Target coverage and connectivity with the sink. To achieve these three aspects, a mobile sensor should move constantly to execute the Target discovery task. Then, it should adjust its movement velocity with sensors in the neighbouring circular paths to satisfy the constraints regarding coverage and connectivity with the sink in order to report the information about the Targetss.

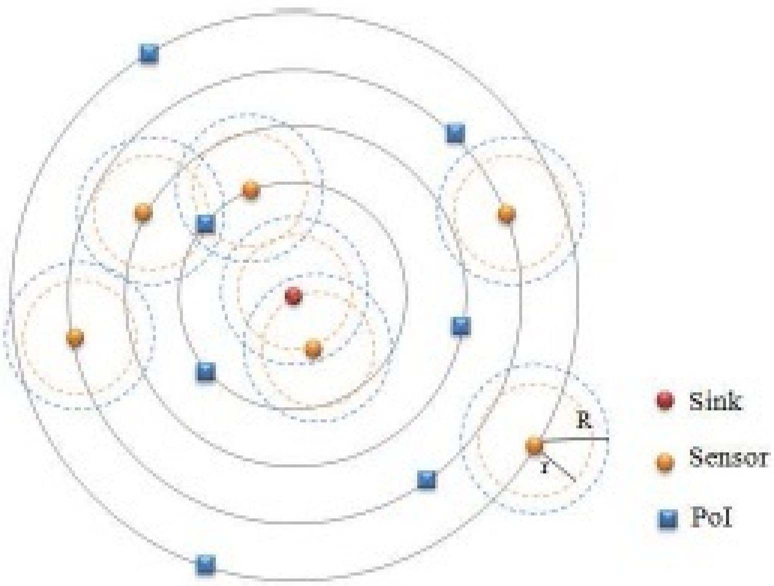


Figure 2.11: Target coverage using Annular Ferries

Temporary coverage of Multiple Targets is studied in [18] and is called the sweep coverage problem as sensor nodes sweep between Targets and cover them periodically. A distributed algorithm DSWEEP is proposed to address this problem. A sensor node covers a Target for a determined duration and then moves to a new one. When a sensor node is moving, it encounters other sensor nodes and exchanges information that serves to decide which Target should be monitored next. This deployment algorithm needs a small number of sensor nodes to cover a large number of Targets. DSWEEP provides temporary coverage and partial network connectivity. In some applications, the Target, as well as the area surrounding it need to be covered.In [19] an localized self deployment algorithm is proposed to meet this goal. This algorithm is based on a virtual triangular lattice grid of edge $\sqrt{3}r$ to maintain connectivity since it is assumed that $R \geq \sqrt{3}r$. Sensor nodes are autonomous and know the position of the Target. They move through the triangular vertexes and organize themselves by respecting rules that avoid collisions between sensors, to reach the vertices around the Target. Based on this principle no coverage holes will occur if all the vertexes around the Target are occupied by sensor nodes.

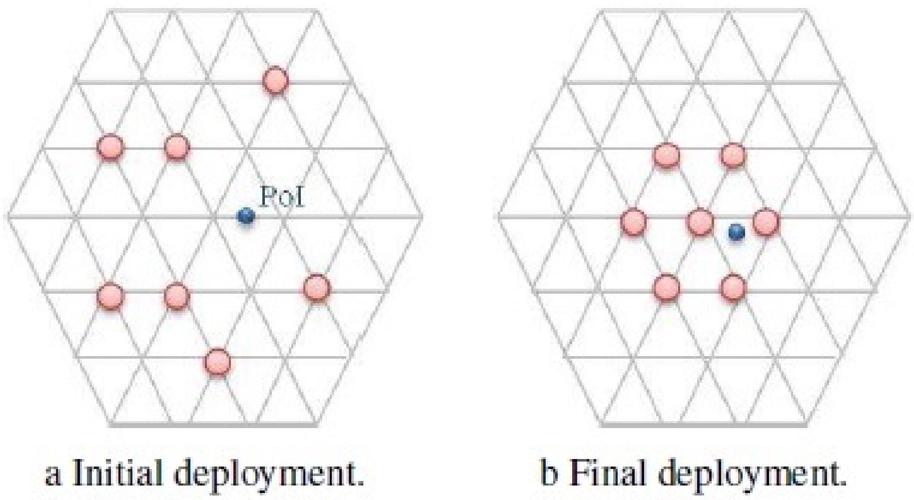


Figure 2.12: Target Coverage using Grid

- **Mobile Targets:** In the case of mobile Targets, there is an author of [9] proposed three strategies to reach the mobile Target:

  • In the first strategy, sensor nodes move back to the sink before deploying toward the new location of the Target. This strategy provides a high coverage quality but increases the deployment duration and the amount of energy consumed.

• In the second strategy, sensors try to move directly toward the new location of the Target without going back to the sink. This strategy reduces the time needed to cover the new Target but also reduces the coverage quality as it needs a greater number of sensors to maintain connectivity.

• In the third strategy, a sensor moves toward the straight line between the sink and the new location of Target, then it moves toward the Target. This strategy provides a higher coverage quality and reduces the time needed to cover the Target.

- **Summary:** Any Target needs only one sensor to be covered. If permanent connectivity is required, a sufficient number of sensor nodes are deployed to ensure connectivity with the sink. However, if intermittent connectivity is sufficient, one sensor node will cover a Target, and a mobile node (that can be the sink or a collector Robot) will operate like a data mule. This can be a solution to deploy a minimum number of sensor nodes and save energy. When the Target is static, a static sensor node can be used to cover it. If the Target is mobile, however, autonomous sensor nodes are deployed to track the Target and avoid the use of a robot that would pick up and deploy sensor nodes each time the position of the Target changes. Notice that Target coverage needs just one sensor to be covered, whereas a zone of interest requires at least one sensor to be covered as the zone may be larger than the sensors sensing range. When many sensor nodes are deployed to cover a zone (area) of interest, they are usually deployed with various densities: high density in the center of the zone of interest and then the density decreases with the distance to the center of the zone.

2.4 Various Algorithms for Lifetime Maximization

P. Khuntia *et al.* [14] discussed the various issues related to the target coverage problem in wireless sensor networks. Also discussed some approaches for covering target in WSNs.

Manju *et al.* [22] discussed various approaches to complete target coverage and maximize the lifetime of the network. They mainly discussed the Energy-Efficient Target Coverage , Target Coverage with Quality of Service Constrains , Target Coverage with Adjustable Sensing Ranges , Distributed and Centralized Approach , Non-Disjoint and

Disjoint Sets Approach .

Zhang *et al.* [32] considered the target coverage problem with 1 local hop presented a Distributed Optimum Coverage Algorithm (DCOA). It initially assumes the base station as a coordinator that performs all the activities. Each and every sensor goes active and sleep round by round. A fix time taken by each round, and time for initialization and monitoring phase. The time taken by the initialization is lesser then round time. Initially Base station coordinates all activities but later sensors construct their local networks, and finally become separate and works within distributed environment.

Jamali *et al.* [11] provided an energy efficient method for target coverage problem that maximizes the lifetime of the wireless sensor networks. This algorithm is for connected set covers (CSC). This algorithm considers the discrete target coverage as well as network connectivity also. This algorithm has three phases are as follows 1) Coverage Phase 2) Connectivity Phase and 3) Redundancy Reduction Phase.

