

# COVERAGE AND DETECTION IN WIRELESS SENSOR NETWORKS

A thesis presented to Indian Statistical Institute  
in partial fulfillment of the thesis requirement for the degree of  
Doctor of Philosophy in Computer Science

by

MRINAL NANDI

under the supervision of  
Professor Bimal Roy



Applied Statistics Unit  
INDIAN STATISTICAL INSTITUTE  
Kolkata, West Bengal, India, 2013



*Dedicated to my parents*



# Abstract

A Wireless Sensor Networks (WSNs), which are two or three dimensional systems, usually consist of a large number of small sensors equipped with some processing circuit, and a wireless transceiver. The sensors have small size, low battery capacity, non-renewable power supply, small processing power, limited buffer capacity and low-power radio. They may measure distance, direction, speed, humidity, wind speed, soil makeup, temperature, chemicals, light, and various other parameters. The sensors are autonomous devices with integrated sensing, processing, and communication capabilities.

In this thesis, we consider ‘coverage problem’ and ‘detection problem’ in Wireless Sensor Networks (WSNs) in grid as well as in continuous domain. Sensor networks aim at monitoring their surroundings for event detection and object tracking. But, due to death of a sensor or due to obstructive, false detection may occur. Also false signal can be transmitted due to noise or faulty sensors. For detection of an event or events in a region, called Region Of Interest (ROI), coverage is essential in WSNs, i.e., every point of ROI should be in the sensing disc of at least one sensor. In case of grid, each vertex and in case of continuous ROI all the points of the ROI should be covered by at least one sensor. When sensors are deployed from air on some previously fixed points (vertices) in the Region of Interest (ROI), they may not fall on the target vertices. So, some part of the ROI may be uncovered by the sensors. In this thesis, we discuss optimal placement of sensors and coverage criteria in  $\mathbb{R}^2$  and  $\mathbb{R}^3$  and for cylindrical grid. We use graph theory and geometry to solve the problems.

Optimal placement of sensors may not be possible due to random deployment. So there may be uncovered area in ROI. We also consider the problem, how one reduce the uncovered area in random deployment scenario. To reduce the uncovered area, extra sensors are usually deployed on some randomly chosen vertices. We develop a new strategy for deployment of extra sensors. Uncovered area i.e., sensing holes can also be repaired using actuators. We develop three strategies for actuator to minimize the travel of the actuator.

Sensor could fail at runtime for various reasons such as power depletion, hardware defects etc. These sensors are known as faulty sensors. We consider the problem of distributed fault detection in wireless sensor network (WSN). In particular, we consider how to take decision regarding fault detection in a noisy environment as a result of false detection or false response of event by some sensors, where the sensors are placed at the center of regular hexagons or square and an event (or events) can occur at any number of hexagons. We propose fault detection schemes that explicitly introduce the error probabilities into the optimal event detection process. We introduce two types of detection probabilities, one for the center node, where the event occurs and the other one for the adjacent nodes. We also introduce probability for correct response and wrong response. We develop two schemes under the model selection and multiple model selection procedure and discuss two interesting special situations. We consider two different scenario: (i) at most one event can occur and (ii) any number of events can occur. We use classical Neyman Pearson hypothesis approach, decision theoretic approach, Bayesian approach, near optimal method and a new method to find solution of detection problem.

# Acknowledgements

*“Does the path know its own end, that the lost flower and silent song make their way to, where in starlight unceasing sorrow has its Festival of Light?” – Rabindra Nath Tagore (Lipika)*

Looking back at the last few years that I have spent at Indian Statistical Institute as a doctoral student, I see several people who have significantly influenced the course of my thesis, as well as my life. I take this opportunity to offer my gratitude to all who stood beside me in the hardest of times.

First and foremost, I extend my heartfelt gratitude to my PhD supervisor, Prof. Bimal Roy, Director, Indian Statistical Institute, Kolkata. Every research problem that we have discussed, debated upon and solved, has enriched my academic outlook; and every morsel of experience that he has kindly shared with me will continue to add value to my life. I will always be grateful to Bimal-sir for being my friend, philosopher and guide during the last few years; primarily because he kept his sanity and sense of humor alive whenever I lost mine! I have always idolized Prof. Bimal Roy whose active support and unparalleled vision has inspired me.

It was a delight working with my collaborators and co-authors, Prof. Anup Dewanji of ISI Kolkata, Prof. Amiya Nayak of university of Ottawa, Prof. Xu Li of Huawei Technologies Ottawa, Dr. Santanu Sarkar of CMI Chennai, Dr. Avishek Adhikari of University of Calcutta, Mr. Subrata Parui of IMBIC at Kolkata and Mr. Anubhab Baksi of University of Calcutta. Their academic insight and vision have taught me a lot over the last few years. I would like to specially thank Subhamoy-da, Mridul and Sourav of ISI kolkata for mentoring and supporting me in numerous ways, academic and otherwise, during the course of this thesis. I can never forget Arnab Lahiri, who wrote many C-codes for me. I can not forget Subhadeep and Sourav, who edited several of

my papers, and this thesis, whenever I requested them to do so.

I would like to express my sincerest gratitude to the esteemed anonymous reviewers of my thesis for their invaluable comments, which helped me improve the technical presentation and editorial quality of the thesis to a great extent.

I am thankful to Applied Statistics Unit, ISI Kolkata for supporting me during the formative years of my doctoral studies; Centre of Excellence in Cryptology, ISI Kolkata, funded by DRDO, Government of India, for supporting parts of my research; and DST, Government of India. I humbly acknowledge the kind support of Cryptology Research Society of India, and that of Mr. Amitabha Sinha (our beloved Amitabha-da), during my stay at ISI Kolkata. I am also thankful to the Director's office, the Dean's office, the ASU, BIRU and CoEC offices, and all administrative sectors of the Institute, for their continuous support to make my life smooth at the Institute.

I am indebted to Rana-da, Palash-da and Kishan-da for several invaluable discussions during the tenure of my PhD. I would also like to thank my past and present colleagues at ISI Kolkata, Sibudha, Somitra, Sumanta, Pinaki, Ashish, Sushmita, Srimanta, Sumit, Tapas, Indranil, Sanjay, Subhabrata, Somindu, Shashank, Samiran, Nilanjan, Avik, Pratyay and Goutam, for providing me with an adorable social support. Special thanks to Rishi, Sourav and Koushik as it was only through the uncountably many philosophical and technical discussions with them that I could see myself through the most arduous times during the last few years. I also like to thank my University colleagues, for their great support. They took my duty as their own whenever I need a break from my duty.

My family has always been the 'happy place' for me; even in the hardest of times. This thesis could not be completed without the ardent support and inspiration from my parents, *Baba* and *Maa*, who made me what I am by resting their unflinching faith in me at every step of my life. Undoubtedly the most significant contribution towards this thesis has been made by *Sujata*, my wife, who painstakingly accepted the thesis as her co-wife, night after night, even during the hard times in her life. Last but not least, I thank, my old friend Shambhu Nath Ghosh, who told me that – “You can do all works after your Ph.D., but you can not do Ph.D. after all your works” ... many many thanks, all of you!



# Contents

Abstract . . . . .	iii
Contents . . . . .	vii
List of Tables . . . . .	xi
List of Figures . . . . .	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Wireless Sensor Network and it's applications . . . . .	1
1.2 Coverage, Deployment and Operation of WSN and WSNAN . . . . .	4
1.3 Motivation . . . . .	9
1.4 Our Contribution and Thesis Plan . . . . .	12
<b>2 Previous Work</b>	<b>19</b>
2.1 Coverage Problem . . . . .	19
2.2 Detection Problem . . . . .	24
2.3 Related Areas . . . . .	26
<b>3 Coverage in Continuous Domain</b>	<b>29</b>
3.1 Introduction . . . . .	29
3.2 Assumptions and Definitions . . . . .	33
3.3 A Result on Coverage Problem . . . . .	36

3.4	Coverage Problem in $\mathbb{R}^2$ . . . . .	38
3.5	Simulation Results . . . . .	46
3.6	Coverage Problem in $\mathbb{R}^3$ and Simulation Results . . . . .	49
3.6.1	Simulation Results . . . . .	50
<b>4</b>	<b>Deterministic Covering of Cylindrical Grid</b>	<b>51</b>
4.1	Introduction . . . . .	51
4.1.1	Definitions . . . . .	51
4.2	Finding the domination numbers of some cylindrical grid graphs	54
4.3	Bounds on the domination numbers of $P_5 \square C_n, n \geq 3$ . . . . .	63
<b>5</b>	<b>Sensor Placement</b>	<b>71</b>
5.1	Introduction . . . . .	71
5.2	Preliminary Assumptions . . . . .	71
5.2.1	Some Parameters . . . . .	72
5.2.2	Some Definitions and Notation . . . . .	73
5.3	Three Algorithms . . . . .	74
5.3.1	First Algorithm . . . . .	75
5.3.2	Second Algorithm . . . . .	75
5.3.3	Third Algorithm . . . . .	76
5.3.4	Simulation Result . . . . .	77
5.3.5	Observations . . . . .	78
5.4	Theoretical Results . . . . .	79
5.4.1	Results on $L_1$ . . . . .	79
5.4.2	Results on $T(i, j)$ . . . . .	82
5.4.3	Results on $N$ . . . . .	82
<b>6</b>	<b>Single Event Detection in a Faulty WSN</b>	<b>87</b>
6.1	Introduction . . . . .	87

6.2	Assumptions . . . . .	88
6.3	Problem Statement and Notations . . . . .	90
6.4	Theoretical Analysis of Fault Detection . . . . .	92
6.4.1	Error Probabilities for Nodes . . . . .	94
6.4.2	Error Probabilities and Test for Base Station . . . . .	95
6.4.3	Boundary Case . . . . .	97
6.4.4	When more sensors can detect the Event Square . . . . .	98
6.5	Calculations and Observations . . . . .	98
6.5.1	Calculation of errors for each sensor and Observations . . . . .	99
6.5.2	Calculation for Bayes Test and Observations . . . . .	100
6.5.3	Calculation for MP Test and Observations . . . . .	103
6.6	Simulation Results . . . . .	105
6.7	Some Special Cases . . . . .	107
6.7.1	When sensors always send the message correctly . . . . .	107
6.7.2	When the sensors can detect the event square without any error . . . . .	107
6.7.3	When one center and four distance-one sensors can de- tect the event square, i.e., $p_3 = 0$ . . . . .	108
6.8	When sensors are placed at the centers of regular hexagons . . . . .	109
<b>7</b>	<b>Detection of event in faulty network system</b>	<b>111</b>
7.1	Introduction . . . . .	111
7.2	Statement of the Problem and Assumptions . . . . .	112
7.2.1	Notations and Assumption . . . . .	113
7.3	Theoretical Analysis of Fault Detection . . . . .	115
7.3.1	Model Selection Approach . . . . .	115
7.3.2	Multiple Model Selection . . . . .	117
7.3.3	Bayesian Model Averaging . . . . .	118

7.4	Some Important Considerations and Error Probabilities . . . . .	119
7.5	Simulation Study . . . . .	121
7.6	Discussion . . . . .	125
7.6.1	Role of the neighboring nodes . . . . .	125
7.6.2	Estimation of the parameters . . . . .	125
7.6.3	Incorporation of heterogeneity and uncertainty in pa- rameters . . . . .	126
7.6.4	When more sensors can detect the event square . . . . .	127
<b>8</b>	<b>Detection of Events in General Situation</b>	<b>129</b>
8.1	Introduction . . . . .	129
8.2	Statement of the Problem and Assumptions . . . . .	131
8.2.1	Notations and Assumptions . . . . .	132
8.3	Theoretical Analysis of fault Detection . . . . .	133
8.3.1	Model Selection Approach . . . . .	134
8.3.2	Algorithm to find Near-Optimal Model . . . . .	134
8.4	A New Approach To Find Event Hexagons . . . . .	138
8.5	Two Special Cases . . . . .	142
8.5.1	Model $\{N_4, N_5\}$ vs Model $\{N_0\}$ . . . . .	143
8.5.2	Model $\{N_1, N_4\}$ vs Model $\{N_0, N_3\}$ . . . . .	144
<b>9</b>	<b>Conclusion and Future Work</b>	<b>147</b>
	<b>Bibliography</b>	<b>153</b>

# List of Tables

1.1	Classification of the Coverage Problem and Previous Works . . .	11
1.2	Classification of the Coverage Problem and Our Contribution . .	15
3.1	Simulation results for the proportion of area covered for two strategies (St. 1 and St. 2). . . . .	47
3.2	Simulation for proportion of coverage area for two strategies in $\mathbb{R}^3$ . . . . .	50
5.1	Simulation Results . . . . .	77
6.1	Calculation of errors for some values of the parameters . . . . .	100
6.2	Calculation of Bayes test for some values of the parameters . . .	102
6.3	Calculation of MP test for some choice of the parameters . . .	104
6.4	Simulated and theoretical values of errors . . . . .	105
6.5	Simulation of proportion of correct detections by Bayes and MP tests . . . . .	106
7.1	Values of $P_1, P_2, c$ and $d$ for $p_c = 0.9$ . . . . .	122
7.2	Simulation of estimated success probabilities and number of searches for different threshold values ( $C$ ) and some values of the parameters with $p_c = p_1 = 0.9$ . . . . .	123
7.3	Simulation of estimated probabilities for some values of parameters	128

8.1	Performance of our algorithms under several models . . . . .	137
8.2	Simulated and exact probabilities of selecting a model when the corresponding model is true . . . . .	146

# List of Figures

1.1	Covering of a rectangular ROI without sensing hole . . . . .	4
3.1	Sphere packing of a rectangular ROI . . . . .	33
3.2	Nodes placed in ROI, which partitioned into regular hexagons .	34
3.3	Sensing discs in different situations . . . . .	36
3.4	Intersection of two discs of different radii, $\theta = \cos^{-1}(\frac{a^2+d^2-r^2}{2ad})$ , $\phi = \cos^{-1}(\frac{r^2+d^2-a^2}{2rd})$ . . . . .	38
3.5	Hexagonal tiling of ROI . . . . .	40
3.6	Intersection of two discs of radius $r$ in two different cases . . . .	42
3.7	One of the 12 triangles which partitioned a regular Hexagon . . .	45
3.8	Proportion Coverage area in $\mathbb{R}^2$ for different distribution. . . . .	48
3.9	Face-Centered Cube (dots are center of spheres) . . . . .	50
4.1	Two different looks of the cylindrical grid graph $P_5 \square C_5$ . . . . .	52
4.2	Modified concatenation of the dominating sets of $P_5 \square C_6$ and $P_5 \square C_3$ to get the dominating set for $P_5 \square C_9$ . . . . .	53
4.3	Vertices with bold circle form a minimum dominating set for $P_3 \square C_3$ and $P_3 \square C_4$ . . . . .	57
4.4	Vertices with bold circles form a minimum dominating set for $P_4 \square C_3$ , $P_4 \square C_5$ and $P_4 \square C_9$ . . . . .	59
4.5	Figure showing non admissibility of $\gamma(P_4 \square C_9) = 9$ . . . . .	61

4.6	Vertices with bold circle form a minimum dominating set for $P_4 \square C_4$ , $P_4 \square C_6$ and $P_4 \square C_7$ . . . . .	62
4.7	Vertices with bold circle form a minimum dominating set for $P_5 \square C_3$ . . . . .	67
4.8	Vertices with bold circle form a minimum dominating set for $P_5 \square C_4$ . . . . .	68
4.9	Vertices with bold circle form a minimum dominating set for $P_5 \square C_5$ . . . . .	69
4.10	Vertices with bold circle form a dominating set for $P_5 \square C_9$ . . . . .	69
4.11	Vertices with bold circle form a dominating set for $P_5 \square C_6$ and $P_5 \square C_7$ . . . . .	70
6.1	Nodes placed in cells of the ROI . . . . .	89
6.2	Detection and response probabilities when ROI is normal . . . . .	92
6.3	Detection and response probabilities when event occurs . . . . .	93
8.1	ROI partitioned into regular hexagons and Nodes placed in centers ( $N_i, i = 1, 2, \dots, 6$ are the adjacent nodes of $N_0$ ) . . . . .	131
8.2	Average search vs. Success and C.v. of search vs. Success graph . . . . .	140
8.3	Nodes placed at the center regular hexagons . . . . .	142



# Introduction

In this chapter, we describe sensor, wireless sensor network, actuator and their applications. We describe the deployment of sensors in a region of interest, relocation of sensors, the operation of wireless sensor network and different modes of wireless sensor network. We also describe the motivational problems which was standing open in literature. Finally we present our contribution in this thesis and the thesis plan.

## **1.1 Wireless Sensor Network and it's applications**

Traditional sensor networks use wired communication, whereas wireless sensor networks (WSNs) provide radically new communication and networking paradigms, and myriad new applications. A WSN usually consists of a large number of small sensors equipped with some processing circuit, and a wireless transceiver. The sensors have small size, low battery capacity, non-renewable power supply, small processing power, limited buffer capacity and low-power radio. They may measure distance, direction, speed, humidity, wind speed, soil makeup, temperature, chemicals, light, and various other parameters. The sensors, also known as nodes, are autonomous devices with integrated sensing, processing, and communication capabilities. There are two types of WSN; one contains one or more actuators (i.e., robots), another does not. A wireless

sensor network with actuator(s) is known as wireless sensor-actuator network (WSAN). The objective of a WSN or WSAN is to detect event(s) within some pre-fixed region, known as the *Region of Interest or ROI*, which may be two or three dimensional. An actuator moves in the ROI, where the sensors are placed or have to be placed. Actuators can move with sensors in hand and can take sensors from ROI or place sensors at a point of ROI [77]. The ROI may be a grid or a connected region.

Sensors or nodes in a sensor network are densely deployed from air at random or placed by actuators using some specific strategy, either very close to or inside the ROI. Once the sensors are deployed, they are expected to self-configure into an operational wireless network, and must work unattended. Since the energy budget of individual sensors is limited, to ensure the longevity of the network, the transmission range needs to be restricted and the redundant sensors should be put in to sleep mode. Since sensors are dropped randomly some portion of the ROI may not be covered by any sensor. On the other hand, some sensors could be marked redundant in terms of local sensing coverage and these sensors are called *passive sensors*. Passive sensors could either be deployed on purpose or determined by some area coverage protocol [30]. Sensors may be placed deterministically; even then due to dead sensors some portion of the ROI may be uncovered. Dead sensors can neither detect an event nor send any information.

A sensor could fail at runtime for various reasons such as power depletion, hardware defects etc. These sensors are known as *faulty sensors*. There are several reasons for a sensor to be faulty. A faulty sensor may send wrong information. Sharma et al. [86] characterized the different types of fault and fault detection methods. They characterized four different detection methods for detecting faults: (i) rule based methods, (ii) estimation methods, (iii) time series analysis based methods and (iv) learning based methods. Ni et al. [78] categorized the three types of fault models, although there exist other models.

Recent advances in wireless communications and electronics have enabled the development of low-cost, low power, and multi-functional wireless sensor nodes which consist of sensing, data processing and communication components. One of the unique features of a WSN is random deployment in inaccessible environments and cooperative effort that offers unprecedented opportu-

nities for a broad spectrum of civilian and military applications, for example industrial automation, environment and habitat monitoring, eco-physiology, condition-based equipment maintenance, disaster management, emergency response, military surveillance, national security, and emergency health care [1, 3, 82]. Sensor Networks are also useful in detecting topological events such as forest fires [25].

Current research and implementation efforts are mostly oriented toward *static sensors* and a single *static sink* that collects information from sensors. A static sensor cannot move without external help whereas a mobile sensor can relocate in the ROI. The sink may be the actuator or may be a *base station*. If the sink is the base station then the information sent to the base station directly from a sensor (known as *single hop*) or via other sensors (known as *multiple hop*). Health care is an example of single-hop network. In a multi-hop scenario, reports from individual sensors are sent to other sensors, where they can be combined with other sensor reading or simply retransmitted to the other sensors until reports are sent to a sink node, which is capable of communicating with the base station. Base stations are usually static, it may be a computer or may be a police station [62].

Sensor networks are used to determine event regions and boundaries in the environment with a distinguishable characteristic [12, 45, 79]. The basic idea of distributed detection is to have each independent sensor make a local decision (typically, a binary one, i.e., whether or not an event has occurred), and then combine these decisions at a *fusion sensor* (a sensor which collects the local information and takes the decision) to generate a global decision or send the information to the base station. Two sensors can communicate to each other either directly or via other sensors. If each sensor can communicate to all other sensors in the WSN then the WSN is called *fully connected*. A sensor can send information to a base station via other sensors. Optimal distributed design to make a decision has been sought under both the Bayesian and the Neyman-Pearson performance criteria [93, 94].

Energy saving of a WSN is also very important. To save energy in a three dimensional ROI, deployments of sensors using a lattice pattern is considered in [1]. Almost all the works show their efficiency in terms of energy, either the consumption battery power of sensors or the length traveled by the actuator(s).

## 1.2 Coverage, Deployment and Operation of WSN and WSNAN

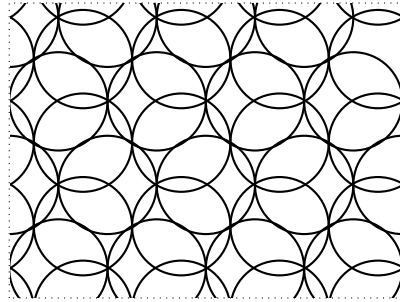


Figure 1.1: Covering of a rectangular ROI without sensing hole

Sensor networks aim at monitoring their surroundings for event detection [3,66]. Because of this surveillance goal, *coverage* is the functional basis of any sensor network. In order to fulfill its designated surveillance tasks, a sensor network must cover the Region of Interest (ROI) without leaving any *internal sensing hole* [4, 5, 19, 27] (see Figure 1.1). Internal sensing hole is a part of the ROI which is not covered by any sensor node for detection of events. The sensors can detect an event inside a surrounding disc (called *sensing disc*) of some radius (called *sensing radius*) and centered at that sensor. Sensors are so small we can think of a sensor geometrically as a point. A point will not be covered by a set of sensors if the point does not belong to any one of the sensing discs of the sensors. The aim of the well known coverage problem is to place sensors on the ROI in such a way that they cover the ROI with minimum number of sensors. If we place the sensor in a deterministic way then it is just a geometrical problem. There are other types of coverage problem. One of them is to cover maximum area of the ROI with a fixed number of sensors. This problem is not exactly the same as the above problem. Coverage of a WSN will be easy if we placed the sensors deterministically.

However, it cannot be expected that sensors are placed in a desired way at initiation as they are often randomly dropped. Sensors are densely deployed in the ROI, in general, which is a bounded subset of  $\mathbb{R}^2$ . Even after the ROI is fully covered by the sensors, wrong information can be sent by some faulty sensors or sensors may fail to detect the event. Moreover a non-faulty sensor can send wrong information due to noise or obstructions. In a WSN, sensors can be placed by robot which is also known as *actuator*. If sensors are mobile, they can place themselves without any external help. But as physical movement consumes a large amount of energy for the sensor nodes, a movement assisted sensor placement scheme is preferred [7, 24, 48, 49, 61].

So far, a number of movement-assisted sensor placement algorithms have been proposed. An exclusive survey on these topics is presented by Li et al. [57]. Analysis also been carried out for maximum distance covered and expected distance covered by the actuator(s) or by the mobile sensors to achieve the full coverage [44]. There is an extensive literature on sensor positioning and repositioning. Younis and Akkaya [97] provide a survey of models, requirements and strategies that affect sensor deployment.

There is another type of coverage problem, called  $k$ -coverage. If every point of the ROI is covered by not less than  $k$  sensors then the ROI is called  $k$ -covered [99]. Sensors may detect the direction of other sensors and the desired event(s). Network consisting of these type of sensors are known as direction sensor networks. One can suitably activate some passive sensors and deactivate the active sensors such a way that the life time of the network is maximized [55]. There are several papers which consider only the coverage problem of sensor networks. Tseng and Huang [39] formulate the problem as a decision problem, whose goal is to determine whether every point in the service area of the sensor network is covered by at least  $k$  sensors, where  $k$  is a predefined value.

Several works have sought to find an efficient algorithm for placing discs to cover a specific convex region in  $\mathbb{R}^2$ , like squares and equilateral triangles. When the set is a convex and bounded set the problem is referred to as a *covering problem* in the literature. Several variations can be found in [20, 35]. Silva et al. present homological criteria for covering in two dimensional space [87].

In a dense network, a target is covered by more than one sensor. The

grade of this depends on the sensing range and density of the network. It is also possible that one sensor covers more than one target. The goal is to keep active only the sensors necessary to cover an area. Taking it a step further, we could schedule sets of sensors, all covering the same area, to be active in turns, saving energy while keeping coverage of the area. To do so one needs to partition the set of all sensors into mutually exclusive subsets, known as *covers*. The first step is to identify the parts of the area covered by different sensor nodes. A field is a set of points which are required to cover. Two points belong to the same field if and only if they are covered by the same set of sensors. The area is modeled as a collection of fields defined above. A field has the property that any point inside the field is covered by the same set of sensors. Seetharaman and Minai [85] gave an algorithm which computes the disjoint covers successively, selecting sensors that cover the critical element (field covered by a minimal number of sensors), giving priority to sensors that cover a high number of uncovered fields.

Coverage of the ROI is an important goal. Moreover the ROI is fully covered or not, detection of event(s) is also an important objective of a WSN or a WSAN. We summarize different aspects of WSNs or WSANs as follows:

A) Detection of events may be classified according to time and space in two main aspects:

1. Spatial Distribution (Localized/Distributed):

The events of interest may be spatially localized. Wildlife tracking, vehicle tracking, perimeter breaches and forest fires are considered as such. They are usually detected by a small number of sensors [8] within whose sensing range the events are taking place. The only concern is to locate the current position of the target and plot the movement path. The information received by the sensors is spatially correlated.

2. Temporal Distribution (Discrete/Continuous):

Measuring temperature is a procedure that can be scheduled at regular intervals during a day. On the other hand, monitoring industrial machinery or seismic data requires active sensing at all times [85].

B) Sensing Models may be classified according to space in two main aspects:

1. Boolean or 0/1 Model:

We may use a circle as an abstraction of the sensors. The sensor's location is the center of the circle and the area of the circle is its sensing disc. The sensor provides full coverage within its sensing disc and none outside it, i.e., a sensor can detect an event if and only if the event occurs in the sensing disc. There is no concept of sensing intensity or ability.

2. Continuous Model:

Taking in consideration that sensing ability diminishes as distance increases and that sensing devices have different hardware features, a more realistic way is to express sensing intensity  $S(s, p)$  at any point  $p$  in the field by a sensor  $s$  at a distance  $d(s, p)$  from the sensor  $s$  is the following:  $S(s, p) = \frac{l}{(d(s, p))^k}$ , where  $l$  and  $k$  are hardware dependent parameters [67].

C) Node Deployment Strategies may also be classified by two main aspects:

1. Deterministic / Manual Placement:

We deploy sensors over a field uniformly, according to a predefined shape. An example of a uniform deterministic coverage is a grid based sensor deployment where sensors are located on the intersection points of a grid (known as vertices). This requires manual placement, which is realistic for small number of nodes, and in an accessible environment. This placement ensures complete coverage of the field with the minimum number of sensors. The number of sensors needed to cover an area  $A$  is given by  $n = \frac{2A}{\sqrt{(27)r^2}}$  Where  $r$  is the sensing radius,  $n$  is the required number of sensors, and  $A$  is the area covered [96].

2. Stochastic Placement:

In hostile or inhospitable environments, it is a necessity to deploy sensors from a plane, in order to gather data of interest. In this case, sensors are deployed randomly, and since they are of low cost, we deploy redundant sensors to increase connectivity, coverage and to prolong network lifetime.

D) There are two types of coverage protocols:

1. Area Coverage:

Given a region and given a set of sensors one has fixed a schedule to activate sensors that guarantees that at any time, all points of the target region are covered by active sensors, maximizing the network lifetime.

2. Point Coverage:

Given a set  $S$  of  $N$  sensors, and a set  $T$  of  $M$  targets, one has fixed a schedule to activate sensors that guarantees that at any time, all targets can be covered by active sensors, maximizing the network lifetime. The maximum disjoint set covers and the maximum lifetime are two different problems. Li and Yin proposes an algorithm that finds the schedule that produces the maximum lifetime, instead of trying to find the maximum number of mutually exclusive sets.

E) Mobility of sensors: Sensors can be classified according to the mobility in two different categories:

1. Static Sensors:

A static sensor cannot move without external help. Usually actuators relocate static sensors to the desired place.

2. Mobile Sensors:

Mobile sensors can move within the ROI without any external force or without any actuator. The potential field technique is one of the common methods to guide the movement of the sensors. Potential field technique [38,42] is used for robotic applications such as local navigation and obstacle avoidance. Mobile sensors and objects in the environment exert virtual repulsive force. The vector of that force is calculated and given as direction to the sensor's mobility system. In this way sensors seem to push away one another and are pushed by obstacles of the environment. The sensors will keep moving till the static equilibrium state is reached. This approach does not require models of the environment or communication between sensors.



## 1.3 Motivation

There are two important problems sensor networks: the coverage problem and the detection problem. Both areas have been the subject of research but several important problems remain open. We discuss some of them in this thesis.

### Coverage Problem

The coverage problem can be considered in two different scenarios:

1. The ROI is a set of discrete points or a grid, which may be a cylindrical grid or a rectangular grid with a square as an unit, in this case we have to cover all the vertices of the grid and an event can occur only at finitely many points.
2. The ROI is a bounded continuous region, in which case we have to cover all the points of the ROI and an event can occur at uncountably many points.

There are two different ways of placement of the sensors:

1. Deterministic placement, in which case ROI will be fully covered with a sufficient number of sensors. If we place the sensor in a deterministic way then the coverage problem is just a geometrical problem in case of continuous region and a graph theoretical problem in case of a grid.
2. Random deployment from air, in which case some points of the ROI may remain uncovered even if we use a large number of sensors.

So in case of random deployment, when the ROI is not covered by a set of sensors, the actuators are used to cover the ROI. If full coverage is not possible, then the actuators are used to minimize the uncovered area. In WSANs, sensors can be placed or relocated by mobile actuator(s). If sensors are mobile, they can place themselves without any external help. But as physical movement consumes a large amount of energy for the sensor nodes, a movement assisted sensor placement scheme is preferred [24, 61]. Uncovered area in the ROI can be covered by activating the passive sensors, when required.

Coverage is the main goal for a WSN, but due to shortage of sensors, random deployment of sensors and dead sensors, we cannot avoid uncovered regions in the ROI. Moreover, an actuator may not be available where needed, or the ROI may be such that we cannot use actuators. In that case we should calculate the uncovered area. To the best of our knowledge there is no work on estimating the uncovered area in the random deployment scenario. Our target is to calculate and develop strategies to reduce the uncovered area of ROI. There are several methods to cover the ROI or minimize uncovered area. There are several works on the problem: “Is the ROI fully covered?”. Further, if the ROI is not covered by the sensors then there are several methods to cover the ROI using actuators or by using mobile sensors, or by activating some passive sensors. But no work has been done on the following problem:

*“How does the uncovered area change with the number of sensors, or how does the uncovered area depend on the strategy of deployment of the sensors when the sensors are deployed on the ROI in a random manner?”*

Even if the number of sensors is enough to cover the ROI in a deterministic manner, there will be some uncovered area if the sensors are deployed randomly. We cannot be sure about the full coverage of the ROI even if we deploy extra sensors, unless we use relocation of the sensors either by mobile sensors or by actuators. If we consider everything static then our target will be to reduce the uncovered area. It is enough to cover each point of ROI by exactly one sensor. If some portion of the ROI is covered by more than one sensor, then we have in some sense ‘wastage’ of sensors. Since the sensing area of a sensor is a circular disc we cannot avoid the wastage, but our target will be reducing the wastage. One idea is to deploy the sensors at pre-determined points such that if they are actually placed on those points, then the wastage is minimum, i.e., the coverage is maximum.

To the best of our knowledge, there is no work on the coverage problem in the random deployment situation with different distributions and different strategies for deployment of the sensors. One variation of the coverage problem available in literature is the case when the centers of the congruent discs are fixed and the objective is to cover a given set of points with the minimum number of discs. Stochastic formulations of this problem can be found in [6,92].

There is a related problem, known as the sphere packing problem. This problem asks: “what fraction of  $\mathbb{R}^n$  can be covered by congruent balls that do not intersect except along their boundaries”. In the two dimensional case, the sphere packing problem is known as the circle packing problem. The circle packing problem is to arrange maximum number circles (of equal or varying radii) on a given region such that no overlapping occurs [16]. We use the idea of the sphere packing problem to find an answer to the coverage problem. We use exactly the same patterns for sphere packing in our covering in two and three dimensional space.

Note that we can classify the coverage problem in twelve different categories, see Table 1.1. There is no work on cylindrical grids. Also there is no work on continuous domain, when extra sensors are used without any actuator.

Table 1.1: Classification of the Coverage Problem and Previous Works

Mode of deployment	Characterization of the ROI		
	Rectangle/Hexagonal grid	Cylindrical grid	Continuous Domain
Deterministic	[10, 22, 23, 26]	Open	[6, 20, 35, 86]
Random deployment using actuator(s)	[1, 28, 57, 69, 98]	Open	[18, 28, 50, 57, 98]
Random deployment using extra sensors	[44]	Open	Open
Random deployment with extra sensors and actuator	[24, 44, 55, 57]	Open	[7, 24, 48, 49, 57, 61]

## Detection Problem

Another fundamental challenge in wireless sensor networks is to detect an event. But the detection is disturbed by the noise and the reliability of sensor nodes. A sensor may fail to detect the event due to natural obstruction. After detecting the event, a sensor can send false message to the base station due to some technical reasons. The sensors are usually low-end inexpensive devices and sometimes exhibit unreliable behavior.

In a large ROI, if events occur at some point of the region then a particular sensor may not determine exactly where the event has happened. Fusion sensors are used in this case. But there are cases where a fusion sensor cannot make a decision or there is no fusion sensor. Consider, for example, a network of sensors that are capable of sensing mines or bombs. We assume that either

no mines (or bombs) are placed or very few mines (or bombs) are placed on a particular area of the ROI. In this case an important query could be; “have bombs been placed”. In that situation, there is no fusion sensor. All sensors have to communicate with the base station, and the base station will take the decision about the query.

In almost all previous work, authors assume that an event occurs over a region and there are fusion sensors that collect the information locally and take a decision. Since they do not introduce the concept of a base station there is no concept of response probability (the probability of correct response of a sensor). Also they assume that informations are spatially correlated. But there is no work in literature when the informations are not spatially correlated.

Most authors introduce only one type of detection probability and simulate the different error probabilities for some specific values of parameters. But there are no works on exact calculation of error probabilities. Also there are no works when there is only one event or few events in the ROI. In almost all previous work, authors assume that the grid is a square grid. But the hexagonal grid is better in the sense that less number of sensors is required to cover the entire ROI [96]. The number of sensors needed to cover an area for the hexagonal grid is given by Williams [96]. In this thesis, we calculate error probabilities. We deal with square as well as hexagonal grids.

## **1.4 Our Contribution and Thesis Plan**

In this thesis we contribute on two different aspects of WSNs: the coverage problem and the detection problem. This thesis is based on six papers [70–74, 76]. *Chapter 1* contains the introduction. In *Chapter 2* we provide a detailed literature survey that is pertinent for the work presented in *Chapters 3 to 8*. Concluding remarks and some open problems are given in *Chapter 9*. *Chapters 3 to 5* deal with the coverage problem and *chapters 6 to 8* deal with detection problem.

## Coverage Problem

Based on the work of [76], in *Chapter 3* we try to solve the coverage problem considering the ROI as a continuous region. After partitioning the ROI in several hexagons with sides equal to the sensing radius, sensors are dropped from the air stochastically at the center of all hexagons. But sensors may be placed at any point of the ROI due to stochastic deployment. We assume that the distance of this point to the target center is a random variable, which we will call the called *sensor displacement*.

We consider the probability distribution of sensor displacement as uniform as well as a more natural normal distribution. We calculate and simulate the proportion of uncovered area for both the distributions and for two different approaches.

If we drop one extra sensor at some randomly chosen centers then the proportion of the uncovered area will be reduced. If we drop some more discs (sensors) at some randomly chosen centers then the proportion of the uncovered area will reduce. Instead of placing these extra discs at randomly chosen squares or hexagons, if we reduce the distance between two neighboring target centers (i.e., reduce size of the regular hexagons keeping the sensing radius fixed) and place exactly one disc at each center, then also the uncovered area will reduce. That is, we use the extra discs in two different ways. In this thesis we compare these two ideas in terms of uncovered area. Two basic differences between these strategies are:

1. In the first strategy we drop one sensor on some hexagon (randomly chosen) and two sensors on the rest but in the second strategy we drop exactly one sensor on the center of each regular hexagon.
2. The length of the side of the regular hexagons is less in the second strategy than that in the first.

But in both strategies, we use the same number of discs to cover the same ROI and the sensing radius is same for both.

In WSNs we consider a bounded convex subsets of  $\mathbb{R}^2$ . More generally, we can consider any bounded subset of  $\mathbb{R}^n$ . In this thesis, we consider  $n = 1, 2, 3$ ,

with extra emphasis on  $\mathbb{R}^2$ . We study the placement of a minimal number of discs to cover  $\mathbb{R}^2$  under some constraints, and develop some theoretical results on the coverage problem in  $\mathbb{R}^2$ . We find the minimum number of sensors needed to cover an area and the points where they have to be placed. Note that a sensing disc is a closed disc in  $\mathbb{R}^2$  and closed ball in  $\mathbb{R}^3$ , with center as the point where a sensor is placed. We also consider more general coverage problem:

*“How can one place  $n$  dimensional balls of radius  $r$  to cover the  $n$  dimensional space  $\mathbb{R}^n$ , or a bounded subset of  $\mathbb{R}^n$  such that the portion of the volume which is common in every pair of balls is minimum?”*

In case of a bounded subset of  $\mathbb{R}^n$ , the problem is same as that of finding the minimum number of sensors to cover the bounded set. One may call the intersection of two balls as wastage. So our problem is to find the deployment of the sensors with minimum wastage, where the wastage is the portion of total volume which is common between pairs of balls. We mainly consider the case when  $n = 1, 2, 3$ , and we give a partial answer to this problem. We show hexagonal deployment is close to optimal for a bounded convex subset of  $\mathbb{R}^2$ , with ‘some’ restrictions.

After random deployment of sensors, actuators relocate the sensors. So far several randomized algorithms are used for relocation. But no deterministic algorithm is used. Also there is no previous works for rectangular grid structure of ROI. Based on the work of [74], in *Chapter 4* we provide an optimal placement of sensors in a rectangular grid ROI partitioned into several squares such that the ROI will be fully covered by these sensors. In general, a sensor is placed at the center of a square, known as center node. This sensor can detect events at the center node along with the four centers of the four adjacent squares which have a common edge with the center square. These four centers are known as distance-one nodes. The sensor placed at the center node cannot detect events placed at the center of the other squares, e.g., distance-two node, which has a common vertex but no common edge with center square. Our objective is to place the minimum number of sensors at the center of some selected squares in such a way that, they can detect the events at the center of all the squares. Then the minimum number of sensors required is same as the domination number of the corresponding rectangular grid and a minimum dominating set will suggest which squares we have to choose. In this chapter,

we deal with the domination number of some special types of graphs, known as cylindrical grid graphs. We found the domination numbers as well as a minimum dominating sets of the some graphs.

In *Chapter 5* we provide algorithms for actuator based on the work of [71]. In this chapter we consider the coverage problem in wireless sensors and actuator networks composed of static sensors dropped stochastically in a region of interest (ROI), which is a rectangular grid. An event can occur at any vertex of the grid and sensors can be placed at any vertex. Sensors are dropped at the vertices of the grid from the air. A sensor may placed at center of the target vertex or at one of neighboring four vertices due to stochastic deployment of sensors. An actuator can take, carry and place the sensors according to some pre-assigned algorithm. We want the ROI to be free of sensing holes. So the actuator will go to the grid and rearrange some of the sensors in such a way that at least one sensor is placed at each node. In this chapter we develop three deterministic algorithms for actuators and compare these algorithms in context with some pre-assigned parameters. We consider only one actuator and this actuator can carry only one sensor with it when it travels from one node to another.

Table 1.2: Classification of the Coverage Problem and Our Contribution

Mode of deployment	Characterization of the ROI		
	Rectangle/Hexagonal grid	Cylindrical grid	Continuous Domain
Deterministic		Chapter 4	Chapter 3
Random deployment using actuator(s)		Still open	
Random deployment using extra sensors	Chapter 5	Still open	Chapter 3
Random deployment with extra sensors and actuator	Chapter 5	Still open	

Our contribution (chapters 3 to 5) to the coverage problem is summarized in the Table 1.2. Our contribution in this field is very useful in the sense that there is no work on cylindrical grid like ROI or on reducing the uncovered area of ROI. We deal with the coverage problem for cylindrical grids using graph theory and for continuous regions using geometry, which is unique in the literature.

## Detection Problem

In this thesis we introduce two detection probabilities, one for the center node and the other for the adjacent nodes. Even if the center node fails to detect the event, the adjacent nodes may detect the event, and vice versa. We consider these probabilities and show that, in various situations, the adjacent nodes play a key role in detecting the event. We calculate the exact error probabilities and the exact tests. It is hard to calculate the exact probabilities and the exact test for the general case when events occur in more than one cell.

In this thesis, we consider both square and hexagonal grids in separate sections. We assume that the ROI is partitioned into a suitable number of identical squares of side  $2a$  (i.e., we consider ROI as a rectangular grid with square cells). We also consider a regular hexagonal grid with regular hexagonal cells. We assume that sensors have already been placed at the center of the squares (or hexagons). In this thesis, we are interested to detecting an event in the ROI. We assume the ROI to be partitioned into suitable number of congruent regular hexagonal or square cells. This physical structure of ROI is not a requirement for the theoretical analysis, we can do similar analysis with other structures as well. Suppose that sensors are placed a priori at the center (known as a node) of every hexagon of the grid. We assume that sensors are connected to their adjacent sensor nodes in the sense that a hexagon will be strongly covered by its center node and weakly covered by the adjacent nodes. If an event occurs in the hexagon where a particular sensor lies, then that particular sensor can detect the event with a greater probability. Whereas, if event occurs in any adjacent hexagon, then the particular sensor can detect the event with a lesser probability. Hence, only one node (center node of the event hexagon) can detect the event hexagon with greater probability and adjacent nodes (six for interior nodes and less for boundary nodes) can detect the event hexagon with lesser probability. In the case of a square cell we assume there are one more detection probability, which is less than the other two. We assume that no other sensor can detect the event hexagon.

In our theoretical analysis, the sensor fault probabilities are introduced into the optimal event detection process. We applied model selection approach, multiple model selection approach and Bayesian model averaging methods [36,



65] to find a solution to the problem. We develop the schemes using the model selection technique. We calculate different error probabilities and find some theoretical results. We propose fault detection schemes that take into account error probabilities in the optimal event detection process. We develop the schemes under the consideration of classical hypothesis testing and the Bayes test. We calculate different error probabilities and find some theoretical results involving different parameters such as probability of false alarm of a sensor, probability of event detection by a sensor, prior probability of occurring a event, etc. Finally, we calculate different error probabilities, Bayes test and Neyman-Pearson Most Powerful (MP) test for some specific values of the parameters and analyze the results.

The detection probabilities of a sensor and the probability of sending information correctly by a sensor cannot be estimated from the real life situation, but we can estimate them experimentally beforehand. The prior probability of the event cannot be estimated. In various situations, it may be known, in which case we apply Bayes test; otherwise, we use Neyman-Pearson MP test.