M. Cardei *et al.* [4] used the dividing all the sensors into various set covers to maximize the lifetime of the network. And then finds the maximum numbers of the set covers (Maximum Set Covers (MSC)), as the number of set covers increase the lifetime also increase. Then they proved that the **MSC problem is a NP-Complete problem**. And the discussed two heuristics for MSC problem as 1) Linear Programming MSC 2) Greedy MSC.

B. DIOP *et al.* [7] used the greedy approach for the sensors scheduling to increase the number of set covers. Here greedy approach select a sensor and check whether it is covering any uncovered target, if yes then it selects the sensor into set cover. This is a quick method to find the set cover.

Mihaela C. *et al.* [5] discussed a sensors scheduling algorithm to maximize the lifetime of the network with adjustable sensing range. Here they introduced the distributed and centralized heuristic for both linier programming and greedy method.

2.5 Approaches to Complete Target Coverage and Maximizes Lifetime of the Network

There are various approaches to maximize lifetime and covering all the targets are spread. Some are as follows

- **1. Energy-Efficient Target Coverage:** Efficient power management can extend the life of the network. This approach finds a sensor cover to cover all the targets based on the energy characteristics. Here the selection of sensor for a sensor-cover based on their battery life. In this approach the sensor covers can be disjoint sets or can be non-disjoint sets. It can select the sensors with high power first or in the last or sensors with low power first or last, or we can also select the combination of both (sensors with low and high power).

- **2. Target Coverage with Quality of Service Constrains:** Sensor networks have the use in the various fields. Different application have different specification and requirements. For example some applications need more degree of the coverage or some applications have some fields that can be underestimate and some fields which are must to be cover.

- **3. Target Coverage with Adjustable Sensing Ranges:** This approach is used where sensors are of heterogeneous type, i.e. the properties and powers of sensors are different. Sensors have different sensing ranges different lifetime etc. Here we schedule the sensors and their sensing ranges to maximize the lifetime of network providing that all the targets are covered by sensors.

- **4. Distributed and Centralized Approach:** These are two types of algorithms distributed and centralized. In the Distributed approach different works and operation are performed on the different nodes. While in the Centralized approach all the operations and works are perform on a single node. Here operations and works can define as the routing scheduling and synchronizing etc.

- **5. Non-Disjoint and Disjoint Sets Approach:** Here the objective is to find the sensor cover to cover all the targets. The sensor cover is the set of sensors which collectively cover all the targets. A sensor cover is selected from the all the sensors i.e. it's a subset of complete sensor set. If it find sensor-covers with the different sensors, i.e. all the sensor-covers are the disjoint means there is no common sensor in any two sensor cover. This type of approach is called as Disjoint sets Approach. This approach is used when all sensors are of the homogeneous type. While if it determine all sensor-covers as there are some common sensors in the different sensor covers is called as Non-Disjoint Sets Approach. This approach is used when sensors are of heterogeneous type.

2.6 Our Approach

To achieve our goal we will use the strategy of dividing the total sensors into the various sets, and those sets are individually work as a sensor cover. So instead of using all the sensors together we will split them into sets and then we will use them one by one. So the lifetime of the network will equal to the sum of lifetime of all the set covers.
Thus to increase the life time of the sensor networks then we will have to find the maximum number of the set covers from the given number of sensors. As it is known that all the sensors are of homogenous type so their all characteristics are equal, their lifetime is also same. We have already discussed that the life time of the set cover is equals to the minimum lifetime of any sensor in the set. So here it clears that the lifetime of a set cover is equals to the lifetime of the lifetime of any sensor in that cover.
Now our objective is to find maximum set covers. We can divide the set of sensors into maximum set covers only if each of the set cover contains minimum sensors, i.e. if each set cover is minimal then we can find the maximum number of set covers, and as the number of set cover increase the lifetime of the network also increase.

Problem Statement

In [5] the author has defined the problem statement with the assumptions are as follows:
Given:

- N sensor nodes

$$S_1, S_2, S_3, \ldots, S_n$$

- M targets

$$T_1, T_2, T_3, \ldots, T_n$$

- P sensing ranges

$$R_1, R_2, r_3, \ldots, R_P$$

  and the corresponding energy consumption

$$E_1, E_2, E_3, \ldots, E_P;$$

- Initial sensor energy E

- The coefficients showing the relationship between sensor, radius and target:
  a[i][p][j] = 1 if sensor S_i with radius R_p covers the target T_j .
  For simplicity, we use the following notations:

- i: i^{th} sensor, when used as index

- j: j^{th} target, when used as index

- p: p^{th} sensing range, when used as index

- k: k^{th} cover, when used as index

Variables

- c[k], boolean variable, for k = 1..K;
  c[k] = 1 if this subset is a set cover, otherwise c[k] = 0.

- x[i][k][p], boolean variable, for i = 1..N, k = 1..K, p = 1..P ;
  x[i][k][p] = 1 if sensor i with range R_p is in cover k,
  otherwise x[i][k][p] = 0.

Maximize c[1] + ... + c[k] subject to

$\sum_{k=1}^{K} \sum_{p=1}^{P} x[i][k][p] * e[p] \leq E$
for i=1,2,3,....,N

$\sum_{k=0}^{n} x[i][k][p] \leq c[k]$

for all i = 1..N,

k = 1..K

$\sum_{i=1}^{N} \sum_{p=1}^{P} x[i][k][p] * a[i][p][j] \geq c[k]$ for all k = 1..K,

j = 1..M

$x[i][k][p] \in 0, 1$

$c[k] \in 0, 1$

Where k represents the maximum number of covers

Chapter 3

Greedy Method for Set-Cover Generation

3.1 Introduction to Greedy Algorithms

A game like chess can be won only by thinking ahead: a player who is focused entirely on immediate advantage is easy to defeat. While there are various other games like scrabble, to win the game we can simply choose the move which is best at time instant, without thinking for the problem of future. This kind of approach is simple and it is effective at various places that makes it a very good algorithm design method. This approach/method find the solution part by part. It always choose the best part of available solution set that provide maximum profit or which is the highest effective as a solution and add it to the final solution. This method may be harmful for some computational tasks. Various problems are there for what this method works very well e.g. Minimum spanning tree etc. The greedy approach is an optimization approach. Greedy algorithms are used to solve optimization problems. When there are various solutions are available and problem need to choose one solution out of many it can use the Greedy approach.

The name ***Greedy*** describes its nature. It selects best solution available at any particular instant of time and consider it as best Global solution or a part of Global solution it global solution contains a series of small solutions. The Greedy approach gives best result when the global solution contains single element. But when Global solution contains a series of elements for any problem then greedy approach may leads to suboptimal solution, i.e. it

doesn't provide the best solution that is required.
Greedy approach is simple and straightforward. It is directly to the point in this strategy the method is to finds the solution just by analysing the current information in the hand without taking care of the future result. It is very easy to use and implement this approach.

3.1.1 Characteristics and Features of Problems solved by Greedy Algorithms

The greedy algorithm consists of four function.

- A function that checks whether chosen set of items provide a solution.
- A function that checks the feasibility of a set.
- The selection function tells which of the candidates is the most promising.
- An objective function, which does not appear explicitly, gives the value of a solution.

3.1.2 Structure Greedy Algorithm

- Initially the set of chosen items is empty i.e., solution set.
- At each step
  - item will be added in a solution set by using selection function.
  - IF the set would no longer be feasible
    - reject items under consideration (and is never consider again).
  - ELSE IF set is still feasible THEN
    - add the current item.