We propose a rule for the base station to take a decision compiling the information coming from the all sensors and find the optimal solutions. We consider two types of error: (1) type I error when an event occurs but the sensors report normal (which is the more serious error) and (2) type II error when the ROI is normal but sensors report an event. We observed that type I and type II errors decrease when detection probabilities increase. If detection probabilities are low then type I error is close to 1. If probability of occurrence of the event is high but detection probabilities are small then type I and type II errors are high, which means sensors are not useful. So, when the probability of occurrence of the event is high, we have to use sensors with high detection probability (i.e., sensors with much better quality). We calculate the MP test and the Bayes test for some specific values of the parameters. We observed that for small values of detection probability and large value of loss, the Bayes test is not applicable. When loss is large, we cannot use sensors with small detection probabilities to decide about the event square using Bayes' test. We also observed that when the size of the test is small we cannot use sensors with small detection probabilities for MP test; we have to use good sensors (sensor with high detection probability) for MP test in this case. For details of loss,

MP test and Bayes test, see [68].

*Chapter 6* is based on the work of [72] on the event detection problem when a sensor can be faulty or a detection can be wrong due to noise. In our theoretical analysis, we propose fault detection schemes that take into account error probabilities in the optimal event detection process. We assume the ROI is partitioned into disjoint squares (or hexagons) and only one event can occur at a specific cell. We develop the schemes under the consideration of classical hypothesis testing and Bayes test. We calculate different error probabilities, Bayes test and Neyman-Pearson Most Powerful(MP) test for some specific values of the parameters, and analyze the calculation results.

*Chapter 7* is based on the work of [73] on the event detection problem when the RIO is considered as a hexagonal grid and sensors are placed at the center of the hexagons. There is at most one event but it may occur at any hexagon. We assume two detection probabilities, one for the center node and other for the adjacent nodes, and they are the same for all sensors. Even if the center node may fail to detect the event, the adjacent nodes may detect the event, and vice versa. We show that, in various situations, the adjacent nodes play a key role in detecting the event. We apply a model selection approach, multiple model selection approach and Bayesian model averaging methods to find a solution to the problem. We calculate error probabilities and find theoretical results.

*Chapter 8* is based on the work of [70] on the event detection problem in most general case. In this chapter, we consider how to take decisions regarding fault detection in a noisy environment, created as a result of false detection or false reporting of an event by some sensors, where the sensors are placed at the center of regular hexagons and events can occur at any number of hexagons. We introduce two types of detection probabilities, one for the center node, where the event occurs and the other one for the adjacent nodes. We develop a scheme under the multiple model selection procedure and discuss two examples. We use a near optimal model selection technique to detect the events. We also introduce a new approach to detect events. We introduce the concept of false response of sensors, i.e., when they send message to the base station, we assume that wrong information can be sent. Our contribution gives a new approach to event detection in the case of a WSN.

# Chapter 2

## Previous Work

There are several works on the coverage and detection problems of WSNs. We briefly summarize some of them which are very much related to our work. We consider the works from 2000 to 2014 only. We also briefly summarize some related areas of our research.

### 2.1 Coverage Problem

One of the major challenges in devising such networks lies in the constrained energy and computational resources available to sensor nodes. These constraints must be taken into account at all levels of the system hierarchy. The deployment of sensor nodes is the first step in establishing a sensor network. Since sensor networks contain a large number of sensor nodes, the nodes must be deployed in clusters, where the location of each particular node cannot be fully guaranteed a priori. Therefore, the number of nodes that must be deployed in order to completely cover the whole monitored area is often higher than if a deterministic procedure were used. In 2001, Slijepcevic and Potkonjak [90] introduced an algorithm that selects mutually exclusive sets of sensor nodes, where the union of these sets completely cover the monitored area. The intervals of activity are the same for all sets, and only one of the sets is active at any time. They achieve a significant energy savings while fully preserving coverage.

In 2002, Akyildiz et. al. [3] described the concept of sensor networks which has been made viable by the convergence of micro-electro-mechanical systems technology, wireless communications and digital electronics. First, the sensing tasks and the potential sensor networks applications are explored, and a review of factors influencing the design of sensor networks is provided. Then, the communication architecture for sensor networks is outlined, and the algorithms and protocols developed for each layer in the literature are explored.

In 2003, C. F. Huang Y. C. Tseng [40] formulated the coverage problem as a decision problem, whose goal is to determine whether every point in the service area of the sensor network is covered by at least  $k$  sensors, where  $k$  is a predefined value. The sensing ranges of sensors can be unit disks or non-unit disks. They presented polynomial time algorithms, in terms of the number of sensors, that can be easily translated to distributed protocols. The result is a generalization of some earlier results where only  $k = 1$  is assumed. Applications of the result include: (i) positioning applications, (ii) situations which require stronger environmental monitoring capability, and (iii) scenarios which impose more stringent fault-tolerant capabilities. Results include determining insufficiently covered areas in a sensor network, enhancing the fault-tolerant capability in hostile regions, and conserving energies of redundant sensors in a randomly deployed network. Their solutions can be easily translated to distributed protocols to solve the coverage problem.

One variation of the coverage problem available in literature is the case when centers of the congruent discs are fixed and the objective is to cover a given set of points with the minimum number of discs. A stochastic formulation of this problem is considered here known as the continuum percolation model. In this model each point of a two-dimensional Poisson point process is the center of a disc of given (or random) radius  $r$ . In 2003, Booth et. al. [6] considered the generalization in which a deterministic algorithm (given the points of the point process) places the discs on the plane, in such a way that each disc covers at least one point of the point process and that each point is covered by at least one disc. This gives a model for WSNs, which was the original motivation to study this class of problems.

In 2004, Z. Zhou et. al. [100] showed, in a randomly deployed sensor networks, how one approach to keep only a small subset of sensors active at any

instant to conserve energy. They considered the problem of selecting a minimum size connected  $K$ -cover, which is defined as a set of sensors  $M$  such that each point in the sensor network is covered by at least  $K$  different sensors in  $M$ , and the communication graph induced by  $M$  is connected. For the above optimization problem, they design a centralized approximation algorithm that delivers a near-optimal (within a factor of  $O(\log n)$ ) solution, and presented a distributed version of the algorithm. They also presented a communication-efficient localized distributed algorithm which is empirically perform well.

In 2006, Mousavi et. al. [69] presented a distributed one step deployment (OSD) algorithm. This algorithm partitions the ROI evenly into two-dimensional square grids, and instructs sensors to occupy all the grid points. The intuition is that if each grid point is occupied by a sensor, then the ROI is fully covered, and the sensors form a connected network. They proposed two methods for the self-deployment of mobile sensors. The first one is a randomized solution that provides both simplicity and applicability to different environments. It improves both speed and energy conservation of the deployment process. The other method is suggested for environments where sensors form a connected graph. They gained improvements over previous works.

In 2005-2007, Silva et. al. [87–89] considered the coverage problems in sensor networks with minimal sensing capabilities. The methods, they introduced, came from persistent homology theory. They assumed the sensors are coordinate free, no localization or orientation capabilities and randomness are there. They demonstrated the robustness of the tools by adapting them to a variety of setting, including static planar coverage, 3-d barrier coverage, and time dependent sweeping coverage. They demonstrated that a stationary collection of sensor nodes with no localization can verify coverage in a bounded domain of unknown topological type. They gave results on hole repair, error tolerance, optimal coverage and variable radii. They also considered coverage problems in robot sensor networks with minimal sensing capabilities. In particular, they demonstrated that a blind swarm of robots with no localization and only a weak form of distance estimation can rigorously determine coverage in a bounded planar domain of unknown size.

In 2007, Zahar et. al. [22] showed the following theorem:

**Theorem 2.1.** *Let  $\gamma(G)$  denote the domination number of a graph  $G$  and*

$C_n \square G$  denote the cartesian product of the cycle of length  $n \geq 3$  and  $G$  then  $\gamma(C_n \square G) = \gamma(C_n)\gamma(G)$  implies  $n \equiv 1 \pmod{3}$ .

In addition they characterize graphs that satisfy the equality when  $n = 4$  and provide infinite classes of such graphs for general  $n \equiv 1 \pmod{3}$ .

In 2009, Filippou et. al. [27] measured the ability of the network to interact with observed phenomena taking place in the ROI. In addition, coverage is associated with connectivity and energy consumption, both of which are important aspects of the design process of a WSN. The results aim at offering a critical overview and presentation of the problem as well as the main strategies developed so far. They classified the environment, coverage protocols, sensors' mobility and placements technique etc. They gave an overview of different scenarios of wireless sensor networks.

The target coverage problem in wireless sensor networks is concerned with maximizing the lifetime of the network while continuously monitoring a set of targets. In 2009, Chaudhary and Pujari [9] studied the target Q-coverage problem. The objective is to maximize sensor network lifetime satisfying Q-coverage requirement. The problem is shown to be NP-complete and there is no known practical algorithm. They proposed a heuristic algorithm to generate Q-covers by prioritizing sensors in terms of the residual battery life and the algorithm assigns a small constant of lifetime to Q-covers so generated. In this process, it allows the sensors to participate in many Q-covers. They observed that the smaller the constant, the closer is the solution to the optimal solution. Through experiments on randomly generated problem instances, they showed that the proposed algorithm yields near-optimal solution.

In 2010, Greg Fletcher et. al. [29] presented randomized algorithms for more than one robot for coverage repair in WSN. They gave two algorithms among these: one for grid based ROIs, and another consider continuous regions. Using these algorithms, mobile robots move within the network to collect redundant sensors and deliver them to reported sensing holes. They simulate the length of the path traveled by the robots for deferent values of parameter like, number of sensors, number of robots etc.

In 2011, Dharma P. Agrawal [2] summarized many underlying design issues of WSNs, starting from the coverage and the connectivity. As batteries provide

energy to sensor nodes, effective ways of power conservation are considered. Advantages of placing sensors in a regular pattern have also been discussed and various trade offs for many possible ways of secured communication in a WSN are summarized. Challenges in deploying WSN for monitoring emissions are briefly covered. Finally, the use of sensors is illustrated in automatically generating music based on a dancer's movements.

Maximizing network lifetime is an important objective for the target-coverage problem. With practical manufacture and cost reduction, directional sensors have been widely used in wireless sensor networks to save energy. In 2012, Li et. al. [56] addressed the target Q-coverage (TQC) problem to prolong the network lifetime with bounded service delay constraint in directional sensor networks. They proposed a protocol to find a collection of coverage sets that satisfy the coverage quality requirement and the bounded service delay constraint, where the target in each coverage set may not be served continuously but can be served with tolerant service delay. By steering some sensors' directional antennas, the protocol could deal with the changes of network topology or monitoring tasks. Simulation results show that the performance of the protocol is close to the upper bound of the optimal solution.

In 2013, Li et. al. [50] described carrier-based sensor relocation by robots to repair sensing holes. They considered grid structure of the ROI and used a virtual force algorithm. Analyses are done for maximum distance covered and expected distance covered by the robot(s) or by the mobile sensors to achieve the full coverage. They considered the problem of repairing sensing holes, while the redundant sensors exist in other areas in the network. Robots move within the network to discover sensing holes. They find redundant sensors by local communication, and transfer the discovered redundant sensors to the encountered sensing hole positions. The authors proposed four algorithms: in one the them robot moves randomly, in another robot movement is restricted to a virtual grid, and the other two are variants of the second.

In 2013, Kranakis et. al. [44] obtained expected sum and maximum displacement for sensors thrown at random in a unit square. They produced a tight bound for the expected maximum displacement, and they proved that:

**Theorem 2.2.** *The expected sum is less than a constant multiple of  $\sqrt{\ln(n)}$ , where  $n$  is the number of sensors.*

The authors investigated the related problem of the expected total and maximum displacement for perimeter coverage (whereby only the perimeter of the region need be covered) of a unit square. They also presented range trade-offs for area coverage.

## 2.2 Detection Problem

In 2004, Krishnamachari and Iyengar [45] proposed a distributed solution for the canonical task in WSN, i.e., the binary detection of interesting environmental events. They explicitly take into account the possibility of sensor measurement faults and develop a distributed Bayesian algorithm for detecting and correcting such faults. They presented two Bayesian algorithms: the randomized decision scheme and the threshold decision scheme, and derive analytical expressions for their performance. Analysis shows that the threshold decision scheme has better performance in terms of minimization of errors. The theoretical and simulation results show that 85-95 percent of faults can be corrected using this algorithm, when 10 percent of the nodes are faulty.

In 2006, Lou et. al. [63] considered two important problems for distributed fault detection in WSN: 1) how to address both the noise-related measurement error and sensor fault simultaneously in fault detection, and 2) how to choose a proper neighborhood size  $n$  for a sensor node in fault correction such that the energy could be conserved. They proposed a fault detection scheme that explicitly introduces the sensor fault probability into the optimal event detection process. They showed that the optimal detection error decreases exponentially with the increase of the neighborhood size. Experiments with both Bayesian and Neyman-Pearson approaches in simulated sensor networks demonstrate that the proposed algorithm is able to achieve better detection and better balance between detection accuracy and energy usage. Their work shows it possible to perform energy-efficient fault-tolerant detection in a WSN.

In 2006, Chen et. al. [11] proposed and evaluated a localized fault detection algorithm to identify the faulty sensors in a WSN where each sensor identifies its own status to be either good or faulty and the claim is then supported or disputed by its neighbors. The proposed algorithm is analyzed using a probabilistic approach. The goal is to locate the faulty sensors in the wireless sensor



networks. they proposed and evaluate a localized fault detection algorithm to identify the faulty sensors. The implementation complexity of the algorithm is low and the probability of correct diagnosis is very high even in the existence of large fault sets. Simulation results show the algorithm can clearly identify the faulty sensors with high accuracy.

In 2009, Ni et. al. [78] presented a detailed study of the sensor faults that occur in deployed sensor networks. They presented a systematic approach to model these faults. They categorized the three types of fault models. (1) CONSTANT fault: The sensor reports a constant value which is either very high or very low compared to the normal sensor reading and uncorrelated to the underlying physical phenomena for a large number of successive samples. (2) SHORT fault: A sharp change in the measured value between two successive samples, i.e., it effects a single sample at a time. (3) NOISE fault: The variance of the sensor reading increases, i.e., it effect a number of successive samples. They used data collected from scientific deployments to develop a set of commonly used features useful in detecting and diagnosing sensor faults. They used this feature set to systematically define commonly observed faults. They provide examples of each of these faults from sensor data collected at recent deployments. There are other fault models also.

In 2010, Sharma et. al. [86] characterized the different type of fault and fault detection methods. They classified the faults as Short, Noise and Constant. While it is not always possible to ascertain the root cause for sensor faults, several system (hardware and software) faults have been known to results in sensor faults. The typical hardware faults that have been observed to cause sensor faults are: damaged sensors, short-circuited connections, low battery and calibration errors. They proposed different algorithm for fault detection considering different types of fault. Some of the methods are statistical, like, using histograms, etc. Both works can only detect the faulty sensors, but not the event. The networks must exclude the faulty sensors to ensure the network quality of service. To identify the faulty sensor is an existing challenge. They used real world datasets to answer the following question: how often are the sensor data fault types observed in real deployment? They used rule-based methods, estimation methods, learning-based methods and time series analysis based methods.

In 2011, Yun et. al. [98] considered the problem of identifying battlefield events using sensors deployed in the area. The goal is to alert centralized headquarters about the occurrence of significant events so that it can respond appropriately to the events. They proposed a mechanism using which the sensors can exchange information using signatures of events instead of data to save on transmission costs. Further, they proposed an algorithm that dynamically generates phases of information exchange based on the cost and selectivity of each filter. They presented simulation results that compare the proposed algorithm to other alternatives. Their results show that the algorithm detects events while minimizing the transmission and processing costs at sensors.

In 2011, Farah et. al. [25] used homology to detect and differentiate between incremental insertion events of interest. Particular homology tools are translated to a distributed environment for 2-dimensional WSN deployments. The result is a distributed algorithm that can compute an incremental insertion event associated with a region comprising  $n$  nodes in  $O(n)$  time, using  $O(n)$  storage, and  $O(n)$  data passed via messages. A small-scale, laboratory testbed is developed to evaluate the algorithm. Deployment results indicate that only nodes in physical proximity to an event are tasked, thereby conserving network resources and allowing multiple disparate events to be simultaneously monitored. Further, transmission cost is showed to vary linearly with the size of the evolving region, confirming one component of the formal analysis.

## 2.3 Related Areas

In data aggregation, sensor measurements from the whole sensory field or a sub-field are collected as a single report at an actor by using aggregate functions such as sum, average, maximum, minimum, count, deviation, and so on. In 2011, Li et. al. [60] proposed a localized delay-bounded and energy-efficient data aggregation (DEDA) protocol for request-driven wireless sensor networks with carrier sense multiple access with collision avoidance run at the media access control layer. This protocol uses a novel two-stage delay model, which measures end-to-end delay by using either hop count or degree sum along a routing path depending on traffic intensity. It models the network as a unit disk graph (UDG) and constructs a localized minimal spanning tree (LMST) sub-

graph. Using only edges from LMST, it builds a shortest-path (thus energy-efficient) tree rooted at the actor for data aggregation. The tree is used without modification if it generates acceptable delay, compared with a given delay bound, otherwise, it is adjusted by replacing LMST sub-paths with UDG edges.

In a broadcasting task, the source node wants to send the same message to all the other nodes in the network. Existing solutions address specific mobility scenarios, e.g. connected dominating set (CDS) based for static networks, blind flooding for moderate mobility, and hyper flooding for highly mobile and frequently partitioned networks. In 2012, Stojmenovic [91] designed a protocol that will seamlessly (without using any parameter) adjust itself to any mobility scenario, and with capability to address various model assumptions and optimality criteria.

WSNs are also used in neural networks. In 2013, Li et. al. [58, 59] found a feasible solution to a class of nonlinear inequalities defined on a graph proposing a recurrent neural network. The convergence of the neural network and the solution feasibility to the defined problem are both theoretically proven. They proposed neural network features as a parallel computing mechanism and a distributed topology isomorphic to the corresponding graph which is suitable for distributed real-time computation. The proposed neural network is applied to range-free localization of WSNs. They showed that feasible solution set to the same problem is often infinity and Laplacian eigenmap is used as heuristic information to gain better performance in the solution. A continuous-time projected neural network, and the corresponding discrete-time projected neural network are both given to tackle this problem iteratively. The effectiveness of the proposed neural networks are compared with others via its applications in the range free localization of WSNs.

In 2013, Li et. al. [46, 52] formulated the sensor network problem as an optimization problem defined on the Blue-tooth network. The solution to this optimization problem is not unique. Heuristic information is employed to improve the performance of the result in the feasible set. They used recurrent neural networks to solve the problem in real time. The convergence of the neural network and the solution feasibility to the defined problem are both theoretically proven. The hardware implementation of the proposed neural network is also explored. Distributed algorithms are also used for a network

dynamic system. Li et. al. studied the decentralized control and kinematic control of multiple redundant manipulators for the cooperative task execution problem. The problem is formulated as a constrained quadratic programming problem and then a recurrent neural network with independent modules is proposed to solve the problem in a distributed manner. They proposed a strategy to solve the problem even though there exists some manipulators unable to access the command signal directly.

Another application of WSN is the winner-take-all (WTA) competition. This is widely observed in both biological media and society. Many mathematical models are proposed to describe the phenomena discovered in different fields. These models are capable of demonstrating the WTA competition. In 2013, Li et. al. [53, 54] make steps in that direction and presented a simple model, which produces the WTA competition by taking advantage of selective positive-negative feedback through the interaction of neurons via  $p$ -norm. They also presented a class of recurrent neural networks to solve quadratic programming problems. Different from most existing recurrent neural networks for solving quadratic programming problems, the proposed neural network model converges in finite time and the activation function is not required to be a hard-limiting function for finite convergence time. The stability, finite-time convergence property and the optimality of the proposed neural network for solving the original quadratic programming problem are proven in theory.

In a mobile Unattended Wireless Sensor Network (UWSN), a trusted sink visits each sensor node periodically to collect data. Data has to be secured until the next visit of the sink. Securing the data from an adversary in UWSN with mobile nodes is a challenging task. In 2012, Reddy et. al. [83] presented two non-cryptographic algorithms to ensure data survivability in mobile UWSN. These algorithms protect against a proactive adversary which compromises nodes before identifying its target and makes the network secure against the reactive adversary which compromises nodes after identifying the target. They analyzed memory overheads and communication costs both mathematically and using simulations. In existing schemes, sensors remain static between visits from the sink, whereas in our scheme sensors can move between successive visits from the sink. They showed that their approaches perform better than known schemes in terms of communication overheads.

## Coverage in Continuous Domain

### 3.1 Introduction

Coverage is the main goal for a WSN but due to shortage of sensors or random deployment of sensors we cannot guarantee full coverage. Moreover, actuators may not be available or the ROI may prevent the use of actuators. To the best of our knowledge there is no work on uncovered area in random deployment scenario. In this chapter our target is to calculate and reduce the uncovered area of ROI using extra sensors. Here we consider two problems:

1. Deterministic deployment of static sensors without any actuator:

How can one place  $n$  dimensional balls of radius  $r$  to cover  $n$  dimensional  $\mathbb{R}^n$  space or a bounded subset of  $\mathbb{R}^n$  such that the portion of the volume which is common in every pair of balls is minimum. In the case of a bounded subset of  $\mathbb{R}^n$  the problem is same as finding the minimum number of sensors to cover the bounded set. One may call the portion of total volume which is common to two balls as wastage. So our problem is to find the deployment of the sensors with minimum wastage. In two and one dimensions we identify volume with area and length respectively. We mainly consider the case when  $n = 1, 2, 3$ . We give a partial answer to this problem. We show hexagonal deployment is close to optimal for a bounded convex subset of  $\mathbb{R}^2$  with ‘some’ restrictions.

2. Random deployment of sensors and use of extra sensors:

Usually the sensors are deployed from the air, so a sensor may not fall onto the right point. Suppose  $P$  is the point where the sensor was intended to drop and  $P'$  is the point where it is actually placed. Then the distance between  $P$  and  $P'$ , called sensor displacement, follows some probability distribution depending on the mechanism of deployment. Note that a sensing disc is a closed disc in  $\mathbb{R}^2$  and closed ball in  $\mathbb{R}^3$  with center as the point where a sensor placed.

More formally, we can formulate the problem as follows: for an index set  $I$ , consider a set of closed unit balls (or equivalently, of radius  $r$ )  $\{B_i \subseteq \mathbb{R}^n : i \in I\}$ , which cover  $\mathbb{R}^n$  or a bounded convex subset of  $\mathbb{R}^n$ , considered as the ROI. Consider a set of  $n$ -dimensional random vectors  $\{X_i : i \in I\}$ . Assume that  $D_i$ , the distance between  $X_i$  and the center of  $B_i$ , for  $i \in I$  are i.i.d. with the density  $f(\cdot)$ . Then some portion of the  $\mathbb{R}^n$  may be uncovered by the set of balls with centers  $X_i, i \in I$ . What portion of  $\mathbb{R}^n$  will be uncovered? How we can reduce this uncovered portion using some extra random vectors (or equivalently balls)? For  $n = 1$  the coverage problem with minimum wastage is trivial. We mainly discuss the case when  $n = 2, 3$  with special emphasis on  $n = 2$ . We consider two type of distributions, uniform and normal, for the distances  $D_i$ , and two different strategies for deployment of extra sensors to reduce the uncovered volume of a bounded convex subset of  $\mathbb{R}^n$ . We consider hexagonal deployment of sensors since it is near optimal for the coverage problem with minimum wastage as well as optimal for the sphere packing problem. One can also think of other types of deployment. If we cover the ROI by circular discs such that the intersection of three or more discs is a point or empty then the ROI is covered by identical disjoint regular hexagons, each of which is inscribed in a disc. Since the deployment is random there will be uncovered area or region after deployment. One can think of several strategies to reduce the uncovered area. In this chapter we consider two different strategies:

**St. 1** : strategy for deployment of extra sensors is as follows: deploy one sensor at each center of the regular hexagons, then choose some centers (according to how many extra sensors we have) and deploy one more sensor there.

**St. 2** : strategy for deployment of extra sensors is as follows: reduce the length

of the side of the regular hexagon (according to how many extra sensors we have) and deploy exactly one sensor at each center, i.e., we reduce the distance among two sensors.

In both strategies, we use the same number of sensors and the same sensing radius. The two basic differences between these two strategies are:

1. in the first strategy we drop one sensor onto some hexagons and two sensors onto the rest (randomly chosen) but in the second strategy we drop exactly one sensor onto the center of regular hexagons; and
2. the length of the side of regular hexagons is less in the second strategy than in the first.

We also discuss some coverage criteria and simulate the uncovered volume in different situations. We observed from simulation results that, the first strategy is better when the variance of the distribution of  $D_i$  is large for both normal and uniform distributions and for both  $\mathbb{R}^2$  and  $\mathbb{R}^3$ . We prove that the hexagonal placement of sensors is near optimal if there is no randomness in deployment. We also calculate theoretically the uncovered area in the uniform situation when there are no extra sensors.

If we consider a convex subset of  $\mathbb{R}^2$ , then the problem is reduced to the coverage problem in the field of sensor networks. Usually the ROI is a bounded convex subset of  $\mathbb{R}^2$ . More generally we can consider the ROI as a union of several bounded convex subsets of  $\mathbb{R}^2$ . Note that placing a sensor in the ROI is equivalent to placing a disc in some convex bounded set in  $\mathbb{R}^2$ .

In this chapter, we write the phrase ‘disc placed at a center of a square or hexagon (or a point)’ instead of writing ‘disc is placed so that the center of the discs placed at a center of a square or hexagon (or a point)’. Note that the placing of a sensor is same as the placing of a disc in  $\mathbb{R}^2$ , or a sphere in  $\mathbb{R}^3$ .

In WSNs we consider bounded convex subsets of  $\mathbb{R}^2$ . More generally, we can consider any bounded subset of  $\mathbb{R}^n$ . We consider  $n = 1, 2, 3$ , with extra emphasis on  $\mathbb{R}^2$ . We study the placement of the minimal number of discs to cover  $\mathbb{R}^2$  under some constraints, stated later, and develop some theoretical results on the coverage problem in  $\mathbb{R}^2$ . The distance between the point where

a disc is targeted to drop and the point where it actually placed is a random variable. We consider the situations when the probability distribution of this random variable is uniform or normal with different parameters. We calculate and simulate the proportion of uncovered area for both the distribution. We introduced two different strategies and compare them in terms of uncovered area.

We prove later, that if we partition  $\mathbb{R}^2$  in regular hexagons of side  $r$  (the radius of discs i.e., sensing radius of sensors) and place one disc at each of the center of these regular hexagons, the placement is optimal in some sense stated latter. One may partition the ROI into identical squares or any other convenient identical smaller areas, such that only one disc is needed to cover each smaller area. In the coverage problem only these two type of partitions are used. It can be proved that partitioning the ROI into squares and placing one disc at each center is not an optimal strategy (to minimize the number of sensors); hexagonal partitioning is better one. But after stochastic deployment of the same number of discs, partitioning the ROI into regular hexagons may or may not be better than the strategy of partitioning the ROI into squares with respect to the covered area. We consider only the regular hexagonal partition.

We also find theoretically the expected proportion of area covered in the ROI when the  $D_i$ 's are independently, identically and uniformly distributed. We also simulate the proportion of covered volume for both uniform and normal distributions with the two strategies for deployment of extra discs.

There is a related problem known as the sphere packing problem (see Figure 3.1). This problem asks, what fraction of  $\mathbb{R}^n$  can be covered by congruent balls that do not intersect except along their boundaries. Analysis of this problem is a very interesting area of research [14, 15]. Linear programming bounds are the most powerful known techniques to produce upper bounds in such problems [21]. The sphere packing problem in  $\mathbb{R}^n$  is trivial for  $n = 1$ . For  $n = 2$ , the standard hexagonal packing is optimal. For  $n = 3$ , Hales has proved that the face-centered cube packing is optimal [32] (see Figure 3.9). Some basic background on the sphere packing problem may be found in [17]. In the two dimensional cases, a sphere packing algorithm is presented in [16]. We use exactly the same patterns of placing circles or spheres to cover two and three dimensional space as in the sphere packing problem.



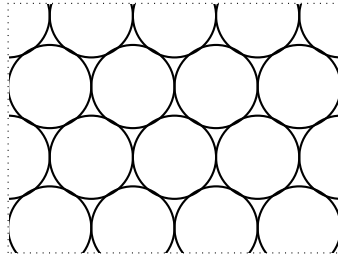


Figure 3.1: Sphere packing of a rectangular ROI

## 3.2 Assumptions and Definitions

Consider the coverage problem in WSNs which are composed of static sensors (equivalently, sensing discs) dropped stochastically in a region of interest (ROI). The ROI is partitioned into several identical regular hexagons of side  $a$ . Although the topology of the ROI may prevent partitioning into hexagons. If we consider the ROI to be the whole of  $\mathbb{R}^2$ , we do not have such a problem. To cover each hexagon by one sensor one should take  $a \leq r$ , where  $r$  is the sensing radius. If  $r = a$  each regular hexagon is covered by the sensor (also known as node) at its center if the sensor is placed exactly at the center of that hexagon. We assume that the ROI is  $\mathbb{R}^2$  or a convex and bounded subset of  $\mathbb{R}^2$ . As the sensors are so small we can think of a sensor as a point.

Now we define some useful terms. *Node* is the point where a sensor or the center of a disc is placed after deployment. In this chapter we use the term node to mean the point as well as the corresponding sensor. A *Vertex* is the point where the center of a disc is targeted to be placed.  $N(V)$  is the node corresponding to the vertex  $V$ , i.e., the center of a disc is placed at  $N(V)$  when the target was to drop at  $V$ . Similarly  $V(N)$  is the corresponding vertex of a node  $N$ . The *Sensing Disc*  $S_N$  of a node  $N$  is a closed disc of radius  $r$  and center  $N$ , which is covered by the disc or sensor placed at that node. The radius  $r$  is known as the *Sensing Radius*, which is assumed to be same for all discs. More generally one can consider discs with different radius. Throughout

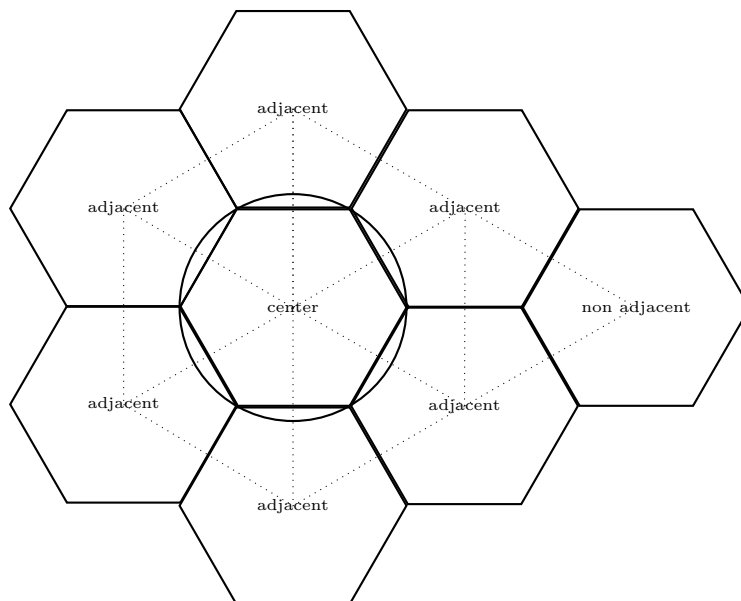


Figure 3.2: Nodes placed in ROI, which partitioned into regular hexagons

the chapter, the word ‘disc’ will refer closed discs only. In higher dimensions we call this the *Sensing Ball*.

The *Adjacent vertex* of a particular vertex means the vertex which is at distance not more than  $2r$  from that particular vertex. Therefore the sensing disc of a vertex has nonempty intersection with the sensing disc of its adjacent vertex and empty intersection with the sensing disc of a non adjacent vertex (which is not an adjacent vertex).

$\mathcal{V}$  is the set of all vertices and  $Adj_V$  is the set of all the adjacent vertices of a vertex  $V$  (see Figure 3.2). Similar definitions and notations apply for nodes also, and the respective notations are  $\mathcal{N}$  and  $Adj_N$  for  $N \in \mathcal{N}$ . Denote the distance between two points  $A$  and  $B$  in  $\mathbb{R}^n$  as  $d(A, B)$ . A point  $P \in \mathbb{R}^n$  is said to be covered by a node  $N$  if  $d(P, N) \leq r$  and the point  $P$  is said to be covered by a set of nodes  $\mathcal{N}$  if  $P$  is covered by at least one node in  $\mathcal{N}$ . A point  $P \in \mathbb{R}^n$  is said to be uncovered by a node  $N$  if it not covered by  $N$  and the point  $P$  is said to be uncovered by  $\mathcal{N}$  if  $P$  is uncovered by all the nodes in  $\mathcal{N}$ . Note that when there is no randomness, then the vertex and the corresponding node are the same, i.e.,  $N(V) = V$  and  $V(N) = N$ . The *Sensing hole* in  $\mathbb{R}^n$  (resp. ROI) is a connected subset of  $\mathbb{R}^n$  (resp. ROI) whose elements are uncovered by  $\mathcal{N}$ . An *Adjacent sensing hole* of a particular node means the sensing hole

whose boundary intersects with the boundary of the sensor disc of that node.  $\mathbb{R}^n$  (resp. ROI) will be called *covered by a set of nodes of sensing radius  $r$*  if every point of  $\mathbb{R}^n$  (resp. ROI) is covered by at least one node. The volume of a set  $B$  will be denoted as  $\text{Vol}(B)$ .

Let  $S$  be a bounded subset of  $\mathbb{R}^n$ , which is covered by a set of finite nodes  $\mathcal{N}$ . Define the wastage in  $S$  for  $\mathcal{N}$  as

$$W_{\mathcal{N}}(S) = \frac{\sum_{N \in \mathcal{N}} \text{Vol}(S \cap S_N) - \text{Vol}(S)}{\sum_{N \in \mathcal{N}} \text{Vol}(S \cap S_N)}.$$

If  $\mathcal{N}$  be such that  $|S_{N_1} \cap S_{N_2} \cap S_{N_3}| \leq 1$  for distinct  $N_1, N_2, N_3 \in \mathcal{N}$  (see figure 3.3a), then

$$W_{\mathcal{N}}(S) = \frac{\sum_{N_1 \neq N_2 \in \mathcal{N}} \text{Vol}(S \cap S_{N_1} \cap S_{N_2})}{\sum_{N \in \mathcal{N}} \text{Vol}(S \cap S_N)}.$$

Intuitively, the denominator represents the sum of the volume (common with  $S$ ) of all spheres. The numerator denotes the difference between the previous volume and the volume that we cover by these spheres, i.e., the volume of the sets whose points are covered by exactly two sensors, which can be thought as the wastage (in a layman sense) of volume. Hence the wastages represents the proportion of wastage to the total volume.

Let  $\mathcal{N}$  be a set of nodes which cover  $\mathbb{R}^n$  such that  $\mathcal{N} \cap S$  is finite for any bounded subset  $S$  of  $\mathbb{R}^n$ . Then wastage in  $\mathbb{R}^n$  for  $\mathcal{N}$  is defined by

$$W_{\mathcal{N}}(\mathbb{R}^n) = \lim_{x \rightarrow \infty} W_{\mathcal{N} \cap B_x}(B_x),$$

where  $B_x$  is the ball in  $\mathbb{R}^n$  of radius  $x$  and centered at the origin (equivalently, at any point). Intuitively, wastage in  $\mathbb{R}^n$  is the proportion of wastage volume in  $\mathbb{R}^n$ . Note that we can take any increasing sequence of sets whose union is  $\mathbb{R}^n$  other than  $B_x$ , e.g., for  $n = 2$  partitioned  $\mathbb{R}^2$  into hexagons or octagons and then take an increasing sequence of union of finitely many such polygons with the property that limit of this sequence is  $\mathbb{R}^2$ . In that case we can similarly define wastage. It can be proved that these two definitions are equivalent.

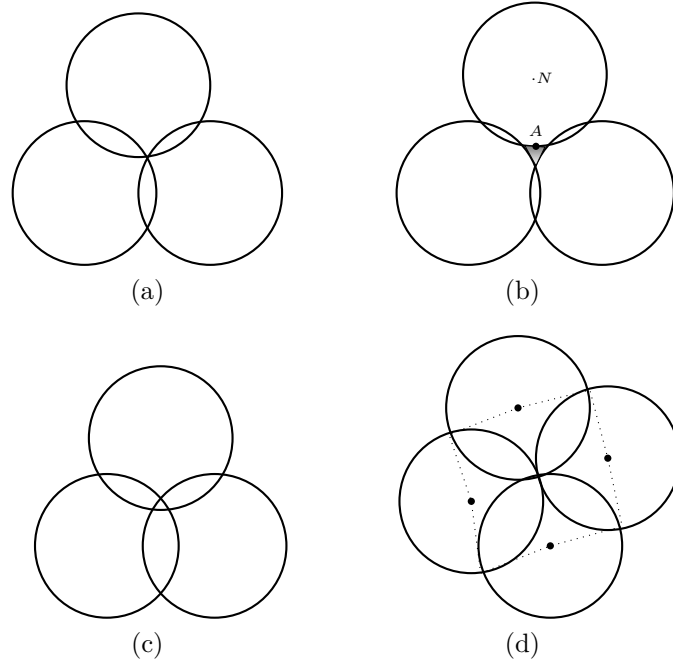


Figure 3.3: Sensing discs in different situations

### 3.3 A Result on Coverage Problem

**Theorem 3.1.** *Let ROI be a bounded and convex subset of  $\mathbb{R}^n$  and let the number of nodes in ROI be finite. Then the ROI is covered by the set of nodes  $\mathcal{N}$  if and only if any interior point of the ROI which also belongs to boundary (i.e., circumference) of a sensing ball belongs to another sensing ball.*

*Moreover, if  $ROI \subset \mathbb{R}^2$  then ROI is covered by a set of nodes if and only if the set of interior points of ROI on the boundary of a sensing ball of a node and which does not belong to the interior of any other sensing ball, is finite.*

*Proof.* Let the ROI be covered by a set of nodes  $\mathcal{N}$ . Suppose there is point  $A$  which belongs to the intersection of the boundary of sensing ball  $S_N$  of node  $N$  (denoted as  $Bd(S_N)$ ) and the interior of the ROI, but does not belong to the any other sensing ball (see Figure 3.3b). Then  $d(A, N) = r$  and  $d(A, N') > r$  for all  $N' \in \mathcal{N} \setminus \{N\}$ . Let  $d = \min\{d(A, N') : N' \in \mathcal{N} \setminus \{N\}\}$ . Therefore,  $d > r$  as  $\mathcal{N}$  is finite. Hence the ball  $B_{(d-r)/2}(A)$  with center  $A$  and radius  $(d-r)/2$  has no intersection with the sensing ball of any node except one node  $N$ . Since  $A \in Bd(S_N), B_{(d-r)/2}(A) \not\subset S_N$ , hence ROI cannot be covered by  $\mathcal{N}$ .

Moreover, if  $\text{ROI} \subset \mathbb{R}^2$ , the set of interior points of ROI that are on the boundary of the sensing disc of a node and do not belong to the interior of any other sensing disc is finite because there are finitely many points which belong to the intersection of boundaries of more than one sensing discs.

Conversely, let ROI be not covered by a set of nodes  $\mathcal{N}$ . Then there is a point  $A$  such that  $d(A, N) > r$  for all  $N \in \mathcal{N}$ . Let,  $d = \min\{d(A, N') : N' \in \mathcal{N}\}$ , then the boundary of the ball  $B_{d-r}(A)$  intersects at most one point with boundary of each sensing ball and there is a sensing ball whose boundary, say  $Bd$ , such that  $Bd \cap B_{d-r}(A)$  is a singleton set, say,  $\{B\}$ . Then  $B$  is a point which belongs to the intersection of the boundary of sensing ball of a node and the interior of the ROI but neither belongs to the interior nor on the boundary of any other sensing ball.

Moreover, if  $\text{ROI} \subset \mathbb{R}^2$ , in that case, the set of interior points of ROI which is on the circumference of sensing disc of a node and does not belong to the interior of any other sensing disc is infinite (an suitable arc containing  $B$ ).  $\square$

*Remark 3.2.* It seems that when  $\mathbb{R}^n$  is covered by a set of nodes with minimum wastage then the intersection of interior of three balls centered at three distinct nodes is empty. In this chapter we consider only the situations like Figure 3.3a and 3.3d but not like Figure 3.3c.

Before going to  $\mathbb{R}^2$  and  $\mathbb{R}^3$ , we state few results on  $\mathbb{R}^1$ . If  $\mathcal{N} = \mathcal{V} = 2r\mathbb{Z} = \{2rn : n \in \mathbb{Z}\}$ , wastage will be 0. Let  $d(V, N(V))$  follows i.i.d. with distribution  $F(\cdot)$  for  $V \in \mathcal{V}$ . Consider a point  $x \in \mathbb{R}$ . Now the probability of covering  $x$  by node whose corresponding vertex  $2nr$  is  $F(x + r - 2rn) - F(x - r - 2rn)$ . Hence  $x$  is uncovered with probability

$$\prod_{n=-\infty}^{\infty} (1 - F(x + r - 2rn) + F(x - r - 2rn)).$$

The intervals  $[n - r, n + r]$  are symmetric with respect to  $\mathcal{V}$  for all  $n \in \mathbb{Z}$ . Wastage is same in all intervals  $[n - r, n + r]$  which is  $W = \frac{1}{2r} \int_{-r}^r \prod_{n=-\infty}^{\infty} (1 - F(x + r - 2rn) + F(x - r - 2rn)) dx$  and hence wastage in  $\mathbb{R}$  is  $W$ .

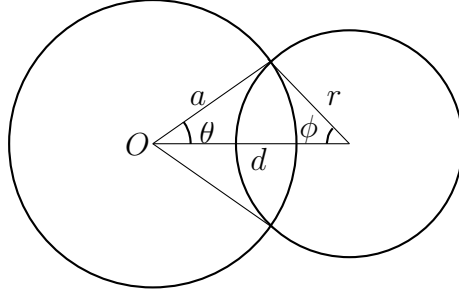


Figure 3.4: Intersection of two discs of different radii,  $\theta = \cos^{-1}\left(\frac{a^2+d^2-r^2}{2ad}\right)$ ,  $\phi = \cos^{-1}\left(\frac{r^2+d^2-a^2}{2rd}\right)$

### 3.4 Coverage Problem in $\mathbb{R}^2$

**Theorem 3.3.** *Let  $r$  be the sensing radius of the sensors. If a sensor in  $\mathbb{R}^2$  can detect the distance and the position (w.r.t. some coordinate system) of adjacent sensors, which are placed at distance  $d$  ( $0 < d < 2r$ ), then sensors can detect the adjacent sensing holes.*

*Proof.* If two nodes are at distance  $d$  ( $< 2r$ ), then the length of the arc which is common to the boundaries of the sensing discs of both the nodes is  $2r \cos^{-1}\left(\frac{d}{2r}\right)$  (see Figure 3.4 with  $a = r$ ). Since the sensor can detect the position of a sensor which is at distance less than  $2r$ , it can detect which part of the circumference of its sensing disc intersects with an adjacent sensing disc. Therefore, it can detect whether any part of the boundary of its sensing disc exists that has no intersection with any other sensing disc or every point of the boundary interest with another sensing disc. Then the proof follows from Theorem 3.1.  $\square$

**Lemma 3.4.** *Let boundaries of three circles  $C_1, C_2$  and  $C_3$  with centers  $A, B$  and  $C$  respectively, have exactly one common point  $O$  as shown in Figure 3.5. Let  $P, Q$  and  $R$  be the common points, other than  $O$ , of boundaries of  $C_1, C_2$ ;*

*$C_2, C_3$  and  $C_3, C_1$  respectively. Then  $\widehat{POQ} + \widehat{QOR} + \widehat{ROP} = 2\pi r$ .*

*Proof.* Let  $\angle OAP = 2\alpha$ ,  $\angle OBQ = 2\beta$  and  $\angle OCR = 2\gamma$ . Then  $\angle AOB = \pi - 2\alpha$ ,  $\angle BOC = \pi - 2\beta$  and  $\angle COA = \pi - 2\gamma$ . Since  $\angle AOB + \angle BOC + \angle COA = 2\pi$ , we have  $\alpha + \beta + \gamma = \frac{\pi}{2}$ .