As we have discussed that to maximize the lifetime of a WSNs, it required to find maximum number of set covers. Now we will discuss the Greedy Algorithms to find maximum number of set covers.
Here we will discuss the set-cover generation algorithms using greedy strategy. These algorithms generates a set cover. To find maximum number of set covers repeat the algorithm again and again until it finds all the possible set covers.

3.2 Types of Greedy Algorithms for Set Cover generation

There are two methods for generating set-covers for a wireless sensor networks using Greedy Method.

- Simple Greedy Set-cover Generation
- Greedy MSSC (Maximum Sensor Set-Cover Generation)

These algorithms takes targets and sensors as input and provide a set-cover. To find maximum number of set-cover call these algorithms again and again until it finds all the possible set covers.

3.3 Simple Greedy Set-cover Generation

3.3.1 Introduction

WSNs are the systems of small, low-powered sensing devices are arranged in a network. These sensing devices are deployed to monitor a particular area or some particular targets according to the application requirements. Coverage problem is one of the most crucial issues of the WSNs. It determines how well all the targets/area is being cover by the sensors. We are working on target coverage problem so we'll discuss the approach for target coverage only. Here Babacar DIOP has given Greedy approach for lifetime maximization problem with respect to target coverage in WSNs. The lifetime maximization problem in target coverage application can be addressed by determining" How to partition sensors into an optimal number of sets and schedule their operating intervals so that coverage requirement can be satisfied and the network lifetime can be maximized." This greedy approach produce disjoint set covers. The simulation result shows the performance of generating the sensor-covers v/s number of sensor used. This method produce the maximum disjoint set-covers those activates one after another.

3.3.2 Assumptions

- The number of targets are fixed and all targets are static.
- The locations of targets are predefined.

- Several low power sensors are deployed randomly.
- This method is using disjoint set approach.
- Sensors are being deploy randomly into the plane.
- All sensors are homogeneous

3.3.3 Set-Cover Generation

- Consider the variables as there are n-sensors as a set $S_0 = S_1, S_2, S_3, \ldots, S_n$.
- There are m targets need to be monitor as $Z_0 = Z_1, Z_2, Z_3, \ldots., Z_m$
- C is a set of sensor covers that will generate.

3.3.4 Greedy-Sensors Set Covers Generation

Algorithm 1: Simple Greedy Set-covers generation

```
Data: Set of targets Z_0 = Z_1, Z_2, ...., Z_n,
And Set of sensor S_0 = S_1, S_2, ...., S_m
Result: C_i = Number of set covers
begin;
find s[m][n]= covering matrix by sensors;
Uncovered = Z_0, C_i = NULL;
while (Uncovered ≠ NULL) or (S_i ≠ NULL) do
    for i = 1 to n do
        if s[i] ∩ Uncoverd ≠ NULL then
            x = i;
            End For Loop;
    C_i = C_i ∪ x;
    p= set of targets being cover by the x;
    Uncovered = Uncovered - p;
    S_i = S_i − x;
Return C_i
```

This algorithm can be explained as it start to finding the sensor covers from initial set of sensor covers and work until all the sensor covers get find. It finds the disjoint sensor covers. Here it use an array to identify the covered targets and other array to identify the used sensors. It checks all the sensors one by one if any sensor is covering any uncovered target it include the sensor into sensor cover. And perform same operation to find all the sensor covers.

3.3.5 Performance Evaluation

We simulate a sensor network which having 20 targets with fixed and predefined location in an area of 100*100 m^2 2-D plane. We are going to take 10 to 100 sensors to cover these targets. All the sensors are of homogeneous type and the sensing range of all the sensors is is varying from 30-50 mtrs. The complexity of the algorithm is $O(n \times m \times n)$

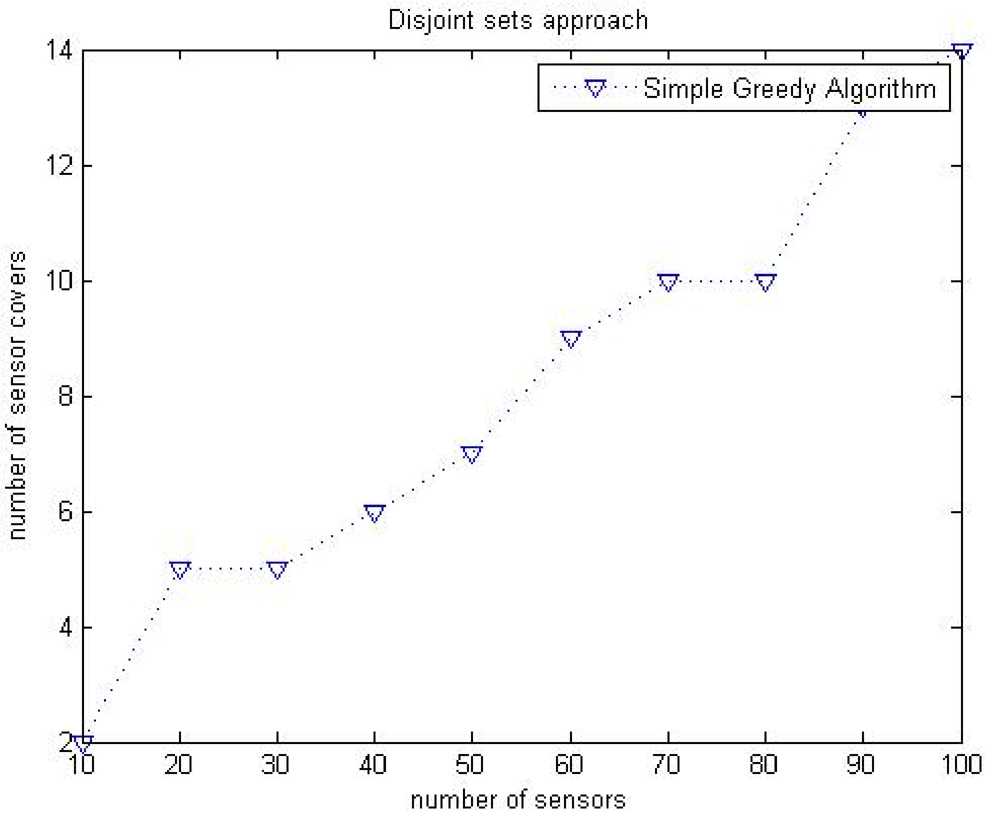


Figure 3.1: Sensing Range = 30mtr.

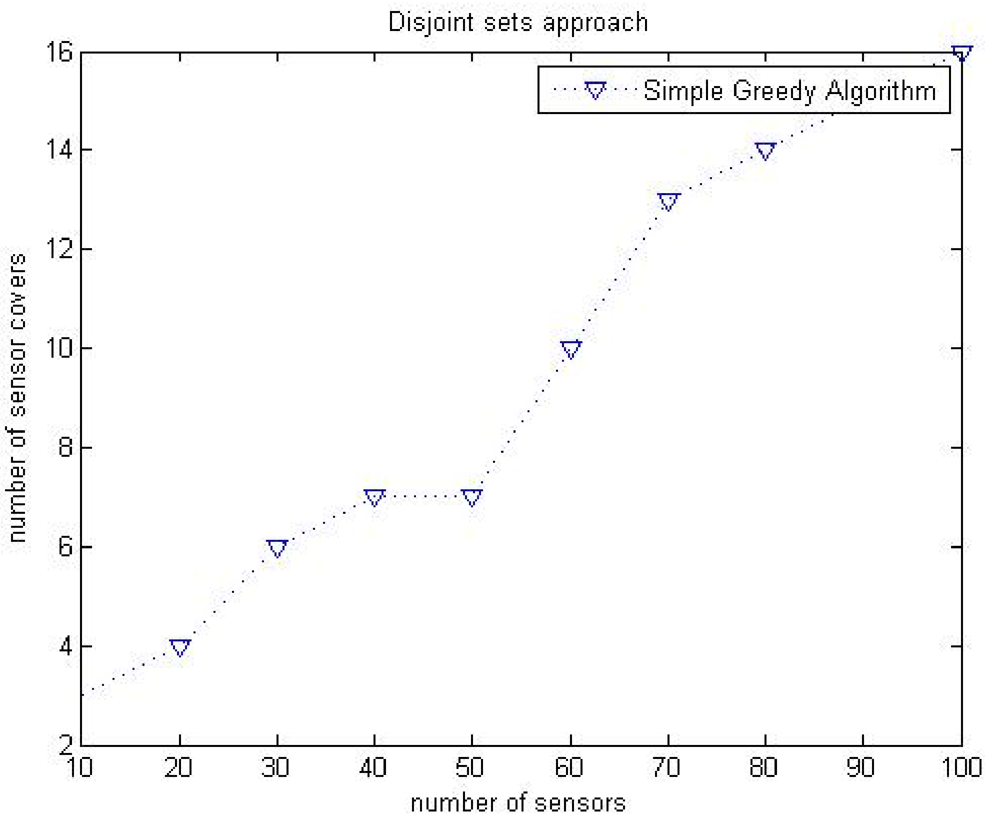


Figure 3.2: Sensing Range = 40mtr.

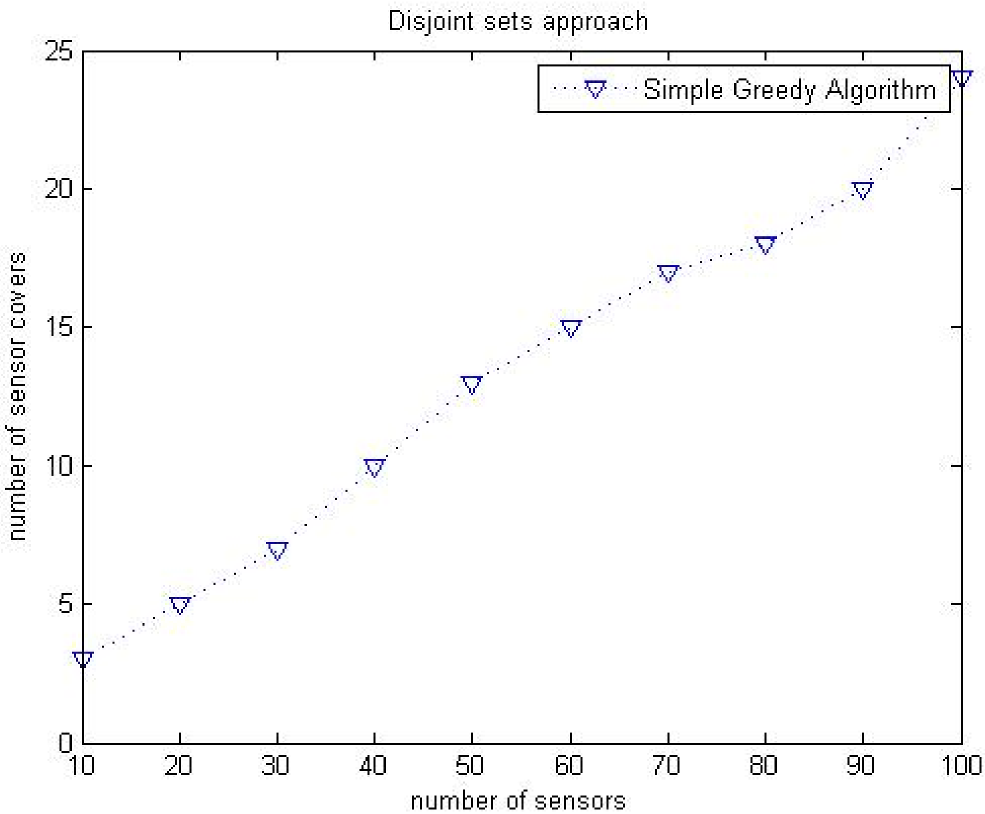


Figure 3.3: Sensing Range = 50mtr.

3.4 Greedy MSSC (Maximum Sensor Set-Cover Generation)

3.4.1 Introduction

In [7] author proposed an improved version of the simple greedy approach we discussed earlier to generate more number of set covers for network's lifetime maximization. In the previous approach it takes every sensor into set-cover which is covering at least one new target, and repeat this process again and again until we find the sensor cover. In this case it does not give best result many time such as when new node is covering maximum nodes which are already covered, thus this approach increase the size of sensor cover.

Here author of [7] has given a method in which it consider all the sensors and find the sensor which is covering maximum uncovered targets, thus this method will include the least sensors into the set cover, and as the size of the set-cover decrease then the number of the sensor-covers increase.

Algorithm 2: Greedy-MSSC Set-covers generation

```
Data: Set of targets Z_0 = Z_1, Z_2, ...., Z_n,
And Set of sensor S_0 = S_1, S_2, ...., S_m
Result: C_i = Number of set covers
begin;
find s[m][n]= covering matrix by sensors;
Uncovered = Z_0, C_i = NULL;
while (Uncovered ≠ NULL) or (S_i ≠ NULL) do
    count = 0;
    max = 0;
    for i = 1 to m do
        uc=|s[i][n]| ; Number of uncovered targets covering by the i^th sensor;
        if uc > max then
            count = i;
            max = uc;
    x= count ; the sensor which is covering maximum uncovered targets.;
    C_i = C_i ∪ x;
    p= set of targets being cover by the x;
    Uncovered = Uncovered -p;
    S_i = S_i − x;
Return C_i
```

3.4.2 Performance Evaluation

For Simulation we are considering a sensor network which having 20 targets with fixed and predefined location in an area of 100×100 m^2 2-D plane. We are going to take 10 to 100 sensors to cover these targets. All the sensors are of homogeneous type and the sensing range of all the sensors is 50 m.

Here we found that improved approach is generating more sensor covers then the traditional approach. And we know that the network lifetime is directly proportional to the number of sensor covers. The improved approach is much better than traditional approach.