Now  $\widehat{POQ} = 2r(\alpha + \beta)$ ,  $\widehat{QOR} = 2r(\beta + \gamma)$  and  $\widehat{ROP} = 2r(\gamma + \alpha)$ . Hence, the result.  $\square$

**Theorem 3.5.** *Consider that  $\mathbb{R}^2$  is covered by a set of nodes such that intersection of interior of any three sensing discs centered at three distinct nodes is empty. Then the placement of nodes to cover  $\mathbb{R}^2$  (under the above restriction) with minimum wastage is as follows: partition  $\mathbb{R}^2$  into equilateral triangles having sides  $\sqrt{3}r$  and take nodes to be the vertices of those equilateral triangles (see figure 3.2). Note that the distance between centers of two adjacent regular hexagons of side  $r$  is  $\sqrt{3}r$ .*

*Proof.* Note that by Theorem 3.1 covering  $\mathbb{R}^n$  is equivalent to covering every point on the boundary of each disc by another disc. Since the set of nodes is such, that intersection of interior of any three sensing discs centered at three distinct nodes is empty, the total arc length of the boundary of a disc covered by other discs is exactly  $2\pi r$ .

Consider the situation when three nodes are placed at  $A, B$  and  $C$  as described in Lemma 3.1, then they cover overall  $2\pi r$  arc length of three discs. Now the area of  $(C_1 \cap C_2) \cup (C_2 \cap C_3) \cup (C_3 \cap C_1)$  is

$$\begin{aligned} \sum_{\theta \in \{\alpha, \beta, \gamma\}} 4 \left( \frac{1}{2} r^2 \theta - \frac{1}{2} (r \sin \theta)(r \cos \theta) \right) &= \sum r^2 (2\theta - \sin 2\theta) \\ &= \pi r^2 - r^2 (\sin 2\alpha + \sin 2\beta + \sin 2\gamma). \end{aligned}$$

Consider a triplet of discs  $(C_1, C_2, C_3)$  such that  $C_1, C_2, C_3$  intersect exactly at one point  $O$ . Let  $H$  is the set of interior points and boundary points of the hexagon  $APBQCR$ , see Figure 3.5. Now

$$\begin{aligned} W_{\mathcal{N} \cap H}(H) &= \frac{\sum_{i=1,2,3} \text{Area}(H \cap C_i) - \text{Area}(H)}{\sum_{i=1,2,3} \text{Area}(H \cap C_i)} = \frac{\pi r^2 - \sum_{\theta \in \{\alpha, \beta, \gamma\}} r^2 \sin 2\theta}{\pi r^2} \\ &= 1 - \frac{1}{\pi} (\sin 2\alpha + \sin 2\beta + \sin 2\gamma), \end{aligned}$$

Note that the denominator is same for all hexagons of the above type (one of

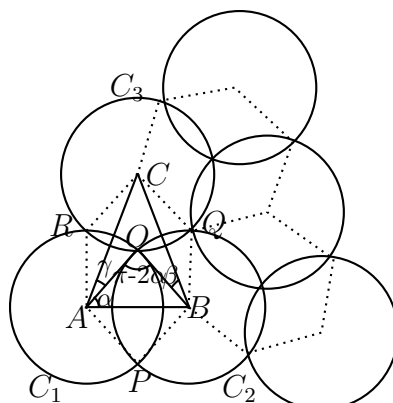


Figure 3.5: Hexagonal tiling of ROI

them is  $APBQCR$ ), which is a partition of  $\mathbb{R}^2$ . Hence the wastage in  $\mathbb{R}^2$  is the limit (as  $x \rightarrow \infty$ ) of the average of wastages in all such hexagons corresponds to triplet of discs which intersect exactly at one point  $P \in B_x$ , where  $B_x$  is the ball in  $\mathbb{R}^2$  of radius  $x$  and centered at origin. So if we minimize the wastage in the corresponding hexagon (one of them is  $APBQCR$ ) for each triplet of discs (one of them is  $(C_1, C_2, C_3)$ ) with the above property, we get the optimal placement of nodes. So we need to maximize  $\sin 2\alpha + \sin 2\beta + \sin 2\gamma$  subject to  $\alpha + \beta + \gamma = \frac{\pi}{2}$ . Using Jensen's inequality one can determine that the wastage in  $H$  is minimum when  $\alpha = \beta = \gamma$ . If we take any other placement then the wastage in the corresponding hexagon is greater than  $1 - \frac{1}{\pi}(\sin 2\alpha + \sin 2\beta + \sin 2\gamma)$ .

Now consider the situation when four discs meet at a point. Consider the octagon joining the centers and the point of intersections of pair of circles, see Figure 3.3d. In this case also the wastage in that octagon is greater than  $1 - \frac{1}{\pi}(\sin 2\alpha + \sin 2\beta + \sin 2\gamma)$ . Note that considering all triplet  $(C_1, C_2, C_3)$  we consider each hexagon and octagon exactly twice. Hence the wastage does not change as it is a proportion. Also note that there are no other situations other than hexagon and octagon. Hence the result.  $\square$

**Hexagonal placement of nodes:** Define the placement (as in Theorem 3.5 with  $\alpha = \beta = \gamma$ ) of nodes in  $\mathbb{R}^2$  as *Hexagonal placement*.

In this case the wastage is  $1 - \frac{3\sqrt{3}}{2\pi}$ , which is approximately 0.16. Now since the area of a regular hexagon with side  $r$  is  $\frac{3\sqrt{3}}{2}r^2$ , the number of sensors needed to cover an area  $A$  which can be partitioned into identical regular hexagons of



side  $r$  is  $n = \frac{2A}{3\sqrt{3}r^2}$ , where  $r$  is the sensing radius.

In practical situations ROI is finite. Suppose we have  $N$  discs each of radius  $r$ . We want to cover ROI as much as possible with these discs. Let the coordinate of the center of the first disc be  $(0,0)$ . In Algorithm 1, we present our placement of the centers of the discs formally.

<p><b>Input:</b> Total number of discs is <math>N</math> with radius <math>r</math>.  <b>Input:</b> Maximum number <math>R</math> of discs can be placed in a row.  <b>Output:</b> Coordinate of center of discs.</p> <pre> 1  <math>S = 2R - 1;</math> 2  <math>X[N], Y[N], i = 0, a = 0;</math> 3  <b>while</b> <math>i &lt; N</math> <b>do</b> 4      <b>if</b> <math>i \bmod S = 0</math> <b>and</b> <math>i &gt; 0</math> <b>then</b> 5          <math>a = a - \sqrt{3}r;</math> 6      <b>end</b> 7      <b>if</b> <math>i \bmod S &lt; R</math> <b>then</b> 8          <math>X[i] = (i \bmod S)3r;</math> 9          <math>Y[i] = a;</math> 10     <b>end</b> 11     <b>else</b> 12         <math>X[i] = ((i + R - 1) \bmod S)3r + \frac{3r}{2};</math> 13         <math>Y[i] = a - \frac{\sqrt{3}r}{2};</math> 14     <b>end</b> 15     <math>i = i + 1;</math> 16 <b>end</b> 17 Report <math>x</math> coordinates <math>X[N]</math> and <math>y</math> coordinates <math>Y[N]</math> of discs; </pre>
--

**Algorithm 1:** Hexagonal placement of discs.

*Remark 3.6.* If  $A$  is any subset of  $\mathbb{R}^2$  and  $A$  is covered by a set of nodes such that the intersection of interiors of any three sensing discs centered at three distinct nodes is empty, then the hexagonal placement of nodes may not be the optimal one. But if  $A$  is convex and large in both length and width with respect to  $r$  (i.e. number of boundary nodes are considerably less than total number of nodes) then this placement may be optimal or very close to optimal in the sense that the wastage is close to minimum. The wastage is greater than the minimum value due to the nodes placed at the boundary of  $A$ . So in WSNs we can use the hexagonal placement of nodes with special care at the boundary of ROI when ROI is convex and large.

**Theorem 3.7.** *Let a node be targeted to be placed at a point  $A$ , but it is placed at point  $C$  on the plane i.e.,  $C = N(A)$ . Let the distribution of the distance between  $A$  and  $C$  have the density  $f(\cdot)$ . Let  $B$  be a point at a distance  $d$  from the point  $A$ . Then the probability  $P_A(B)$  that the point  $B$  is in the sensing disc of the node  $C$  is*

$$P_A(B) = \begin{cases} \int_0^{r-d} f(x)dx + \frac{1}{\pi} \int_{r-d}^{r+d} \cos^{-1}\left(\frac{d^2+x^2-r^2}{2xd}\right) f(x)dx, & \text{if } 0 \leq d \leq r, \\ \frac{1}{\pi} \int_{d-r}^{d+r} \cos^{-1}\left(\frac{d^2+x^2-r^2}{2xd}\right) f(x)dx, & \text{if } d > r. \end{cases}$$

*Proof.* Note that, the point  $B$  is in the sensing disc of the node  $C$  if and only if  $C$  belongs to the disc of radius  $r$  and center  $B$ . Call this disc  $D$ . For  $0 \leq d < r$  the intersection of  $D$  and the boundary of the disc of radius  $x$  and center  $A$  is an arc of length  $2x \cos^{-1}\left(\frac{d^2+x^2-r^2}{2xd}\right)$  if  $r-d < x < r+d$ . Hence, the conditional probability that  $C$  falls in  $D$  given that  $C$  is at a distance  $x$  from  $A$  is

$$\frac{2x \cos^{-1}\left(\frac{d^2+x^2-r^2}{2xd}\right)}{2\pi x} = \frac{1}{\pi} \cos^{-1}\left(\frac{d^2+x^2-r^2}{2xd}\right) \text{ if } r-d < x < r+d.$$

Now the conditional probability that  $C$  falls in  $D$  given that  $C$  is at a distance  $x$  from  $A$  is 1 if  $0 \leq x \leq r-d$  and 0 if  $x > r+d$ . Hence the theorem for  $d < r$  (see Figure 3.6a). The proof is similar for  $r \leq d$  (see Figure 3.6b).  $\square$

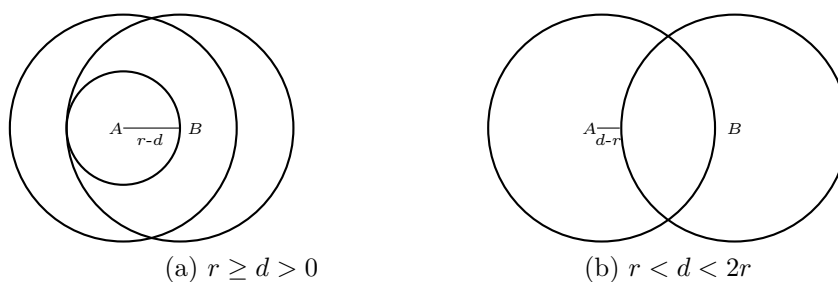


Figure 3.6: Intersection of two discs of radius  $r$  in two different cases

**Corollary 3.8.** *Let a node, targeted to be placed at a point  $A$ , be placed at a randomly chosen point,  $C$  in the disc of radius  $a$  and center  $A$ . Let  $B$  be a point at a distance  $d$  from the point  $A$ . Then the probability that the point  $B$  is in the sensing disc of the node  $C$  is  $\frac{I}{\pi a^2}$ , where  $I$  is the intersecting area of the two discs of radii  $a$  and  $r$  whose centers are at a distance  $d$ .*

Moreover, if  $a = r$  then the above probability is  $\frac{2}{\pi} \cos^{-1}\left(\frac{d}{2r}\right) - \frac{d}{2\pi r^2} \sqrt{4r^2 - d^2}$  if  $d \leq 2r$ , and 0 otherwise.

*Proof.* Applying the Theorem 3.7 and using integration by parts one can obtain the required probability is  $\frac{I}{\pi a^2}$ , where  $I$  is as follows: when  $a > r$ ,

$$I = \begin{cases} 0, & \text{for } d \geq a + r, \\ \pi r^2, & \text{for } d \leq a - r, \\ \frac{a^2}{2}(2\theta - \sin(2\theta)) + \frac{r^2}{2}(2\phi - \sin(2\phi)) & \text{otherwise,} \end{cases}$$

where  $\cos \theta = \frac{a^2 + d^2 - r^2}{2ad}$  and  $\cos \phi = \frac{r^2 + d^2 - a^2}{2rd}$ , and when  $a < r$ , one can calculate the required probability just interchanging  $a$  and  $r$ , as the expression for  $I$  is a symmetric function of  $a$  and  $r$ .

Note that,  $I$  is same as the intersecting area of the two discs of radii  $a$  and  $r$  whose centers are at a distance  $d$ .

If  $a = r$  then  $\cos \theta = \cos \phi = d/2r$ , hence the result. We can prove this independently as follows, since the point  $C$  is chosen randomly in the disc of radius  $r$  and center  $A$ ,  $f(x) \propto (2\pi x)I_{(0,r)}$ , as the length of the circumference of a circle of radius  $x$  is  $2\pi x$ , where  $I_{(0,r)}$  is the indicator function of the interval  $(0, r)$ . As  $f(\cdot)$  is density function we have  $f(x) = \frac{2x}{r^2}I_{(0,r)}$ . Applying the Theorem 3.7 and using integration by parts one can obtain the result.  $\square$

**Theorem 3.9.** *Let  $n$  nodes be targeted to be placed at centers  $A_i$  for  $i = 1, 2, \dots, n$  of  $n$  regular hexagons in ROI such that the  $i^{\text{th}}$  node is targeted to be placed at point  $A_i$ , but it is placed at a randomly chosen point  $C_i$  on the plane. Let  $P_{A_i}(B)$  denote the probability that the point  $B$  is in the sensing disc of  $C_i$ . Suppose  $D_i = d(A_i, C_i)$  be i.i.d. with density function  $f(\cdot)$ . Let  $B$  be a point at a distance  $d_i$  from the point  $A_i$ . Define  $\mathcal{A} = \{A_i : i = 1, \dots, n\}$ . Then the probability that the point  $B$  is in the sensing disc of at least one of the  $n$  nodes  $C_i$  for  $i = 1, 2, \dots, n$  is  $P_{\mathcal{A}}(B) = 1 - \prod_{i=1}^n (1 - P_{A_i}(B))$ . The expected area of the region whose points are in the sensing disc of at least one node is*

$$\text{Area}(\text{ROI}) - \int_{B \in \text{ROI}} \prod_{i=1}^n (1 - P_{A_i}(B)),$$

where  $\text{Area}(\text{ROI})$  denotes the total area of ROI.

*Proof.* It is clear from the Theorem 3.7 that the probability of the point  $B$  not being in the sensing disc of the node  $C_i$  is  $1 - P_{A_i}(B)$ . Since the distributions of the distances between  $A_i$  and  $C_i$  are identical, the probability that the point  $B$  is not in the sensing disc of all the  $n$  nodes  $C_i$  for  $i = 1, 2, \dots, n$  is  $\prod_{i=1}^n (1 - P_{A_i}(B))$ , Hence the first part of the lemma.

For  $B \in ROI$ , let  $X_B$  be the random variable which takes the value 1 if  $B$  is in the sensing disc of at least one of the  $n$  nodes  $C_i$  for  $i = 1, 2, \dots, n$  and 0 otherwise. Now  $P(X_B = 1) = 1 - \prod_{i=1}^n (1 - P_{A_i}(B))$ , hence  $E(X_B) = 1 - \prod_{i=1}^n (1 - P_{A_i}(B))$ , where  $E(\cdot)$  denotes the expectation. Now the expected area of the region whose points are in the sensing disc of at least one node is

$$\int_{B \in ROI} (1 - \prod_{i=1}^n (1 - P_{A_i}(B))) = Area(ROI) - \int_{B \in ROI} \prod_{i=1}^n (1 - P_{A_i}(B)).$$

Hence the result. □

*Remark 3.10.* If  $C_i$  are distributed independently and uniformly in the disc of radius  $r$  and center  $A_i$  and  $B$  is a point in a regular hexagon centered at the point  $A_1$  then  $P_A(B) = 1 - \prod_{i=1}^7 (1 - P_{A_i}(B))$ , where the points  $A_i$  for  $i = 2, 3, \dots, 7$  are the centers of the six regular hexagons adjacent to the hexagon centered at  $A_1$ , because the vertices other than  $A_i$  for  $i = 1, 2, \dots, 7$  are at a distance greater than  $2r$  from any point inside the regular hexagon centered at  $A_1$ .

**Theorem 3.11.** *Let  $n$  nodes be targeted for placement at centers  $A_i$  for  $i = 1, 2, \dots, n$  of the  $n$  regular hexagons which partitions  $ROI$  such that  $i^{th}$  node is targeted to place at point  $A_i$ , but is placed at a randomly chosen point  $C_i$  on the plane. Let  $C_i$ 's be distributed independently and uniformly in the disc of radius  $r$  and center  $A_i$ . Then the proportion of the expected area of  $ROI$  which is covered by at least one of the nodes is close to*

$$1 - \frac{8}{\sqrt{3}} \int_{x=0}^{\frac{1}{2}} \int_{y=0}^{\sqrt{3}(\frac{1}{2}-x)} \prod_{i=1}^7 \left( 1 - \frac{2}{\pi} \left( \cos^{-1}(d_i) - d_i \sqrt{1 - d_i^2} \right) I_{(0,1)}(d_i) \right) dy dx,$$

where  $2d_i$  is the distance between the point  $B$  and  $P_i$  and the coordinates of  $B$

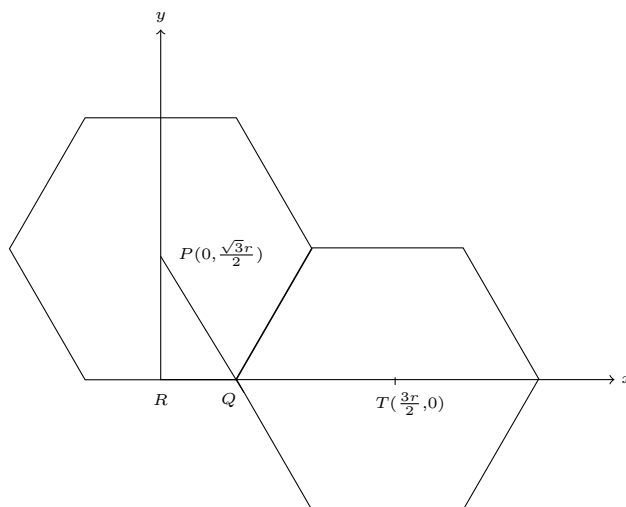


Figure 3.7: One of the 12 triangles which partitioned a regular Hexagon

and  $P_i$ 's are  $(x, y)$ ,  $(0, \sqrt{3}/2)$ ,  $(0, -\sqrt{3}/2)$ ,  $(0, 3\sqrt{3}/2)$ ,  $(3/2, 0)$ ,  $(3/2, \sqrt{3})$ ,  $(-3/2, 0)$ ,  $(-3/2, \sqrt{3})$  respectively for  $i = 1, 2, \dots, 7$ .

*Proof.* Note that a point is not covered by  $C_i$  with probability 1 if and only if the point is at a distance more than  $2r$  from the point  $A_i$ . Since the disc of radius  $2r$  and center at a vertex  $V$  intersects only with the seven hexagons corresponding to  $V$  and its six adjacent vertices, the points inside a hexagon can be covered by the node corresponding to the adjacent 6 vertices and that node corresponds to that hexagon only. Now by symmetry, the proportion of the area of a hexagon which is covered by at least one node is the same for all the interior hexagons and for the boundary hexagon it is less than that of an interior hexagon. So if  $n$  is large the required area is almost the same as (slightly less than and if we are able to partition the ROI into regular hexagons then equal to) the proportion of the expected area of an interior hexagon which is covered by at least one of the six nodes corresponding to the six adjacent vertices and the node corresponds to that particular interior hexagon.

An interior regular hexagon can be partitioned into 12 congruent triangles. One of them say,  $PQR$ , where  $P$  is the center of the above hexagon,  $Q$  is a vertex of the above hexagon and  $R$  is the midpoint of a side whose one vertex is  $Q$  (see Figure 3.7). Note that these 12 triangles are symmetric in the context of covering. Hence, the required proportion of expected area is same as the

proportion of the expected area of the triangle  $PQR$  which is covered by at least one of the seven nodes corresponding to the six adjacent vertices and the center of the above hexagon.

Now by the Theorem 3.9 and Corollary 3.8 the proportion of the expected area of the triangle  $PQR$  which is uncovered by the seven aforesaid nodes is  $\frac{8}{\sqrt{3}r^2} \times$

$$\int_{x=0}^{\frac{r}{2}} \int_{y=0}^{\sqrt{3}(\frac{r}{2}-x)} \prod_{i=1}^7 \left( 1 - \left( \frac{2}{\pi} \cos^{-1} \left( \frac{d_i}{2r} \right) - \frac{d_i}{2\pi r^2} \sqrt{4r^2 - d_i^2} \right) I_{(0,2r)}(d_i) \right) dydx$$

where  $d_i$  is the distance between the point  $B$  and  $A_i$  and the coordinates of  $B$  and  $A_i$ 's are  $(x, y), (0, \sqrt{3}r/2), (0, -\sqrt{3}r/2), (0, 3\sqrt{3}r/2), (3r/2, 0), (3r/2, \sqrt{3}r), (-3r/2, 0), (-3r/2, \sqrt{3}r)$  respectively for  $i = 1, 2, \dots, 7$ .

On changing the variables  $x$  to  $rx$  and  $y$  to  $ry$  and  $d_i$  to  $rd_i$  we see that the above expression is same as

$$\frac{8}{\sqrt{3}} \int_{x=0}^{\frac{1}{2}} \int_{y=0}^{\sqrt{3}(\frac{1}{2}-x)} \prod_{i=1}^7 \left( 1 - \left( \frac{2}{\pi} \cos^{-1} \left( \frac{d_i}{2} \right) - \frac{d_i}{2\pi} \sqrt{4 - d_i^2} \right) I_{(0,2)}(d_i) \right) dydx,$$

where  $d_i$  is the distance between the point  $B$  and  $P_i$  and the coordinates of  $B$  and  $P_i$ 's are  $(x, y), (0, \sqrt{3}/2), (0, -\sqrt{3}/2), (0, 3\sqrt{3}/2), (3/2, 0), (3/2, \sqrt{3}), (-3/2, 0), (-3/2, \sqrt{3})$  respectively for  $i = 1, 2, \dots, 7$ .

Now on changing  $d_i$  to  $2d_i$  we get the results. □

### 3.5 Simulation Results

Since the sensing radius  $r$  has no effect on the simulation results we consider 10000 discs with sensing disc of radius 1 and 10000 vertices. We consider ROI as a regular hexagonal grid and target to drop a node on the center of each unit regular hexagon of the grid. Note that area of ROI is  $10000 \times \frac{3\sqrt{3}}{2}$  unit. The distance between two adjacent vertices is  $\sqrt{3}$  unit. 100 nodes are arranged in rows. Now due to stochastic deployment, a node may not fall on the corresponding vertex but on a neighboring point (node). Let the distance between the vertices and corresponding nodes are i.i.d. uniform or normal. Let  $p\%$  more nodes be used, i.e., total number of nodes is  $10000(1 + \frac{p}{100})$  where

$p \in [0, 100]$ . We simulate the uncovered area using two different strategies.

The first strategy (Strategy 1 or St. 1) is as follows: choose  $100p$  vertices randomly from 10000 vertices and generate 2 nodes corresponding to each of these vertices and 1 node for each of the other  $100(100 - p)$  vertices. Then simulate the uncovered area of ROI by choosing 100000 points from ROI and counting the number of points which are covered by at least one of these  $10000(1 + \frac{p}{100})$  nodes and then divide that number by 100000. We repeat this whole procedure 1000 times and find the average of the ratio.

The second strategy (Strategy 2 or St. 2) is as follows: consider ROI as a hexagonal grid of  $10000(1 + \frac{p}{100})$  regular hexagon of side  $\sqrt{\frac{100}{100+p}}$ . Consider  $10000(1 + \frac{p}{100})$  centers of these hexagons as vertices and generate 1 node for each vertex. Note that the area of ROI is  $10000(1 + \frac{p}{100}) \times \frac{3\sqrt{3}}{2} \left(\sqrt{\frac{100}{100+p}}\right)^2$ , which is same as the previous one. Then we simulate the uncovered area of ROI as before.

Table 3.1: Simulation results for the proportion of area covered for two strategies (St. 1 and St. 2).

	$U(0.5)$		$U(1)$		$N(0, 0.10)$		$N(0, 0.25)$		$N(0, 0.50)$	
$p$	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2
0.00	0.9061	0.9092	0.8167	0.8145	0.9863	0.9867	0.9280	0.9260	0.8532	0.8505
0.05	0.9148	0.9131	0.8270	0.8238	0.9878	0.9901	0.9320	0.9372	0.8619	0.8577
0.10	0.9198	0.9188	0.8384	0.8283	0.9884	0.9931	0.9372	0.9454	0.8660	0.8638
0.15	0.9221	0.9230	0.8489	0.8423	0.9903	0.9941	0.9416	0.9490	0.8768	0.8773
0.20	0.9299	0.9282	0.8566	0.8516	0.9897	0.9960	0.9454	0.9535	0.8851	0.8881
0.25	0.9359	0.9351	0.8652	0.8552	0.9903	0.9963	0.9490	0.9586	0.8950	0.8912
0.50	0.9550	0.9546	0.9068	0.8904	0.9937	0.9981	0.9656	0.9764	0.9261	0.9201
0.75	0.9716	0.9737	0.9387	0.9135	0.9956	0.9988	0.9805	0.9861	0.9539	0.9411
1.00	0.9866	0.9856	0.9653	0.9319	0.9974	0.9985	0.9912	0.9915	0.9756	0.9553

Let  $D_i$  be the distances between vertices and the corresponding node. In simulations we consider five different distributions for  $D_i$ 's. For both strategies, we consider that the distances between vertex and its corresponding nodes, follow five different distributions  $U(0.5), U(1), N(0, 0.10), N(0, 0.25), N(0, 0.50)$  and they are independent. Here  $U(t)$  denotes the distribution whose density function is  $f(x) = \frac{2x}{t^2} I_{(0,t)}$  and  $N(0, t^2)$  be the normal distribution with mean 0 and s.d.  $t$ . We simulate the uncovered area separately for these 5 different

distributions and compare them with respect to different values of  $p$ . We also simulate and draw ‘proportion of covered area  $\delta$ ’ vs. ‘ $p$ ’ graphs for 5 different distributions (see Figure 3.8a to Figure 3.8e).

From the experimental results (see Figure 3.8a-3.8e and Table 3.1) it is noted that Strategy 1 is better in the case of  $U(1)$  and  $N(0, 0.5)$  but Strategy 2 is better for the other three distributions. So we can conclude that Strategy 1 is better when s.d. of the distributions is high and Strategy 2 is better when s.d. of the distributions is low. We also numerically calculate the coverage area in case of  $U(1)$  when  $p = 0$  using numerical integration method (see Theorem 3.11). We see that this value is close enough to the simulated value.

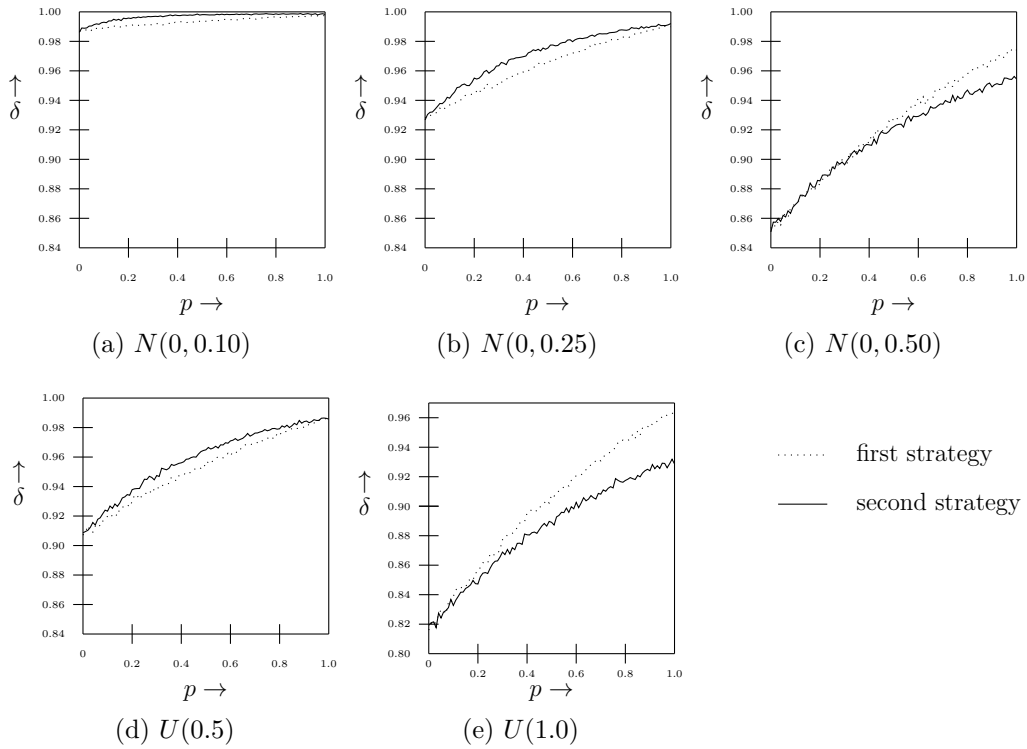


Figure 3.8: Proportion Coverage area in  $\mathbb{R}^2$  for different distribution.



## 3.6 Coverage Problem in $\mathbb{R}^3$ and Simulation Results

The hexagonal placement is optimal for the sphere packing problem in  $\mathbb{R}^2$ . We discuss the hexagonal placement of nodes for the coverage problem in  $\mathbb{R}^2$  in previous two sections. It is well known that face-centered cube packing is optimal for the sphere packing problem in  $\mathbb{R}^3$  and for  $n > 3$  optimal placement is not known [32]. In various situations WSNs may be three dimensional. In this section we discuss a similar type of placement of nodes (similar to face-centered cube packing) to cover  $\mathbb{R}^3$ . Consider the set  $\mathcal{N} = \{(2k, 2l, 2m) : k, l, m \in \mathbb{Z}\} \cup \{(2k+1, 2l+1, 2m) : k, l, m \in \mathbb{Z}\} \cup \{(2k+1, 2l, 2m+1) : k, l, m \in \mathbb{Z}\} \cup \{(2k, 2l+1, 2m+1) : k, l, m \in \mathbb{Z}\}$ . We partition  $\mathbb{R}^3$  as a cube grid and take the nodes at the 8 corners and the center of the 6 faces of all the cubes. If  $r$  be the sensing radius then we consider set of nodes  $\{rN : N \in \mathcal{N}\}$ . The placement of nodes in our case is similar to the choice of center of spheres in face centered cube packing, the only difference being that the distance between the two nodes is less in our case which confirms the covering.

**Theorem 3.12.** *Consider a partition of a cube  $C$  of side  $2nr$  unit into  $n^3$  many cubes of side  $2r$  unit each. Let the nodes be placed as discussed above. Then the number of nodes required to cover the cube  $C$  is  $4n^3 + 6n^2 + 3n + 1$  and the proportion of wastage volume for sufficiently large  $n$  is approximately  $1 - \frac{3}{2\pi}$ .*

*Proof.* Clearly there are  $(n+1)^3$  corner nodes and  $n^2(n+1)$  nodes at the center of faces parallel to one of the three coordinate planes. Hence the number of nodes is  $(n+1)^3 + 3n^2(n+1) = 4n^3 + 6n^2 + 3n + 1$ .

We need  $4n^3 + 6n^2 + 3n + 1$  spheres of radius  $r$  to cover the cube of side  $2nr$ . The total volume of the spheres is  $(4n^3 + 6n^2 + 3n + 1)\frac{4}{3}\pi r^3$  and they cover volume of  $8n^3r^3$  units. Hence the proportion of wastage volume is  $\frac{(4n^3+6n^2+3n+1)\times\frac{4}{3}\pi r^3-8n^3r^3}{(4n^3+6n^2+3n+1)\times\frac{4}{3}\pi r^3} = 1 - \frac{8n^3}{(4n^3+6n^2+3n+1)\times\frac{4}{3}\pi}$ . For large  $n$ ,  $1 - \frac{8n^3}{(4n^3+6n^2+3n+1)\times\frac{4}{3}\pi} = 1 - \frac{8n^3}{n^3\times\frac{4}{3}\pi}$ . Hence the result.  $\square$

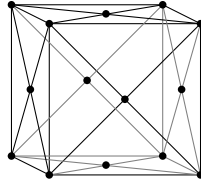


Figure 3.9: Face-Centered Cube (dots are center of spheres)

### 3.6.1 Simulation Results

We consider  $n = 13$  and  $r = 1$ . So, we have 9842 many nodes. Hence the volume of ROI is  $13^3 \times 2^3$  unit. We simulate the proportion for the covered volume using two strategies, Strategy 1 (St. 1) and Strategy 2 (St. 2), which are exactly same as in case of  $\mathbb{R}^2$ . If we use  $p\%$  extra nodes then for Strategy 2, we have to partition ROI of volume  $13^3 \times 2^3$  unit into  $m^3$  many cubes where  $4m^3 + 6m^2 + 3m + 1 = 9842 \times (1 + \frac{p}{100})$ . We simulate the proportion of coverage for two different strategies and for five different distributions as described in previous the section (in case of  $\mathbb{R}^2$ ). The uniform distribution with parameter  $t$  has the density function  $f(x) = \frac{3x^2}{t^3} I_{(0,t)}$ . It is noted from the simulation results that, St. 1 is better than St. 2 in the higher variance cases. For lower variance cases St. 2 is better for most values of  $p$ . This observation is almost same as in the two dimensional case. So we can conclude that Strategy 1 is better for distributions with higher variance and Strategy 2 is better for distributions with lower variance.

Table 3.2: Simulation for proportion of coverage area for two strategies in  $\mathbb{R}^3$

	$U(0.5)$		$U(1)$		$N(0, 0.10)$		$N(0, 0.25)$		$N(0, 0.50)$	
$p$	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2	St. 1	St. 2
0.00	0.9709	0.9709	0.9235	0.9233	0.9658	0.9642	0.9385	0.9347	0.9348	0.9330
0.05	0.9690	0.9710	0.9290	0.9292	0.9685	0.9688	0.9440	0.9455	0.9408	0.9327
0.10	0.9759	0.9737	0.9338	0.9335	0.9695	0.9708	0.9470	0.9475	0.9438	0.9456
0.15	0.9775	0.9781	0.9423	0.9368	0.9723	0.9779	0.9540	0.9505	0.9528	0.9458
0.20	0.9811	0.9825	0.9460	0.9450	0.9797	0.9765	0.9618	0.9550	0.9530	0.9518
0.25	0.9853	0.9850	0.9568	0.9439	0.9823	0.9787	0.9645	0.9577	0.9620	0.9600
0.50	0.9918	0.9915	0.9730	0.9625	0.9913	0.9859	0.9787	0.9705	0.9803	0.9669
0.75	0.9955	0.9945	0.9903	0.9707	0.9958	0.9936	0.9930	0.9817	0.9870	0.9785
1.00	0.9985	0.9975	0.9945	0.9795	0.9980	0.9952	0.9953	0.9865	0.9943	0.9827

# Deterministic Covering of Cylindrical Grid

## 4.1 Introduction

In this chapter we consider ROI as a cylindrical grid. We assume that an event can occur at any vertices of the grid and sensors can be placed only at vertices. Sensor at any vertex  $v$  can detect an event occurred at vertex  $v$  and four (or three, in case of boundary) adjacent vertices of  $v$ . Finding a set of sensors which cover all vertices of a cylindrical grid is similar to finding a dominating set of that cylindrical grid graph. Several works have been done on domination number of rectangular graph but no theoretical results are there for cylindrical graphs. We find theoretical results in some special cases. This is the area of Graph theory. The graphs, considered here, are finite, nonempty, connected, undirected, without loops and without multiple edges. Besides these, any undefined terms in this chapter may be found in Harary [33].

### 4.1.1 Definitions

Let  $G$  be a simple graph whose vertex set and edge set are  $V(G)$  and  $E(G)$ , respectively. A set  $D \subseteq V(G)$  of a simple graph  $G$  is called a *dominating set* if every vertex  $v \in V(G) \setminus D$  is adjacent to some vertex  $u \in D$ . The *domination number* of  $G$  is the cardinality of a smallest dominating set of the graph  $G$  and

it is usually denoted by  $\gamma(G)$ . In addition, a smallest such dominating set is called a *minimum dominating set* of  $G$ .

For any two graphs  $G$  and  $H$ , the *Cartesian product*  $G \square H$  is the graph with vertex set  $V(G) \times V(H)$  and edge set  $E(G \times H)$  such that  $(u_1, v_1)(u_2, v_2) \in E(G \times H)$ , when  $v_1 = v_2$  and  $u_1u_2 \in E(G)$ , or  $u_1 = u_2$  and  $v_1v_2 \in E(H)$  [43].

Throughout the chapter, the following notation and terminology are used. The numbers  $0, 1, 2, \dots, n - 1$  always denote the vertices of the path  $P_n$  or the cycle  $C_n$ . Also, let  $\gamma(P_m \square P_n)$  and  $\gamma(P_m \square C_n)$  denote the domination numbers of Cartesian product graphs  $P_m \square P_n$  and  $P_m \square C_n$ , respectively. The graph  $P_m \square C_n$  can be termed as *cylindrical grid graph* as shown in Figure 4.1 in two different looks.

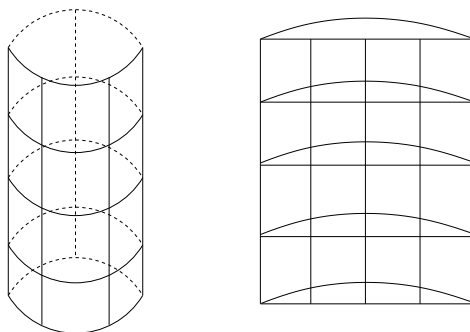


Figure 4.1: Two different looks of the cylindrical grid graph  $P_5 \square C_5$ .

Let  $(G)_v = G \times \{v\}$  where  $v \in V(H)$  and  $(H)_u = \{u\} \times H$  where  $u \in V(G)$ .  $(G)_v$  and  $(H)_u$  are called the *layers* of  $G$  and  $H$ , respectively. Moreover, the layer of a dominating set means  $D \cap (P_m)_i$  for  $i \in V(C_n)$ . Throughout the chapter, the leftmost column in all figures denotes the layer  $(P_m)_0$ . Now, we define the term *modified concatenation*, of two dominating sets of  $P_m \square C_{n_1}$  and  $P_m \square C_{n_2}$ . If  $D_1$  and  $D_2$  are two dominating sets of  $P_m \square C_{n_1}$  and  $P_m \square C_{n_2}$ , respectively then the modified concatenation of  $D_1$  and  $D_2$ , denoted by  $D_1 || D_2$ , is a subset  $D$  of  $P_m \square C_{n_1+n_2}$  such that  $D \cap (P_m)_i = D_1 \cap (P_m)_i$ ,  $i = 0, 1, \dots, n_1 - 1$  and  $D \cap (P_m)_{n_1+i} = D_2 \cap (P_m)_i$ ,  $i = 0, 1, \dots, n_2 - 1$ , i.e., the  $i$ th  $(P_m)$ -layer of  $D$  is coming from the  $i$ th  $(P_m)$ -layer of  $D_1$  if  $0 \leq i \leq n_1 - 1$  and from the  $i - n_1$ th  $(P_m)$ -layer of  $D_2$  if  $n_1 \leq i \leq n_1 + n_2 - 1$ . The illustration is shown in the Figure 4.2.

One of the most challenging problems concerning the domination numbers of Cartesian products of graphs is the proof of the Vizing Conjecture, namely

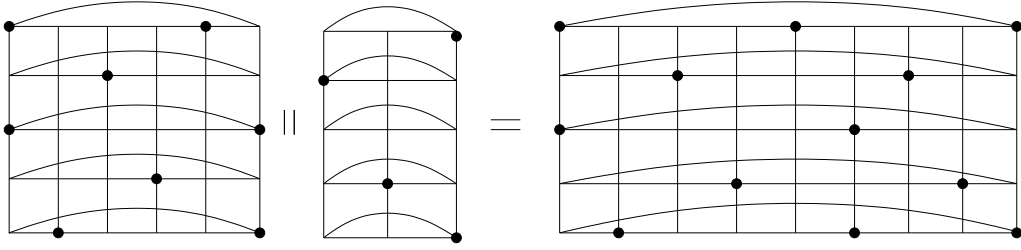


Figure 4.2: Modified concatenation of the dominating sets of  $P_5 \square C_6$  and  $P_5 \square C_3$  to get the dominating set for  $P_5 \square C_9$ .

$\gamma(G \square H) \geq \gamma(G) \cdot \gamma(H)$  [95]. Despite numerous results showing its validity in some special cases, the conjecture remains an open problem. Partial works have been made towards finding the domination numbers of some particular Cartesian product of graphs. This problem also seems to be a difficult one and the authors of [13] proved that even for subgraphs of  $P_m \square P_n$ , this problem is NP-complete. In [41], Jacobson and Kinch established the following results : For all  $n \geq 1$ ,

1.  $\gamma(P_2 \square P_n) = \lfloor \frac{n+2}{2} \rfloor$ .
2.  $\gamma(P_3 \square P_n) = \lfloor \frac{3n+4}{4} \rfloor$ .
3.  $\gamma(P_4 \square P_n) = \begin{cases} n+1, & \text{for } n = 1, 2, 3, 5, 6, 9, \\ n, & \text{otherwise.} \end{cases}$

In [10], Chang and Clark established the following results :

$$\gamma(P_5 \square P_n) = \begin{cases} \lfloor \frac{6n+6}{5} \rfloor, & \text{for } n = 2, 3, 7, \\ \lfloor \frac{6n+8}{5} \rfloor, & \text{otherwise.} \end{cases}$$

In [43], the authors established the following results regarding the Cartesian product of two cycles:

1. For  $n \geq 4$ ,  $\gamma(C_3 \square C_n) = n - \lfloor \frac{n}{4} \rfloor$ .
2. For  $n \geq 4$ ,  $\gamma(C_4 \square C_n) = n$ .
3. For  $n \geq 5$ ,  $\gamma(C_5 \square C_n) = \begin{cases} n, & n = 5k, k \geq 1, \\ n+2 & n = 5k+3, k \geq 1, \\ n+1, & \text{otherwise.} \end{cases}$

More works may be found in [23], [22], [26] and [34].

In this chapter, the topic of interest is to find the domination numbers of cylindrical grid graphs  $P_m \square C_n$ ,  $m \geq 2$ ,  $n \geq 3$ .

Towards finding the answer, some partial results in this direction are obtained. The domination numbers as well as minimum dominating sets of the graphs  $P_m \square C_n$ , for  $m = 2, 3, 4$  and  $n \geq 3$  are found. Bounds on  $\gamma(P_m \square C_n)$  for  $m = 5$  and  $n \geq 3$  are also proposed. As a brief summary, results that are proved in the subsequent sections are stated as follows.

For all  $n \geq 3$ ,

$$1. \gamma(P_2 \square C_n) = \begin{cases} \lceil \frac{n+1}{2} \rceil, & \text{when } n \text{ is not a multiple of } 4, \\ \frac{n}{2}, & \text{when } n \text{ is a multiple of } 4. \end{cases}$$

$$2. \gamma(P_3 \square C_n) = \lceil \frac{3n}{4} \rceil.$$

$$3. \gamma(P_4 \square C_n) = \begin{cases} n + 1, & \text{for } n = 3, 5, 9, \\ n, & \text{otherwise.} \end{cases}$$

$$4. \gamma(P_5 \square C_3) = 4, \gamma(P_5 \square C_4) = 5 \text{ and } \gamma(P_5 \square C_5) = 7.$$

$$\text{Moreover, for } n \geq 6, n + \lceil \frac{n}{5} \rceil \leq \gamma(P_5 \square C_n) \leq n + \lceil \frac{n}{4} \rceil.$$

Throughout the chapter, we use the arithmetic operations of the indices over modulo  $n$ .