The complexity of the algorithm is **O(**$n \times m \times n$**)**

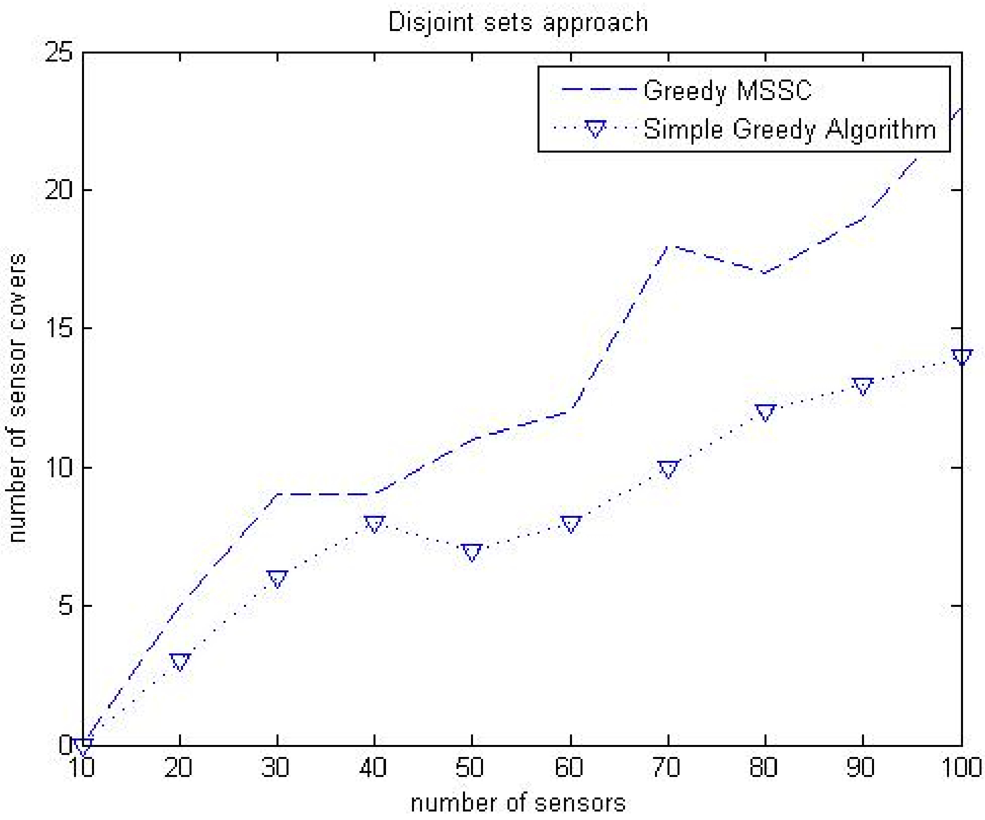


Figure 3.4: Sensing Range = 30

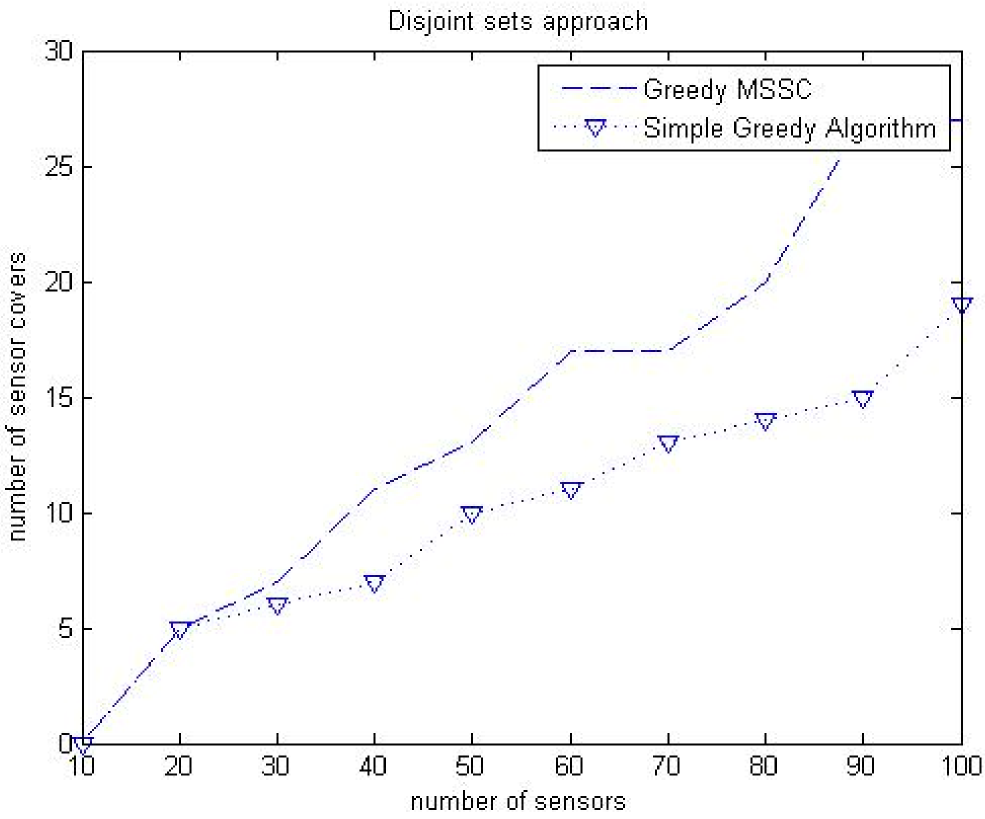


Figure 3.5: Sensing Range = 40

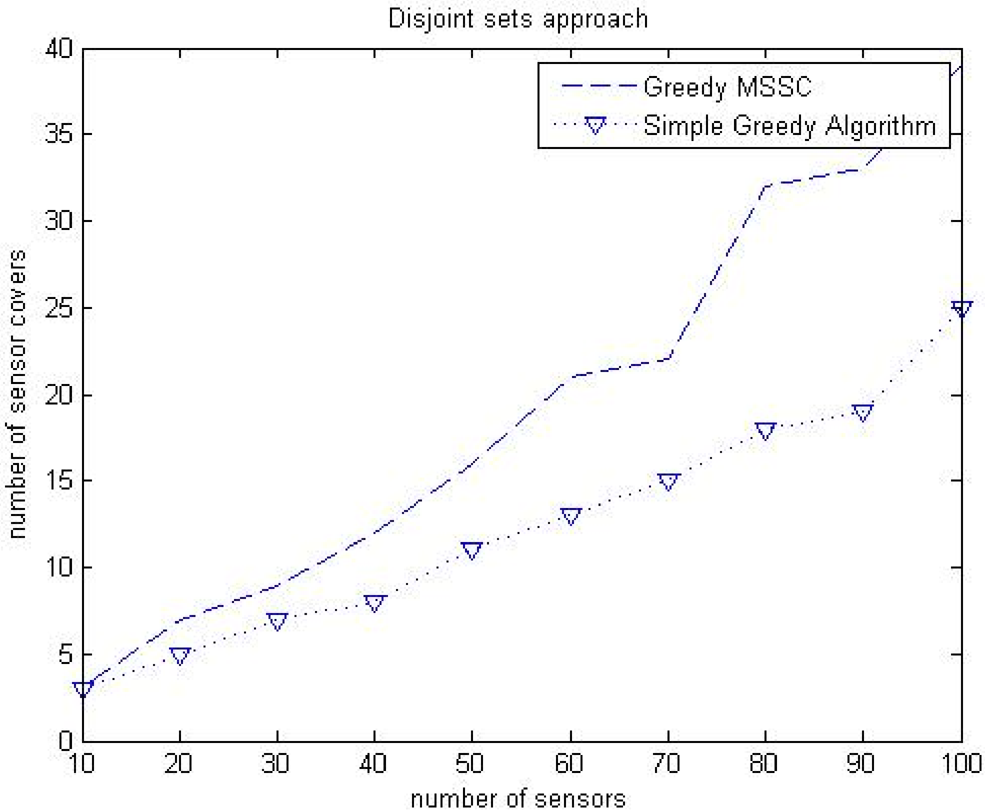


Figure 3.6: Sensing Range = 50

Chapter 4

Improvised Greedy Method for Set-Cover Generation

4.1 Introduction

Here we present more improved version of the greedy approach than Greedy MSSC, we discussed earlier. In the Greedy MSSC method it analyse all the sensors and find the sensor which is covering maximum uncovered targets, and add it to the set-cover. Then it repeat this process until either all targets get cover i.e. until it finds a set cover or all sensors comes are over. In some cases it may not give best result many times.

For example we have 4 sensors S_1, S_2, S_3, S_4 and 7 targets as $T_1, T_2, T_3, T_4, T_5, T_6, T_7$. Consider the sensors are covering the targets as follows:

- $S_1 = T_1, T_2, T_3, T_4$
- $S_2 = T_4, T_5, T_6, T_7$
- $S_3 = T_5, T_6, T_7$
- $S_4 = T_1, T_2, T_3$
- **If we go through the Greedy MSSC for set cover generation:**
- In first sensor it has to select either S_1 or S_2 because both are covering 4 uncovered targets, let us consider it select S_1 in first iteration now we have sensor-set as S_1. And covered targets as T_1, T_2, T_3, T_4, while uncovered targets as T_5, T_6, T_7.

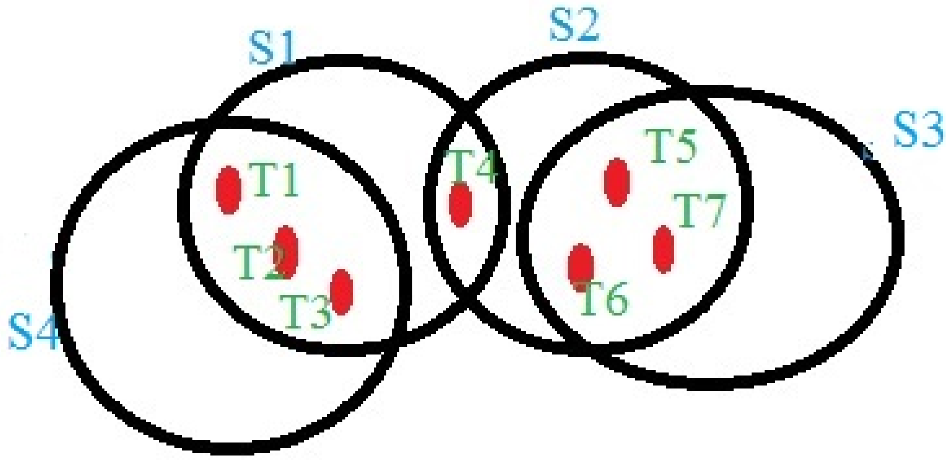


Figure 4.1: Example

- Now for second sensor selection this method have two choices S_2 and S_3 because both are covering same number of uncovered targets 3. If it selects S_2 then it gives sensors-set S_1, S_2. Now covered targets are $T_1, T_2, T_3, T_4, T_5, T_6, T_7$, and set of uncovered targets is NULL.
- If we go for another set cover generation, it is not possible.
- This method gave one set cover.

Now we propose an improved method of Greedy MSSC that always gives either better or equal result than to Greedy MSSC. In Greedy MSSC it finds all the sensors which are covering maximum number of uncovered targets, and selects one either first one or randomly and add it to set-cover and repeat this process until either set-cover is generated or all sensors are over. While here we also finds all the sensors which are covering maximum number of uncovered targets. If only one sensor is covering maximum targets then it selects that sensor and add it to set-cover, Else if more than one sensors are covering same number of targets then we see the total numbers of targets being covered by the each sensor and then select the sensor which is covering minimum targets in comparison to other sensors.

For Example Consider the above example.

Let us generate set cover using Improvised Greedy MSSC:

- In first it has two choices so it has to select either S_1 or S_2 because both are covering 4 uncovered targets, let us consider it select S_1 in first iteration now we have sensor-set as S_1. And covered targets as T_1, T_2, T_3, T_4, while uncovered targets as T_5, T_6, T_7
- For second sensor selection this algorithm have 2 choices S_2 and S_3 both are covering 3 uncovered targets. Now we have two sensors so we finds the total number of targets being cover by the S_2 and S_3, that is as follows:
  - S_2 covering 4 targets.
  - S_3 covering 3 targets.
- This method will select S_3 as second sensor and add it to set cover. Now it gives sensors-set/set-cover S_1, S_3. Now covered targets are $T_1, T_2, T_3, T_4, T_5, T_6, T_7$, and set of uncovered targets is NULL.
- If we go for another set cover generation, this method will again generate another set cover as S_2, S_4.
- This method gives 2 set cover while previous one is giving only 1.

4.2 Improvised Greedy MDSC(Maximum Disjoint Set Covers) Generation Algorithm

- All assumptions are as same as the previous approach.
- All the variables are also as same as the greedy approach.

Algorithm 3: Improvised Greedy-MDSC Set-covers generation

Data: Set of targets $Z_0 = Z_1, Z_2, ..., Z_n,$

And Set of sensor $S_0 = S_1, S_2, ..., S_m$

Result: C_i = Number of set covers

begin;

find s[m][n]= covering matrix by sensors;

Uncovered = Z_0, C_i = NULL;

while *(Uncovered ≠ NULL)or* ($S_i \neq NULL$) **do**

    P_i = Find all the sensors which are covering same as well as maximum number of uncovered targets.;

    if $P_i == 1$ **then**

        ind = index of sensor;

    if $P_i > 1$ **then**

        X[P_i,2]= *Index* and *Number* of targets being covered by the sensors;

        ind = *Index* of sensor covering minimum number of sensors;

x= find the sensor which is covering maximum uncovered s.;

$C_i = C_i \cup S_{ind}$;

tar = set of targets being cover by the ind;

Uncovered = *Uncovered* - *tar*;

$S_i = S_i - x$;

Return C_i

4.3 Performance Evaluation

We simulate a sensor network which having 20 targets with fixed and predefined location in an area of 100*100 m^2 2-D plane. We are going to take 10 to 100 sensors to cover these targets. All the sensors are of homogeneous type and the sensing range of all the sensors is 50 m. Here we found that improvised approach is generating more sensor covers then the traditional approach. And we know that the network lifetime is directly proportional to the number of sensor covers. The improved approach is much better than traditional approach.