## 4.2 Finding the domination numbers of some cylindrical grid graphs

In this section, the domination numbers as well as minimum dominating sets of particular cylindrical grid graphs of the form  $P_m \square C_n$ , for all  $n \geq 3$  and for  $m = 2, 3$  and  $4$  are found. To prove the result we need the following Lemmas.

**Lemma 4.1.** *Let  $m \geq 2$ . Then, there exists a minimum dominating set  $D$  of  $P_m \square C_n$  such that for every  $i \in V(C_n)$ ,  $|(P_m)_i \cap D| \leq m - 1$ .*

**Proof.** Let  $D'$  be one of the minimum dominating sets of  $P_m \square C_n$ . Suppose further that,  $|(P_m)_i \cap D'| = m$  holds for  $k$   $P_m$ -layers  $(P_m)_i$ , i.e., for  $k$ -many  $i$ 's in  $\{0, 1, \dots, n-1\}$ ,  $0 \leq k \leq n-1$ .

Let us assume that  $|(P_m)_i \cap D'| = m$  for some  $i \in \{0, 1, 2, \dots, n-1\}$ . In addition, if  $(P_m)_{i-1} \cap D' = (P_m)_{i+1} \cap D' = \phi$ , then we consider  $D = (D' \cup \{(0, i-1), (1, i+1)\}) \setminus \{(0, i), (1, i)\}$ .

Next, we assume that the layer  $(P_m)_{i+1}$  has nonempty intersection with  $D'$  and let  $(j, i+1) \in (P_m)_{i+1} \cap D'$ . Then, it is clear that  $(j, i-1)$  and at least one of  $(j+1, i-1)$  and  $(j-1, i-1) \notin (P_m)_{i-1} \cap D'$ , otherwise,  $D' \setminus \{(j, i)\}$  would be a dominating set, contradicting the minimality of  $D'$  and hence  $|(P_m)_{i-1} \cap D'| < m-1$ . Then, we consider  $D = (D' \cup \{(j, i-1)\}) \setminus \{(j, i)\}$ . Now,  $D$  is a minimum dominating set with  $k-1$  many  $P_m$ -layers having  $m$  vertices in common with  $D$ . Repeating this construction we get the result.  $\square$

**Lemma 4.2.** *Let  $n \geq 3$ . Then, there exists a minimum dominating set  $D$  of  $P_m \square C_n$  such that for every  $i \in V(P_m)$ ,  $|D \cap (C_n)_i| \leq n-1$ .*

**Proof.** Just interchanging the roles of  $P_m$  and  $C_n$  in the proof of Lemma 4.1, we have similar proof of this lemma.

**Lemma 4.3.** *There cannot be two consecutive  $P_m$ -layers having empty intersection with a minimum dominating set of  $P_m \square C_n$ , for  $m \geq 3$  and  $n \geq 4$ .*

**Proof.** If possible, let there be a minimum dominating set  $D$  having empty intersection with two consecutive layers  $(P_m)_i$  and  $(P_m)_{i+1}$  for some  $i \in \{0, 1, 2, \dots, n-1\}$ . Define  $(P_m)_{-1} = (P_m)_{n-1}$ ,  $(P_m)_{-2} = (P_m)_{n-2}$  and  $(P_m)_n = (P_m)_0$ ,  $(P_m)_{n+1} = (P_m)_1$ . Thus,  $|(P_m)_i \cap D| = |(P_m)_{i+1} \cap D| = 0$  gives  $|(P_m)_{i-1} \cap D| = |(P_m)_{i+2} \cap D| = m$ .

Now, consider the following cases:

**Case 1.** For  $n = 4, 5$ , we set  $D' = D \cup \{(0, i)\} \setminus \{(0, i-1), (1, i-1)\}$ .

**Case 2.** For  $n \geq 6$ , we set  $D' = (D \cup \{(0, i), (1, i-2), (1, i+3), (2, i)\}) \setminus \{(0, i-1), (0, i+2), (1, i-1), (2, i-1), (2, i+2)\}$ .

In each case we find that  $D'$  is a dominating set and  $|D'| < |D|$ , which is a contradiction. Therefore, there cannot have two consecutive layers  $(P_m)_i$ ,  $(P_m)_{i+1}$  having empty intersection with a minimum dominating set.  $\square$

**Lemma 4.4.** For every dominating set  $D$  of  $P_m \square C_n$ ,

$$x_{i-1} + 3x_i + x_{i+1} \geq n, \quad \forall i = 0, \dots, m-1,$$

where  $x_i = |(C_n)_i \cap D|$  for  $i = 0, 1, \dots, m-1$  and  $x_{-1} = x_m = 0$ .

Moreover, if  $x_{i-1} + 3x_i + x_{i+1} = n$ , then there does not exist any pair of vertices from  $((C_n)_{i-1} \cup (C_n)_i \cup (C_n)_{i+1}) \cap D$  such that they dominate a common vertex of  $(C_n)_i$ . Finally,  $\sum_{i=0}^{m-1} x_i = |D| \geq \gamma(P_m \square C_n)$ .

**Proof.** The result follows from the fact that any vertex of  $D$  from  $(C_n)_i$  dominates three vertices of  $(C_n)_i$  including itself and any vertex of  $D$  from  $(C_n)_{i-1}$  or  $(C_n)_{i+1}$  dominates one vertex of  $(C_n)_i$ .  $\square$

*Remark 4.5.* The similar result of Lemma 4.4 holds for  $y_i = |(P_m)_i \cap D|$ , where  $i \in \{0, 1, \dots, n-1\}$ .

Using the above Lemmas we now prove the following theorems.

**Theorem 4.6.** For  $n \geq 3$ ,  $\gamma(P_2 \square C_n) = \begin{cases} \lceil \frac{n+1}{2} \rceil, & \text{when } 4 \nmid n \\ \frac{n}{2}, & \text{when } 4 \mid n. \end{cases}$

**Proof.** We consider a set  $D$  with

$$D = \begin{cases} D_1, & \text{if } n \equiv 2 \pmod{4} \\ D_2, & \text{otherwise.} \end{cases}$$

Where,  $D_1 = \{(0, i) : i \equiv 2 \pmod{4}\} \cup \{(1, i) : i \equiv 0 \pmod{4}\} \cup \{(0, n-1)\}$  and  $D_2 = \{(0, i) : i \equiv 2 \pmod{4}\} \cup \{(1, i) : i \equiv 0 \pmod{4}\}$ . Then,  $D$  is a dominating set of  $P_2 \square C_n$  and that  $|D| = \lceil \frac{n+1}{2} \rceil$ , whenever  $n$  is not a multiple of 4 and  $|D| = \frac{n}{2}$ , whenever  $n$  is a multiple of 4. Hence,  $\gamma(P_2 \square C_n) \leq \lceil \frac{n+1}{2} \rceil$ , or  $\gamma(P_2 \square C_n) \leq n/2$  for different values of  $n$ .

To show that  $\gamma(P_2 \square C_n) \geq \lceil \frac{n}{2} \rceil$ , let  $D'$  be a minimum dominating set. Let  $|(C_n)_i \cap D'| = x_i$ , for  $i = 0, 1$ , i.e.,  $x_i$  is the number of vertices of  $D'$  from the layer  $(C_n)_i$ . Then, using Lemma 4.4 we have

$$3x_0 + x_1 \geq n \quad (i)$$

$$x_0 + 3x_1 \geq n \quad (ii)$$



$$x_0 + x_1 = \gamma(P_2 \square C_n) \quad (iii)$$

Therefore,  $4(x_0 + x_1) \geq 2n$  which gives  $x_0 + x_1 \geq \frac{n}{2}$ .

$$\text{Consequently, } \gamma(P_2 \square C_n) \geq \begin{cases} \frac{n+1}{2}, & \text{if } n \text{ is odd,} \\ \frac{n}{2}, & \text{if } n \text{ is even.} \end{cases}$$

Now, let  $n = 4k + 2$ ,  $k \geq 1$ . We will show that  $\gamma(P_2 \square C_n) \geq \frac{n}{2} + 1$ , i.e.,  $\gamma(P_2 \square C_n) \geq 2k + 2$ . If possible, let  $\gamma(P_2 \square C_n) \leq \frac{n}{2} = 2k + 1$ . Then,  $x_0 + x_1 \leq 2k + 1$ . Therefore, one of  $x_0$  or  $x_1$  must be less than or equals  $k$ . Without loss of generality, let  $x_0 \leq k$ . Then,  $3x_0 + x_1 \leq 3x_0 + (2k + 1 - x_0) = 2x_0 + 2k + 1 \leq 4k + 1 < n$ , which contradicts (i). Hence,  $\gamma(P_2 \square C_n) \geq \frac{n}{2} + 1$ .  $\square$

*Remark 4.7.* For  $n \geq 3$ ,  $\gamma(P_2 \square C_n) \geq \gamma(P_2) \cdot \gamma(C_n) = 1 \cdot \lceil \frac{n}{3} \rceil$ , the equality holds only when  $n = 4$ .

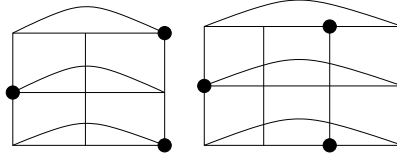


Figure 4.3: Vertices with bold circle form a minimum dominating set for  $P_3 \square C_3$  and  $P_3 \square C_4$ .

**Theorem 4.8.** For  $n \geq 3$ ,  $\gamma(P_3 \square C_n) = \lceil \frac{3n}{4} \rceil$ .

**Proof.** Consider a set  $D$  with

$$D = \begin{cases} D_1, & \text{if } n \equiv 2(\text{mod } 4) \\ D_2, & \text{otherwise.} \end{cases}$$

Where,  $D_1 = \{(0, i), (2, i) : i \equiv 2(\text{mod } 4)\} \cup \{(1, i) : i \equiv 0(\text{mod } 4)\} \cup \{(1, n-1)\}$  and  $D_2 = \{(0, i), (2, i) : i \equiv 2(\text{mod } 4)\} \cup \{(1, i) : i \equiv 0(\text{mod } 4)\}$ . Then,  $D$  is a dominating set of  $P_3 \square C_n$  and that  $|D| = \lceil \frac{3n}{4} \rceil$ . For  $n = 3$  and  $n = 4$ , it is illustrated in Figure 4.3. Therefore,  $\gamma(P_3 \square C_n) \leq \lceil \frac{3n}{4} \rceil$ .

Next, we show that  $\gamma(P_3 \square C_n) \geq \lceil \frac{3n}{4} \rceil$ . Let  $n = 4k + t$ , where  $k \geq 0$ ,  $3 \geq t \geq 0$  and let  $D$  be one of the minimum dominating sets satisfying the property as stated in Lemma 4.1. Let  $s$  be the number of  $P_3$ -layers which

have empty intersection with  $D$ . Then, by Lemma 4.3, since no two empty layers are adjacent, we have  $s \leq \lfloor \frac{n}{2} \rfloor = 2k + \lfloor \frac{t}{2} \rfloor$ . Now, as every empty  $P_3$ -layer is dominated by exactly two other  $P_3$ -layers, there are at least  $\lceil \frac{s}{2} \rceil$   $P_3$ -layers with precisely two vertices from  $D$ . Hence,  $|D| \geq 2 \lceil \frac{s}{2} \rceil + (n - \lceil \frac{s}{2} \rceil - s) = n - (s - \lceil \frac{s}{2} \rceil)$ . Also,  $(s - \lceil \frac{s}{2} \rceil)$  is maximum when  $s = 2k + \lfloor \frac{t}{2} \rfloor$ . So,  $|D| \geq n - (2k + \lfloor \frac{t}{2} \rfloor - \lceil \frac{2k + \lfloor \frac{t}{2} \rfloor}{2} \rceil) = (4k + t) - (2k + \lfloor \frac{t}{2} \rfloor - \lceil \frac{2k + \lfloor \frac{t}{2} \rfloor}{2} \rceil) = 3k + t - \lfloor \frac{t}{2} \rfloor + \lceil \frac{\lfloor \frac{t}{2} \rfloor}{2} \rceil = 3k + t$  (since  $0 \leq t \leq 3$ )  $= \lceil \frac{3n}{4} \rceil$ .  $\square$

*Remark 4.9.* For  $n \geq 3$ ,  $\gamma(P_3 \square C_n) > \gamma(P_3) \cdot \gamma(C_n) = 1 \cdot \lceil \frac{n}{3} \rceil$ .

**Theorem 4.10.**  $\gamma(P_4 \square C_n) = n + 1$ , for  $n = 3, 5, 9$ .

**Proof.** The proof of the theorem for three values of  $n$  are given successively.

**n=3.** Consider the set  $D_1 = \{(0, 2), (1, 0), (2, 2), (3, 1)\}$ . Then,  $D_1$  is a dominating set for  $P_4 \square C_3$  as shown in Figure 4.4. Since  $|D_1| = 4$ ,  $\gamma(P_4 \square C_3) \leq 4$ . Now, if possible, let there exist a dominating set  $D'_1$  such that  $|D'_1| \leq 3$ . Let  $|(C_3)_i \cap D'_1| = x_i$ , where  $x_i$  is the number of vertices of  $D'_1$  from the layer  $(C_3)_i$ , for  $i = 0, 1, 2, 3$ . Therefore,

$$\sum_{i=0}^3 x_i \leq 3 \quad (i)$$

$$3x_0 + x_1 \geq 3 \quad (ii)$$

$$x_0 + 3x_1 + x_2 \geq 3 \quad (iii)$$

$$x_1 + 3x_2 + x_3 \geq 3 \quad (iv)$$

$$x_2 + 3x_3 \geq 3 \quad (v)$$

Now, adding (ii) and (v) and subtracting (i) we get

$$x_0 + x_3 \geq 2 \quad (vi)$$

which gives

$$x_1 + x_2 \leq 1 \quad (vii)$$

Therefore, at least one of  $x_1$  and  $x_2$  must be zero. Without loss of generality, let  $x_1 = 0$ . Then, from (iii) we have

$$x_0 + x_2 \geq 3 \quad (viii)$$

and hence from (i)  $x_3 = 0$ . Therefore, from (v) we get  $x_2 = 3$  and from (i)  $x_0 = 0$  which contradicts (ii). So, we have  $|D'| \geq 4$  and therefore  $\gamma(P_4 \square C_3) = 4$ .

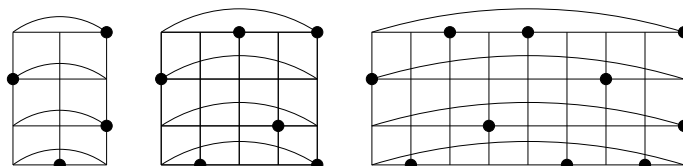


Figure 4.4: Vertices with bold circles form a minimum dominating set for  $P_4 \square C_3$ ,  $P_4 \square C_5$  and  $P_4 \square C_9$ .

**n=5.** Consider the set  $D_2 = \{(0, 2), (0, 4), (1, 0), (2, 3), (3, 1), (3, 4)\}$ . Then,  $D_2$  is a dominating set of  $P_4 \square C_5$  with  $|D_2| = 6$  as shown in Figure 4.4. Therefore,  $\gamma(P_4 \square C_5) \leq 6$ . We now show that,  $\gamma(P_4 \square C_5) \geq 6$ . If possible, let there exist a dominating set  $D'_2$  such that  $|D'_2| \leq 5$ . Let  $|(C_5)_i \cap D'_2| = x_i$ , where  $x_i$  is the number of vertices of  $D'_2$  from the layer  $(C_5)_i$  for  $i = 0, 1, 2, 3$ . Therefore,

$$\sum_{i=0}^5 x_i \leq 5 \tag{i}$$

Also, from Lemma 4.4 we have

$$3x_0 + x_1 \geq 5 \tag{ii}$$

$$x_0 + 3x_1 + x_2 \geq 5 \tag{iii}$$

$$x_1 + 3x_2 + x_3 \geq 5 \tag{iv}$$

$$x_2 + 3x_3 \geq 5 \tag{v}$$

Now, adding (ii) and (v) and subtracting (i) we get

$$x_0 + x_3 \geq 3 \tag{vi}$$

which implies

$$x_1 + x_2 \leq 2 \tag{vii}$$

Now, we claim that  $x_1 \neq 0$ . Otherwise, if  $x_1 = 0$ , then from (iii) we get  $x_0 + x_2 \geq 5$ . Therefore,  $x_0 + x_2 = 5$  and from (i) we have  $x_3 = 0$ . Therefore, from (v) we get  $x_2 \geq 5$ , i.e.,  $x_2 = 5$ . This gives  $x_0 = 0$ , which contradicts (ii). Therefore, we have  $x_1 \neq 0$ . Similarly,  $x_2 \neq 0$ . So, we have from (vii),  $x_1 = x_2 = 1$ . For these values of  $x_1$  and  $x_2$  we get from (v),  $x_3 \geq 2$  and from (ii),  $x_0 \geq 2$ , contradicting (i). Therefore, we cannot have  $|D'_2| \leq 5$ . So,  $|D'_2| \geq 6$  and hence  $\gamma(P_4 \square C_5) = 6$ .

**n=9.** Consider the set  $D_3 = \{(0, 2), (0, 4), (0, 8), (1, 0), (1, 6), (2, 3), (2, 8), (3, 1), (3, 5), (3, 7)\}$ . Then  $D_3$  is a dominating set of  $P_4 \square C_9$  with  $|D_3| = 10$  as shown in Figure 4.4. Therefore,  $\gamma(P_4 \square C_9) \leq 10$ . Now, if possible, let there exists a dominating set  $D'_3$  such that  $|D'_3| \leq 9$  and let  $|(C_9)_i \cap D'_3| = x_i$ , where  $x_i$  is the number of vertices of  $D'_3$  from the layer  $(C_9)_i$  for  $i = 0, 1, 2, 3$ . Therefore,

$$\sum_{i=0}^9 x_i \leq 9 \quad (i)$$

Also, from Lemma 4.4 we have

$$3x_0 + x_1 \geq 9 \quad (ii)$$

$$x_0 + 3x_1 + x_2 \geq 9 \quad (iii)$$

$$x_1 + 3x_2 + x_3 \geq 9 \quad (iv)$$

$$x_2 + 3x_3 \geq 9 \quad (v)$$

Now, adding (ii) and (v) and subtracting (i) we get

$$x_0 + x_3 \geq 5 \quad (vi)$$

which gives

$$x_1 + x_2 \leq 4 \quad (vii)$$

Now, we claim that  $x_1 \neq 0$ . If  $x_1 = 0$  then from (iii) we have  $x_0 + x_2 \geq 9$  and hence from (i),  $x_3 = 0$ . Therefore, from (v),  $x_2 = 9$  and from (i),  $x_0 = 0$  which contradicts (ii). Therefore,  $x_1 \neq 0$ . Similarly, we can show that  $x_2 \neq 0$ . Again, we claim that, either  $x_1 \neq 2$  or  $x_2 \neq 2$ . If possible, let  $x_1 = x_2 = 2$ . Then, from (ii) and (v), we get  $x_0 \geq 3$  and  $x_3 \geq 3$ , contradicting (i). Therefore, either  $x_1 \neq 2$  or  $x_2 \neq 2$ .

Now, from (vii), we get one of  $x_1$  or  $x_2$  must be 1. Without loss of generality, let  $x_1 = 1$ . Therefore, from (iii) we get  $x_0 + x_2 \geq 6$  and from (i) we get  $x_1 + x_3 \leq 3$ . This gives  $x_3 \leq 2$ . We now claim that  $x_3 \neq 0$ . If possible, let  $x_3 = 0$ , then from (v) we get  $x_2 \geq 9$  which contradicts (i). So,  $x_3 \neq 0$ . Next, we show that  $x_3 \neq 1$ . If possible, let  $x_3 = 1$ . Then, from (v) we get  $x_2 \geq 6$ . Therefore, from (i),  $x_0 \leq 1$  which contradicts (ii). Consequently,  $x_3 \neq 1$  and hence  $x_3 = 2$ . This implies  $x_0 + x_2 = 6$ . Now, from (v),  $x_2 \geq 3$  and from (ii),  $x_0 \geq 3$ . Therefore,  $x_0 = 3$ ,  $x_1 = 1$ ,  $x_2 = 3$ , and  $x_3 = 2$ .

Lastly, we show that  $x_0 = 3$ ,  $x_1 = 1$ ,  $x_2 = 3$ , and  $x_3 = 2$  cannot give a dominating set  $D'_3$  with  $|D'_3| = 9$ . Without loss of generality, let us assume  $(1, 0) \in D'_3$ .

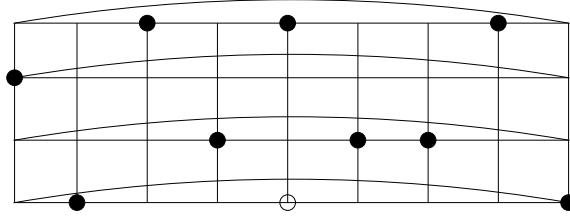


Figure 4.5: Figure showing non admissibility of  $\gamma(P_4 \square C_9) = 9$ .

Then,  $(1, i) \notin D'_3$  for all  $i = 1, 2, \dots, 8$ . Again,  $(0, 1) \notin D'_3$  because if  $(0, 1) \in D'_3$  then  $(1, 1)$  is dominated by  $(0, 1)$  and  $(1, 0)$  which will be a contradiction by the Lemma 4.4. Similarly,  $(0, 0), (0, 8) \notin D'_3$ . Therefore, to dominate  $(0, 1)$  and  $(0, 8)$  we must take  $(0, 2), (0, 7) \in D'_3$ . Till now, the vertices  $(0, 4)$  and  $(0, 5)$  in the layer  $(C_9)_0$  are not dominated by any vertices of  $D'_3$ . Therefore, either  $(0, 4) \in D'_3$  or  $(0, 5) \in D'_3$ . Without loss of generality, let  $(0, 4) \in D'_3$ . Again, the vertices  $(1, 3), (1, 5), (1, 6)$  in the layer  $(C_9)_1$  are not dominated by any vertices of  $D'_3$ . To dominate these vertices we must take the vertices  $(2, 3), (2, 5), (2, 6)$  in  $D'_3$ . In a similar way, to dominate the vertices  $(2, 1), (2, 8)$  of the layer  $(C_9)_2$  we must take the vertices  $(3, 1), (3, 8)$  in  $D'_3$ . Still  $(3, 4)$  is not dominated by any vertex of  $D'_3$  as shown in Figure 4.5. Hence,  $D'_3$  can not be a dominating set. This contradiction shows that  $\gamma(P_4 \square C_9) \geq 10$ .  $\square$

**Theorem 4.11.** For  $n \geq 3$ ,  $\gamma(P_4 \square C_n) = n$ , for  $n \neq 3, 5, 9$ .

**Proof.** The three diagrams (i), (ii) and (iii) shown in Figure 4.6 give dominating sets for  $P_4 \square C_4$ ,  $P_4 \square C_6$  and  $P_4 \square C_7$ , respectively. Note that,

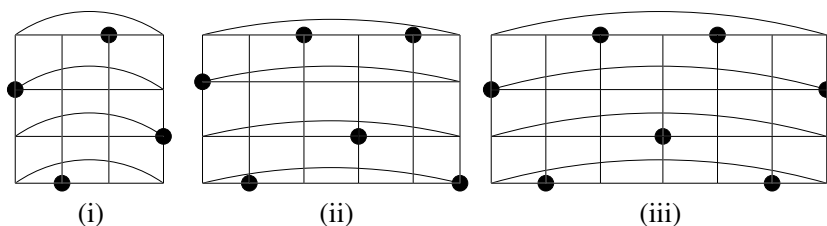


Figure 4.6: Vertices with bold circle form a minimum dominating set for  $P_4 \square C_4$ ,  $P_4 \square C_6$  and  $P_4 \square C_7$ .

a dominating set for  $P_4 \square C_8$  can be obtained by the modified concatenation of the two copies of the diagram (i). A dominating set for  $P_4 \square C_{10}$  can be constructed by the modified concatenation of the diagrams (i) and (ii), a dominating set for  $P_4 \square C_{11}$  can be constructed by the modified concatenation of the diagrams (i) and (iii), a dominating set for  $P_4 \square C_{12}$  can be constructed by the modified concatenation of the two copies of the diagram (ii) and a dominating set for  $P_4 \square C_{13}$  can be constructed by modified concatenation of the diagrams (ii) and (iii). Finally, the dominating set for  $n \geq 14$  can be constructed by modified concatenation of the diagram for  $n = 4$  with the diagrams for  $n = 10, 11, 12, 13$ . Hence, we have  $\gamma(P_4 \square C_n) \leq n$ .

It now remains to show that  $\gamma(P_4 \square C_n) \geq n$ . To prove this, consider a minimum dominating set  $D$  which has no empty  $(P_4)_i$  layer. Then,  $|D| \geq n$  and hence the theorem follows. Now, consider a minimum dominating set  $D'$  with  $s$  empty  $(P_4)_i$  layers. Further, let there be  $t$  nonempty  $(P_4)_i$  layers each adjacent to exactly one empty  $(P_4)_i$  layer. Then, there are  $k = \frac{2s-t}{2}$  nonempty  $(P_4)_i$  layers each adjacent to two empty layers. Since there do not exist two consecutive empty  $(P_4)_i$  layers, the number of nonempty  $(P_4)_i$  layers which have no adjacent empty layer is  $p = n - t - \left(\frac{2s-t}{2}\right) - s = n - 2s - \frac{t}{2}$ .

Now, let  $x'_1, x'_2, \dots, x'_t$  be the numbers of elements of  $D'$  which are contained in  $t$  nonempty  $(P_4)_i$  layers, respectively such that each of these  $(P_4)_i$  layers is adjacent with exactly one empty  $(P_4)_i$  layer. Further, we define  $x''_1, x''_2, \dots, x''_k$  and  $x^0_1, x^0_2, \dots, x^0_p$  in an analogous way. Now, by Remark 4.5,  $2(x''_1 + x''_2 + \dots + x''_k) + x'_1 + x'_2 + \dots + x'_t \geq 4s$ .

Therefore,  $2(x''_1 + x''_2 + \dots + x''_k + x'_1 + x'_2 + \dots + x'_t) \geq 4s + t$ .

This implies  $2(x^0_1 + x^0_2 + \dots + x^0_p + x''_1 + x''_2 + \dots + x''_k + x'_1 + x'_2 + \dots + x'_t) \geq 4s + t + 2p = 4s + t + 2n - 4s - t = 2n$ . Consequently,  $x_1 + x_2 + \dots + x_n \geq n$

where  $x_i = |(P_4)_i \cap D'|$ . □

*Remark 4.12.* For  $n \geq 3$ ,  $\gamma(P_4 \square C_n) > \gamma(P_4) \cdot \gamma(C_n) = 2 \cdot \lceil \frac{n}{3} \rceil$ .

### 4.3 Bounds on the domination numbers of $P_5 \square C_n$ , $n \geq 3$

In this section, the exact domination numbers are given for  $P_5 \square C_3$ ,  $P_5 \square C_4$  and  $P_5 \square C_5$ . Furthermore, bounds on the domination numbers are also proposed for  $P_5 \square C_n$ ,  $n \geq 6$ . The following Lemmas are essential for proving the subsequent theorems.

**Lemma 4.13.** *For  $n \geq 3$ , there exists a minimum dominating set  $D$  of  $P_5 \square C_n$  such that for every  $i \in V(C_n)$ ,  $|(P_5)_i \cap D| \leq 3$ .*

**Proof.** Let  $D$  be a minimum dominating set of  $P_5 \square C_n$  such that for every  $i \in V(C_n)$ ,  $|(P_5)_i \cap D| \leq 4$ . Such a  $D$  exists by Lemma 4.1. Suppose further that  $|(P_5)_i \cap D| = 4$  holds for  $k$  many  $P_5$ -layers. We now construct a dominating set  $D'$  with  $|D'| = |D|$  such that only  $k - 1$  many  $P_5$ -layers have 4 vertices in common with  $D$ .

Assume that  $|(P_5)_i \cap D| = 4$  for some  $i \in V(C_n)$ . We claim that  $|(P_5)_{i-1} \cap D| \leq 2$ , and  $|(P_5)_{i+1} \cap D| \leq 2$ . If possible, let  $|(P_5)_{i+1} \cap D| \geq 3$ . Then,  $D'' = (D \setminus ((P_5)_i \cup (P_5)_{i+1})) \cup \{(2, i-1), (0, i), (4, i), (1, i+1), (0, i+2), (3, i+2)\}$  is a dominating set with  $|D''| < |D|$  which contradicts the minimality of  $D$ . Therefore,  $|(P_5)_{i+1} \cap D| \leq 2$ . Similarly,  $|(P_5)_{i-1} \cap D| \leq 2$ . Now, we set  $D' = (D \setminus (P_5)_i) \cup \{(2, i-1), (0, i), (4, i), (2, i+1)\}$ . Then,  $D'$  is a dominating set with  $|D'| \leq |D|$  but since  $D$  is a minimum dominating set we have  $|D'| = |D|$ . Also,  $|(P_5)_j \cap D'| < 4$  for  $j = i-1, i, i+1$  and  $|(P_5)_j \cap D'| = |(P_5)_j \cap D|$  for other values of  $j$ . Repeating this process we get the desired minimum dominating set. □

**Lemma 4.14.** *For  $n \geq 5$ , there exists a minimum dominating set  $D$  of  $P_5 \square C_n$  such that for every  $i \in V(C_n)$  either (a)  $|(P_5)_i \cap D| \leq 2$  or (b)  $|(P_5)_i \cap D| = 3$  with  $(P_5)_{i-1} \cap D = \phi$  and  $(P_5)_{i+1} \cap D = \phi$  for  $i \in V'$  and  $|(P_5)_i \cap D| \leq 2$  for all  $i \notin V'$ , for some  $V' \subseteq V(C_n)$ .*

**Proof.** Lemma 4.13 shows there exists a minimum dominating set  $D$  such that  $|(P_5)_i \cap D| \leq 3$  for all  $i \in V(C_n)$ .

Now, let  $|(P_5)_i \cap D| = 3$  hold for  $k$  many  $P_5$ -layers and among these  $k$  many  $P_5$ -layers there exist  $k'$  many  $P_5$ -layers with the property that both the adjacent  $P_5$ -layers of such a layer have empty intersection with  $D$ .

Now, if  $k' = k$  then  $D$  has the desired property. When  $k' < k$  we construct a dominating set  $D'$  with the property that  $D'$  has  $k-1$  or less  $P_5$ -layers having 3 vertices in common with  $D$  and  $|D'| = |D|$ . Let  $|(P_5)_i \cap D| = 3$  for some  $i \in V(C_n)$ . We claim that, both of  $|(P_5)_{i-1} \cap D|$  and  $|(P_5)_{i+1} \cap D|$  cannot be simultaneously equal to 2.

On contrary, if possible, let  $|(P_5)_{i-1} \cap D| = |(P_5)_{i+1} \cap D| = 2$ . Now, we consider the following sets

$$\begin{aligned} A_1 &= \{(1, i-2), (3, i-1), (4, i-1), (0, i), (2, i+1), (4, i+2)\}, \\ A_2 &= \{(3, i-2), (0, i-1), (1, i-1), (4, i), (2, i+1), (0, i+2)\}, \\ A_3 &= \{(2, i-2), (4, i-2), (0, i-1), (3, i), (1, i+1), (4, i+2)\}, \\ A_4 &= \{(2, i-2), (0, i-2), (4, i-1), (1, i), (3, i+1), (0, i+2)\} \\ \text{and } A_5 &= \{(2, i-2), (0, i-1), (4, i-1), (2, i), (0, i+1), (4, i+1)\}. \end{aligned}$$

Let  $B_{kl} = \{(k, i+1), (l, i+1)\}$ , where  $k, l = 0, 1, 2, 3, 4$ .

Now, if  $(P_5)_{i+1} \cap D = B_{01}$  or  $B_{02}$  or  $B_{12}$  then let  $D' = (D \setminus \{(P_5)_{i-1} \cup (P_5)_i \cup (P_5)_{i+1}\}) \cup A_2$ ,

if  $(P_5)_{i+1} \cap D = B_{03}$  or  $B_{13}$  then let  $D' = (D \setminus \{(P_5)_{i-1} \cup (P_5)_i \cup (P_5)_{i+1}\}) \cup A_4$ ,

if  $(P_5)_{i+1} \cap D = B_{04}$  then let  $D' = (D \setminus \{(P_5)_{i-1} \cup (P_5)_i \cup (P_5)_{i+1}\}) \cup A_5$ ,

if  $(P_5)_{i+1} \cap D = B_{14}$  then let  $D' = (D \setminus \{(P_5)_{i-1} \cup (P_5)_i \cup (P_5)_{i+1}\}) \cup A_3$ ,

and if  $(P_5)_{i+1} \cap D = B_{23}$  or  $B_{24}$  or  $B_{34}$  then let  $D' = (D \setminus \{(P_5)_{i-1} \cup (P_5)_i \cup (P_5)_{i+1}\}) \cup A_1$ .

Then, in every case  $D'$  will be a dominating set with  $|D'| < |D|$ , which is a contradiction. Hence, the claim follows.

Now, we consider the all other possible cases:

**Case 1.** Let  $|(P_5)_{i+1} \cap D| = 3$ . Then, we claim that  $|(P_5)_{i+2} \cap D| < 2$ . To show this if possible, let  $|((P_5)_i \cup (P_5)_{i+1} \cup (P_5)_{i+2}) \cap D| \geq 8$ . Then,  $D'' = (D \setminus \{(P_5)_i \cup (P_5)_{i+1} \cup (P_5)_{i+2}\}) \cup \{(0, i-1), (4, i-1), (2, i), (0, i+1), (3, i+2), (4, i+2), (1, i+3)\}$  is a dominating set with  $|D''| < |D|$ ,



contradicting the minimality of  $D$ . Therefore,  $|(P_5)_{i+2} \cap D| < 2$ . Similarly,  $|(P_5)_{i-1} \cap D| < 2$ .

Now, we construct  $D' = (D \setminus \{(P_5)_i \cup (P_5)_{i+1}\}) \cup \{(2, i-1), (0, i), (4, i), (0, i+1), (4, i+1), (2, i+2)\}$ . Then  $|D'| \leq |D|$  and  $D'$  is a dominating set. Hence,  $D'$  is a minimum dominating set having  $(k-2)$  many  $P_5$ -layers, each of which has three vertices in common with  $D$ .

Similar argument for  $|(P_5)_{i-1} \cap D| = 3$ .

**Case 2.** Let  $|(P_5)_{i-1} \cap D| = 2$  and  $|(P_5)_{i+1} \cap D| \leq 1$ . Then, we can construct  $D'$  such that  $|(P_5)_{i-1} \cap D'| = 2$ ,  $|(P_5)_i \cap D'| = 2$  and  $|(P_5)_{i+1} \cap D'| = |(P_5)_{i+1} \cap D| + 1$ . Similarly, we can construct  $D'$  for  $|(P_5)_{i-1} \cap D| \leq 1$  and  $|(P_5)_{i+1} \cap D| = 2$ .

**Case 3.** Let  $|(P_5)_{i-1} \cap D| = 1$  and  $|(P_5)_{i+1} \cap D| \leq 1$ .

**subcase 3.1.** When  $|(P_5)_i \cap D| = \{(0, i), (1, i), (2, i)\}$  or  $\{(2, i), (3, i), (4, i)\}$ , then construct  $D' = (D \setminus \{(0, i), (1, i)\}) \cup \{(0, i-1), (1, i+1)\}$  for the first one. Similarly, we can construct  $D'$  for the other one.

**subcase 3.2.** Other than the above case we can construct  $D'$  such that  $|(P_5)_{i-1} \cap D'| = 1$ ,  $|(P_5)_i \cap D'| = 2$ ,  $|(P_5)_{i+1} \cap D'| = |(P_5)_{i+1} \cap D| + 1$ .

Similar construction will be made when  $|(P_5)_{i+1} \cap D| = 1$  and  $|(P_5)_{i-1} \cap D| \leq 1$ . Then,  $D'$  is a minimum dominating set having  $k-1$  many or less  $P_5$ -layers each of which has three vertices in common with  $D$ .

Repeating this replacement, we will get the minimum dominating set with the desired property as stated in this Lemma.  $\square$

**Lemma 4.15.** *For  $n \geq 5$ , there cannot be a dominating set  $D$  with five consecutive  $P_5$ -layers having exactly one vertex in common with  $D$ .*

**Proof.** On contrary, let there exist a dominating set  $D$  and there exists  $i \in V(C_n)$  such that  $|(P_5)_j \cap D| = 1$  for all  $j = i, i+1, i+2, i+3, i+4$ . Note that,  $(0, j) \notin D$  and  $(4, j) \notin D$  for all  $j = i+1, i+2, i+3$ . Therefore,  $(1, i+2) \in D$  and  $(3, i+2) \in D$ , a contradiction.  $\square$

**Lemma 4.16.** *Let  $D$  be one of the minimum dominating sets with the property as stated in Lemma 4.14. Again, let  $(P_5)_i$  and  $(P_5)_j$  be two layers having two vertices in common with  $D$  and  $|(P_5)_{i+1} \cap D| \neq 2$ ,  $|(P_5)_{i+2} \cap D| \neq 2, \dots, |(P_5)_{j-1} \cap D| \neq 2$ . Then either (a)  $|(P_5)_{i+1} \cap D| = 0$ ,  $|(P_5)_{i+2} \cap D| = 3$ ,  $|(P_5)_{i+3} \cap D| = 0$ ,  $|(P_5)_{i+4} \cap D| = 3, \dots, |(P_5)_{j-1} \cap D| = 0$ , or (b)  $j-i \leq 5$  and  $|(P_5)_l \cap D| = 1$  for all  $l = (i+1), (i+2), \dots, (j-1)$ .*

**Proof.** If  $|(P_5)_l \cap D| = 0$ , then  $|(P_5)_{l-1} \cap D| > 1$ . Otherwise,  $|(P_5)_{l+1} \cap D| \geq 4$ . Hence,  $|(P_5)_{l-1} \cap D| = 2$  or  $3$ . Similarly,  $|(P_5)_{l+1} \cap D| = 2$  or  $3$ . Therefore, using Lemma 4.14 and Lemma 4.15 we get the desired result.

**Theorem 4.17.** For  $n \geq 6$ ,  $\gamma(P_5 \square C_n) \geq n + \lceil \frac{n}{5} \rceil$ .

**Proof.** Let  $D$  be one of the minimum dominating sets with the property as stated in Lemma 4.14. Let us call the collection of  $P_5$ -layers  $\{(P_5)_k : k = i + 1, i + 2, \dots, j - 1\}$  as a block where  $i$  and  $j$  as in the above Lemma 4.16. Let  $x_0, x_1, x_3$  be the number of  $P_5$ -layers having 0, 1, 3 vertices, respectively in common with  $D$ . Let  $x'_2$  be the number of blocks in which every  $P_5$ -layer has exactly one vertex in common with  $D$  and  $x''_2$  be the number of blocks in which every  $P_5$ -layer has either 0 or 3 vertices in common with  $D$  and  $x_2^0$  be the number of blocks where  $j = i + 1$ , i.e., when the block contains no  $P_5$ -layer. Then, clearly,  $x_2 = x_2^0 + x'_2 + x''_2$  be the number of  $P_5$ -layers having 2 vertices in common with  $D$  and we have therefore  $x_3 \geq x''_2$ ,  $x_0 = x''_2 + x_3$ ,  $x_1 \leq 4x'_2$  and  $x_0 + x_1 + x_2^0 + x'_2 + x''_2 + x_3 = n$ . Now,  $|D| - \frac{6n}{5} = 3x_3 + 2x_2 + x_1 - \frac{6n}{5} = 3x_3 + 2x_2^0 + 2x'_2 + 2x''_2 + x_1 - \frac{6(x_0 + x_1 + x_2^0 + x'_2 + x''_2 + x_3)}{5} = \frac{1}{5}(9x_3 + 4x_2^0 + 4x'_2 + 4x''_2 - x_1 - 6x_0) = \frac{1}{5}(9x_3 + 4x_2^0 + 4x'_2 + 4x''_2 - x_1 - 6x''_2 - 6x_3) = \frac{1}{5}(3x_3 - 2x''_2 + 4x_2^0 + 4x'_2 - x_1) = \frac{1}{5}\{x_3 + 2(x_3 - x''_2) + (4x'_2 - x_1) + 4x_2^0\} \geq 0$ . Therefore, we have  $|D| \geq \frac{6n}{5}$ . Thus,  $\gamma(P_5 \square C_n) \geq n + \lceil \frac{n}{5} \rceil$ .

*Remark 4.18.*  $|D| = \frac{6n}{5} + \frac{x_3 + 2(x_3 - x''_2) + (4x'_2 - x_1) + 4x_2^0}{5}$ .

**Theorem 4.19.** (a)  $\gamma(P_5 \square C_3) = 4$ , (b)  $\gamma(P_5 \square C_4) = 5$ , and (c)  $\gamma(P_5 \square C_5) = 7$ .

**Proof.** (a) Let  $D$  be a minimum dominating set of  $P_5 \square C_3$ . Further, let  $|(C_n)_i \cap D| = x_i$ , for  $i = 0, 1, 2, 3, 4$ . Then, by Lemma 4.4 we have

$$3x_0 + x_1 \geq 3 \quad (i)$$

$$x_0 + 3x_1 + x_2 \geq 3 \quad (ii)$$

$$x_1 + 3x_2 + x_3 \geq 3 \quad (iii)$$

$$x_2 + 3x_3 + x_4 \geq 3 \quad (iv)$$

$$x_3 + 3x_4 \geq 3 \quad (v)$$

If possible, let

$$|D| = x_0 + x_1 + x_2 + x_3 + x_4 \leq 3 \quad (vi)$$

Now,  $x_0 \neq 0$ . Otherwise,  $x_1 = 3$  would imply  $x_2 = x_3 = x_4 = 0$ , contradicting (iv). Similarly,  $x_4 \neq 0$ . Next, if possible, let  $x_0 = 2$  and  $x_4 = 1$ . Then, we have  $x_1 = x_2 = x_3 = 0$ , contradicting (ii). Again, taking  $x_0 = 1$  and  $x_4 = 2$  we arrived at the same contradiction. Now, if possible, let  $x_0 = x_4 = 1$ . Adding (ii), (iii) and (iv) we get  $x_0 + 4x_1 + 5x_2 + 4x_3 + x_4 \geq 9$ . Therefore,  $4x_1 + 5x_2 + 4x_3 \geq 7$ . This implies  $5x_1 + 5x_2 + 5x_3 \geq 7$  and hence  $x_1 + x_2 + x_3 \geq 2$ , contradicting (vi). Consequently,  $|D| \geq 4$  and therefore  $\gamma(P_5 \square C_3) \geq 4$ . Now,  $D = \{(0, 1), (2, 0), (2, 2), (4, 1)\}$  is a dominating set of  $P_5 \square C_3$  as shown in Figure 4.7. Therefore,  $\gamma(P_5 \square C_3) \leq 4$  and hence  $\gamma(P_5 \square C_3) = 4$ . This completes the proof of (a).

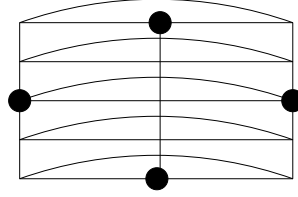


Figure 4.7: Vertices with bold circle form a minimum dominating set for  $P_5 \square C_3$ .

(b) Let  $D$  be a minimum dominating set of  $\gamma(P_5 \square C_4)$ . Further, let  $|(C_n)_i \cap D| = x_i$  for  $i = 0, 1, 2, 3, 4$ . Then, by Lemma 4.4,

$$3x_0 + x_1 \geq 4 \quad (i)$$

$$x_0 + 3x_1 + x_2 \geq 4 \quad (ii)$$

$$x_1 + 3x_2 + x_3 \geq 4 \quad (iii)$$

$$x_2 + 3x_3 + x_4 \geq 4 \quad (iv)$$

$$x_3 + 3x_4 \geq 4 \quad (v)$$

If possible, let

$$|D| = x_0 + x_1 + x_2 + x_3 + x_4 \leq 4 \quad (vi)$$

Then, as in the proof of (a) we have  $x_0 \neq 0$  and  $x_4 \neq 0$ . From (i) and (v) we get  $3x_0 + x_1 + x_3 + 3x_4 \geq 8$ . Therefore,  $2x_0 + 2x_4 \geq 4 + x_2$ . Now, if

$x_2 \neq 0$  then  $x_0 + x_4 \geq 3$ . Hence,  $x_1 + x_2 + x_3 \leq 1$  which implies  $x_2 = 1$  and  $x_1 = x_3 = 0$ . Therefore, from (ii) and (iv)  $x_0 + x_2 \geq 4$  and  $x_2 + x_4 \geq 4$ . Hence,  $x_0 \geq 3$  and  $x_4 \geq 3$ , contradicting (vi). If  $x_2 = 0$  then from (iii),  $x_1 + x_3 \geq 4$ . Now, from (vi) we have  $x_0 = x_4 = 0$ . From (i) we have  $x_1 \geq 4$  and hence  $x_3 = 0$ , contradicting (iv). Therefore, we have  $\gamma(P_5 \square C_4) \geq 5$ . Now,  $D = \{(0, 2), (1, 0), (2, 3), (3, 1), (4, 3)\}$  is a dominating set of  $P_5 \square C_4$  as shown in Figure 4.8. Therefore,  $\gamma(P_5 \square C_4) \leq 5$  and hence  $\gamma(P_5 \square C_4) = 5$ .  $\square$

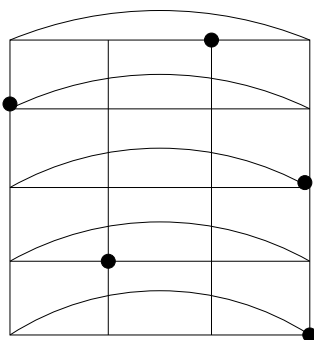


Figure 4.8: Vertices with bold circle form a minimum dominating set for  $P_5 \square C_4$ .

(c) Let  $D$  be a minimum dominating set of  $\gamma(P_5 \square C_5)$ . Further, let  $|(C_n)_i \cap D| = x_i$  where  $i = 0, 1, 2, 3, 4$ . Then, from Lemma 4.4 we have

$$3x_0 + x_1 \geq 5 \quad (i)$$

$$x_0 + 3x_1 + x_2 \geq 5 \quad (ii)$$

$$x_1 + 3x_2 + x_3 \geq 5 \quad (iii)$$

$$x_2 + 3x_3 + x_4 \geq 5 \quad (iv)$$

$$x_3 + 3x_4 \geq 5 \quad (v)$$

If possible, let

$$|D| = x_0 + x_1 + x_2 + x_3 + x_4 \leq 6 \quad (vi)$$

From (i) and (v) we get  $3x_0 + x_1 + x_3 + 3x_4 \geq 10$ . Therefore,  $2(x_0 + x_4) \geq 4 + x_2$ . Now, if possible, let  $x_2 = 0$ . Then,  $x_0 + x_4 \geq 2$  and  $x_1 + x_3 \geq 5$  (by (iii)), contradicting (vi). Therefore,  $x_2 \geq 1$  and hence  $x_0 + x_4 \geq 3$ .

Again,  $x_0 \neq 0$  and  $x_4 \neq 0$  as in (a). If possible, let  $x_0 + x_4 \geq 4$ . Then,  $x_1 + x_3 \leq 1$  implies either  $x_1 = 0$  or  $x_3 = 0$ . But this is a contradiction (since

$x_1 = 0$  implies  $x_0 + x_2 \geq 5$  implies  $x_0 = 4$  implies  $x_2 = 1$  implies  $x_3 \geq 2$ , contradicting (vi). Similar contradiction for  $x_3 = 0$ . Therefore,  $x_0 + x_4 = 3$ ,  $x_2 \geq 1$  and  $x_1 + x_3 \leq 2$ . Again, if  $x_1 = x_3 = 0$  then the only possibility is  $x_0 = 1$ ,  $x_1 = 0$ ,  $x_2 = 4$ ,  $x_3 = 0$  and  $x_4 = 1$  which contradicts  $x_0 + x_4 = 3$ . If  $x_1 = 1$ ,  $x_2 = 1$ ,  $x_3 = 1$  and  $x_0 + x_4 = 3$  then it will contradict either (i) or (v). Finally, if  $x_1 = 1$ ,  $x_3 = 0$  then  $x_2 = 2$  and  $x_0 + x_4 = 3$ , contradicting (i) or (v). Similar contradiction for  $x_1 = 0$ ,  $x_3 = 1$ . Hence, we have  $\gamma(P_5 \square C_5) \geq 7$ . Now,  $D = \{(0, 0), (0, 4), (1, 2), (2, 0), (3, 3), (3, 4), (4, 1)\}$  is a dominating set of  $P_5 \square C_5$  as shown in Figure 4.9. Therefore,  $\gamma(P_5 \square C_5) \leq 7$  and hence  $\gamma(P_5 \square C_5) = 7$ .  $\square$

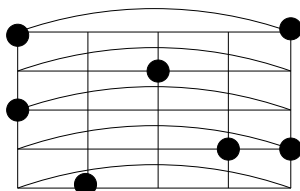


Figure 4.9: Vertices with bold circle form a minimum dominating set for  $P_5 \square C_5$ .

**Lemma 4.20.**  $9 + \lceil \frac{9}{5} \rceil \leq \gamma(P_5 \square C_9) \leq 9 + \lceil \frac{9}{4} \rceil$ .

**Proof.** The result follows from the Theorem 4.17 and the Figure 4.10.

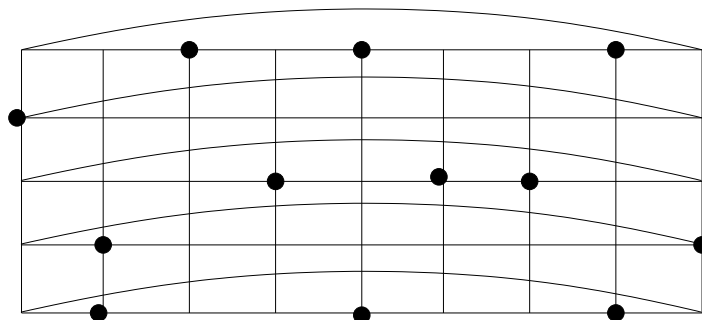


Figure 4.10: Vertices with bold circle form a dominating set for  $P_5 \square C_9$ .

**Theorem 4.21.** For  $n \geq 6$ ,  $n + \lceil \frac{n}{5} \rceil \leq \gamma(P_5 \square C_n) \leq n + \lceil \frac{n}{4} \rceil$ .

**Proof.** For  $n = 9$ , the result follows from Lemma 4.20. For  $n \geq 6$ , the inequality in the left side is already being proved in Theorem 4.17. For the other part of the inequality, let us consider the minimum dominating sets for  $P_4 \square C_6$  and

$P_4 \square C_7$  as shown in Figure 4.6. Now adding the vertices  $\{(4, 3), (4, 5)\}$  and  $\{(4, 3), (4, 6)\}$  to the dominating sets of  $P_4 \square C_6$  and  $P_4 \square C_7$ , respectively, the dominating sets for  $P_5 \square C_6$  and  $P_5 \square C_7$  are obtained (see Figure 4.11).

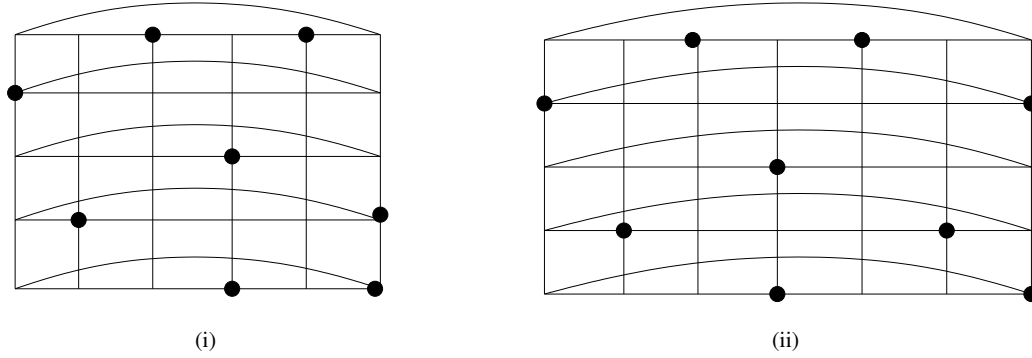


Figure 4.11: Vertices with bold circle form a dominating set for  $P_5 \square C_6$  and  $P_5 \square C_7$ .

Finally, using the modified concatenation suitably as in the case of  $P_4 \square C_n$ ,  $n \geq 3$ ,  $n \neq 3, 5, 9$ , (Theorem 4.11), among these two dominating sets and the minimum dominating set of  $P_5 \square C_4$ , as shown in Figure 4.8, we get the dominating sets for  $P_5 \square C_n$ ,  $n \geq 6$ ,  $n \neq 9$ , with required cardinality.  $\square$

# Chapter 5

## Sensor Placement

### 5.1 Introduction

In this chapter we consider the coverage problem in WSNs composed of static sensors, deployed stochastically in a rectangular grid which defines the ROI. We are interested in placing sensors at the vertices of a rectangular grid. Sensors are dropped at the vertices of the grid from air. However the ROI may not be fully covered as the sensors may not be placed at the target vertices. An uncovered vertex is called sensing hole. An actuator is assigned to carry and place the misplaced sensors according to some pre-assigned algorithm so as to fill in the sensing holes. The actuator rearranges some of the sensors in such a way that at least one sensor should be placed at each node. In this chapter, we consider that there is only one actuator and it can carry only one sensor when it travels from one vertex to another. We develop three algorithms for the actuator and compare these algorithms in context with some pre-assigned parameters.

### 5.2 Preliminary Assumptions

Consider a  $(m + 2) \times (n + 2)$  rectangular grid whose nodes are labeled as  $(i, j)$ , with  $i = 0, 1, \dots, m + 1$  and  $j = 0, 1, \dots, n + 1$ , where  $m, n$  are two parameters of our problem. Suppose that the sensors have the communication radius  $r_c$ ,

and the sensing radius  $r_s$ , where  $r_c \geq \sqrt{2}r_s$ . Then, the distance between two adjacent nodes is  $\sqrt{2}r_s$ . At least one sensor has to be placed at each  $(i, j)$  nodes for  $i = 1, 2, \dots, m$ ; and  $j = 1, 2, \dots, n$ . Sensors are deployed from air by helicopter. But sensors may not be placed in the proper node for various reasons. It may be placed in one of the four adjacent nodes. Each sensor has an ID number.

Helicopter will deploy one sensor with probability  $(1 - \mu)$  or two sensors with probability  $\mu$  at each node  $(i, j)$  for  $i = 1, 2, \dots, m$ , and  $j = 1, 2, \dots, n$ , where  $\mu$  is a parameter of our problem. Let the ID number(s) of the sensor(s) which is(are) supposed to be deployed on the grid  $(i, j)$  be  $ID(i, j)$ . We assume that the sensor(s) with ID number  $ID(i, j)$  will be placed at correct node i.e. at node  $(i, j)$  with probability  $p$  or any one of the adjacent nodes i.e., node  $(i - 1, j)$  or  $(i + 1, j)$  or  $(i, j - 1)$  or  $(i, j + 1)$  with probability  $q$  each. Where  $q = \frac{1-p}{4}$  is also a parameter.

After deployment of sensors, the actuator will go to the node  $(1, 1)$  and start its tour according to some pre-assigned algorithms which we have developed in this section. An actuator is a robot which can travel, carry sensors, identify the ID numbers of sensors and count the number of sensors at a node. We assume that there is only one actuator. We also assume that an actuator can carry at most one sensor with itself when it travels from one node to another node. The actuator can travel along the path (horizontal or vertical) of the grid. If the actuator is standing at node  $(i, j)$  then it can recognize the total number of sensors and their ID numbers which are placed at the node  $(i, j)$  and also at the adjacent four nodes.

### 5.2.1 Some Parameters

The traversed length  $L$  of the actuator is the most important parameter of our problem. It is the length traveled by the actuator starting from the node  $(1, 1)$  to the node  $(m, n)$  or  $(m, 1)$  according as  $m$  is odd or even. The distance traveled by the actuator in going from one node to its adjacent node is taken as the unit of traversed length. The parameters of our problem are as follows.

We assume  $m$  and  $n$  are large such that the product of  $m$  and  $n$  is closed 10000, or more; and  $4q \leq 0.5$ , i.e.,  $p \geq 0.5$ . Given  $m, n$  and  $q$  one can find the



Grid size	$m \times n$
Error probability	$q$
Repetition probability	$\mu$
Traversed length	$L$
Traversed length with sensor in hand	$L_S$
Number of empty nodes after deployment	$N$

relationship between  $\mu$  and  $L$  and can minimize  $\mu$  for fixed  $L$ .

Here we have developed three different algorithms for the actuator. We compare  $L$  and  $L_S$  obtained from three different algorithms for several different values of parameters  $q$  and  $\mu$  by simulation. We find the expected value of  $L$  in terms of other parameters for the first algorithm. We also find the expected value, an approximate distribution and some theoretical results for  $N$ .

### 5.2.2 Some Definitions and Notation

The actuator will start from node  $(1, 1)$  and terminate at  $(m, n)$  or  $(m, 1)$  according as  $n$  is odd or even. Define,  $(\bar{i}, \bar{j})$  as the next node of  $(i, j)$ , where

$$(\bar{i}, \bar{j}) = \begin{cases} (i, j + 1) & \text{if } i \text{ is odd and } j \leq (n - 1) \\ (i, j - 1) & \text{if } i \text{ is even and } j \geq 2 \\ (i + 1, j) & \text{otherwise} \end{cases}$$

The actuator will move from a node  $(i, j)$  toward the next node  $(\bar{i}, \bar{j})$ , as defined above. If  $(i, j)$  is the next node of  $(i', j')$ , then we call  $(i', j')$  the previous node of  $(i, j)$ . Let us define the following in a similar direction.

$$\begin{aligned} (i', j') & \quad \text{previous node of } (i, j) \\ (i'', j'') & \quad \text{previous node of } (i', j') \\ (i - 1, j) & \quad \text{the north node of } (i, j) \end{aligned}$$

We can similarly define east, west and south nodes. Let us denote  $(i - 2, j)$ ,  $(i, j - 2)$ ,  $(i, j + 2)$ ,  $(i - 1, j - 1)$ ,  $(i - 1, j + 1)$  as the two distance nodes of  $(i, j)$ . Also define  $(i, j)$  as a external node if  $i = 0$  or  $i = m + 1$  or  $j = 0$  or

$j = n + 1$ .

Let  $X(i, j)$  denote the number of sensor(s) with ID number  $(i, j)$  which are correctly placed at the node  $(i, j)$ . We define  $X(i, j) \uparrow$  to be the number of sensor(s) with ID number  $(i, j)$  which are placed at the node  $(i - 1, j)$ . Similarly, we define  $X(i, j) \rightarrow$ ,  $X(i, j) \downarrow$ ,  $X(i, j) \leftarrow$ . Again, we define

$$\bar{X}(i, j) = \begin{cases} X(i, j) \rightarrow & \text{if } (\bar{i}, \bar{j}) = (i, j + 1) \\ X(i, j) \downarrow & \text{if } (\bar{i}, \bar{j}) = (i + 1, j) \\ X(i, j) \leftarrow & \text{if } (\bar{i}, \bar{j}) = (i, j - 1). \end{cases}$$

That is,  $\bar{X}(i, j)$  is the number of sensor(s) with ID number  $(i, j)$  which are placed at the next node  $(\bar{i}, \bar{j})$ . Similarly, we define

$$X'(i, j) = \begin{cases} X(i, j) \leftarrow & \text{if } (i', j') = (i, j - 1) \\ X(i, j) \uparrow & \text{if } (i', j') = (i - 1, j) \\ X(i, j) \rightarrow & \text{if } (i', j') = (i, j + 1). \end{cases}$$

That is,  $X'(i, j)$  is the number of sensor(s) with ID number  $(i, j)$  which are placed at the previous node  $(i', j')$ . Let us also define  $T(i, j)$  as the number of sensor(s) (of any ID number) which are placed at the node  $(i, j)$ , and  $H(i, j)$  as the number of sensor(s) with the actuator when it stands at the node  $(i, j)$ . As we have presented the notation we need for our work, we can now move on to the technical details of the proposed algorithms.

### 5.3 Three Algorithms

We state here three algorithms. One is based on the ID numbers of the sensors and other two do not. For the first one we further assume that the sensor with ID number  $ID(i, j)$  should be placed by the actuator either at the node  $(i, j)$  or one of the four adjacent nodes. For the second and third algorithms we assume that sensors have no ID number. The third algorithm is a slight modification of the second one. In the second algorithm, the actuator looks only at the one distance neighbors, while in the third algorithm, the actuator

looks at the two distance neighbors, thus resulting in an improvement.

### 5.3.1 First Algorithm

We assume here that each sensor has an ID number.

For  $i = 1$  to  $m$  and  $j = 1$  to  $n$ ,

when the actuator is standing on the node  $(i, j)$ , do the following:

- If  $(H(i, j) = 1)$   
place sensor there and continue to case  $(H(i, j) = 0)$
- If  $(H(i, j) = 0)$  the actuator will do the following:
  1. If  $(X(i, j) = 2, X(\bar{i}, \bar{j}) = 0)$  move to  $(\bar{i}, \bar{j})$  with one sensor
  2. If  $(X(i, j) = 2, X(\bar{i}, \bar{j}) \geq 1)$  move to  $(\bar{i}, \bar{j})$  with no sensor
  3. If  $(X(i, j) = 1, X(\bar{i}, \bar{j}) \geq 1)$  move to  $(\bar{i}, \bar{j})$  with no sensor
  4. If  $(X(i, j) = 1, X(\bar{i}, \bar{j}) = 0, X'(\bar{i}, \bar{j}) > 0)$   
move to  $(\bar{i}, \bar{j})$  with one sensor whose ID number is  $ID(\bar{i}, \bar{j})$
  5. If  $(X(i, j) = 1, X(\bar{i}, \bar{j}) = 0, X'(\bar{i}, \bar{j}) = 0)$   
move to  $(\bar{i}, \bar{j})$  with no sensor
  6. If  $(X(i, j) = 0$  and  $[\bar{X}(i', j') \geq 1$  or  $X(i - 1, j) \downarrow > 0$  or  $(\bar{X}(i, j) > 0$  and  $X'(\bar{i}, \bar{j}) > 0)])$   
move to  $(\bar{i}, \bar{j})$  with no sensor
  7. Else the actuator will go to the node where the sensor with ID number  $ID(i, j)$  is placed, take the sensor, come back to  $(i, j)$ , place the sensor there and continue to case  $X(i, j)=1$ .

### 5.3.2 Second Algorithm

In this case, we assume that all the sensors are identical, and possess no ID.

For  $i = 1$  to  $m$  and  $j = 1$  to  $n$

when the actuator is standing on the node  $(i, j)$ , do the following:

- If  $(T(i, j) \geq 2 \text{ and } H(i, j) = 1)$   
the actuator will move the next node  $(\bar{i}, \bar{j})$  with the sensor in hand.
- If  $(T(i, j) \geq 2 \text{ and } H(i, j) = 0 \text{ and } T(\bar{i}, \bar{j}) \leq 1)$   
the actuator will take one sensor and move to the next node.
- If  $(T(i, j) \geq 2 \text{ and } H(i, j) = 0 \text{ and } T(\bar{i}, \bar{j}) \geq 2)$   
the actuator will move to the next node with no sensor.
- If  $(T(i, j) = 1 \text{ and } H(i, j) = 1)$   
the actuator will move to the next node  $(\bar{i}, \bar{j})$  with the sensor in hand.
- If  $(T(i, j) = 1 \text{ and } H(i, j) = 0 \text{ and } T(\bar{i}, \bar{j}) = 0)$   
the actuator will check the other three adjacent nodes whether the numbers of sensors placed there is greater than 1 (or 0 for external nodes) if the actuator finds such a node (or nodes) it will go to that node (or one of the nodes), take one sensor and back to the node  $(i, j)$  and then move to the next node with that sensor.
- If  $(T(i, j) = 1 \text{ and } H(i, j) = 0 \text{ and } T(\bar{i}, \bar{j}) \geq 1)$   
the actuator will move to the next node with no sensor.
- If  $(T(i, j) = 0 \text{ and } H(i, j) = 1)$   
the actuator will place the sensor on the node  $(i, j)$  and do the same job as in case of  $T(i, j) = 1 \text{ and } H(i, j) = 0$
- If  $(T(i, j) = 0 \text{ and } H(i, j) = 0)$   
the actuator will check the adjacent four nodes (order of checking is north-east-west-south node of the node  $(i, j)$  whether the numbers of sensors placed there is greater than 1 (or 0 for external nodes). If so it will go there and take one sensor, place that sensor on the present node  $(i, j)$  and do the same job as in case of  $T(i, j) = 1 \text{ and } H(i, j) = 0$
- Else go to the next node with no sensor.

### 5.3.3 Third Algorithm

This algorithm is nearly same as the second one except the following cases.

**Case 1:  $T(i, j) = H(i, j) = 0$** 

At first the actuator will do the same job as in the second algorithm but if the actuator cannot find such nodes then before going to the next node it will check the same condition and do the same job for all two distance node of the node  $(i, j)$  through which it has already traveled.

**Case 2:  $T(i, j) + H(i, j) > 1$** 

At first the actuator will do the same job as in the case of the second algorithm but before going to the next node the actuator again check whether  $T(i, j) + H(i, j) > 1$ . If  $T(i, j) + H(i, j) > 1$ , the actuator checks whether  $T(i'', j'') \geq 1$  for all the two distance nodes (through which it already traveled) of the node  $(i'', j'')$ . If  $T(i'', j'') = 0$  for one of the two distance nodes the actuator shifts one sensor from the node  $(i, j)$  to  $(i'', j'')$  and continues this process until one of the two conditions is false.

**5.3.4 Simulation Result**

For  $m = 100, n = 100$ , we simulate the different parameters of the problem. We also calculate the expected value of  $N$  and  $L_1$  from the theoretical results. There will be no empty nodes after placement sensors using the first algorithm.

Simulation results for $p = 0.5$											
$\mu$	$N$	$E(N)$	$E(L_1)$	$L_1$	$L_2$	$L_3$	$N_2$	$N_3$	$L_{S_1}$	$L_{S_2}$	$L_{S_3}$
0.2	2373	2387	15284	14949	11437	11955	113	2	3584	7170	7240
0.3	2171	2138	14688	14083	11187	11425	52	0	3280	7376	7376
0.4	1996	1909	14122	13731	10945	11001	15	1	3051	7543	7511
0.5	1708	1698	13588	12843	10603	10635	8	0	2700	7708	7639
0.6	1528	1501	13084	12447	10449	10465	4	0	2454	7798	7741
Simulation results for $p = 0.6$											
0.2	2148	2130	14206	14057	11257	11633	81	2	2993	7270	7273
0.3	1969	1905	13660	13355	11071	11163	37	2	2713	7549	7514
0.4	1679	1694	13150	12813	10647	10713	17	0	2426	7854	7808
0.5	1471	1496	12676	12173	10443	10509	4	0	2176	7922	7865
0.6	1308	1311	12236	11755	10363	10375	3	0	1938	8077	8017

Table 5.1: Simulation Results

The simulation is performed using a C-program, and required random numbers are generated using the standard C-library. In Table 5.1,  $L_i$  denotes the value of  $L$  for the  $i$ -th algorithm, where  $i = 1, 2, 3$ ,  $N$  is the number of

empty nodes after deployment of the sensors by helicopter, and  $E(L_1)$  is the approximate expected value of traversed length of the actuator using the first algorithm (calculated in the next section).  $E(N)$  is the expected number of empty nodes after deployment of sensors by the helicopter (calculated in the next section),  $N_i$  is the number of empty nodes after placement of sensors by the actuator using the  $i$ -th algorithm ( $i = 2, 3$ ), and  $L_{S_i}$  is the traversed length with sensor in hand of the actuator using the  $i$ -th algorithm ( $i = 1, 2, 3$ ).

### 5.3.5 Observations

A few immediate observations from Table 5.1 are as follows.

1. The number of empty nodes after placement of sensors by the actuator using the third algorithm is almost zero. This is because in the third algorithm, the actuator considers two distant nodes. If the two distant nodes are empty or if they have more than one sensor, then the actuator tries to equalize the number of sensors as much as possible. For much larger values of  $n$  and  $m$ , the number of empty nodes may be high. Then we can use an improved algorithm in the actuator will consider three or more distant nodes.
2. Simulated values of  $L_1$  are always less than  $E(L_1)$  in each cases, because we have slightly modified the algorithm (to get more efficiency) when we write the C code for simulation. Difference between these two values is less than 5%.
3. Difference between the simulated values of  $N$  and the expected values of  $N$  is less than 3%. It means that the simulation is good enough, and we can use the expected value in practical situations.
4. Simulated values of  $N, L_i, N_i$  decrease as the values of  $p$  or  $\mu$  increase. This is because an increment of  $p$  means that the error probability is lower, an increment of  $\mu$  means that the repetition probability is higher. In both the cases number of empty nodes after deployment is reduced, and hence, traversed length of the actuator ( $L$ ) and the number of empty nodes at the end of the tour of the actuator will be less.

5. Simulated values of  $L_{S_1}$  decrease as the values of  $p$  or  $\mu$  increase, whereas the simulated values of  $L_{S_2}, L_{S_3}$  decrease as the values of  $p$  or  $\mu$  increase. The reason is same as in the previous observation.
6. For fixed  $p$  and  $\mu$ , values of  $L_1$  are larger than those of  $L_2, L_3$ , and values of  $L_3$  are slightly higher than those of  $L_2$ . The reason is same as in observation 4.
7. For fixed  $p$  and  $\mu$ , values of  $L_{S_1}$  are very low compared to  $L_2, L_3$ , and values of  $L_3$  are more or less equal to that of  $L_2$ .

## 5.4 Theoretical Results

### 5.4.1 Results on $L_1$

**Result 5.1.** *The expectation of  $L_1$  is*

$$(mn - 1) + 2(t_1 + 3t_2 + 2(m + n - 4)t_3 + (m - 2)(n - 2)t_4)$$

where

$$t_1 = d - 2ab + cb^2$$

$$t_2 = d - a - ab - de + cb + ae + abe - cbe$$

$$t_3 = d - a - 2ab - de + 2cb + ae + cb^2 + 2ab^2e - 2cbe - cb^2e^2$$

$$t_4 = d - a - 3ab - de + 2cb + 3ab + ae + 3cb^2 - 3abe - 3cbe - 3cb^2e$$

with  $a = 7\mu q^2 + (1 - \mu)q$ ,  $b = \mu(2 - q)q + (1 - \mu)q$ ,  $c = 2\mu q^2$ ,  $d = 4q(1 - \mu(1 - 4q))$ , and  $e = \mu(1 - 4q)^2$ .

*Proof.* Following the notational convention of this section, we have

$$P(X(i, j) = 0) = \mu(4q)^2 + (1 - \mu)(4q) = (1 - p)(1 - \mu p) = d$$

$$P(X(i, j) = 1) = 2\mu(1 - 4q)(4q) + (1 - \mu)(1 - 4q) = (1 - 4q)(1 + 8\mu q - \mu)$$

$$P(X(i, j) = 2) = \mu p^2 = e$$

$$P(X(i, j) \uparrow = 0) = \mu(1 - q)^2 + (1 - \mu)(1 - q) = (1 - q)(1 - \mu q)$$

$$P(X(i, j) \uparrow = 1) = 2\mu(1 - q)q + (1 - \mu)q$$

$$P(X(i, j) \uparrow = 2) = \mu q^2$$

$$P(X(i, j) = 0, X(i, j) \uparrow \geq 1) = 7\mu q^2 + (1 - \mu)q = a$$

$$P(X(i, j) \uparrow \geq 1) = \mu(2 - q)q + (1 - \mu)q = b$$

$$P(X(i, j) \uparrow = 1, X(i, j) \downarrow = 1) = 2\mu q^2 = c.$$

Note that,  $X(i, j)$ ,  $X(i, j) \uparrow$ ,  $X(i, j) \rightarrow$ ,  $X(i, j) \downarrow$  and  $X(i, j) \leftarrow$  are identically distributed for all  $i, j$ . Also,  $X(i, j)$ ,  $X(i, j) \uparrow$ ,  $X(i, j) \rightarrow$ ,  $X(i, j) \downarrow$  and  $X(i, j) \leftarrow$  are independent to  $X(k, l)$ ,  $X(k, l) \uparrow$ ,  $X(k, l) \rightarrow$ ,  $X(k, l) \downarrow$  and  $X(k, l) \leftarrow$  for  $(i, j) \neq (k, l)$ .

Let,  $L_1(i, j)$  is the excess length travel by the actuator at the node  $(i, j)$  to place sensor at the node  $(i, j)$ . Therefore,  $L_1 = (mn - 1) + \sum L(i, j)$ .

$$P(L_1(1, 1) = 2) = P(X(1, 1) = 0) - P[(X(1, 1) = 0)$$

$$\text{and } ((X(1, 1) \rightarrow \geq 1, X(1, 2) \leftarrow \geq 1)$$

$$\text{or } (X(1, 1) \downarrow \geq 1, X(2, 1) \uparrow \geq 1))]$$

The right hand side, following a detailed calculation, gives

$$P(X(1, 1) = 0) - P[(X(1, 1) = 0, X(1, 1) \rightarrow \geq 1, X(1, 2) \leftarrow \geq 1)$$

$$\text{or } (X(1, 1) = 0, X(1, 1) \downarrow \geq 1, X(2, 1) \uparrow \geq 1)]$$

$$= P(X(1, 1) = 0)$$

$$- P(X(1, 1) = 0, X(1, 1) \rightarrow \geq 1, X(1, 2) \leftarrow \geq 1)$$

$$- P(X(1, 1) = 0, X(1, 1) \downarrow \geq 1, X(2, 1) \uparrow \geq 1)$$

$$+ P[X(1, 1) \rightarrow \geq 1, X(1, 2) \leftarrow \geq 1, X(1, 1) \downarrow \geq 1, X(2, 1) \uparrow \geq 1]$$

$$= P(X(1, 1) = 0)$$

$$- P(X(1, 1) = 0, X(1, 1) \rightarrow \geq 1)P(X(1, 2) \leftarrow \geq 1)$$

$$- P(X(1, 1) = 0, X(1, 1) \downarrow \geq 1)P(X(2, 1) \uparrow \geq 1)$$

$$+ P(X(1, 1) \rightarrow \geq 1, X(1, 1) \downarrow \geq 1)P(X(1, 2) \leftarrow \geq 1)$$

$$P(X(2, 1) \uparrow \geq 1)$$

$$= d - ab - ab + cb^2 = d - 2ab + cb^2 = t_1$$



Again, we have

$$\begin{aligned}
P(L_1(1, n) = 2) &= P(X(1, n) = 0) \\
&\quad - P[(X(1, n) = 0, X(1, n) \leftarrow \geq 1) \\
&\quad \text{or } (X(1, n) = 0, X(1, 1) \downarrow \geq 1, X(2, 1) \uparrow \geq 1) \\
&\quad \text{or } (X(1, n) = 0, X(1, n-1) = 2)] \\
&= d - a - ab - de + cb + ae + abe - cbe = t_2
\end{aligned}$$

Similarly,  $P(L_1(m, 1) = 2) = P(L_1(m, n) = 2) = t_2$ .

$$\begin{aligned}
&P(L_1(1, 2) = 2) \\
&= P(X(1, 2) = 0) - P[(X(1, 2) = 0) \text{ and } ((X(1, 2) \leftarrow \geq 1) \\
&\quad \text{or } (X(1, 2) \downarrow \geq 1, X(2, 2) \uparrow \geq 1) \\
&\quad \text{or } (X(1, 2) \rightarrow \geq 1, X(1, 3) \leftarrow \geq 1) \text{ or } (X(1, 1) = 2))] \\
&= d - a - 2ab - de + 2cb + ae + cb^2 + 2ab^2e - 2cbe - cb^2e^2 = t_3
\end{aligned}$$

Similarly,  $P(L_1(1, j) = 2) = P(L_1(m, j) = 2) = t_3$  for  $j = 2, 3, \dots, n-1$  and  $P(L_1(i, 1) = 2) = P(L_1(i, n) = 2) = t_3$  for  $i = 2, 3, \dots, m-1$ .

$$\begin{aligned}
P(L_1(2, 2) = 2) &= P(X(2, 2) = 0) \\
&\quad - P[(X(2, 2) = 0) \text{ and } ((X(2, 2) \rightarrow \geq 1) \\
&\quad \text{or } (X(2, 2) \leftarrow \geq 1, X(2, 1) \rightarrow \geq 1) \\
&\quad \text{or } (X(2, 2) \uparrow \geq 1, X(1, 2) \downarrow \geq 1) \\
&\quad \text{or } (X(2, 2) \downarrow \geq 1, X(3, 2) \uparrow \geq 1) \\
&\quad \text{or } (X(2, 3) = 2))] \\
&= d - a - 3ab - de + 2cb + 3ab + ae + 3cb^2 \\
&\quad - 3abe - 3cbe - 3cb^2e = t_4
\end{aligned}$$

Similarly,  $P(L_1(i, j) = 2) = t_4$  for  $i = 2, 3, \dots, m-1$  and  $j = 2, 3, \dots, n-1$ . Hence the Expectation of  $L_1$  is

$$\begin{aligned}
&(mn - 1) + 2 \sum P(L_1(i, j) = 2) \\
&= (mn - 1) + 2(t_1 + 3t_2 + 2(m + n - 4)t_3 + (m - 2)(n - 2)t_4).
\end{aligned}$$

Hence the result. □

### 5.4.2 Results on $T(i, j)$

**Result 5.2.** *Probability that the node  $(i, j)$  is empty ( $P(T(i, j) = 0)$ ) is:*

$$\begin{aligned}
 &P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^2 \quad \text{for } i = 1, m \text{ and } j = 1, n \\
 &P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^3 \quad \text{for } i = 1, m \text{ and } j = 2, 3, \dots, n - 1 \\
 &P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^3 \quad \text{for } i = 2, 3, \dots, m - 1 \text{ and } j = 1, n \\
 &P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^4 \quad \text{otherwise}
 \end{aligned}$$

where  $P(X(i, j) = 0) = \mu(4q)^2 + (1 - \mu)(4q) = (1 - p)(1 - \mu p)$ , and  $P(X(i, j) \uparrow = 0) = \mu(1 - q)^2 + (1 - \mu)(1 - q) = (1 - q)(1 - \mu q)$ .

*Proof.* Follows from the first algorithm. □

### 5.4.3 Results on $N$

**Result 5.3.** *Let  $N(i, j)$  be defined as*

$$N(i, j) = \begin{cases} 1 & \text{if } T(i, j) = 0 \\ 0 & \text{otherwise} \end{cases}$$

*Then,  $N(i, j)$ 's are Bernoulli random variables with parameter  $P(T(i, j) = 0)$  and they are dependent. For all values of  $i \neq 1, m$  and  $j \neq 1, n$ ,  $P(T(i, j) = 0)$ 's are equal. Again,  $P(T(i, j) = 0)$ 's are equal for  $(i, j) = (1, 1), (1, n), (m, 1)$  and  $(m, n)$ . Further  $P(T(i, j) = 0)$ 's are equal for all the other boundary nodes, and  $N = \sum N(i, j)$ .*

*Proof.* Clearly, for all  $i, j$ ,  $N(i, j)$ 's are Bernoulli random variables with parameter  $P(T(i, j) = 0)$ . Since  $T(i, j)$ 's are dependent  $N(i, j)$ 's are also dependent. The four corner nodes  $(1, 1), (1, n), (m, 1)$  and  $(m, n)$  are similar in terms of the different probabilities when sensors are deployed from the air. Hence  $P(T(i, j) = 0)$ 's are equal. Similar arguments hold for other two cases also.

Now  $N(i, j) = 1$  means  $(i, j)$ -th node is empty hence  $\sum N(i, j)$  is the total number of empty nodes.  $\square$

**Result 5.4.** *The expectation of  $N$  is approximately equal to  $4mnpq(1 - \mu + 4\mu^2pq)$  which is less than or equal to  $\frac{mn(1-\mu+\mu^2/4)}{4}$ .*

*Proof.* The expectation of  $N$  is

$$\begin{aligned} E(N) &= E(\sum N(i, j)) = \sum P(T(i, j) = 0) \\ &= 4(P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^2) \\ &\quad + 2(m + n - 4)(P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^3) \\ &\quad + (m - 2)(n - 2)(P(X(i, j) = 0)P(X(i, j) = 0)^4) \end{aligned}$$

using result 3.4.3.

Approximate value of  $E(N)$  for large  $m, n$  is

$$\begin{aligned} &mn(P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^4) \\ &= mn(1 - p)(1 - \mu p)(1 - q)^4(1 - \mu q)^4 \\ &\approx mn(1 - p)(1 - \mu p)(1 - 4q)(1 - 4\mu q) \\ &= mn(1 - p)p(1 - \mu p - 4\mu q + 4\mu^2pq) \\ &= 4mnpq(1 - \mu + 4\mu^2pq) \leq \frac{mn(1 - \mu + \mu^2/4)}{4}, \end{aligned}$$

since  $4pq \leq (\frac{p+4q}{2})^2 = 1/4$ . Hence the result.  $\square$

**Result 5.5.** *For  $i \neq 1, m$  and  $j \neq 1, n - 1, n$ , we have*

$$P(T(i, j) = 0, T(i, j + 1) = 0) \leq P(T(i, j) = 0)^2$$

*Proof.* We have  $P(T(i, j) = 0) = P(X(i, j) = 0)P(X(i, j) \uparrow = 0)^4 = 4q(1 - \mu + 4q\mu)(1 - q)^4(1 - \mu q)^4$ , and  $P(T(i, j) = 0, T(i, j + 1) = 0) = (3q)^2(1 - \mu + 3q\mu)^2(1 - q)^6(1 - \mu q)^6$ . Now,

$$\begin{aligned} &(3q)(1 - \mu + 3q\mu) \leq (4q)(1 - \mu + 4q\mu)(7/8)^2 \\ &\leq (4q)(1 - \mu + 4q\mu)(1 - q)^2 \quad (\text{since, } q \leq 1/8) \\ &\leq (4q)(1 - \mu + 4q\mu)(1 - q)(1 - \mu q) \end{aligned}$$

Therefore,  $(3q)^2(1-\mu+3q\mu)^2(1-q)^6(1-\mu q)^6 \leq (4q)^2(1-\mu+4q\mu)^2(1-q)^8(1-\mu q)^8$ , and hence the result.  $\square$

**Result 5.6.** For  $i \neq 1, m$  and  $j \neq 1, n-1, n$ , we have

$$P(T(i, j) = 0)^3 \leq P(T(i, j) = 0, T(i, j+1) = 0)$$

*Proof.* Following the regular notation of this chapter, we have

$$\begin{aligned} & (p^2 + q^2 + 5pq) \leq p + q \\ \Rightarrow & (p^2 + q^2 + 5pq)\mu \leq p + q \\ \Rightarrow & 3\mu pq \leq (p + q) - \mu(p + q)^2 \\ \Rightarrow & \mu^2 pq \leq (\mu(p + q)/3)(1 - \mu(p + q)) \\ \Rightarrow & 1 - \mu(p + q) + \mu^2 pq \leq (1 + \mu(p + q)/3)(1 - \mu(p + q)) \\ \Rightarrow & (1 - \mu p)(1 - \mu q) \leq 1 - \mu(p + q)^{-1/3}(1 - \mu(p + q)) \\ \Rightarrow & (1 - \mu p)^3(1 - \mu q)^3 \leq (1 - \mu(p + q))^2 \\ \Rightarrow & (1 - \mu p)^3(1 - \mu q)^6 \leq (1 - \mu + 3q\mu)^2 \end{aligned}$$

Also,  $(4q)^3 \leq (3q)^2$ . Therefore,  $(4q)^3(1-q)^6 \leq (3q)^2$ , and hence,

$$(4q)^3(1-\mu p)^3(1-q)^6(1-\mu q)^6 \leq (3q)^2(1-\mu+3q\mu)^2$$

This gives us  $(4q)^3(1-\mu p)^3(1-q)^{12}(1-\mu q)^{12} \leq (3q)^2(1-\mu+3q\mu)^2(1-q)^6(1-\mu q)^6$ , and hence the result.  $\square$

**Result 5.7.** Using the usual notation, we have

$$-0.23 \leq Cor(N(i, j), N(i, j+1)) = Cor(N(i, j), N(i+1, j)) \leq 0$$

for  $i \neq 1, m-1, m$  and  $j \neq 1, n-1, n$ , and  $Cor(N(i, j), N(i, j+3)) = Cor(N(i, j), N(i+3, j)) = 0$  for  $i \neq m-2, m-1, m$  and  $j \neq n-2, n-1, n$ .

*Proof.* We have, for  $i \neq 1, m-1, m$  and  $j \neq 1, n-1, n$ ,

$$\begin{aligned} Cor(N(i, j), N(i, j+1)) &= Cor(N(i, j), N(i+1, j)) \\ &= \frac{Cov(N(i, j), N(i+1, j))}{Var(N(i, j))} \end{aligned}$$

since  $\text{Var}(N(i, j)) = \text{Var}(N(i + 1, j))$ . Now,

$$\begin{aligned} & \text{Cov}(N(i, j), N(i + 1, j)) \\ &= P(N(i, j) = 1, N(i + 1, j) = 1) - P(N(i, j) = 1)P(N(i + 1, j) = 1) \\ &= P(T(i, j) = 0, T(i + 1, j) = 0) - P(T(i, j) = 0)P(T(i + 1, j) = 0) \\ &= P(T(i, j) = 0, T(i + 1, j) = 0) - P(T(i, j) = 0)^2 \end{aligned}$$

Hence, by the Results 3.4.11 and 3.4.13 we have,

$$-P(T(i, j) = 0)^2[1 - P(T(i, j) = 0)] \leq \text{Cov}(N(i, j), N(i, j + 1)) \leq 0$$

Since  $\text{Var}(N(i, j)) = P(N(i, j) = 1) - P(N(i, j) = 1)^2$ . We get, for  $i \neq 1, m - 1, m$  and  $j \neq 1, n - 1, n$ ,

$$\begin{aligned} -P(T(i, j) = 0) &\leq \text{Cor}(N(i, j), N(i, j + 1)) \\ &= \text{Cor}(N(i, j), N(i + 1, j)) \leq 0 \end{aligned}$$

Also,  $P(T(i, j) = 0) = (1 - p)(1 - \mu p)(1 - q)^4(1 - \mu q)^4$  for  $i \neq 1, m$  and  $j \neq 1, n$ .

Now,  $(1 - p)(1 - q)^4$  and  $(1 - \mu p)(1 - \mu q)^4$  are increasing functions of  $q$  on  $[0, 1/5]$ . Hence, for  $q \leq 1/8$ , we have

$$(1 - p)(1 - q)^4 \leq 2401/8192$$

and for  $\mu > 0.2$ , we have

$$(1 - \mu p)(1 - \mu q)^4 \leq (1 - \mu/2)(1 - \mu/8)^4 \leq 0.81.$$

Therefore,  $P(T(i, j) = 0) \leq 0.23$ , and hence the result.  $\square$

**Result 5.8.**  *$N$  approximately follows normal distribution, for large  $m, n$  and for  $p \geq 1/2$  and  $\mu \geq 0.2$ .*

*Proof.*  $N = \sum N(i, j)$  and  $N(i, j)$ 's are Bernoulli random variables with parameter  $P(T(i, j) = 0) \leq 0.23$  for  $i \neq 1, m$  and  $j \neq 1, n$ ; and for other values of  $i$  and  $j$ ,  $N(i, j)$ 's are Bernoulli random variables.