From the performance analysis and results we can conclude that the Improvised Greedy Algorithm always gives either better or equal to the result of Greedy MSSC algorithm. But It will never give worse result than Greedy

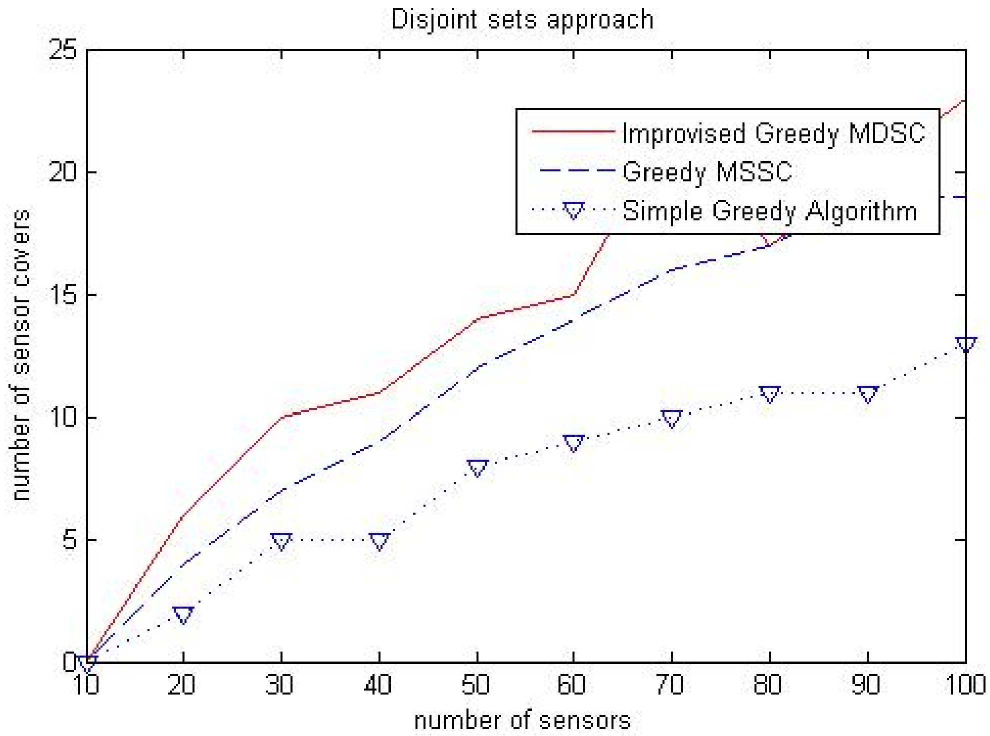


Figure 4.2: Sensing Range =30

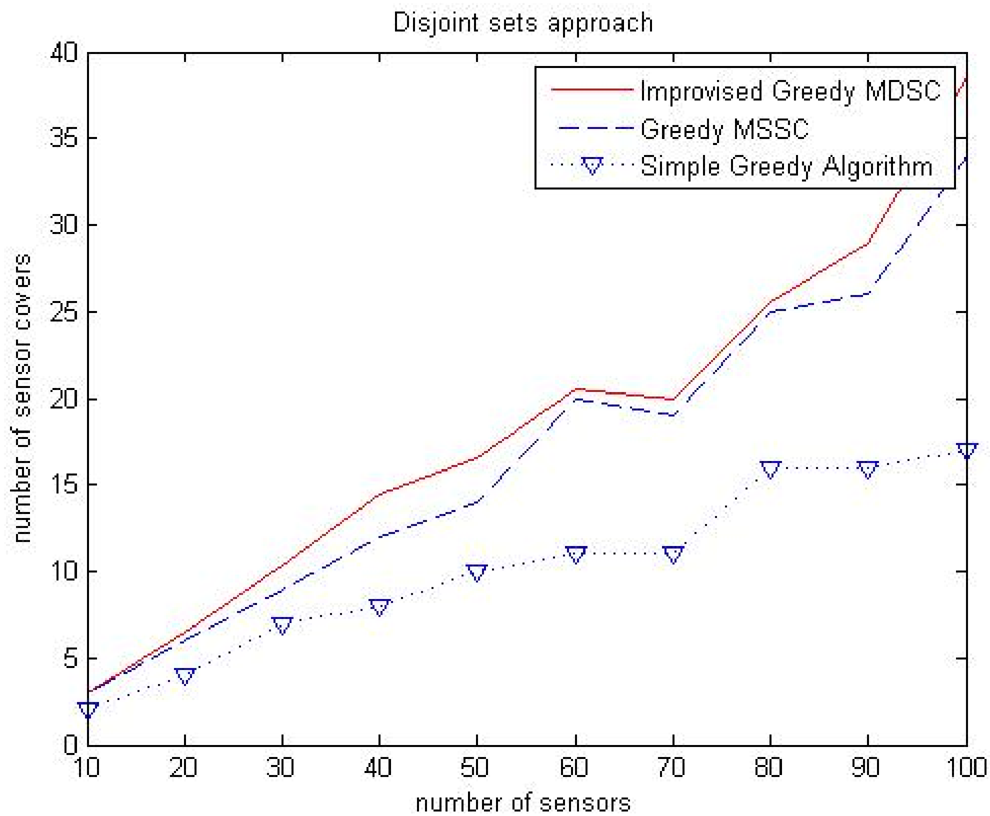


Figure 4.3: Sensing Range =40

MSSC.

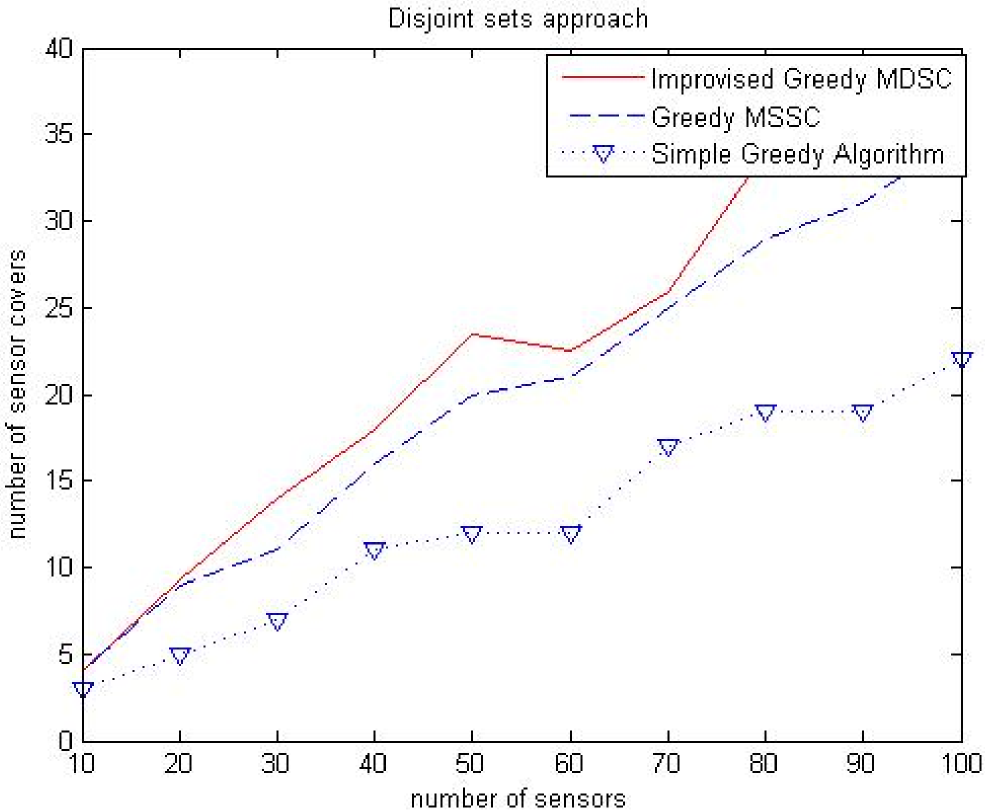


Figure 4.4: Sensing Range =50

Chapter 5

Conclusions and Future Work

Here we have find the way to maximize the lifetime of wireless sensor networks using Greedy about maximum uncovered targets with total covering targets. This work discussed the scheduling of the sensor's for WSNs with the coverage constraint is that all the targets must be covered all the time continuously. There are many schemas for maximizing the life time of the wireless sensor networks, and found dividing the sensors into various disjoint sets is an effective and good approach. Find the maximum number of the disjoint sensor covers from set of all sensor, and let the sensor cover active/work one by one. As the number of sensor cover increase the lifetime also increase. This problem can be discussed using combinatorial optimization theory.