Note that, for large  $m, n$  number of  $N(i, j)$  for  $i \neq 1, m$  and  $j \neq 1, n$  is

much larger than number of other  $N(i, j)$  and correlations are small hence by CLT,  $N$  is approximately Normally distributed with approximate expectation  $mnp(1 - p)(1 - \mu + 4\mu^2pq)$  and approximate variance  $mnp(1 - p)(1 - \mu + 4\mu^2pq)[1 - p(1 - p)(1 - \mu + 4\mu^2pq)]$ .  $\square$

# Single Event Detection in a Faulty WSN

## 6.1 Introduction

One fundamental challenge in the event detection problem for a sensor network is the detection accuracy which is limited by the amount of noise associated with the measurement and the reliability of sensor nodes. The sensors are usually low-end inexpensive devices and sometimes exhibit unreliable behavior. For example, a faulty sensor node may issue an alarm even though it has not received any signal for an event. On the other hand, it may fail to detect an actual event. Moreover, a sensor may be dead in which case it cannot send any alarm.

The event region may be large, and if an event occurs at a particular point of the region then the sensor may not determine exactly where the event has happened. There are cases where a fusion sensor cannot make a decision. Consider, for example, a network of sensors that are capable of sensing mines or bombs. We assume that either no mines (or bombs) are placed or very few mines (or bombs) are placed on a particular area of the ROI. In this case an important query could be; “are bombs present or not”. In that situation all sensors have to communicate with the base station, and base station will take the decision about the query.

In this chapter, we are interested in determining the occurrence of an event, when the position where the event may have occurred is already known. We assume that the ROI is partitioned into suitable number of identical squares of side  $2a$ . We consider ROI as a rectangular grid with square cells. We also consider a regular hexagonal grid with regular hexagonal cells in a separate section. We also assume that sensors have already been placed at the centers (call them *nodes*) of the squares (or hexagons).

In this chapter, we propose a rule for the base station to take a decision compiling the information coming from the all sensors and find the optimal solutions. We consider two types of error: 1) type I error when an event occurs but the sensors report normal (which is the more serious error) and 2) type II error when the ROI is normal but sensors report an event. We observed that type I and type II errors decrease when detection probabilities increase. If detection probabilities are low then type I error is close to 1. If probability of occurrence of the event is high but detection probabilities are small then type I and type II errors are high, which means there is no utility of sensors. So, when the probability of occurrence of the event is high, we have to use sensors with high detection probability (i.e., sensors with much better quality). We calculate the MP test and the Bayes test for some specific values of the parameters. We observed that for small values of detection probability and large value of loss, the Bayes test is not applicable. When loss is large, we cannot use sensors with small detection probabilities to determine the event's originating square (here after called the event square) using Bayes' test. We also observed that when the size of the test is small we cannot use sensors with small detection probabilities for the MP test; we have to use good sensors (sensors with high detection probability) for the MP test in this case. For details of loss, MP test and Bayes test, see [68].

## 6.2 Assumptions

In this section, we describe the assumptions we make; some are new and some are identical to the ones made by other researchers. The new assumptions lead to a new type of problem statement and a new approach to solve the problem.



1. We assume that (i) if the event occurs then it occurs at only one particular square of the grid which will be known as event square; (ii) there is no fusion sensor to take the decision locally; (iii) all sensors communicate with the base station, which takes the decision and (iv) the sensors have same sensing radius. If an event does not occur we say the ROI is *normal*. Sensors are deployed or manually placed over the ROI in such a way that they cover the entire ROI. We assume that sensors are placed a priori, at the center (which are known as nodes) of every square cell. We also consider a regular hexagonal grid with regular hexagonal cells in a separate section. Each sensor node can determine its location through a beacon-assisted positioning mechanism [7].

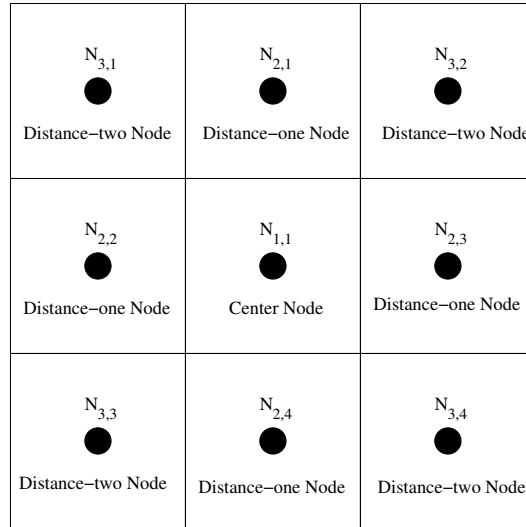


Figure 6.1: Nodes placed in cells of the ROI

2. We assume that there are only 9 sensors which can detect the event, see figure 6.1. These are (i) one sensor placed at the center of the event square (call it the center node), (ii) four sensors placed at the centers of adjacent squares with a common side (call them distance-one nodes) and (iii) four sensors placed at the centers of adjacent squares with a common vertex (call them distance-two nodes). Consider a  $3 \times 3$  square grid. If an event occurs, it occurs at the center square only. Our problem is to find whether the center square is actually an event square.
3. We assume that if an event occurs in the event square then the sensor,

lying in the event square, can detect the event with probability  $p_1$ . The distance-one nodes can detect the event with a lesser probability  $p_2$  and the distance-two nodes can detect the event with the lowest probability  $p_3$  due to different noise, distance and obstruction characteristic, etc. We assume that no other sensor can detect the event.

4. We also assume that there is prior probability of a particular square being the event square. Even if an event is detected by a sensor, it may not respond or send the information to the base station due to some technical fault (we call that sensor a *faulty sensor*) with some probability. Conversely, if the event was not detected or if there is no occurrence of the event (i.e., the normal situation), then a faulty sensor can falsely respond or send the wrong information to the base station with some probability. A sensor is called a *dead sensor* if the sensor does not work (detect) at all. A dead sensor sends no response in either case. If a sensor is dead or the ROI is normal then the sensors send no information. We also assume that the sensors work independently, i.e., detection and response of different sensors are independent.

### 6.3 Problem Statement and Notations

Our problem is to find various error probabilities, e.g., probability of false response when the ROI is normal or probability of no response when a particular square is the event square, etc. We want to develop schemes for the base station to take the decision and find the error probabilities of two different wrong situations: 1) the base station decides that the ROI is normal whereas the event has occurred and 2) the base station decides that the ROI is not normal, i.e., the event has occurred, but there is actually no event. We want to develop the schemes and find the error probabilities under two different considerations: a) classical hypothesis testing and b) a decision theoretic approach (i.e., Bayes test). In the consideration of decision theoretic approach, we introduce risk factor for two different wrong situations. We also calculate error probabilities for some values of different parameters like the probability of a false alarm of a sensor, probability of event detection by a sensor, prior probability of existence of the event etc. Our problem is to give an optimal test for the base station.

The node which is placed at the center square is the nearest node and hence can detect the event square with highest probability. We denote this node as  $N_{1,1}$ . The 4 nodes, whose distances are  $2a$  from the previous node, are the second nearest nodes and hence can detect the event square with second highest probability. We denote these nodes as  $N_{2,j}; j = 1, 2, 3, 4$ .

The 4 nodes, whose distances are  $2\sqrt{2}a$  from the center node, are the farthest nodes which can detect the event square and hence can detect the event square with lowest probability. We denote these nodes as  $N_{3,j}; j = 1, 2, 3, 4$ , see Figure 6.1.

For  $(i, j) \in \{(1, 1)\} \cup (\{2, 3\} \times \{1, 2, 3, 4\})$ , let  $y_{ij} = 1$  if the node  $N_{i,j}$  detects the center square as the event square,  $y_{ij} = 0$  if the node  $N_{i,j}$  detects the center square as normal,  $x_{ij} = 1$  if the node  $N_{i,j}$  responds, i.e., the node informs the base station that the center square is the event square, and  $x_{ij} = 0$  if the node  $N_{i,j}$  does not respond, i.e., the node informs the base station that the center square is normal.

Here we make one natural assumption, for  $k, l = 0, 1$ ,  $\Pr(x_{ij} = k \mid y_{ij} = l, \text{Normal}) = \Pr(x_{ij} = k \mid y_{ij} = l)$  and  $\Pr(x_{ij} = k \mid y_{ij} = l, \text{Event}) = \Pr(x_{ij} = k \mid y_{ij} = l)$ , i.e., the response of a sensor is independent of the event occurrence.

Note that, detection of event by a sensor does not mean that the sensor informs the base station that the center square is the event square; if the sensor is faulty, it can send a normal report. Similar thing can happen if sensor does not detect the event square. Also note that,  $y_{ij}$ 's are not independent, but  $y_{ij}$ 's are independent when it is given that the event occurs or not.

Let, for all possible values of  $i$  and  $j$ ,  $p_i = \Pr(y_{ij} = 1 \mid \text{Event})$ ;  
 $p_e = \Pr(\text{Event}) = \Pr(\text{event occurs})$ ;  $p_n = \Pr(\text{Normal}) = \Pr(\text{ROI is normal})$ ;  
 $p_c = \Pr(x_{ij} = 1 \mid y_{ij} = 1)$  and  $p_w = \Pr(x_{ij} = 1 \mid y_{ij} = 0)$ .

These are important parameters of the problem. Clearly,  $\Pr(y_{ij} = 1 \mid \text{Normal}) = 0$  for all possible values of  $i$  and  $j$ .

## List of Notations

$N_{i,j}$  is the  $(i, j)$ -th node.

$r_c$  = communication radius of the sensors.

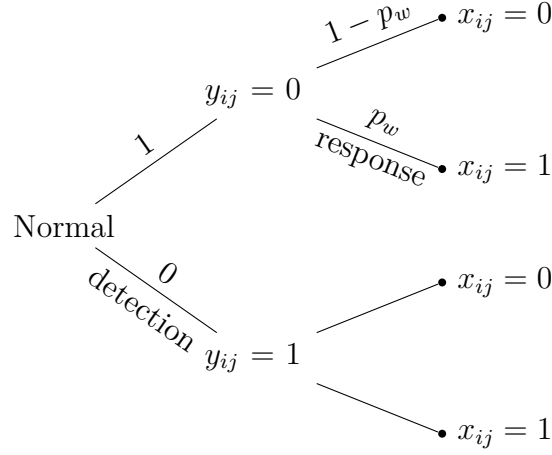


Figure 6.2: Detection and response probabilities when ROI is normal

$r_s$  = sensing radius of the sensors.

$2a$  = distance between two nodes= length of a square of the grid.

$p_1$  = probability of detection of the event by the node at event square.

$p_2$  = probability of detection of the event by distance-one nodes.

$p_3$  = probability of detection of the event by distance-two nodes.

$p_c$  = probability of correct response.

$p_w$  = probability of wrong response.

$d = p_c - p_w$ .

$p_e$  = probability of the event occurs in ROI.

$p_n$  = probability of ROI is normal.

$l_e$  is the loss when event occurs but base station takes decision as normal.

$l_n$  = is the loss when ROI is normal but base station takes decision as event.

$l = l_e/l_n$  = ratio of the losses.

## 6.4 Theoretical Analysis of Fault Detection

In this section, we derive various error probabilities for all nodes and then propose a rule for the base station to take a decision compiling the information

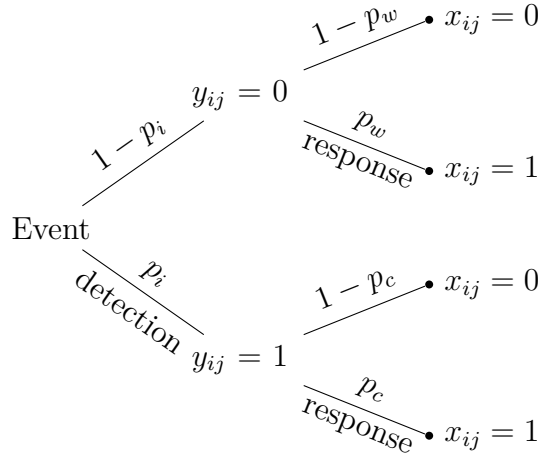


Figure 6.3: Detection and response probabilities when event occurs

coming from all the 9 nodes and to find the optimal solution. Finally, we calculate the error probabilities and the tests for the base station.

Let us consider the testing problem  $H_0$ : Event vs.  $H_1$ : Normal. There are two types of error: type I error when event occurs but sensor reports normal and type II error when ROI is normal but sensor reports Event. We consider ‘Event’ as null hypothesis because type I error should be the more serious error than type II error. If we reject the null hypothesis when it is true, i.e., if Event occurs but base station decides Normal, then that will be the more serious error than the other one.

Throughout the section, we consider  $i = 1, j = 1$  and  $j = 1, 2, 3, 4$  when  $i = 2, 3$ . There are eight possible scenarios for a particular node  $N_{i,j}$ , see Figure 6.2 and 6.3 (the numbers which are above the line segments are corresponding probabilities).

1. Normal,  $y_{ij} = 0, x_{ij} = 0$  (sensor correctly detects a normal reading and sends the correct message to the base station),
2. Normal,  $y_{ij} = 0, x_{ij} = 1$  (sensor correctly detects a normal reading but sends the wrong message to the base station due to fault),
3. Normal,  $y_{ij} = 1, x_{ij} = 0$  (sensor wrongly detects a normal reading as event but sends the normal message to the base station due to fault),
4. Normal,  $y_{ij} = 1, x_{ij} = 1$  (sensor wrongly detects a normal reading as

event but sends the correct message to the base station),

5. Event,  $y_{ij} = 0, x_{ij} = 0$  (sensor wrongly detects an event reading and sends the wrong message i.e. normal message to the base station),

6. Event,  $y_{ij} = 0, x_{ij} = 1$  (sensor wrongly detects an event reading but sends the correct message to the base station),

7. Event,  $y_{ij} = 1, x_{ij} = 0$  (sensor correctly detects an event reading but sends the wrong message to the base station due to fault), and

8. Event,  $y_{ij} = 1, x_{ij} = 1$  (sensor correctly detects an event reading and sends the correct message to the base station).

### 6.4.1 Error Probabilities for Nodes

$$\begin{aligned}
 \text{Let, } P_N &= \Pr(x_{ij} = 0 \mid \text{Normal}) = \Pr(x_{ij} = 0 \mid y_{ij} = 0, \text{Normal}) \times \\
 &\Pr(y_{ij} = 0 \mid \text{Normal}) + \Pr(x_{ij} = 0 \mid y_{ij} = 1, \text{Normal}) \Pr(y_{ij} = 1 \mid \text{Normal}) \\
 &= \Pr(x_{ij} = 0 \mid y_{ij} = 0) \Pr(y_{ij} = 0 \mid \text{Normal}) \\
 &+ \Pr(x_{ij} = 0 \mid y_{ij} = 1) \Pr(y_{ij} = 1 \mid \text{Normal}) = 1 - p_w, \\
 \text{and, } P_{E,i} &= \Pr(x_{ij} = 1 \mid \text{Event}) = \Pr(x_{ij} = 1 \mid y_{ij} = 0, \text{Event}) \times \\
 &\Pr(y_{ij} = 0 \mid \text{Event}) + \Pr(x_{ij} = 1 \mid y_{ij} = 1, \text{Event}) \Pr(y_{ij} = 1 \mid \text{Event}) \\
 &= \Pr(x_{ij} = 1 \mid y_{ij} = 0) \Pr(y_{ij} = 0 \mid \text{Event}) + \Pr(x_{ij} = 1 \mid y_{ij} = 1) \times \\
 &\Pr(y_{ij} = 1 \mid \text{Event}) = p_w(1 - p_i) + p_c p_i = p_w + p_i(p_c - p_w).
 \end{aligned}$$

Hence the probability of type I error for the node  $N_{i,j}$  is  $Q_{E,i} = \Pr(x_{ij} = 0 \mid \text{Event}) = 1 - \Pr(x_{ij} = 1 \mid \text{Event}) = (1 - p_w) - p_i(p_c - p_w)$  and probability of type II error for the node  $N_{i,j}$  is  $\Pr(x_{ij} = 1 \mid \text{Normal}) = 1 - P_N = p_w$ .

Now other types of errors may be as follows:

$$\begin{aligned}
 P_{1,i} &= \Pr(\text{Event} \mid x_{ij} = 0) = \frac{p_e Q_{E,i}}{p_n P_N + p_e Q_{E,i}} \\
 \text{and } P_{2,i} &= \Pr(\text{Normal} \mid x_{ij} = 1) = \frac{p_n p_w}{p_n p_w + p_e P_{E,i}}.
 \end{aligned}$$

### 6.4.2 Error Probabilities and Test for Base Station

Now, let us consider the detection problem for the base station. After the observations about  $x_{ij}$ 's are made, at the base station they are combined to make a final decision regarding the hypotheses ( $H_0$ : Event vs  $H_1$ : Normal). When  $H_0$  is true,  $x_{ij}$  follows  $\text{Ber}(P_{E,i})$ , and when  $H_1$  is true  $x_{ij}$  follows  $\text{Ber}(p_w)$ , where  $\text{Ber}(p_w)$  be the Bernoulli distribution with parameter  $p_w$ . Let the probability mass function of  $x_{ij}$  when  $H_k$  is true be  $\Pr(x_{ij} | H_k)$  for  $k = 0, 1$ . We make one more natural assumption  $p_c > p_w$ , which is equivalent to say  $P_{E,i} > p_w$  for all  $i$ . This is needed for a result in the next section.

#### The Neyman-Pearson Approach

In many practical situations, the prior probabilities may be unknown in which case the decision theoretic approach is not appropriate. So, we employ the Neyman-Pearson criterion. In that case, the most powerful (MP) test of size  $\alpha$  is to

$$\text{reject } H_0 \text{ when } \Pi \Pr(x_{ij} | H_1) > \lambda'' \Pi \Pr(x_{ij} | H_0)$$

and reject  $H_0$ , with probability  $k$ , when equality holds in place of greater than, where  $\lambda''$  and  $k$  can be found from the size  $\alpha$  of the test.

Since, when  $H_0$  is true  $x_{ij}$  follows  $\text{Ber}(P_{E,i})$  and, when  $H_1$  is true,  $x_{ij}$  follows  $\text{Ber}(p_w)$ , we can simplify the MP test as to

$$\text{reject } H_0 \text{ when } p_w^{\sum x_{ij}} (1 - p_w)^{\sum (1 - x_{ij})} >$$

$$\lambda'' P_{E,1}^{x_{11}} (1 - P_{E,1})^{(1 - x_{11})} P_{E,2}^{\sum x_{2j}} (1 - P_{E,2})^{\sum (1 - x_{2j})} P_{E,3}^{\sum x_{3j}} (1 - P_{E,3})^{\sum (1 - x_{3j})},$$

and reject  $H_0$  with probability  $k$  when equality holds in place of greater than,

$$\text{i.e., reject } H_0 \text{ when } \frac{1}{\lambda''} > \left( \frac{P_{E,1}}{p_w} \right)^{x_{11}} \left( \frac{1 - P_{E,1}}{1 - p_w} \right)^{(1 - x_{11})} \times$$

$$\left( \frac{P_{E,2}}{p_w} \right)^{\sum x_{2j}} \left( \frac{1 - P_{E,2}}{1 - p_w} \right)^{(4 - \sum x_{2j})} \left( \frac{P_{E,3}}{p_w} \right)^{\sum x_{3j}} \left( \frac{1 - P_{E,3}}{1 - p_w} \right)^{(4 - \sum x_{3j})}$$

and reject  $H_0$  with probability  $k$  when equality holds in place of greater than.

Hence, reject  $H_0$  when

$$x_{11} \ln \left( \frac{P_{E,1}}{p_w} \right) + (1 - x_{11}) \ln \left( \frac{1 - P_{E,1}}{1 - p_w} \right) + \Sigma x_{2j} \ln \left( \frac{P_{E,2}}{Q_N} \right) +$$

$$(4 - \Sigma x_{2j}) \ln \left( \frac{1 - P_{E,2}}{1 - p_w} \right) + \Sigma x_{3j} \ln \left( \frac{P_{E,3}}{p_w} \right) + (4 - \Sigma x_{3j}) \ln \left( \frac{1 - P_{E,3}}{1 - p_w} \right) < \lambda'$$

and reject  $H_0$  with probability  $k$  when equality holds in place of less than.

i.e., reject  $H_0$  when

$$x_{11} \ln \left( \frac{P_{E,1}(1 - p_w)}{(1 - P_{E,1})p_w} \right) + \Sigma x_{2j} \ln \left( \frac{P_{E,2}(1 - p_w)}{(1 - P_{E,2})p_w} \right) + \Sigma x_{3j} \ln \left( \frac{P_{E,3}(1 - p_w)}{(1 - P_{E,3})p_w} \right) < \lambda$$

and reject  $H_0$  with probability  $k$  when equality holds in place of less than.

Hence, we get the MP test as to reject  $H_0$  when

$$\Sigma x_{ij} \ln \left( \frac{P_{E,i}(1 - p_w)}{(1 - P_{E,i})p_w} \right) < \lambda \dots (R)$$

and reject  $H_0$  with probability  $k$  when equality holds in place of less than, where,  $\lambda$  and  $k$  can be found from the relation  $\Pr(H_0 \text{ reject} \mid H_0 \text{ true}) = \alpha$ , i.e.,

$$\Pr(\Sigma x_{ij} \ln(d_i) < \lambda) + k \Pr(\Sigma x_{ij} \ln(d_i) = \lambda) = \alpha,$$

where,  $d_i = \frac{P_{E,i}(1-p_w)}{(1-P_{E,i})p_w}$  and  $x_{ij}$  follows  $\text{Ber}(P_{E,i})$ .

Since we assume  $P_{E,i} > p_w$ ,  $\ln \left( \frac{P_{E,i}(1-p_w)}{(1-P_{E,i})p_w} \right) > 0$  for all  $i$ .

Based on the given error bound  $\alpha$  and sensor fault probabilities, the base station will take the decision given by the rule (R).

### Decision Theoretic Approach

A test  $T_g$  of  $H_0 : \theta = \theta_0$  vs  $H_1 : \theta = \theta_1$  is defined to be a Bayes test with respect to the prior distribution  $\Pr(H_1) = g$  if and only if

$$(1 - g)R_{T_g}(\theta_0) + gR_{T_g}(\theta_1) \leq (1 - g)R_T(\theta_0) + gR_T(\theta_1)$$



for any other test  $T$ , where  $R_T(\theta)$  is the risk function of the test  $T$ . The Bayes test is the test which seeks a critical region that minimizes the overall risk. If loss function is not available then we can assume the losses are 0 or 1. It can be proved that the Bayes test is to Reject  $H_0$  when

$$\frac{L_0}{L_1} < \frac{gl(d_0; \theta_1)}{(1-g)l(d_1; \theta_0)},$$

where  $L_0$  and  $L_1$  are the likelihoods for  $\theta = \theta_0$  and  $\theta = \theta_1$ , respectively;  $l(d_0; \theta_1)$  is the loss when null hypothesis is accepted but it is false, and  $l(d_1; \theta_0)$  is the loss when null hypothesis is rejected but it is true [68].

Let the losses be as follows:  $l_e$  when an event occurs but base station takes decision as normal,  $l_n$  when the ROI is normal but base station takes decision as event, and loss is 0 when the base station takes the correct decision. Hence, under the Bayesian setup, i.e., when the prior distribution  $(p_n, p_e)$  are available, the Bayes test with respect to the prior distribution  $\Pr(H_0) = p_e$  and  $\Pr(H_1) = p_n$  can be derived as follows:

$$\text{Reject } H_0 \text{ when } p_w^{\sum x_{ij}} (1 - p_w)^{\Sigma(1-x_{ij})} >$$

$$\frac{p_e l_e}{p_n l_n} P_{E,1}^{x_{1j}} (1 - P_{E,1})^{(1-x_{1j})} P_{E,2}^{\sum x_{2j}} (1 - P_{E,2})^{\Sigma(1-x_{2j})} P_{E,3}^{\sum x_{3j}} (1 - P_{E,3})^{\Sigma(1-x_{3j})},$$

$$\text{i.e., Reject } H_0 \text{ when } \sum x_{ij} \ln \left( \frac{P_{E,i}(1-p_w)}{(1-P_{E,i})p_w} \right) <$$

$$\ln \left( \frac{p_n l_n}{p_e l_e} \right) + \ln \left( \frac{1 - p_w}{1 - P_{E,1}} \right) + 4 \ln \left( \frac{1 - p_w}{1 - P_{E,2}} \right) + 4 \ln \left( \frac{1 - p_w}{1 - P_{E,3}} \right)$$

### 6.4.3 Boundary Case

In the above discussion, we assume that the center square is an interior one. If we consider the boundary squares, then the expression for the error probabilities for each sensor changed and consequently the expression for the error probabilities for the base station are also changed. In that case, if we consider the corner squares then  $j$  takes value 1 for  $i = 1, 3$  and  $j$  takes values 1, 2 for  $i = 2$ , and if we consider the boundary square other than a corner one then  $j$  takes value 1 for  $i = 1$ ;  $j$  takes values 1, 2, 3 for  $i = 2$ , and  $j$  takes values 1, 2

for  $i = 3$ . The theoretical analysis is similar as in the case of interior squares.

#### 6.4.4 When more sensors can detect the Event Square

We may consider the situation when sensing radius has larger value, and then more sensors can detect the event square but with different probabilities. In this case, we classify all the nodes as follows: two sensors belong to the same class if they have the same distance from the event square and hence have the same detection probability to detect the event. Let sensors in the  $i$ -th class detect the event square with probability  $p_i, i = 1, 2, 3, \dots$ . Then, the expressions of the error probabilities for the sensors and the test (MP and Bayes) are similar to the ones previously discussed, but now the summation in the left side of the expression of the test is changed; instead of three, there will be more terms.

### 6.5 Calculations and Observations

We have the independent set of parameters of the problem as follows:

$p_e, p_i, p_c, p_w$  for all possible values of  $i$  and  $j$ ,

$l = \text{ratio of losses} = l_e/l_n$  and size of the test  $= \alpha$ .

The type I error for  $N_{i,j}$  is  $(1 - p_w) - p_i(p_c - p_w)$  and the type II error for  $N_{i,j}$  is  $p_w$  for all possible values of  $i$  and  $j$ .

Let  $d = p_c - p_w > 0$ . Other types of errors for  $N_{i,j}$ 's are

$$P_{1,i} = \frac{p_e(1 - p_w - p_i d)}{1 - p_w - p_e p_i d} \quad \text{and} \quad P_{2,i} = \frac{(1 - p_e)p_w}{p_w + p_e p_i d}$$

for all possible values of  $i$  and  $j$ .

Let,  $t_i = \ln \left( \frac{P_{E,i}(1-p_w)}{(1-P_{E,i})p_w} \right)$  and  $k = \left( \frac{1-p_e}{lp_e} \right)$ .

Therefore,  $t_i = \ln \left( \frac{(p_w + p_i d)(1 - p_w)}{p_w(1 - p_w - p_i d)} \right) = \ln \left( 1 + \frac{p_i d}{p_w(1 - p_w - p_i d)} \right)$ .

$$\begin{aligned}
\text{Let, } t &= \ln\left(\frac{p_n l_n}{p_e l_e}\right) + \ln\left(\frac{1-p_w}{1-P_{E,1}}\right) + 4\ln\left(\frac{1-p_w}{1-P_{E,2}}\right) + 4\ln\left(\frac{1-p_w}{1-P_{E,3}}\right) \\
&= \ln(k) + \ln\left(\frac{1-p_w}{1-p_w-p_1 d}\right) + 4\ln\left(\frac{1-p_w}{1-p_w-p_2 d}\right) + 4\ln\left(\frac{1-p_w}{1-p_w-p_3 d}\right) \\
&= \ln\left[k\left(1 + \frac{p_1 d}{1-p_w-p_1 d}\right)\left(1 + \frac{p_2 d}{1-p_w-p_2 d}\right)^4\left(1 + \frac{p_3 d}{1-p_w-p_3 d}\right)^4\right],
\end{aligned}$$

### 6.5.1 Calculation of errors for each sensor and Observations

In this subsection, we calculate different error probabilities for the sensors, for some specific values of parameters. We choose two set of values of  $p_1, p_2, p_3, p_w$  and  $p_c$ , one is for a good reliable network and other is for a less reliable network. We choose five different values of  $p_e$ . These values are chosen just to give an idea of the errors and the tests. One can easily calculate different error probabilities for any other values of the parameters.

*Observations:* A few immediate observations from the theoretical results (which can be verified from Table 6.1) are as follows:

1.  $P_{1,i}$  decreases when  $p_i$  and  $p_c$  increase and independent of  $p_j$  when  $i \neq j$ .
2.  $P_{1,i}$  increases when  $p_e$  and  $p_w$  increase.
3.  $P_{2,i}$  decreases when  $p_i, p_e$  and  $p_c$  increase and independent of  $p_j$  when  $i \neq j$ .
4.  $P_{2,i}$  increases when  $p_w$  increases.
5. If detection probability  $p_i$  is low then type I error is close to 1. In that case, the network is not reliable.
6. If  $p_e$  is high but  $p_i$  is small then other types of errors are high; that means, there is no use of sensors. So when  $p_e$  is high, we have to use sensors with high detection probability (i.e. better quality sensors).

Table 6.1: Calculation of errors for some values of the parameters

Calculation of type I error							
$p_1$	$p_2$	$p_3$	$p_c$	$p_w$	$Q_{E,1}$	$Q_{E,2}$	$Q_{E,3}$
0.9	0.5	0.3	0.9	0.1	0.1800	0.5000	0.6600
0.7	0.3	0.1	0.8	0.2	0.3800	0.6200	0.7400
When $p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$							
$p_e$	$P_{1,1}$	$P_{1,2}$	$P_{1,3}$	$P_{2,1}$	$P_{2,2}$	$P_{2,3}$	
0.1	0.0217	0.0581	0.0753	0.5233	0.6429	0.7258	
0.2	0.0476	0.1220	0.1550	0.3279	0.4444	0.5405	
0.3	0.0789	0.1923	0.2391	0.2215	0.3182	0.4070	
0.4	0.1176	0.2702	0.3284	0.1546	0.2308	0.3061	
0.5	0.1667	0.3571	0.4231	0.1087	0.1667	0.2273	
When $p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$							
0.1	0.0501	0.0793	0.0932	0.7438	0.8257	0.8738	
0.2	0.1061	0.1623	0.1878	0.5634	0.6780	0.7547	
0.3	0.1691	0.2493	0.2839	0.4294	0.5511	0.6422	
0.4	0.2405	0.3407	0.3814	0.3261	0.4412	0.5357	
0.5	0.3220	0.4366	0.4805	0.2439	0.3448	0.4348	

7. When  $p_e$  and  $p_w$  are small then

$$P_{1,i} = \frac{p_e(1 - p_w - p_i d)}{1 - p_w - p_e p_i d} = p_e \left(1 - \frac{p_i d}{1 - p_w}\right) \left(1 - \frac{p_e p_i d}{1 - p_w}\right)^{-1}$$

$$\approx p_e \left(1 - \frac{p_i d}{1 - p_w}\right) \left(1 + \frac{p_e p_i d}{1 - p_w}\right) \approx p_e \left(1 - \frac{p_i d}{1 - p_w} + \frac{p_e p_i d}{1 - p_w}\right).$$

## 6.5.2 Calculation for Bayes Test and Observations

Let,  $x_1 = x_{11}$  and  $x_i = x_{i1} + x_{i2} + x_{i3} + x_{i4}$  for  $i = 2, 3$ .

The Bayes test is to reject  $H_0$  when

$$\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 < 1,$$

where,  $\lambda_i = t_i/t$  with

$$t_i = \ln \left(1 + \frac{p_i d}{p_w(1 - p_w - p_i d)}\right), i = 1, 2, 3.$$

and  $t =$

$$\ln \left[ k \left( 1 + \frac{p_1 d}{1 - p_w - p_1 d} \right) \left( 1 + \frac{p_2 d}{1 - p_w - p_2 d} \right)^4 \left( 1 + \frac{p_3 d}{1 - p_w - p_3 d} \right)^4 \right],$$

where  $k = \left( \frac{1-p_e}{lp_e} \right)$ .

Note that we always accept  $H_0$  if  $t \leq 0$

$$\text{As } \ln \left( 1 + \frac{p_i d}{1 - p_w - p_i d} \right) > 0, \text{ for all } i,$$

$t$  is negative if  $\ln \left( \frac{lp_e}{1-p_e} \right) >$

$$\ln \left( 1 + \frac{p_1 d}{1 - p_w - p_1 d} \right) + 4 \ln \left( 1 + \frac{p_2 d}{1 - p_w - p_2 d} \right) + 4 \ln \left( 1 + \frac{p_3 d}{1 - p_w - p_3 d} \right),$$

i.e., if  $l >$

$$\left( \frac{1 - p_e}{p_e} \right) \left( 1 + \frac{p_1 d}{1 - p_w - p_1 d} \right) \left( 1 + \frac{p_2 d}{1 - p_w - p_2 d} \right)^4 \left( 1 + \frac{p_3 d}{1 - p_w - p_3 d} \right)^4.$$

In this case, the Bayes test is not applicable, i.e., if  $l$  (ratio of the losses) is large then we have to use good quality sensors, i.e., sensor with high detection probabilities such that  $\frac{p_1 d}{1 - p_w - p_1 d}$  is so large that  $l <$

$$\left( \frac{1 - p_e}{p_e} \right) \left( 1 + \frac{p_1 d}{1 - p_w - p_1 d} \right) \left( 1 + \frac{p_2 d}{1 - p_w - p_2 d} \right)^4 \left( 1 + \frac{p_3 d}{1 - p_w - p_3 d} \right)^4.$$

*Observations:* A few immediate observations from the theoretical results (which can be verified from Table 6.2) are as follows:

1.  $\lambda_i$ 's are the weights of  $x_i$ 's in the Bayes test which means that the value of  $\lambda_i$  tell us how much weight the base station has to give to  $x_i$  while taking the decision about the event square, e.g., consider the Bayes test for  $p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9, p_e = 0.1$  and  $l = 5$ ,

$$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 5.789,$$

Here,  $\lambda_1 : \lambda_2 : \lambda_3 \approx 5 : 3 : 2$  means 3 distance-two sensors is equivalent

Table 6.2: Calculation of Bayes test for some values of the parameters

$p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$			
$p_e$	$l$	$t$	Bayes test
0.1	5	5.789	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 5.789$
0.3	5	4.439	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 4.439$
0.5	5	3.592	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 3.592$
0.1	20	4.403	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 4.403$
0.3	20	3.053	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 3.053$
0.5	20	2.205	$3.714x_1 + 2.197x_2 + 1.534x_3 \leq 2.205$
$p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$			
0.1	5	2.664	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq 2.664$
0.3	5	1.314	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq 1.314$
0.5	5	0.466	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq 0.466$
0.1	20	1.277	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq 1.277$
0.3	20	-0.073	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq -0.073$
0.5	20	-0.920	$1.876x_1 + 0.897x_2 + 0.340x_3 \leq -0.920$

to 2 distance-one sensors in the context of detecting an event and so on.

2. Let  $\lambda_i/\lambda_k = t_i/t_k$  be the ratio of the weights which tells us how many  $N_{k,j}$  nodes are equivalent to one  $N_{i,j}$  in the context of detecting an event.
3.  $t$  increases when  $p_i$  and  $p_c$  increase.
4.  $t$  decreases when  $p_w, l$  and  $p_e$  increase.
5. For  $p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8, l = 20$  and  $p_e = 0.3$  (resp. 0.5), the Bayes test is (Table 6.2) to reject  $H_0$  when  $1.876x_1 + 0.897x_2 + 0.340x_3 \leq -0.0734$  (resp.  $1.876x_1 + 0.897x_2 + 0.340x_3 \leq -0.920$ ), i.e., we accept  $H_0$  for all values of  $x_i$ 's. So in this situation the Bayes test is not applicable. This indicates that for small values of  $p_i$ 's and large values of  $l$  the Bayes test is not applicable. When  $l$  is large we have to use sensors with high detection probabilities to decide about the event square using Bayes test.

### 6.5.3 Calculation for MP Test and Observations

The most powerful (MP) test of size  $\alpha$  is to reject  $H_0$  when

$$t_1x_1 + t_2x_2 + t_3x_3 < \lambda$$

and reject  $H_0$  with probability  $k$ , when equality holds in place of less than,

$$\text{where, } t_i = \ln \left( 1 + \frac{p_i d}{p_w(1 - p_w - p_i d)} \right), i = 1, 2, 3.$$

$\lambda$  and  $k$  can be found from the relation

$$\Pr(t_1x_1 + t_2x_2 + t_3x_3 < \lambda) + k \Pr(t_1x_1 + t_2x_2 + t_3x_3 = \lambda) = \alpha.$$

Note that,  $x_1$  follows  $\text{Ber}(p_w + p_1d)$  and  $x_i$  follows  $\text{Bin}(4, p_w + p_id)$  for  $i = 2, 3$ , where  $\text{Bin}(n, p)$  is the Binomial distribution with parameters  $n$  and  $p$ . Also  $x_i$ 's are independent when it is known that  $H_0$  is true.

To simplify calculations, we take the approximate values of  $t_1 : t_2 : t_3$  and  $p_w + p_id$ . For  $p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$ , we take  $t_1 : t_2 : t_3 \approx 5 : 3 : 2$  and  $p_w + p_1d \approx 0.8, p_w + p_2d \approx 0.5$  and  $p_w + p_3d \approx 0.35$ . And for  $p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$ , we take  $t_1 : t_2 : t_3 \approx 10 : 5 : 2$  and  $p_w + p_1d \approx 0.6, p_w + p_2d \approx 0.4$  and  $p_w + p_3d \approx 0.25$ .

To calculate  $\lambda$  and  $k$ , we first set  $\lambda = 0$  and calculate the probability of  $t_1x_1 + t_2x_2 + t_3x_3 < \lambda$ ,

if the probability is less than  $\alpha$ , we increase the value of  $\lambda$  by 1 and do the same as above. If, for  $\lambda = \lambda'$  the probability is less than  $\alpha$ , and for  $\lambda > \lambda'$  the probability is greater than  $\alpha$ , we take that  $\lambda'$  as the value of  $\lambda$ , and then, calculate

$$k = \frac{\alpha - \Pr(X < \lambda')}{\Pr(X \leq \lambda') - \Pr(X < \lambda')},$$

where,  $X = t_1x_1 + t_2x_2 + t_3x_3$ .

*Observations:* A few immediate observations from the theoretical results (which can be verified from Table 6.3) are as follows:

Table 6.3: Calculation of MP test for some choice of the parameters

$p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$			
$\alpha$	$\lambda$	$k$	MP test
0.900	8	0.04	$5x_1 + 3x_2 + 2x_3 \leq 8$
0.950	6	0.15	$5x_1 + 3x_2 + 2x_3 \leq 6$
0.975	5	0.19	$5x_1 + 3x_2 + 2x_3 \leq 5$
0.990	3	0.33	$5x_1 + 3x_2 + 2x_3 \leq 3$
$p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$			
$\alpha$	$\lambda$	$k$	MP test
0.900	7	0.24	$10x_1 + 5x_2 + 2x_3 \leq 7$
0.950	5	0.02	$10x_1 + 5x_2 + 2x_3 \leq 5$
0.975	2	0.39	$10x_1 + 5x_2 + 2x_3 \leq 2$
0.990	0	0.61	$10x_1 + 5x_2 + 2x_3 \leq 0$

1.  $t_i$ 's are the weights of the  $x_i$ 's in the MP test

$$5x_1 + 3x_2 + 2x_3 \leq 8, \text{ here } \lambda_1 : \lambda_2 : \lambda_3 \approx 5 : 3 : 2,$$

roughly means 3 distance-two sensors is equivalent to 2 distance-one sensors in the context of detecting event and so on.

2.  $t_i$  is independent of  $p_e$  and  $l$ .
3.  $t_i$  increases when  $p_i$  and  $p_c$  increase.
4.  $t_i$  decreases when  $p_w$  increases.
5. As  $t_i$  increases, critical region (set of all the values of  $x_i$ 's for which we reject the null hypothesis) is going to be smaller.
6. Let  $\lambda_i/\lambda_k = t_i/t_k$  be the ratio of the weights which tells us how many  $N_{kj}$  node are equivalent to one  $N_{ij}$  in the context of detecting an event.
7. For  $p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$  and  $\alpha = 0.990$  the MP test is reject  $H_0$  when  $10x_1 + 5x_2 + 2x_3 \leq 0$ , i.e., we accept  $H_0$  in almost all cases. So in this situation, the MP test is not applicable. This indicates that for small values of  $p_i$ 's and large values of  $\alpha$ , the MP test is not applicable. When  $\alpha$  is large we cannot use sensors with small  $p_i$  values for the MP test. Hence, when the size of the MP test is small we have to use better sensors for MP test.



## 6.6 Simulation Results

Table 6.4: Simulated and theoretical values of errors

Simulation of type I error					
$Q_{E,1}$	$q_{E,1}$	$Q_{E,2}$	$q_{E,2}$	$Q_{E,3}$	$q_{E,3}$
0.1800	0.1791	0.5000	0.4964	0.6600	0.6574
0.3800	0.3823	0.6200	0.6173	0.7400	0.7398
Simulation of other type of error					
$P_{1,1}$	$p_{1,1}$	$P_{1,2}$	$p_{1,2}$	$P_{1,3}$	$p_{1,3}$
0.0217	0.0214	0.0581	0.0570	0.0753	0.0738
0.0476	0.0469	0.1219	0.1203	0.1549	0.1546
0.0789	0.0789	0.1923	0.1918	0.2391	0.2387
0.1176	0.1176	0.2703	0.2690	0.3283	0.3275
0.1667	0.1677	0.3571	0.3546	0.4231	0.4235
0.0501	0.0514	0.0793	0.0808	0.0932	0.0964
0.1061	0.1039	0.1623	0.1585	0.1878	0.1833
0.1691	0.1677	0.2493	0.2499	0.2839	0.2831
0.2405	0.2402	0.3407	0.3366	0.3814	0.3782
0.3220	0.3234	0.4366	0.4380	0.4805	0.4813
Simulation of another type of error					
$P_{2,1}$	$p_{2,1}$	$P_{2,2}$	$p_{2,2}$	$P_{2,3}$	$p_{2,3}$
0.5233	0.5262	0.6429	0.6433	0.7286	0.7258
0.3279	0.3298	0.4444	0.4492	0.5405	0.5337
0.2215	0.2195	0.3182	0.3200	0.4070	0.4083
0.1546	0.1561	0.2308	0.2370	0.3061	0.3080
0.1087	0.1106	0.1667	0.1684	0.2273	0.2225
0.7438	0.7399	0.8257	0.8196	0.8738	0.8739
0.5634	0.5566	0.6780	0.6731	0.7547	0.7512
0.4294	0.4299	0.5512	0.5550	0.6422	0.6393
0.3261	0.3230	0.4412	0.4382	0.5357	0.5349
0.2439	0.2490	0.3448	0.3420	0.4348	0.4385

For  $m = 200, n = 250$ , we simulate the different probabilities and the tests of the problem. We also simulate the number of times the Bayes test and the MP test give the correct decision. The simulation is performed using a C-program with required random numbers generated using the standard C-library.

In the following table,  $q_{E,i}$  denotes the corresponding simulated values of  $Q_{E,i}$  and  $p_{k,i}$  denotes the corresponding simulated values of  $P_{k,i}$  where  $k = 1, 2$  and  $i = 1, 2, 3$ .

*Observations:* A few immediate observations from the theoretical results (which can also be seen in Table 6.4 and Table 6.5) are as follows:

1. The simulated and theoretical values of the different errors are close enough; they differ by at most 2%. The simulated and theoretical values of type I error of the MP test are differ by at most 0.3%.
2. The powers of the Bayes and MP test are very close for the same type I error, e.g., for  $p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$  type I error for the MP test and the Bayes test are 0.9014 and 0.9016, respectively, and the corresponding type II errors are 0.9437 and 0.9436. This indicates that both test are good and equally powerful. In case of the MP test, the type I error (i.e.,  $1 - \alpha$ ) has to be chosen before the test, but in case of Bayes test if the ratio of the losses is known then type I error is automatically fixed and the overall loss is minimized.

Table 6.5: Simulation of proportion of correct detections by Bayes and MP tests

$p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$			
$p_e$	$l$	1- type I error (simulated)	Power (simulated)
0.1	5	0.9016	0.9436
0.3	5	0.9417	0.9131
0.5	5	0.9819	0.7554
0.1	20	0.9465	0.9090
0.3	20	0.9866	0.7358
0.5	20	0.9931	0.7291
$p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$			
0.1	5	0.5927	0.8537
0.3	5	0.8520	0.5963
0.5	5	0.9610	0.2682
0.1	20	0.8367	0.6001
0.3	20	1.0000	0.0
0.5	20	1.0000	0.0
$p_1 = 0.9, p_2 = 0.5, p_3 = 0.3, p_w = 0.1, p_c = 0.9$			
MP test	$\alpha$	1- type I error (simulated)	Power (simulated)
	0.900	0.9014	0.9437
	0.950	0.9500	0.8814
	0.975	0.9767	0.7897
	0.990	0.9906	0.6231
$p_1 = 0.7, p_2 = 0.3, p_3 = 0.1, p_w = 0.2, p_c = 0.8$			
	0.900	0.8875	0.4985
	0.950	0.9458	0.3296
	0.975	0.9735	0.1856
	0.990	0.9891	0.0882

## 6.7 Some Special Cases

With fewer sensors, e.g., only center node or when sensors always send the message correctly with probability 1, we can simplify the error probabilities and the tests. In this subsections we discuss some special cases of that nature.

### 6.7.1 When sensors always send the message correctly

We have,  $p_w = 0$  and  $p_c = 1$ . Hence,  $d = 1$  since  $d = p_c - p_w$ .

The type I and type II errors for  $N_{ij}$  are  $1 - p_i$  and 0, respectively, and

$$P_{1,i} = \frac{p_e(1 - p_i)}{1 - p_e p_i} \approx p_e(1 - p_i)(1 + p_e p_i) \approx p_e(1 - p_i),$$

for small  $p_e$  and  $P_{2,i} = 0$  for all possible values of  $i, j$ . If

$$\left(\frac{p_n}{p_e l}\right) \left(\frac{1}{1 - p_1}\right) \left(\frac{1}{1 - p_2}\right)^4 \left(\frac{1}{1 - p_3}\right)^4 > 1,$$

then the Bayes test is to reject  $H_0$  only when  $x_1 = x_2 = x_3 = 0$ . Otherwise, the Bayes test is to accept  $H_0$  for all values of  $x_1, x_2, x_3$ . Therefore, if

$$l > \frac{p_n}{p_e} (1 - p_1)^{-1} (1 - p_2)^{-4} (1 - p_3)^{-4},$$

the Bayes test is to reject  $H_0$  when  $x_1 = x_2 = x_3 = 0$ . Otherwise, the Bayes test is to accept  $H_0$  for all values of  $x_1, x_2, x_3$ . If  $(1 - p_1)(1 - p_2)^4(1 - p_3)^4 \leq \alpha$ , then the MP test of size  $\alpha$  is to reject  $H_0$  when  $x_1 = x_2 = x_3 = 0$ . Otherwise, the MP test of size  $\alpha$  is to accept  $H_0$  for all values of  $x_1, x_2, x_3$ .

### 6.7.2 When the sensors can detect the event square without any error

We have the detection probability is 1, i.e.,  $p_i = 1$  for  $i = 1, 2, 3$ . Then the type I and type II error for  $N_{i,j}$  are  $1 - p_c$  and  $p_w$ , respectively.

$$P_{1,i} = \frac{p_e(1 - p_w - d)}{1 - p_w - p_e d} \text{ and } P_{2,i} = \frac{(1 - p_e)p_w}{p_w + p_e d}$$

for all possible values of  $i$  and  $j$ .

The Bayes test is to reject  $H_0$  when

$$x_1 + x_2 + x_3 < \frac{\ln\left(\frac{1-p_e}{lp_e}\right) + 9\ln\left(1 + \frac{d}{1-p_c}\right)}{\ln\left(1 + \frac{d}{p_w(1-p_c)}\right)}.$$

The MP test of size  $\alpha$  is to reject  $H_0$  when  $x_1 + x_2 + x_3 < \lambda$  and reject  $H_0$  with probability  $k$ , when  $x_1 + x_2 + x_3 = \lambda$ .

$\lambda$  and  $k$  can be found from the relation  $\Pr(x_1 + x_2 + x_3 < \lambda) + k\Pr(x_1 + x_2 + x_3 = \lambda) = \alpha$ , where  $x_1 + x_2 + x_3$  follows  $\text{Bin}(9, p_c)$ .

### 6.7.3 When one center and four distance-one sensors can detect the event square, i.e., $p_3 = 0$

If sensors have less sensing power (i.e., small sensing radius), then small numbers of sensors can detect the event square. Assume that only center and four adjacent sensors can detect the event square. Then, we consider only five squares: one center square and four one-distanced adjacent squares.

The Bayes test is to reject  $H_0$  when  $\lambda_1 x_1 + \lambda_2 x_2 < 1$ , where, for  $i = 1, 2$ ,

$$\lambda_i = \frac{\ln\left(1 + \frac{p_i d}{p_w(1-p_w-p_i d)}\right)}{\ln\left(\frac{1-p_e}{lp_e}\right) + \ln\left(1 + \frac{p_1 d}{1-p_w-p_1 d}\right) + 4\ln\left(1 + \frac{p_2 d}{1-p_w-p_2 d}\right)}.$$

The MP test of size  $\alpha$  is to reject  $H_0$  when  $t_1 x_1 + t_2 x_2 < \lambda$  and reject  $H_0$  with probability  $k$ , when  $t_1 x_1 + t_2 x_2 = \lambda$ ,

$$\text{where, } t_i = \ln\left(1 + \frac{p_i d}{p_w(1-p_w-p_i d)}\right), i = 1, 2.$$

$\lambda$  and  $k$  can be find from the relation

$$\Pr(t_1 x_1 + t_2 x_2 < \lambda) + k\Pr(t_1 x_1 + t_2 x_2 = \lambda) = \alpha,$$

where  $x_1$  follows  $\text{Ber}(p_w + p_1 d)$ ,  $x_2$  follows  $\text{Bin}(4, p_w + p_2 d)$  and they are independent when  $H_0$  is true.

## 6.8 When sensors are placed at the centers of regular hexagons

In this section, we assume the ROI is partitioned into congruent regular hexagons (which are known as cells) with side  $a$ , i.e., we can think ROI as a hexagonal grid with regular hexagonal cells. We consider that sensors are placed at the center of each cell of the hexagonal grid. We assume that the sensor network covers the entire ROI. Instead of three detection probabilities (as in the case of square grid), we assume there are two detection probabilities  $p_1, p_2$ , where  $p_1 > p_2$ . Note that there are six adjacent nodes of a particular node, see Fig. 3.2. Hexagonal grid is better in the sense that less number of sensors is required to cover the entire ROI, but square grid is mostly used in literature.

We define  $N_{i,j}, x_{ij}, y_{ij}, x_i, p_e, p_n, p_c, p_w, t_i, l$  as in the case of square grid. Hypotheses are also same as in the case of square grid. The Bayes test is to reject  $H_0$  when

$$\lambda_1 x_1 + \lambda_2 x_2 < 1, \text{ where, } \lambda_i = \frac{\ln \left( 1 + \frac{p_i d}{p_w(1-p_w-p_i d)} \right)}{\ln(c)}, \text{ for } i = 1, 2;$$

$$\text{with } c = \left( \frac{1-p_e}{lp_e} \right) \left( 1 + \frac{p_1 d}{1-p_w-p_1 d} \right) \left( 1 + \frac{p_2 d}{1-p_w-p_2 d} \right)^6.$$

The MP test of size  $\alpha$  is to reject  $H_0$  when  $t_1 x_1 + t_2 x_2 < \lambda$  and reject  $H_0$  with probability  $k$ , when  $t_1 x_1 + t_2 x_2 = \lambda$ ,

$$\text{where, } t_i = \ln \left( 1 + \frac{p_i d}{p_w(1-p_w-p_i d)} \right), i = 1, 2.$$

$\lambda$  and  $k$  can be found from the relation

$$\Pr(t_1 x_1 + t_2 x_2 < \lambda) + k \Pr(t_1 x_1 + t_2 x_2 = \lambda) = \alpha,$$

$x_1$  follows Ber  $(p_w + p_1 d)$ ,  $x_2$  follows Bin  $(6, p_w + p_2 d)$  and  $x_1$  and  $x_2$  are independent under  $H_0$ .

We find the Bayes test is to reject  $H_0$  when

$$t_1x_1 + t_2x_2 + t_3x_3 < t,$$

$$\text{where, } t_i = \ln \left( 1 + \frac{p_i d}{p_w(1 - p_w - p_i d)} \right), i = 1, 2, 3;$$

$$t = \ln \left( \frac{1 - p_e}{lp_e} \right) + \ln \left( 1 + \frac{p_1 d}{1 - p_w - p_1 d} \right) +$$

$$4\ln \left( 1 + \frac{p_2 d}{1 - p_w - p_2 d} \right) + 4\ln \left( 1 + \frac{p_3 d}{1 - p_w - p_3 d} \right)$$

and  $\lambda_i = t_i/t$ .

We observed  $t$  increases when  $p_i$  and  $p_c$  increase.  $t$  decreases when  $p_w$ ,  $l$  and  $p_e$  increase. For small values of  $p_i$ 's and large values  $l$  the Bayes test is not applicable. When  $l$  is large we cannot use sensors with small  $p_i$  values.

The most powerful (MP) test of size  $\alpha$  is reject  $H_0$  when

$$t_1x_1 + t_2x_2 + t_3x_3 < \lambda$$

and reject  $H_0$  with probability  $k$ , when

$$t_1x_1 + t_2x_2 + t_3x_3 = \lambda,$$

$$\text{where, } t_i = \ln \left( 1 + \frac{p_i d}{p_w(1 - p_w - p_i d)} \right), i = 1, 2, 3.$$

$\lambda$  and  $k$  can be find from the relation  $\Pr(X < \lambda) + k \Pr(X = \lambda) = \alpha$ , where,  $X = t_1x_1 + t_2x_2 + t_3x_3$  and  $x_1$  follows  $\text{Ber}(p_w + p_1d)$ , and  $x_i$  follows  $\text{Bin}(4, p_w + p_i d)$  for  $i = 2, 3$ .

Also  $x_i$ 's are independent when it is known that event occurs or not, i.e.,  $H_0$  is true or not.

We observe  $t_i$  independent of  $p_e$  and  $l$ .  $t_i$  increase when  $p_i$  and  $p_c$  increase.  $t_i$  decrease when  $p_w$  increase. When  $t_i$  increase critical region (set of all the values of  $x_i$ 's for which we reject the null hypothesis) decrease. For small values of  $p_i$ 's and large values  $\alpha$  the MP test is not suitable. When  $\alpha$  is large we cannot use sensors with small  $p_i$  values.

# Detection of event in faulty network system

## 7.1 Introduction

In this chapter, we are interested in one particular query: determining event in the environment (i.e., ROI) with a distinguishable characteristic. We assume the ROI to be partitioned into suitable number of congruent regular hexagonal cells (i.e., we can think ROI as a regular hexagonal grid). This physical structure of ROI is not a requirement for the theoretical analysis, we can do similar analysis with other structures also. Suppose that sensors are placed a priori at the center (which are known as nodes) of every hexagon of the grid. We assume that the sensors are connected to its adjacent sensor nodes in the sense that a hexagon will be strongly covered by its center node and weakly covered by the adjacent nodes. If event occurs in the hexagon where a particular sensor lies, then that particular sensor can detect the event with a greater probability whereas, if event occurs in any adjacent hexagon, then the particular sensor can detect the event with a lesser probability. Hence, only one node (center node of the event hexagon) can detect the event hexagon with greater probability, say  $p_1$ , and adjacent nodes (six for interior nodes and less for boundary nodes) can detect the event hexagon with lesser probability, say  $p_2$ , with  $p_1 > p_2$ . We assume that no other sensor can detect the event hexagon. We also assume that if the event occurs then it occurs at only one hexagon of

the grid which will be known as event hexagon and there is no fusion sensor. All sensors can communicate with the base station and the base station takes the decision about the query. In previous chapter we assume only one event can occur at a particular hexagon (event hexagon) and all other hexagons are normal, in this chapter we assume at most one event but this may be occurs at any hexagon, not merely at a particular hexagon.

One fundamental challenge in the event detection problem for a sensor network is the detection accuracy which is disturbed by the noise associated with the detection and the reliability of sensor nodes. A sensor may fail to detect the event due to natural obstruction or any other causes. After detecting the event, a sensor can send false message to the base station due to some technical reasons.

## **7.2 Statement of the Problem and Assumptions**

In this section, we describe the problem in more specific terms and state the assumptions that we make.

Sensors are deployed, or manually placed, over the ROI to perform event detection (i.e., to detect whether an event of interest has happened or not) in the ROI. If sensors are deployed from air then, using actuator-assisted sensor placement or by movement-assisted sensor placement, sensors are so placed that sensor network covers the entire ROI. This ROI is partitioned into suitable number of regular hexagons (i.e., we can think of the ROI as a regular hexagonal grid), as shown in Figure 3.2. Sensors are placed, a priori, at every center (which are known as nodes) of the regular hexagons. Sensors have two detection probabilities. The sensor network covers the entire ROI and there is only one event hexagon, as discussed before.

Each sensor node determines its location through beacon positioning mechanisms [7] or by exploiting the Global Positioning System (GPS). Through a broadcast or acknowledge protocol, each sensor node is also able to locate the neighbors within its communication radius. Sensors are also able to communicate with the base station. The base station takes the decision by combining



the information received from all the sensors.

There are two phases in the whole process. The first one is detection phase, when the sensor at the center of a regular hexagon tries to detect the event. The sensor at the center of the event hexagon can detect the event hexagon with greater probability  $p_1$  and the sensors at the adjacent nodes (see Figure 3.2) can detect the event hexagon with lesser probability  $p_2$ . We also assume that there is a prior probability that a particular hexagon is an event hexagon. The next phase is response phase, in which sensors send message to the base station. Even if the event hexagon is detected by a faulty sensor, it may not respond with some positive probability. Conversely, if an event hexagon is not detected, or there is no event hexagon at all (i.e., ROI is normal), then also a faulty sensor can send the wrong information to the base station with some probability.

Each sensor sends information to the base station. As the sensors may send wrong information, the base station takes the important role in identifying the event hexagon. The base station will collect all the information and take a decision about the event hexagon according to a rule which we have to find out. Our job is to find a rule for the base station such that base station works most efficiently.

### 7.2.1 Notations and Assumption

Our problem is to develop a strategy for the base station to take decision about the event hexagon (i.e., which hexagon of the ROI is the event hexagon, if at all). Let  $R$  be the set of all nodes and  $R'$  be the set of all interior nodes (a node which has six adjacent nodes). For  $N \in R$ , define  $B(N)$ , as the set of adjacent node(s) of  $N$  and let  $k(N)$  be the number of adjacent node(s) of  $N$ . Hence,  $0 \leq k(N) \leq 6$ . Call a node  $N$  interior if  $k(N) = 6$ . Let  $S_N$  be the sensor that is placed at the node  $N$  and  $H_N$  be the hexagon where the node  $N$  is placed (i.e.,  $N$  is the center of  $H_N$ ). For  $N \in R$ , let  $X_N$  denote the true status of the node  $N$ . That is,  $X_N = 1$  if event occurs at  $H_N$ , and 0 otherwise. Also define  $Y_N = 0$  if  $S_N$  detects no event, and 1 if  $S_N$  detects the event in  $H_N$  or  $H_{N'}$ , for  $N' \in B(N)$ . Finally define  $Z_N = 0$  if  $S_N$  does not respond, i.e., the sensor informs the base station that event does not occur at  $H_N$  or

$H_{N'}$  for  $N' \in B(N)$ , and  $Z_N = 1$  if  $S_N$  responds, i.e., the sensor  $S_N$  informs the base station that the event has occurred in  $H_N$  or  $H_{N'}$ , for  $N' \in B(N)$ .

Now we make one natural assumption that, once the detection phase is completed, the response of a sensor depends only on what it detects but not on whether the event has actually occurred or not, i.e.,  $P(Z_N = k|Y_N, X_N) = P(Z_N = k|Y_N)$ , for  $k = 0, 1$ . We also assume that the sensors work independently and identically.

Since we assume that there is at most one event hexagon,  $\sum_{N \in R} X_N = 1$  or 0.

The possible true scenarios are, therefore, represented by the following  $|R| + 1$  different models:

$$\mathcal{M}_0 : (X_N = 0 \text{ for all } N \in R),$$

and, for each  $N \in R$ ,

$$\mathcal{M}_N : (X_N = 1 \text{ and } X_{N'} = 0 \text{ for all } N' \in R \setminus N).$$

Let  $\Pr(\mathcal{M}_0) = P(\text{ROI is normal}) = p_{norm}$

and, for all  $N \in R$ ,  $\Pr(\mathcal{M}_N) = \Pr(\text{event occurs at the hexagon } H_N) = p_N$ .

In particular, we may assume  $p_N$ 's to be same for all  $N$ . We denote any probability under the model  $\mathcal{M}_0$  as  $P_{\mathcal{M}_0}(\cdot)$  and under the model  $\mathcal{M}_N$  as  $P_{\mathcal{M}_N}(\cdot)$ .

We also make the followings assumptions:

1. For all  $N \in R$ ,  $P_{\mathcal{M}_0}(Y_N = 1) = 0$  and  $P_{\mathcal{M}_N}(Y_N = 1) = p_1$ .
2. For all  $N' \in B(N)$ ,  $P_{\mathcal{M}_N}(Y_{N'} = 1) = p_2$ , and  
for all  $N' \in R \setminus [B(N) \cup \{N\}]$ ,  $P_{\mathcal{M}_N}(Y_{N'} = 1) = 0$ .
3. For all  $N \in R$ ,  $P(Z_N = 1|Y_N = 1) = p_c$  and  $P(Z_N = 1|Y_N = 0) = p_w$ .
4.  $Z_N$  and  $Y_{N'}$  are independent for  $N \neq N'$ .
5. The responses from different nodes are independent under a particular model, i.e.,  $Z_N$ 's are independent under  $\mathcal{M}_{N'}$  for a fixed  $N' \in R$ .

## 7.3 Theoretical Analysis of Fault Detection

In this section we discuss some theoretical results. In real situations,  $|R|$  may be very large. Given the network of the sensor nodes and some prior knowledge about the nature of event, one may have fairly good idea about the set of feasible regions for the event. Formally, instead of all possible models, one may be able to restrict to a set containing all the feasible models. For example, if the event is known to take place in a particular region, we can restrict our models accordingly.

### 7.3.1 Model Selection Approach

$$\begin{aligned}
 & \text{For all } N \in R, P_{\mathcal{M}_0}(Z_N = 1) \\
 &= P_{\mathcal{M}_0}(Z_N = 1|Y_N = 0)P_{\mathcal{M}_0}(Y_N = 0) + P_{\mathcal{M}_0}(Z_N = 1|Y_N = 1)P_{\mathcal{M}_0}(Y_N = 1) \\
 &= P(Z_N = 1|Y_N = 0)P_{\mathcal{M}_0}(Y_N = 0) + P(Z_N = 1|Y_N = 1)P_{\mathcal{M}_0}(Y_N = 1) = p_w.
 \end{aligned}$$

Hence, under the model  $\mathcal{M}_0$ ,  $Z_N$  follows  $\text{Ber}(p_w)$ , for all  $N \in R$ , and the likelihood of the data  $\{Z_N = z_N, \text{ for all } N \in R\}$ , under the model  $\mathcal{M}_0$ , is

$$\begin{aligned}
 L_0 &= P_{\mathcal{M}_0}(Z_N = z_N, \text{ for all } N \in R) \\
 &= \prod_{N \in R} p_w^{z_N} (1 - p_w)^{(1-z_N)} = (p_w)^{\sum_{N \in R} z_N} \times (1 - p_w)^{\sum_{N \in R} (1-z_N)}.
 \end{aligned}$$

$$\text{So } \ln L_0 = \sum_{N \in R} z_N \ln p_w + \sum_{N \in R} (1 - z_N) \ln(1 - p_w).$$

$$\begin{aligned}
 & \text{For any } N \in R, \text{ we have } P_{\mathcal{M}_N}(Z_N = 1) \\
 &= P_{\mathcal{M}_N}(Z_N = 1|Y_N = 0)P_{\mathcal{M}_N}(Y_N = 0) \\
 &\quad + P_{\mathcal{M}_N}(Z_N = 1|Y_N = 1)P_{\mathcal{M}_N}(Y_N = 1) \\
 &= P(Z_N = 1|Y_N = 0)P_{\mathcal{M}_N}(Y_N = 0) + P(Z_N = 1|Y_N = 1)P_{\mathcal{M}_N}(Y_N = 1) \\
 &= p_w(1 - p_1) + p_c p_1 = p_1(p_c - p_w) + p_w = P_1, \text{ say.}
 \end{aligned}$$

Hence, for all  $N \in R$ , under  $\mathcal{M}_N$ ,  $Z_N$  follows  $\text{Ber}(P_1)$ . Similarly, for all  $N' \in B(N)$ , under  $\mathcal{M}_N$ ,  $Z_{N'}$  follows  $\text{Ber}(P_2)$ , where  $P_2 = p_2(p_c - p_w) + p_w$  and, under  $\mathcal{M}_N$ ,  $Z_{N'}$  follows  $\text{Ber}(p_w)$  for all  $N' \in R \setminus [B(N) \cup \{N\}]$ . Note that  $P_1 > P_2$  since

$p_1 > p_2$ . Hence the likelihood for the model  $\mathcal{M}_N$ , given  $Z_{N'} = z_{N'}, N' \in R$ , is

$$\begin{aligned}
 L_N &= P_{\mathcal{M}_N}(Z_{N'} = z_{N'}, \text{ for all } N' \in R) \\
 &= P_1^{z_N} (1 - P_1)^{(1-z_N)} \prod_{N' \in B(N)} P_2^{z_{N'}} (1 - P_2)^{(1-z_{N'})} \\
 &\quad \times \prod_{N' \in R \setminus [B(N) \cup \{N\}]} p_w^{z_{N'}} (1 - p_w)^{(1-z_{N'})} \\
 &= P_1^{z_N} (1 - P_1)^{(1-z_N)} P_2^{\sum_{N' \in B(N)} z_{N'}} (1 - P_2)^{\sum_{N' \in B(N)} (1-z_{N'})} \\
 &\quad \times p_w^{\sum_{N' \in R \setminus [B(N) \cup \{N\}]} z_{N'}} (1 - p_w)^{\sum_{N' \in R \setminus [B(N) \cup \{N\}]} (1-z_{N'})}.
 \end{aligned}$$

Let  $T_N = \sum_{N' \in B(N)} Z_{N'}$ , so that  $\sum_{N' \in B(N)} (1 - Z_{N'}) = k(N) - T_N$

with the corresponding observed values denoted by

$$t_N = \sum_{N' \in B(N)} z_{N'} \quad \text{and} \quad \sum_{N' \in B(N)} (1 - z_{N'}) = k(N) - t_N.$$

Therefore,  $\ln L_N =$

$$\begin{aligned}
 & z_N \ln P_1 + (1 - z_N) \ln(1 - P_1) + t_N \ln P_2 + (k(N) - t_N) \ln(1 - P_2) \\
 & + \sum_{N' \in R \setminus [B(N) \cup \{N\}]} z_{N'} \ln p_w + \sum_{N' \in R \setminus [B(N) \cup \{N\}]} (1 - z_{N'}) \ln(p_w(1 - p_w)) \\
 & = \ln L_0 + z_N \ln \frac{P_1}{p_w} + (1 - z_N) \ln \frac{1 - P_1}{1 - p_w} + t_N \ln \frac{P_2}{p_w} + (k(N) - t_N) \ln \frac{1 - P_2}{1 - p_w} \\
 & = \ln L_0 + z_N \ln \frac{P_1(1 - p_w)}{p_w(1 - P_1)} + t_N \ln \frac{P_2(1 - p_w)}{p_w(1 - P_2)} + \ln \frac{1 - P_1}{1 - p_w} + k(N) \ln \frac{1 - P_2}{1 - p_w} \\
 & = a + b(cz_N + t_N - dk(N)), \text{ say,}
 \end{aligned}$$

$$\text{where, } a = \ln L_0 + \ln \frac{1 - P_1}{1 - p_w}, b = \ln \frac{P_2(1 - p_w)}{p_w(1 - P_2)} > 0,$$

$$c = \frac{\ln \frac{P_1(1-p_w)}{p_w(1-P_1)}}{\ln \frac{P_2(1-p_w)}{p_w(1-P_2)}} \quad \text{and} \quad d = \frac{\ln \frac{1-p_w}{1-P_2}}{\ln \frac{P_2(1-p_w)}{p_w(1-P_2)}} \text{ are independent of } N.$$

In the model selection approach, the model resulting in the maximum value of the likelihood is selected. Note that, since there is no parameter being estimated, this is equivalent to the well-known Akaike Information Crite-

tion(AIC) [31]. Therefore, the base station will accept the model  $\mathcal{M}_0$  if

$$= \ln \frac{1 - P_1}{1 - p_w} + b(cz_N + t_N - dk(N)) < 0, \text{ for all } N \in R.$$

Otherwise, as  $b$  is positive, accept the model  $\mathcal{M}_N$  for which  $(cz_N + t_N - dk(N))$  is maximum among all  $N \in R$ . If values of  $(cz_N + t_N - dk(N))$  are equal for more than one  $N$ , then we can select one of the corresponding models with equal probability. If we want to maximize the likelihood for the models  $\mathcal{M}_N$  corresponding to the interior nodes only, so that  $k(N)$  is fixed, then we need to maximize  $(cz_N + t_N)$  among all  $N \in R$ .

### 7.3.2 Multiple Model Selection

Instead of selecting one particular model, one may want to select more than one model with approximately similar log likelihood values to the maximum one. We can consider the set of models

$$\{\mathcal{M}_K : \frac{L_K}{\max_{N \in R} L_N} > C\},$$

where  $0 < C < 1$  is a suitable constant close to 1. This  $C$  is usually chosen according to the resource available. This is similar to the idea of Occam's window [64] in the context of Bayesian model selection [36]. This may be interpreted as the interval estimation for the true model.

Note that  $L_N$  is an increasing function of  $cz_N + t_N - dk(N)$ , as  $b$  is positive. We consider only the following set of models

$$\{\mathcal{M}_K : Q_K > C^* \cdot \max_{N \in R} Q_N\},$$

where  $Q_N = cz_N + t_N - dk(N)$ , for all  $N \in R$ , with  $0 < C^* < 1$ . In particular, if we consider the interior nodes only, then we consider the set of models given by

$$\{\mathcal{M}_K : cz_K + t_K > C^* \cdot \max_{N \in R} \{cz_N + t_N\}\}.$$

We can select multiple models using some other criteria. One such may be to select all the models (one or more) for which the maximum value of the likelihood is attained. Let  $\mathcal{N}_{\max}$  be the set of nodes corresponding to all these

models, including ‘ $N = 0$ ’ corresponding to  $\mathcal{M}_0$  if it has the maximum value of the likelihood. Then this method select all the models  $\mathcal{M}_N$  with  $N \in \mathcal{N}_{\max}$ . By another criterion, one may select the models  $\mathcal{M}_{N'}$ , for  $N' \in \mathcal{N}_{\max} \cup [\cup_{N \in \mathcal{N}_{\max}} B(N)]$ ; that is,  $N'$  be a node in  $\mathcal{N}_{\max}$  or any of the neighboring nodes of a node in  $\mathcal{N}_{\max}$ . Note that  $B(N)$  for  $N = 0$  is the empty set. One can combine these two types of criteria and come up with many others.

### 7.3.3 Bayesian Model Averaging

Bayesian model averaging is an effective method to solve a decision problem when there are many alternative hypotheses or models, which are complicated [36]. Suppose  $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k$  are the models considered and  $D$  denotes the given data. The posterior probability for model  $\mathcal{M}_k$  is given by

$$\Pr(\mathcal{M}_k|D) = \frac{\Pr(D|\mathcal{M}_k) \Pr(\mathcal{M}_k)}{\sum \Pr(D|\mathcal{M}_l) \Pr(\mathcal{M}_l)},$$

where  $\Pr(D|\mathcal{M}_k)$  denotes the probability of observing data  $D$  under the model  $\mathcal{M}_k$  (which is essentially the likelihood  $L_k$  under  $\mathcal{M}_k$ ) and  $\Pr(\mathcal{M}_k)$  is the prior probability that  $\mathcal{M}_k$  is the true model (assuming one of the models is true).

In this work, the data  $D$  is  $\{Z_N = z_N : N \in R\}$  and the models are  $\mathcal{M}_0, \mathcal{M}_N, N \in R$  as defined in Section 3.2. Hence, The posterior probability for model  $\mathcal{M}_N$  is

$$\Pr(\mathcal{M}_N|Z_N = z_N, N \in R) = \frac{p_N L_N}{\sum_{l \in R} p_l L_l + p_{norm} L_0},$$

$$\text{and that for } \mathcal{M}_0 \text{ is } \frac{p_{norm} L_0}{\sum_{l \in R} p_l L_l + p_{norm} L_0}.$$

We select the model  $\mathcal{M}_0$  if  $p_{norm} L_0$  is greater than  $p_N L_N$ , for all  $N \in R$ ; otherwise, select  $\mathcal{M}_N$  for which  $p_N L_N$  is maximum among all  $N \in R$ . Hence, if  $p_N$ 's are all equal, then Bayesian approach is same as the likelihood approach.

## 7.4 Some Important Considerations and Error Probabilities

In this section, we consider some important issues related to the problem of fault detection and the proposed methodology including calculation of errors (e.g., false detection, etc.) and detection probabilities.

The following probabilities give some idea about the role of neighboring nodes, along with the center node, in detection, or false detection, of an event. For example,  $P_{\mathcal{M}_0}(T_N = 0, Z_N = 1)$  gives the probability of a false detection by the  $N^{\text{th}}$  node, and not by the neighboring nodes, while  $P_{\mathcal{M}_N}(T_N = 6, Z_N = 0)$  gives the probability of a false negative by the  $N^{\text{th}}$  node, with all the neighboring nodes detecting the event. Since, given a particular model,  $T_N$  and  $Z_N$  are independent, calculation of such probabilities is simple as given in the following. For any  $N \in R$  and  $i = 0, 1, \dots, k(N)$ ,

1.  $P_{\mathcal{M}_0}(T_N = i, Z_N = 0) = \binom{k(N)}{i} p_w^i (1 - p_w)^{k(N)-i+1}$
2.  $P_{\mathcal{M}_0}(T_N = i, Z_N = 1) = \binom{k(N)}{i} p_w^{i+1} (1 - p_w)^{k(N)-i}$
3.  $P_{\mathcal{M}_N}(T_N = i, Z_N = 0) = \binom{k(N)}{i} P_2^i (1 - P_2)^{k(N)-i} (1 - P_1)$
4.  $P_{\mathcal{M}_N}(T_N = i, Z_N = 1) = \binom{k(N)}{i} P_2^i (1 - P_2)^{k(N)-i} P_1.$

$$\begin{aligned} & \text{Note that, for } N \in R, P_{\mathcal{M}_0}(L_N > L_0) = P_{\mathcal{M}_0}(\ln L_N > \ln L_0) = \\ & P_{\mathcal{M}_0} \left( Z_N \ln \frac{P_1(1 - p_w)}{p_w(1 - P_1)} + T_N \ln \frac{P_2(1 - p_w)}{p_w(1 - P_2)} + \ln \frac{1 - P_1}{1 - p_w} + k \ln \frac{1 - P_2}{1 - p_w} > 0 \right) \\ & = P_{\mathcal{M}_0} \left( Z_N \ln \frac{P_1(1 - p_w)}{p_w(1 - P_1)} + T_N \ln \frac{P_2(1 - p_w)}{p_w(1 - P_2)} > k \ln \frac{1 - p_w}{1 - P_2} + \ln \frac{1 - p_w}{1 - P_1} \right), \end{aligned}$$

where,  $k = K(N)$ .

which can be numerically obtained using the joint distribution of  $T_N$  and  $Z_N$  under the model  $\mathcal{M}_0$ . The maximum of these probabilities over all  $N$  gives a lower bound for the probability that a node is considered to be an event node when the ROI is normal. On the other hand, the sum over all  $N$  gives

an upper bound for the same. Similarly, for  $N \in R$ ,  $P_{\mathcal{M}_N}(L_N < L_0) =$

$$P_{\mathcal{M}_N} \left( Z_N \ln \frac{P_1(1-p_w)}{p_w(1-P_1)} + T_N \ln \frac{P_2(1-p_w)}{p_w(1-P_2)} < k(N) \ln \frac{1-p_w}{1-P_2} + \ln \frac{1-p_w}{1-P_1} \right),$$

which can be again numerically obtained using the joint distribution of  $T_N$  and  $Z_N$  under the model  $\mathcal{M}_N$ . This probability gives some idea about the error that, when  $N^{\text{th}}$  node is the event node and it is not detected.

For  $N, N' \in R$ ,  $N \neq N'$ ,  $P_{\mathcal{M}_N}(L_{N'} > L_N)$

$$= P_{\mathcal{M}_N} \left( (Z_{N'} - Z_N) \ln \frac{P_1(1-p_w)}{p_w(1-P_1)} + (T_{N'} - T_N) \ln \frac{P_2(1-p_w)}{p_w(1-P_2)} > (k(N') - k(N)) \ln \frac{1-p_w}{1-P_2} \right).$$

As noted in Section 7.3.1, we select the model  $\mathcal{M}_N$  for which  $Q_N$  is the maximum, for  $N \in R$ . The random variable  $Q_N$  is, therefore, of some interest, the distribution of which under different models is useful in calculating many error probabilities. We first find the distribution of  $Q_N$  under the model  $\mathcal{M}_N$ . Note that  $Q_N$  takes values  $ci + j - dk(N)$ , corresponding to  $Z_N = i$  and  $T_N = j$ , for  $i = 0, 1$ , and  $j = 0, 1, 2, \dots, k(N)$ . Assume that, for convenience, the values of  $Q_N$  for different  $i$  and  $j$  are all distinct. Therefore, for  $i = 0, 1$  and  $j = 0, 1, \dots, k(N)$ ,

$$P_{\mathcal{M}_N}(Q_N = ci + j - dk(N)) =$$

$$\binom{k(N)}{j} (P_1)^i (1-P_1)^{(1-i)} (P_2)^j (1-P_2)^{(k(N)-j)}$$

$$\text{and, } P_{\mathcal{M}_0}(Q_N = ci + j - dk(N)) = \binom{k(N)}{j} (p_w)^{i+j} (1-p_w)^{(1-i+k(N)-j)}.$$

For  $N' \in B(N)$ , or  $N' \in R \setminus [B(N) \cup \{N\}]$ , one can find  $P_{\mathcal{M}_{N'}}(Q_N = ci + j - dk(N))$  in similar manner, although the calculation is very tedious as there are many sub-cases. Ideally, one is interested in probability of errors occurring at the level of base station. For example, the two important errors are: (1) not selecting  $\mathcal{M}_0$  when  $\mathcal{M}_0$  is true (false positive), and (2) selecting  $\mathcal{M}_0$  when  $\mathcal{M}_N$  is true for some  $N \in R$  (false negative). Theoretical calculation of these error probabilities is complicated. We, therefore, use simulation to estimate these and similar error probabilities.



## 7.5 Simulation Study

We consider a  $32 \times 32$  hexagonal grid and we run the programme 10000 times. The simulation is performed using the C-code, and required random numbers are generated using the standard C-library.

In our simulation study, we consider different criteria, as discussed in Sections 7.3.1 and 7.3.2, for estimating the error probabilities, or equivalently, the success rate. First consider the probability of selecting  $\mathcal{M}_0$ , when it is true. Let  $S_1$  denote the proportion of correct detection of the normal situation, when model  $\mathcal{M}_0$  is true, using the model selection method. That is,  $S_1$  gives an estimate of  $P_{\mathcal{M}_0}(0 \in \mathcal{N}_{max} \text{ and } 0 \text{ is selected by randomization})$ . Then  $1 - S_1$  gives an estimate of the false positive rate.

When  $\mathcal{M}_N$  is true for some  $N \in R$ , let  $S_2$  denote the proportion of correct decision for the event node using the model selection method of Section 7.3.1, so that it estimates  $P_{\mathcal{M}_N}(N \in \mathcal{N}_{max} \text{ and is selected by randomization})$ . Note that, for each simulation run, the event hexagon is chosen randomly so that  $S_2$  gives an average value over all  $N$ . In this context, this probability is same for all the interior nodes. Then,  $1 - S_2$  gives an estimate of the corresponding error probability of not selecting  $\mathcal{M}_N$ , when it is true.

Note that, in this problem of fault detection with a single event node, the likelihood value, for a given observed data configuration, may be equal for more than one model. Therefore, quite often, the maximum value of the likelihood may be attained by more than one model. The model selection method of Section 7.3.1, which selects one of these models randomly in such cases, may often not select the correct model. Therefore, the method of Section 7.3.2, which selects more than one model having similar likelihood value, may be preferred and will have a better chance of selecting the correct model. We now consider some of those methods in the following.

Let us first consider the method in which all the models corresponding to the maximum value of the likelihood are selected. Let  $S_3$  denote the proportion of correct selection of the model  $\mathcal{M}_N$ , when it is true, by this method. Then  $S_3$  estimates the probability  $P_{\mathcal{M}_N}(N \in \mathcal{N}_{max})$ , which is always more than or equal to the quantity estimated by  $S_2$ , as remarked before. We also consider the method in which all the models having maximum likelihood along with their

Table 7.1: Values of  $P_1, P_2, c$  and  $d$  for  $p_c = 0.9$

parameters		$p_w = 0.1$				$p_w = 0.2$			
$p_1$	$p_2$	$P_1$	$P_2$	$c$	$d$	$P_1$	$P_2$	$c$	$d$
0.7	0.3	0.66	0.34	1.865	0.202	0.69	0.41	2.139	0.298
	0.4	0.66	0.42	1.526	0.234	0.69	0.48	1.674	0.330
	0.5	0.66	0.50	1.302	0.268	0.69	0.55	1.378	0.363
	0.6	0.66	0.58	1.135	0.302	0.69	0.62	1.166	0.397
0.8	0.3	0.74	0.34	2.114	0.202	0.76	0.41	2.484	0.298
	0.4	0.74	0.42	1.730	0.234	0.76	0.48	1.944	0.330
	0.5	0.74	0.50	1.476	0.268	0.76	0.55	1.600	0.363
	0.6	0.74	0.58	1.287	0.302	0.76	0.62	1.354	0.397
0.9	0.3	0.82	0.34	0.241	0.202	0.83	0.41	2.907	0.298
	0.4	0.82	0.42	1.981	0.234	0.83	0.48	2.275	0.330
	0.5	0.82	0.50	1.690	0.268	0.83	0.55	1.873	0.363
	0.6	0.82	0.58	1.474	0.302	0.83	0.62	1.584	0.397

neighborhood models are selected. A model  $\mathcal{M}_{N'}$  is a neighborhood model of the model  $\mathcal{M}_N$  if  $N'$  is a neighboring node of  $N$ . If  $S_4$  denotes the proportion of correct selection of the model  $\mathcal{M}_N$ , when it is true, by this method, then  $S_4$  estimates  $P_{\mathcal{M}_N}(N \in \mathcal{N}_{\max} \cup \{\cup_{N' \in \mathcal{N}_{\max}} B(N')\})$ . Clearly,  $S_4 \geq S_3 \geq S_2$ . Similarly, if  $S_5$  denotes the proportion of correct selection of the model  $\mathcal{M}_N$ , when it is true, by selecting all those models with likelihood value being more than 90% of the maximum likelihood (that is, the method of Section 7.3.2 with  $C = 0.9$ ) then  $S_5$  estimates the probability  $P_{\mathcal{M}_N}(L_N > 0.9L_{max})$  with  $L_{max}$  denoting the maximum value of the likelihood.

Suppose  $N_i$  denotes the average number of selected nodes to be searched corresponding to  $S_i$ ,  $i = 1, 2, \dots, 5$ . Clearly,  $N_1 = 1 - S_1$  because we need no search when  $\mathcal{M}_0$  is selected. When an event occurs and we consider only one  $N$  from  $\mathcal{N}_{\max}$ , we need at most one search (since no search is needed if  $\mathcal{M}_0$  is selected) and we have  $N_2 \leq 1$ . In our simulation, we find  $N_2 = 1$  in all the cases; that means, in simulation,  $\mathcal{M}_0$  has not been selected when the event occurred. Note that  $N_3 \geq 1$  since we consider all  $N$ 's in  $\mathcal{N}_{\max}$  for searching. Again, as before,  $N_4 > N_3 \geq 1 \geq N_2$ . Also, by definition,  $N_5 \geq 1$ . Table 7.3 presents the different  $S_i$ 's and  $N_i$ 's based on simulation for different values of  $p_1, p_2, p_c$  and  $p_w$  with  $p_1$  and  $p_c$  taking values 0.9 and 0.99,  $p_w$  taking values 0.01 and 0.001 and  $p_2$  taking values 0.0, 0.3, 0.4, 0.5 and 0.6. The choice of  $p_1$  and  $p_c$  reflects the corresponding high probability, whereas that of  $p_w$  reflects small

Table 7.2: Simulation of estimated success probabilities and number of searches for different threshold values ( $C$ ) and some values of the parameters with  $p_c = p_1 = 0.9$

other parameters		$C = 0.6$		$C = 0.7$		$C = 0.8$		$C = 0.9$	
$p_2$	$p_w$	success	search	success	search	success	search	success	search
0.0	0.01	0.81	18.21	0.81	18.25	0.81	18.21	0.81	18.17
0.3	0.01	0.87	13.72	0.78	9.13	0.75	6.64	0.73	5.70
0.4	0.01	0.89	8.86	0.85	6.47	0.82	5.46	0.80	5.12
0.5	0.01	0.93	6.88	0.90	5.69	0.89	5.05	0.86	4.88
0.6	0.01	0.97	5.27	0.96	4.91	0.93	4.04	0.92	3.90
0.0	0.001	0.80	3.27	0.80	3.21	0.80	3.17	0.80	3.18
0.3	0.001	0.91	4.15	0.91	3.65	0.87	3.31	0.86	3.06
0.4	0.001	0.94	4.25	0.94	3.69	0.93	3.31	0.89	3.03
0.5	0.001	0.97	4.24	0.97	3.64	0.96	3.26	0.93	2.96
0.6	0.001	0.99	3.96	0.98	3.18	0.98	3.04	0.96	2.79

probability, which is desirable in a good sensor. Since the primary interest is to study the effect of detection by neighboring nodes, we consider  $p_2$  as 0 (which means there is no effect of neighboring nodes) and some positive values less than  $p_1$ .

Note that the probability of correct detection under  $\mathcal{M}_0$  depends only on  $p_w$ . This is also evident in Table 7.3. Intuitively, if  $p_w$  is high then the proportion  $S_1$  of correct detection in normal situation is low. In Table 7.3, we see that  $S_1$  is 0 for  $p_w = 0.01$ , varies from 0.35 to 0.37 for  $p_w = 0.001$  and varies from 0.90 to 0.91 for  $p_w = 0.0001$  (not shown in Table 7.3). If we consider smaller values of  $p_w$  then the success probability  $S_1$  will be higher. Hence  $p_w$  must be low as the number of hexagons is high to get better results in the normal situation.