In this work three sensor cover scheduling algorithms are presented, the Simple greedy heuristic, Greedy MSSC, and the Improvised Greedy MDSC algorithm, to solve the DSC problem that is known to be NP-complete. The simulation has performed for the various number of sensors to observe the effect of the algorithm on the performance towards the lifetime maximization problem. The results of simulation shows that the Improvised Greedy schema generates more number of the sensor-covers then Greedy MSSC one, and increase the lifetime of the network.

For the future work we are trying to combine Improvised greedy Algorithm MDSC with sensors placement techniques for reaching closer to more optimal solution. We are also trying PSO technique to maximize lifetime of the network with disjoint sets approach.

Bibliography

[1] N Aziz, K Aziz, and W Ismail. Coverage strategies for wireless sensor networks', world academy of science, engineering and technology. *International Science Index*, 26(3):2, 2009.

[2] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten H Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 131–142. ACM, 2006.

[3] Novella Bartolini, Tiziana Calamoneri, Emanuele Guido Fusco, Annalisa Massini, and Simone Silvestri. Push & pull: autonomous deployment of mobile sensors for a complete coverage. *Wireless Networks*, 16(3):607–625, 2010.

[4] Mihaela Cardei, My T Thai, Yingshu Li, and Weili Wu. Energy-efficient target coverage in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1976–1984. IEEE, 2005.

[5] Mihaela Cardei, Jie Wu, Mingming Lu, and Mohammad O Pervaiz. Maximum network lifetime in wireless sensor networks with adjustable sensing ranges. In *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on*, volume 3, pages 438–445. IEEE, 2005.

[6] Jiming Chen, Shijian Li, and Youxian Sun. Novel deployment schemes for mobile sensor networks. *Sensors*, 7(11):2907–2919, 2007.

[7] Babacar Diop, Dame Diongue, and Ousmane Thiare. Managing target coverage lifetime in wireless sensor networks with greedy set cover. In *Multimedia, Com-*

puter Graphics and Broadcasting (MulGraB), 2014 6th International Conference on, pages 17–20. IEEE, 2014.

[8] Milan Erdelj, Valeria Loscri, Enrico Natalizio, and Tahiry Razafindralambo. Multiple point of interest discovery and coverage with mobile wireless sensors. *Ad Hoc Networks*, 11(8):2288–2300, 2013.

[9] Milan Erdelj, Tahiry Razafindralambo, and David Simplot-Ryl. Covering points of interest with mobile sensors. *Parallel and Distributed Systems, IEEE Transactions on*, 24(1):32–43, 2013.

[10] Amitabha Ghosh and Sajal K Das. Coverage and connectivity issues in wireless sensor networks. *Mobile, wireless, and sensor networks: Technology, applications, and future directions*, pages 221–256, 2006.

[11] MA Jamali, Navid Bakhshivand, Mohammad Easmaeilpour, and Davood Salami. An energy-efficient algorithm for connected target coverage problem in wireless sensor networks. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 9, pages 249–254. IEEE, 2010.

[12] Koushik Kar and Suman Banerjee. Node placement for connected coverage in sensor networks. In *WiOpt'03: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pages 2–pages, 2003.

[13] Ines Khoufi, Pascale Minet, Anis Laouiti, and Saoucene Mahfoudh. Survey of deployment algorithms in wireless sensor networks: coverage and connectivity issues and challenges. *International Journal of Autonomous and Adaptive Communications Systems (IJAACS)*, page 24, 2014.

[14] Purnima Khuntia and Prasant Kumar Pattnaik. Some target coverage issues of wireless sensor network. *International Journal of Instrumentation, Control & Automation (IJICA)*, 1(1):96–98, 2011.

[15] Yong-hwan Kim, Chan-Myung Kim, Dong-Sun Yang, Young-jun Oh, and Youn-Hee Han. Regular sensor deployment patterns for p-coverage and q-connectivity in wireless sensor networks. In *Information Networking (ICOIN), 2012 International Conference on*, pages 290–295. IEEE, 2012.

[16] Deying Li and Hai Liu. Sensor coverage in wireless sensor networks. *International Journal of Sensor Networks*, 2, 2009.

[17] Jun Li, Baihai Zhang, Lingguo Cui, and Senchun Chai. An extended virtual force-based approach to distributed self-deployment in mobile sensor networks. *International Journal of Distributed Sensor Networks*, 2012, 2012.

[18] Mo Li, Weifang Cheng, Kebin Liu, Yuan He, Xiangyang Li, and Xiangke Liao. Sweep coverage with mobile sensors. *Mobile Computing, IEEE Transactions on*, 10(11):1534–1545, 2011.

[19] Xu Li, Hannes Frey, Nicola Santoro, and Ivan Stojmenovic. Strictly localized sensor self-deployment for optimal focused coverage. *Mobile Computing, IEEE Transactions on*, 10(11):1520–1533, 2011.

[20] Saoucene Mahfoudh, Ines Khoufi, Pascale Minet, and Anis Laouiti. Relocation of mobile wireless sensors in the presence of obstacles. In *Telecommunications (ICT), 2013 20th International Conference on*, pages 1–5. IEEE, 2013.

[21] Pillwon Park, Sung-Gi Min, and Youn-Hee Han. A grid-based self-deployment schemes in mobile sensor networks. In *2010 Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications*, pages 1–5, 2010.

[22] Arun K Pujari et al. High-energy-first (hef) heuristic for energy-efficient target coverage problem. *arXiv preprint arXiv:1103.4769*, 2011.

[23] Guang Tan, Stephen A Jarvis, and A-M Kermarrec. Connectivity-guaranteed and obstacle-adaptive deployment schemes for mobile sensor networks. *Mobile Computing, IEEE Transactions on*, 8(6):836–848, 2009.

[24] Bang Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 43(4):32, 2011.

[25] Hung-Lung Wang and Wei-Ho Chung. The generalized k-coverage under probabilistic sensing model in sensor networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 1737–1742. IEEE, 2012.

[26] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 28–39. ACM, 2003.

[27] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *Wireless Internet, 2005. Proceedings. First International Conference on*, pages 114–121. IEEE, 2005.

[28] Jun Xiao, Song Han, Yan Zhang, and Gaobin Xu. Hexagonal grid-based sensor deployment algorithm. In *Control and Decision Conference (CCDC), 2010 Chinese*, pages 4342–4346. IEEE, 2010.

[29] L Xu, A Nayak, and I Stojmenovic. Back tracking based sensor deployment by a robot team. *Sensor Mesh and Ad Hoc Communications and Networks, USA*, 2010.

[30] Ziqiu Yun, Xiaole Bai, Dong Xuan, Ten H Lai, and Weijia Jia. Optimal deployment patterns for full coverage and k-connectivity wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 18(3):934–947, 2010.

[31] Honghai Zhang and Jennifer C Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2):89–124, 2005.

[32] Hongwu Zhang, Hongyuan Wang, and Hongcai Feng. A distributed optimum algorithm for target coverage in wireless sensor networks. In *Information Processing, 2009. APCIP 2009. Asia-Pacific Conference on*, volume 2, pages 144–147. IEEE, 2009.

[33] Yi Zou and Krishnendu Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293–1303. IEEE, 2003.