We see that the estimated false negative rate, that is an estimate of  $P_{\mathcal{M}_N}(\mathcal{M}_0 \text{ is selected})$ , is often 0 in our simulation (not shown in Table 7.3). This is because, if the event occurs at  $N$ , then detection of the event by at least one of the nodes belonging to  $\{N\} \cup B(N)$  is highly probable. Furthermore, since the grid size is large, one of the nodes belonging to  $R \setminus (\{N\} \cup B(N))$  may respond wrongly, though it cannot detect the event. So, under  $\mathcal{M}_N$ , there is a small probability to select the ROI as normal. If we take  $p_w$  and the detection probabilities  $p_1$  and  $p_2$  to be very small, then we may get some positive false

negative rate but this is not a desired condition for a good sensor.

From simulation, we see that, as  $p_2$  increases (for positive  $p_2$ ),  $S_i$  values increase whereas  $N_i$  decrease. As  $p_2$  increases, it helps to differentiate between the likelihood values resulting in lower cardinality of the set  $\mathcal{N}_{\max}$  and lower values of  $N_i$ 's. However, since the neighboring nodes help to detect the event, the success probability increases. From simulation, we find that, as  $p_1$  increases, success probabilities also increase, but the effect of  $p_2$  is more prominent than that of  $p_1$ . On the other hand, success probabilities also change with  $p_w$  and  $p_c$ . Since  $p_2 = 0$  means  $P_2 = p_w$ , so there is little variability in the likelihood values leading to larger size of  $\mathcal{N}_{\max}$ . To increase the success probability we consider a threshold value  $C$  where  $0 < C < 1$ . If  $L_{max}$  be the maximum likelihood among all the likelihoods, consider the set of nodes having likelihood greater than  $C.L_{max}$  and we search whether the event node belongs to this set or not.

When  $p_w = 0.01$ , the effect of  $p_2$  on  $S_3, S_4, S_5$  and  $N_3, N_4, N_5$  seems to be significant, whereas the same cannot be said for  $p_w = 0.001$ . There is sudden change in  $S_i$ 's and  $N_i$ 's, when we shift from  $p_2 = 0$  to  $p_2 = 0.3$ , for  $p_w = 0.01$ , but not  $p_w = 0.001$ . So, when  $p_w$  is small, the effect of the neighborhood seems to be less.

When  $p_w = 0.01$   $N_i$ 's values are quite high and  $S_i$ 's are more for  $p_2 = 0$  than that of  $p_2 = 0.3$ . But When  $p_w = 0.001$   $N_i$ 's values are more and  $S_i$ 's are less for  $p_2 = 0$  than that of  $p_2 = 0.3$ .

The values of  $S_3$  and  $S_4$  are very similar for different values of the parameters; but large increment in  $N_4$  than  $N_3$  suggests that the idea of neighboring search is not effective. But  $S_3$  is much higher than  $S_2$ ; so the method of searching all the nodes in  $\mathcal{N}_{\max}$  is a better idea than that of searching a random node from  $\mathcal{N}_{\max}$ .

We estimate the success probability  $P_{\mathcal{M}_N}(L_N > C.L_{max})$  by simulation for different values of the threshold  $C$  ranging from 0.5 to 0.9 (see Table 7.2). Note that  $S_5$  corresponds to the threshold value  $C = 0.9$ . We consider  $p_1 = 0.99, p_w = 0.001, p_c = 0.9$  and four values of  $p_2 = 0.3, 0.4, 0.5, 0.6$ . From Table 7.2, we see that the success probability increases as the threshold value  $C$  decreases and  $p_2$  increases. The number of search decreases with  $C$  and  $p_2$ .

## 7.6 Discussion

One prime object is to show the effect of the neighboring nodes in detection of an event. In this section, we discuss the role of the neighboring nodes, some other related issues and make remarks.

### 7.6.1 Role of the neighboring nodes

Since  $\ln L_N = a + b(cz_N + t_N - dk(N))$ , where  $a, b, c$  and  $d$  are as defined in Section 7.3.1,  $c$  denotes the weight of the central node compared to the neighboring nodes in the corresponding likelihood. Note that, since  $P_1 > P_2$ , we have  $c > 1$  and, if  $c$  is close to 1, then the six neighboring nodes are as important as the event node. So, as the value of  $c$  increases, the importance of the neighboring nodes decreases. Also,  $d$  gives some idea about the role of the number of adjacent nodes, i.e.,  $k(N)$ . Recall that  $P_1$  and  $P_2$  are the probabilities of responding (i.e., reporting the node  $N$  as the event hexagon) by the sensors  $S_N$  and  $S_{N'}$ , respectively, when  $N$  is the event hexagon and  $N'$  is a neighboring node of  $N$ . So, we numerically calculate the quantities  $P_1, P_2, c$  and  $d$  for some values of the parameters (see Table 7.1).

From the theoretical results in Section 7.3.1 we see that,  $P_1$  and  $c$  increase as  $p_1$  increases, while  $P_2$  and  $d$  do not depend on  $p_1$ . On the other hand,  $P_2$  increases and  $c, d$  decreases with  $p_2$ . Also  $P_1$  is independent of  $p_2$ . Therefore, the importance of the neighboring nodes decreases with  $p_1$  and increases with  $p_2$  (see Table 7.1), as expected.

### 7.6.2 Estimation of the parameters

In practice, the parameters  $p_1, p_2, p_w$  and  $p_c$  may be unknown. We can, however estimate the parameters by some experimentation.

Note that, under  $\mathcal{M}_0, Z_N$  follows  $\text{Ber}(p_w)$  for all  $N \in R$ . Hence,  $p_w$  is the expected value of  $Z_N$  given  $\mathcal{M}_0$ . So we perform the experiment by keeping the ROI normal. The proportion of  $Z_N$ 's having value 1 gives an estimate of  $p_w$ . Repeat this experiment several times so that the average of the proportions over the repeated experiments can be taken as an estimate of  $p_w$ .

Note that,  $p_1$  is the expected value of  $Y_N$  under  $\mathcal{M}_N$ . So, we perform the experiment by causing an event in some node  $N$  of the ROI. The proportion of  $Y_N$ 's having value 1 gives an estimate of  $p_1$ . Repeat this experiment several times so that the average of the proportions over the repeated experiments can be taken as an estimate of  $p_1$ . Similar experiments will give estimates of  $p_2$  and  $p_c$  as well.

### 7.6.3 Incorporation of heterogeneity and uncertainty in parameters

Let  $\theta = (p_1, p_2, p_c, p_w)$  denote the set of parameters, which has been assumed to be the same for all the nodes. While, in practice there is no reason why the parameters should be same for all the nodes, it is also not clear how these would be different across  $N$ . This unexplained heterogeneity can be incorporated by assuming the  $\theta$ 's, for different  $N$ , to be independent realizations from a common distribution.

Let  $\theta_N = (p_{1N}, p_{2N}, p_{cN}, p_{wN})$  denote the set of parameters for node  $N$ . We assume that  $\theta_N, N \in R$ , are i.i.d. from some distribution, say,  $g(\theta)$ . Also assume that, given  $\theta_N, N \in R$ ,  $Z_N$ 's are independent. Note that  $g(\theta)$  denotes the joint distribution of the four parameters. For simplicity, we may assume them to be independent so that  $g(\theta)$  can be written as  $g(\theta) = g_1(p_1)g_2(p_2)g_c(p_c)g_w(p_w)$ . In this situation, the likelihood for the model  $\mathcal{M}_0$  is

$$\prod_{N \in R} \int p_{wN}^{z_N} (1 - p_{wN})^{(1-z_N)} g_w(p_{wN}) dp_{wN},$$

where the integration is over the range of  $p_{wN}$ . Similarly, the likelihood for the model  $\mathcal{M}_N$  can be written as

$$\prod_{N' \in R} \int L_N^{(N')}(\theta_N) g(\theta_N) d\theta_N,$$

where the integral is over the four-dimensional space given by the range of  $\theta_N$ , and  $L_N^{(N')}(\theta_N)$  is the contribution of the  $N'$ th node to the likelihood  $L_N$ , given the value  $\theta_N$ , as described in Section 7.3.1.

Similar technique can also be used to incorporate parameter uncertainty. Even though the parameters can be assumed to be same for all the nodes, there

may be reasonable uncertainty about the constancy of the parameter values. As in the Bayesian paradigm, the set of parameters may be assumed to be a realization from a distribution, say,  $g(\theta)$ . Then, the likelihoods for the model  $\mathcal{M}_0$  and  $\mathcal{M}_N$  are

$$\int \prod_{N \in R} p_w^{z_N} (1 - p_w)^{(1 - z_N)} g_\theta(p_w) dp_w \text{ and } \int \prod_{N' \in R} L_N^{(N')}(\theta) g(\theta) d\theta \text{ respectively.}$$

The choice of  $g(\theta)$  may be a difficult one. However, sometimes there may be specific information available regarding the distribution of  $\theta$ , which can be incorporated in the model.

#### 7.6.4 When more sensors can detect the event square

We may consider the situation when sensing radii are larger and more sensors can detect the event hexagon but with different probabilities. With respect to a particular node, classify the remaining nodes with respect to the probability of detecting the event at that node, which may as well depend on the distance from the particular node. Suppose that the sensors in the  $i$ -th class detect the event hexagon with probability  $p_i, i = 1, 2, 3, \dots$ . The theoretical analysis is similar, but having more probability terms.

Table 7.3: Simulation of estimated probabilities for some values of parameters

other parameters			Simulation of different probabilities with $p_c = 0.9$							
$p_1$	$p_2$	$p_w$	$S_1$	$S_2$	$S_3$	$N_3$	$S_4$	$N_4$	$S_5$	$N_5$
0.9	0.0	0.01	0.00	0.08	0.81	18.16	0.81	59.85	0.82	18.17
0.9	0.3	0.01	0.00	0.47	0.69	5.44	0.70	14.81	0.73	5.70
0.9	0.4	0.01	0.00	0.60	0.79	5.11	0.78	13.51	0.80	5.12
0.9	0.5	0.01	0.00	0.70	0.85	4.64	0.85	11.50	0.86	4.88
0.9	0.6	0.01	0.00	0.79	0.90	3.82	0.91	08.18	0.92	3.90
0.9	0.0	0.001	0.35	0.50	0.81	3.17	0.81	7.17	0.81	3.18
0.9	0.3	0.001	0.36	0.59	0.82	3.03	0.83	6.34	0.86	3.06
0.9	0.4	0.001	0.35	0.67	0.87	2.89	0.87	6.18	0.89	3.03
0.9	0.5	0.001	0.36	0.75	0.90	2.89	0.89	5.85	0.93	2.96
0.9	0.6	0.001	0.36	0.83	0.94	2.74	0.93	5.33	0.96	2.79
0.99	0.0	0.01	0.00	0.08	0.89	17.43	0.90	56.20	0.89	17.70
0.99	0.3	0.01	0.00	0.51	0.73	5.18	0.73	12.98	0.79	5.77
0.99	0.4	0.01	0.00	0.62	0.81	5.09	0.82	13.01	0.84	5.21
0.99	0.5	0.01	0.00	0.73	0.88	4.97	0.88	12.69	0.89	5.04
0.99	0.6	0.01	0.00	0.81	0.92	3.66	0.92	7.35	0.93	3.57
0.99	0.0	0.001	0.35	0.57	0.89	3.19	0.89	6.69	0.89	3.20
0.99	0.3	0.001	0.35	0.62	0.88	2.99	0.87	6.00	0.90	3.02
0.99	0.4	0.001	0.36	0.70	0.90	2.91	0.91	5.83	0.93	2.97
0.99	0.5	0.001	0.36	0.79	0.93	2.82	0.94	5.67	0.95	2.83
0.99	0.6	0.001	0.36	0.84	0.95	2.75	0.95	5.31	0.97	2.68

other parameters			Simulation of different probabilities with $p_c = 0.99$							
$p_1$	$p_2$	$p_w$	$S_1$	$S_2$	$S_3$	$N_3$	$S_4$	$N_4$	$S_5$	$N_5$
0.9	0.0	0.01	0.00	0.08	0.90	18.2	0.89	55.62	0.90	17.58
0.9	0.3	0.01	0.00	0.54	0.76	5.14	0.76	13.08	0.79	5.64
0.9	0.4	0.01	0.00	0.67	0.85	5.05	0.85	12.92	0.87	5.12
0.9	0.5	0.01	0.00	0.77	0.91	4.86	0.90	12.10	0.91	5.02
0.9	0.6	0.01	0.00	0.86	0.94	3.57	0.93	7.24	0.95	3.57
0.9	0.0	0.001	0.36	0.57	0.90	3.18	0.89	6.61	0.89	3.19
0.9	0.3	0.001	0.36	0.65	0.88	2.98	0.89	6.34	0.92	3.02
0.9	0.4	0.001	0.36	0.73	0.92	2.87	0.92	5.91	0.94	2.91
0.9	0.5	0.001	0.35	0.81	0.94	2.81	0.94	5.51	0.96	2.82
0.9	0.6	0.001	0.37	0.88	0.96	2.72	0.96	5.24	0.97	2.90
0.99	0.0	0.01	0.00	0.09	0.98	16.9	0.98	51.40	0.98	17.6
0.99	0.3	0.01	0.00	0.58	0.83	5.66	0.83	14.73	0.87	5.69
0.99	0.4	0.01	0.00	0.69	0.90	5.43	0.91	14.38	0.92	5.69
0.99	0.5	0.01	0.00	0.80	0.94	4.61	0.94	11.19	0.95	4.73
0.99	0.6	0.01	0.00	0.87	0.96	3.26	0.97	6.26	0.96	3.35
0.99	0.0	0.001	0.35	0.62	0.98	3.20	0.98	6.32	0.98	3.28
0.99	0.3	0.001	0.36	0.69	0.94	2.90	0.94	5.88	0.97	3.00
0.99	0.4	0.001	0.36	0.76	0.95	2.89	0.96	5.59	0.98	2.85
0.99	0.5	0.001	0.36	0.83	0.97	2.70	0.97	5.48	0.98	2.80
0.99	0.6	0.001	0.36	0.89	0.98	2.68	0.98	5.19	0.99	2.69



# Detection of Events in General Situation

## 8.1 Introduction

In this chapter, we are interested in one particular query: determining an event in the environment (i.e., ROI) with a distinguishable characteristic. We assume the ROI is to be partitioned into suitable number of congruent regular hexagonal cells (i.e., we can consider ROI as a regular hexagonal grid). This physical structure of ROI is not a requirement for the theoretical analysis, we can also do the similar analysis with other structures like a square grid. We choose a hexagonal grid since it is an optimal placement of sensors in some sense [76]. Suppose that sensors are placed a priori at the center (which are known as nodes) of every hexagon of the grid. We assume that if an event occurs in the hexagon (call it the event hexagon) where a particular sensor lies, then that particular sensor can detect that event with a greater probability; whereas, if an event occurs in any adjacent hexagon, then the particular sensor can detect that event with a lesser probability (due to greater distance). Hence, only one node (center node of the event hexagon) can detect an event hexagon (where an event occurs) with greater probability, say  $p_1$ , and adjacent nodes (six for interior nodes and less for boundary nodes) can detect the event hexagon with lesser probability, say  $p_2$ , with  $p_1 > p_2$ , see Figure 3.2. We assume that no other sensor can detect an event hexagon. We also assume

that there are one or more event hexagons in ROI (all the hexagons may be event hexagons). We also assume there is no fusion sensor and all the sensors can communicate with the base station and the base station takes the decision about the query. As an example, consider a network of devices that are capable of sensing mines or bombs, if we assume that mines or bombs are placed over a region (ROI). Information from these devices can be sent to a nearby police station, or a central facility (the base station). Then, an important query in this situation could be whether a particular hexagon is an event hexagon or not (i.e., whether mines or bombs are placed there or not).

If at most  $n$  events occur in the ROI then there will be  $1 + |R| + \binom{R}{2} + \dots + \binom{R}{n}$  models, where  $|R|$  is the number of hexagons in ROI. If number of models is not so large then we can apply model selection or multiple model selection techniques using likelihood, also Bayesian model averaging method can be used as in the case of  $n = 1$  (see Chapter 7). But if  $n$  is large then the number of models is also very large; and hence calculation of all the likelihood values is infeasible; in that case these techniques can not be applied.

In our theoretical analysis, the sensor fault probabilities are introduced into the optimal event detection process. We apply near-optimal model selection approach and a new method to find a solution of the problem. We also discuss two interesting situations. We simulate different situations with different parameters.

We introduce two detection probabilities,  $p_1$  and  $p_2$ , one for the center node and other for the adjacent nodes. Even if the center node fails to detect the event, the adjacent nodes may detect an event, and vice versa. We consider these probabilities and show that, in various situations, the adjacent nodes play a key role to detect the event. One can introduce more detection probabilities and analyze the situation in similar manner. We introduce the probability model in two different stages; firstly, when a sensor detects the event and, secondly, when a sensor sends the message to the base station. We assume the ROI is an hexagonal grid since it is optimal in some sense (see Chapter 3). We consider the most general situation of fault detection problem.

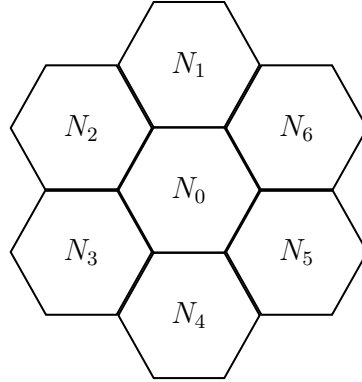


Figure 8.1: ROI partitioned into regular hexagons and Nodes placed in centers ( $N_i, i = 1, 2, \dots, 6$  are the adjacent nodes of  $N_0$ )

## 8.2 Statement of the Problem and Assumptions

In this section, we describe the problem in more specific terms and state the assumptions that we make.

Sensors are deployed, or manually placed, over the ROI to perform event detection (i.e., to detect whether an event of interest has happened or not) in the ROI. If sensors are deployed from air then sensors are so placed (using actuator-assisted sensor placement or by movement-assisted sensor placement) that the sensor network covers the entire ROI. The ROI is partitioned into a suitable number of regular hexagons (i.e., we can think of the ROI as a regular hexagonal grid), as shown in Figure 8.1. Sensors are placed a priori at every center (which are known as nodes) of the regular hexagons. Sensors have two detection probabilities. There may be any number of event hexagons and we assume that there is a prior probability that a particular hexagon is an event hexagon.

Through a broadcast or acknowledge protocol, each sensor node is also able to locate the neighbors within its communication radius. All sensors can communicate with the base station and the base station takes the decision by combining the information received from all the sensors.

There are two phases in the whole process. The first one is the detection phase, when the sensor at the center of a regular hexagon tries to detect an

event. The next phase is response phase, in which sensors send messages to the base station. Even if the event hexagon is detected by a sensor, it may not respond due to some technical fault, with some positive probability. On the other hand, if event hexagon is not detected, or there is no event in the detection range, then also a faulty sensor can send the wrong information to the base station with positive probability.

As the sensors may send wrong information to the base station, the base station takes an important role in identifying the event hexagon. The base station will collect all the information and take a decision about event hexagons according to some rule which we are going to discuss. Our objective is to find a rule for the base station such that the base station works most efficiently.

### 8.2.1 Notations and Assumptions

Our problem is to develop a strategy for the base station to take decision about event hexagons (i.e., which hexagon of the ROI is the event hexagon, if at all). Let  $R$  be the set of all nodes and  $R'$  be the set of all interior nodes (the nodes which have six adjacent nodes). For  $N \in R$ , define  $B(N)$ , as the set of adjacent node(s) of  $N$  and  $k(N) = |B(N)|$ , be the number of adjacent node(s) of  $N$ . Hence,  $0 \leq k(N) \leq 6$ . Let  $S_N$  be the sensor which is placed at the node  $N$  and  $H_N$  be the hexagon where the node  $N$  is placed (i.e.,  $N$  is the center of  $H_N$ ). For  $N \in R$ , let  $X_N$  denote the true status of the node  $N$ . That is,  $X_N = 1$  if event occurs at  $H_N$ , and 0 otherwise. Also define  $Y_N = 0$  if  $S_N$  detects no event, and 1 if  $S_N$  detects at least one event in  $H_N$  or  $H_{N'}$ , for  $N' \in B(N)$ . Finally define  $Z_N = 0$  if  $S_N$  does not respond, i.e., the sensor informs the base station that event does not occur at  $H_N$  or  $H_{N'}$  for  $N' \in B(N)$ , and  $Z_N = 1$  if  $S_N$  responds, i.e., the sensor  $S_N$  informs the base station that at least one event has occurred in  $H_N$  or  $H_{N'}$ , for  $N' \in B(N)$ .

Now we make a natural assumption that, once the detection phase is completed, the response of a sensor depends only on what it detects but not on whether the event has actually occurred or not, i.e.,  $P(Z_N = k|Y_N, X_N) = P(Z_N = k|Y_N)$ , for  $k = 0, 1$ . We also assume that the sensors work independently and hence the  $X_N$ 's are independent.

Since we assume that there are arbitrarily number of event hexagons, the

possible true scenarios are, therefore, represented by the following  $2^{|R|}$  different models: for  $S \subset R$ ,

$$\mathcal{M}_S : (X_N = 1 \text{ for all } N \in S \text{ and } X_N = 0 \text{ for all } N \in R \setminus S).$$

Let  $P(\mathcal{M}_\phi) = P(\text{ROI is normal}) = p_{norm}$  and for all  $N \in R$ ,  $P(\text{an event occurs at the hexagon } H_N) = p_N$ . In particular, we may assume  $p_N$ 's to be same for all  $N$ . We denote any probability under the model  $\mathcal{M}_S$  as  $P_S(\cdot)$ . We also make the followings assumptions:

1. For all  $N \in R$ ,  $P_\phi(Y_N = 1) = 0$  and  $P_{\{N\}}(Y_N = 1) = p_1$ .
2. For all  $N' \in B(N)$ ,  $P_{\{N\}}(Y_{N'} = 1) = p_2$ , and for all  $N' \in R \setminus [B(N) \cup \{N\}]$ ,  $P_{\{N\}}(Y_{N'} = 1) = 0$ .
3. For all  $N \in R$ ,  $P(Z_N = 1|Y_N = 1) = p_c$  and  $P(Z_N = 1|Y_N = 0) = p_w$ .
4.  $Z_N$  and  $Y_{N'}$  are independent for  $N \neq N'$ .
5. The responses from different nodes are independent under a particular model, i.e.,  $Z_N$ 's are independent under  $\mathcal{M}_{\{N'\}}$  for a fixed  $N' \in R$ .

### 8.3 Theoretical Analysis of fault Detection

In this section we discuss some theoretical results. For any two subsets  $S_1$  and  $S_2$  of  $R$  we define distance between the two models  $\mathcal{M}_{S_1}$  and  $\mathcal{M}_{S_2}$  as  $|S_1 \Delta S_2|$ . Note that this distance is a metric on the set of all models. For a model  $\mathcal{M}_S$  we define the neighborhood models of that model as set of all models  $\mathcal{M}_{S'}$  for which distance between  $\mathcal{M}_S$  and  $\mathcal{M}_{S'}$  is 1.

### 8.3.1 Model Selection Approach

For all  $S \subset R$ , define  $P_{N|S} = P_S(Z_N = 1)$ .

$$\begin{aligned}
 & \text{Hence, } P_{N|S} = P_S(Z_N = 1) \\
 & = P_S(Z_N = 1|Y_N = 0)P_S(Y_N = 0) + P_S(Z_N = 1|Y_N = 1)P_S(Y_N = 1) \\
 & = P(Z_N = 1|Y_N = 0)P_S(Y_N = 0) + P(Z_N = 1|Y_N = 1)P_S(Y_N = 1) \\
 & = p_w(1 - p_1)^{|S \cap \{N\}|}(1 - p_2)^{|S \cap B(N)|} \\
 & \quad + p_c \left( 1 - (1 - p_1)^{|S \cap \{N\}|}(1 - p_2)^{|S \cap B(N)|} \right) \\
 & = p_c - (p_c - p_w)(1 - p_1)^{|S \cap \{N\}|}(1 - p_2)^{|S \cap B(N)|}.
 \end{aligned}$$

Note that,  $P_{N|\phi} = p_w$ ,  $P_{N|\{N\}} = p_1(p_c - p_w) + p_w$ ,  $P_{N|\{N'\}} = p_2(p_c - p_w) + p_w$  for  $N' \in B(N)$

$P_{N|\{N'\}} = p_w$  for  $N' \notin B(N) \cup \{N\}$ ,  $P_{N|S} = p_c - (p_c - p_w)(1 - p_1)(1 - p_2)^{|S \cap B(N)|}$  for  $N \in S$  and

$P_{N|S} = p_c - (p_c - p_w)(1 - p_2)^{|S \cap B(N)|}$  for  $N \notin S$ .

Also note that, if  $S_1 \subset S_2$  then  $P_{N|S_1} < P_{N|S_2}$ .

Now, under the model  $\mathcal{M}_S$ ,  $Z_N$  follows  $\text{Ber}(P_{N|S})$ , for all  $N \in R$ , and the likelihood of the data  $\{Z_N = z_N, \text{ for all } N \in R\}$ , under the model  $\mathcal{M}_S$ , is  $L_S = P_S(Z_N = z_N, \text{ for all } N \in R) = \prod_{N \in R} P_{N|S}^{z_N} (1 - P_{N|S})^{(1 - z_N)}$ .

Therefore,  $\ln L_S = \sum_{N \in R} (z_N \ln P_{N|S} + (1 - z_N) \ln(1 - P_{N|S})) = \sum_{N \in S} \ln P_{N|S} + \sum_{N \notin S} \ln(1 - P_{N|S})$ .

Now, one can select the model for which  $L_S$  is maximum. If  $S_1 \subset S_2$  and  $L_{S_1} \geq L_{S_2}$  then obviously  $\mathcal{M}_{S_1}$  is a better choice than  $\mathcal{M}_{S_2}$ . Note that this ordering forms an partial order on the set of all models.

### 8.3.2 Algorithm to find Near-Optimal Model

The model selection approach of the previous subsection to obtain an optimal model is reasonable and numerically manageable as long as  $|R|$  is small. But if  $|R|$  is large, then the number of possible models is  $2^{|R|}$  which may be very large and it is hard to calculate all the likelihoods. In that case we can use different algorithms to find a near optimal model, i.e., a model whose corresponding

likelihood is ‘very close’ to the maximum one. One such algorithm can be found in [32]. We can use the same algorithm to find a near-optimal model.

To construct an optimal model with maximum likelihood or a near-optimal model with near-maximum likelihood, an initial model is needed. We can select that model as  $\mathcal{M}_I$  with  $I = \{N \in R : z_N = 1\}$ , where the data set  $\{Z_N = z_N, \text{ for all } N \in R\}$  is received by the base station. Define the model  $\mathcal{M}_I$  to be the *initial model*. Initial model is that model where we consider that hexagons to be event hexagons for which corresponding nodes respond to occurrences of an event. After selecting the initial model, a variable neighborhood (VNS) search algorithm is proposed to arrive at the optimal model or near-optimal model.

Let us denote the neighborhood of  $\mathcal{M}_S$  by  $N(\mathcal{M}_S) = \{\mathcal{M}_{S'} : |S\Delta S'| = 1\}$ . Note that  $\mathcal{M}_{S'} \in \mathcal{M}_{N(S)}$  then  $\mathcal{M}_S \in \mathcal{M}_{N(S')}$ . Now all the models are connected through this neighborhood concept in the sense that any model can be connected to another model through successive neighborhoods. The proposed algorithm is as follows:

*Step 1.* Start with the initial model  $\mathcal{M}_I$  and corresponding likelihood  $L_I$ .

*Step 2.* Consider  $\mathcal{M}_{S'} \in \mathcal{M}_{N(I)}$  and compute the corresponding likelihood  $L_{S'}$ .

*Step 3.* If  $\max\{L_{S'}, |S'\Delta S| = 1\} > L_I$ , choose the next improved model to be  $\mathcal{M}_{S_1}$ , where  $S_1 = \arg(\max\{L_{S'}, |S'\Delta S| = 1\})$ . Otherwise, choose  $S_1$  randomly from among the models given by  $\{I\} \cup N(I)$  with corresponding probabilities given by

$$p_0 = \frac{L_I}{L_I + \sum_{S' \in N(I)} L_{S'}}, \quad p_{S'} = \frac{L_{S'}}{L_I + \sum_{S' \in N(I)} L_{S'}}, \quad \text{for } |S'\Delta S| = 1.$$

*Step 4.* Stop if there is insignificant improvement, which may be characterized by insignificant change over a certain number of iterations, return to the same model. Otherwise, go back to *Step 2* with  $S_1$  replacing  $I$ .

Note that, because of the connectedness property the algorithm has the potential to cover all the  $2^{|R|}$  models. However, the algorithm may get stuck at a local maximum. For this, we may restart the algorithm with an initial model  $\mathcal{M}_{S_2}$ , whose distance is 2 from the current optimal model. One can think

of some other concept of neighborhood and carry out the same algorithm.

We simulate the above algorithm for different values of parameters starting from the initial model. We also consider two more models, *initial neighborhood model* and *near optimal neighborhood model*. The first model contains all hexagons from the initial model along with their adjacent hexagons and the second model contains all hexagons from the near optimal model along with their adjacent hexagons. Note that the initial neighborhood model is not same as a neighborhood model of the initial model, same argument for near optimal neighborhood model also. Here we consider all over four models. The initial model is the model which takes no time to compute for any ROI. Since this model may not capture all the event hexagons of the true model, we consider the initial neighborhood model, which may contain more event hexagons than initial model. We also consider near optimal neighborhood model for the same reason. Define the *number of searches* by a model  $\mathcal{M}_S$  as the number of elements in  $S$ , so that this many searches may be required for monitoring to detect event hexagons.

We consider a  $16 \times 16$  hexagonal grid and  $p_N = 0.025$ . Hence on an average there are 6.4 event hexagons. We run the program for 16 different choices of parameters and collect data after running our algorithm 100 times for each choice separately. We simulate a number of event hexagons (say,  $n$ ), number of event hexagons detected by initial model (say,  $n_I$ ), number of searches by initial model (say,  $n_{IS}$ ), number of event hexagons detected by near optimal model (say,  $n_O$ ), number of searches by near optimal model (say,  $n_{OS}$ ), number of event hexagons detected by initial neighborhood model (say,  $n_N$ ), number of searches by initial neighborhood model (say,  $n_{NS}$ ), number of event hexagons detected by near optimal neighborhood model (say,  $n_{ON}$ ) and number of searches by near optimal neighborhood model (say,  $n_{ONS}$ ). The simulation is performed using Python-code. Results obtained from simulation are summarized in Table 8.1. The entries of the Table 8.1 are the average values of corresponding quantities over 100 simulations. For example, in the 4th row of the first sub-table, with  $p_1 = 0.9, p_2 = 0.8, p_c = 0.9, p_w = 0.1$ , we find the followings:

1. Average number of event hexagons is 6.00.



Table 8.1: Performance of our algorithms under several models

$p_c = 0.9, p_w = 0.1$										
$p_1$	$p_2$	$n$	$n_I$	$n_{IS}$	$n_O$	$n_{OS}$	$n_N$	$n_{NS}$	$n_{ON}$	$n_{ONS}$
0.9	0.5	6.92	6.08	30.40	4.12	11.04	6.87	91.80	4.91	43.45
0.9	0.6	6.25	5.31	33.23	3.87	10.55	6.25	97.77	3.93	39.70
0.9	0.7	6.10	4.79	32.63	3.72	8.43	6.05	91.56	3.82	39.38
0.9	0.8	6.00	5.32	36.32	4.51	9.67	6.00	94.98	4.51	40.64
0.8	0.5	5.76	4.33	26.50	3.16	10.75	5.50	84.66	4.38	38.28
0.8	0.6	6.33	4.92	31.00	4.17	9.08	6.30	91.58	5.06	40.48
0.8	0.7	6.42	4.50	35.25	3.58	10.50	6.41	93.41	4.17	38.91
0.8	0.8	5.92	4.61	34.57	3.92	8.90	5.92	92.48	3.67	35.93

$p_c = 0.8, p_w = 0.2$										
$p_1$	$p_2$	$n$	$n_I$	$n_{IS}$	$n_O$	$n_{OS}$	$n_N$	$n_{NS}$	$n_{ON}$	$n_{ONS}$
0.9	0.5	6.96	5.21	41.25	2.87	14.76	6.96	111.41	5.29	57.50
0.9	0.6	5.83	4.50	38.00	2.58	11.25	5.83	109.34	4.19	44.71
0.9	0.7	5.60	4.59	39.42	2.43	11.70	5.60	110.07	4.03	43.00
0.9	0.8	6.42	5.08	43.17	3.56	12.21	6.42	110.50	4.11	44.02
0.8	0.5	6.93	5.15	38.05	3.19	13.47	6.78	103.80	5.68	52.65
0.8	0.6	6.99	5.26	40.75	3.19	13.37	6.75	108.72	4.90	51.45
0.8	0.7	6.76	4.78	43.23	3.65	13.53	6.75	110.91	5.07	52.32
0.8	0.8	5.82	4.80	42.01	3.34	11.89	5.82	108.75	3.78	38.48

2. On average, number of searches (i.e., the size) of the initial model is 36.32 and detects 5.32 many event hexagons.
3. On average, our proposed near optimal model detects 4.51 of the event hexagons and it searches 9.67 hexagons.
4. On average, the initial neighborhood model, out of 94.98 searches, it detects 6.00 hexagons (that means, all the event hexagons).

## Observations

1. The near optimal model detects fewer number of events hexagons than the initial model but number of searches is less for the near optimal model.
2. For a better network (higher value of  $p_c$  and lesser value of  $p_w$ ) the number of events hexagons detected using the near optimal model is larger and the number of searches is fewer.

3. The initial neighborhood model finds (almost) all event hexagons, but it requires an excessive number of searches.

## 8.4 A New Approach To Find Event Hexagons

In this section we discuss a different approach which is not based on likelihoods. In previous chapter we show that if there are only one event hexagon, selection of the event hexagon is just based on the quantities  $Q_N = cz_N + t_N - dk(N)$ , for all nodes  $N$ , where  $z_N = 1$  if the node  $N$  responses positively and  $z_N = 0$  otherwise;  $t_N$  be the number of neighboring nodes of  $N$  which respond positively;  $k(N)$  be the number of neighboring nodes of  $N$  and

$$c = \frac{\ln \frac{P_1(1-p_w)}{p_w(1-P_1)}}{\ln \frac{P_2(1-p_w)}{p_w(1-P_2)}} \quad \text{and} \quad d = \frac{\ln \frac{1-p_w}{1-P_2}}{\ln \frac{P_2(1-p_w)}{p_w(1-P_2)}} \quad \text{are independent of } N.$$

Note that the quantity  $c$  can be interpreted as weight of a node  $N$  with respect to its neighboring node in order to suggest  $H_N$  as the event hexagon. For example, if  $c = 3$  then  $z_N = 1, t_N = 0$  and  $z_N = 0, t_N = 3$  gives same  $Q_N$ , i.e., positive responses of  $N$  is equivalent to positive response by 3 of its neighboring nodes to suggest  $H_N$  as the event hexagon. In the previous chapter we show that, the base station selects the node  $N$  for which  $Q_N$  is maximum or the base station selects multiple nodes for which  $Q$ -values are greater than a fixed threshold value. We will use similar idea in this section, when there may be more than one event hexagon, the base station selects those hexagons as event hexagons, for which corresponding  $Q$  values are higher than that of those hexagons not selected by the base station as event hexagons. We have simulated  $Q$ -values for all the nodes (for different choices of parameters). We consider a  $32 \times 32$  hexagonal grid and perform the simulation 10000 times using Python-code.

Note that, for a fixed choice of the parameters,  $c, d$  are fixed. Our guess is that, if  $Q_N > Q_{N'}$  then chance of  $H_N$  to be an event hexagon is greater than that of  $H_{N'}$ . Now we sort all nodes with respect to decreasing order of  $Q$  values of corresponding nodes. If  $Q$  values are equal for more than one node then we order them arbitrarily. After sorting we rename all the nodes as  $(N_1, N_2, \dots)$ ,

i.e.,  $Q_{N_i} \geq Q_{N_j}$  if and only if  $i \geq j$ .

Now if the base station selects a model  $\mathcal{M}_{S_y}$ , where  $S_y$  is defined as  $\{N_1, N_2, \dots, N_y\}$  and  $y$  will be determined by the resource of the base station then the base station will search  $y$  many hexagons and find  $|S_y \cap S|$  many event hexagons, where  $\mathcal{M}_S$  is the true model. We define  $y$  as the number of searches and  $|S_y \cap S|$  as the number of successes. Next for  $k \in \mathbb{N}$  we take the smallest natural number  $y$  for which  $X_{N_1} + X_{N_2} + \dots + X_{N_y} = k$ . So the base station needs  $y$  searches to detect  $k$  event hexagons. So there are two approaches:

1. Fix  $k$  then find  $y$ , i.e., the base station first fix how many event hexagons are to be detected (say,  $k$  many) and then search the hexagons correspond to the nodes  $\{N_1, N_2, \dots, N_k\}$ . If the base station finds  $k$  many event hexagons, i.e., if  $X_{N_1} + X_{N_2} + \dots + X_{N_k} = k$ , then stop the search, and select the model  $\mathcal{M}_{S_k}$  with  $S_k$  being  $\{N_1, N_2, \dots, N_k\}$ . Now, if  $X_{N_1} + X_{N_2} + \dots + X_{N_k} < k$  then search one more hexagon  $H_{N_{k+1}}$  and check whether  $X_{N_1} + X_{N_2} + \dots + X_{N_{k+1}} = k$  or not. If equal, stop and select the model  $\mathcal{M}_{S_{k+1}}$  and if not carry out further searches until  $k$  many event hexagons are detected.
2. Fixed  $y$ , i.e., number of searches depending on resource, and then find the number of events that we are detected.

We simulate number of searches ( $y$ ) for different values of success ( $k$ ). Results obtained from simulation are summarized in Figure 8.2. We draw three graphs for three different sets of parameters with  $k$  on the horizontal axis and average values (from 10000 simulations) of  $y$  on vertical axis. We call the  $y$  vs.  $k$  curves as  $Q$  curves. By setting  $p_N = 0.05$ , we collect data after a run of 10000 simulations for each of the following combinations:

In the left graph,  $p_1 = p_c = 0.99, p_w = 0.01$  and  $p_2 = 0.6, 0.72, 0.84$ ; in the middle graph,  $p_1 = p_c = 0.99, p_2 = 0.65$  and  $p_w = 0.01, 0.05, 0.15$ ; in right graph,  $p_2 = 0.65, p_w = 0.01$  and three different combinations of  $p_1$  and  $p_c$ :  $(p_1, p_c) = (0.99, 0.99), (0.8, 0.99), (0.99, 0.8)$ .

Again for initial model  $I$ , the base station has  $n_I$  successes at the cost of  $n_{IS}$  searches. We put the point  $(n_I, n_{IS})$ , one for each of the combinations of

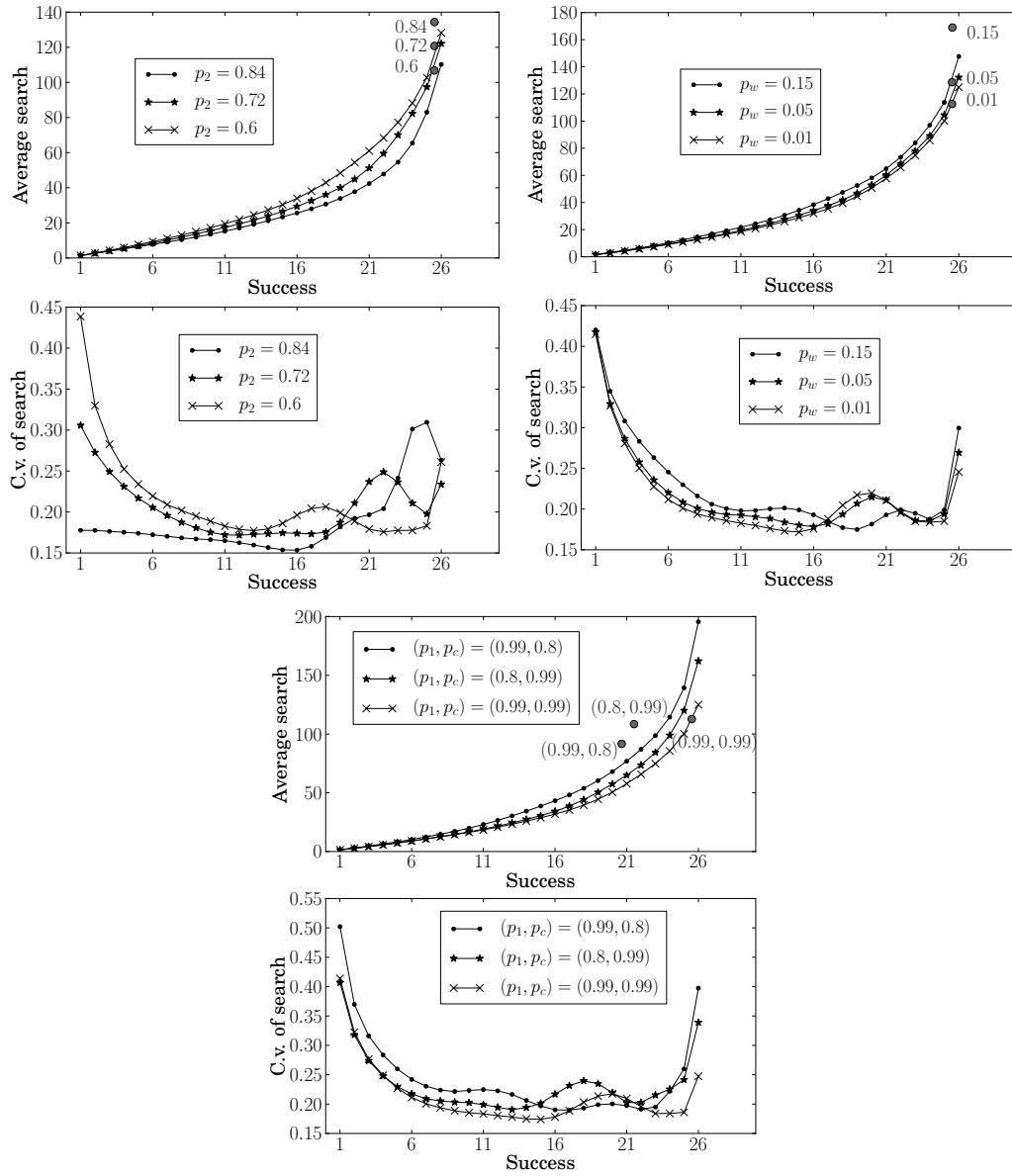


Figure 8.2: Average search vs. Success and C.v. of search vs. Success graph

the parameters (which is obtained from the average of 10000 runs) in the three ‘average search ( $y$ ) vs. success ( $k$ ) graphs’ (in Figure 8.2). Since the position of  $(n_I, n_{IS})$  are above the  $y$  vs.  $k$  curves, our new approach seems to be better than the initial model because it needs fewer searches to find the same number of event hexagons. We also draw the coefficient of variations of searches vs. successes graph.

## Observations

1. From Table 8.1, we note that (average) cardinality of the initial model is generally less than the number of event hexagons. Thus, to find all (or, even any number of event hexagons greater than the cardinality of the initial model) occurrences of events, one needs an exhaustive search on 512 grids (if he has only the initial model at disposal) and in this case a model based on  $Q$  values is surely better than the initial model (as it gives all event hexagons in fewer searches).
2. Note that the  $Q$  curves are convex upward. That means, it most often lies under the line segment joining the origin and the corresponding  $(n_I, n_{IS})$  point. Now, if someone needs to find, say, only 20 event hexagons (let us pretend that it is less than the cardinality of the initial model) then the model based on  $Q$  values is better (as, in the initial model one will choose one point on the line segment with abscissa 20; which lies above that point on  $Q$  curve having the same abscissa).
3. If the point  $(n_I, n_{IS})$  lies anywhere on the left side of the  $Q$  curves, then  $Q$  model is better than the initial model because on equal number of searches it gives more successes.
4. It is evident from Figure 8.2 (first graph) that there is a trend. For larger  $p_2$  larger search in either model is required. But, it is also evident that these two models perform almost same for  $p_2 = 0.72$ .  $Q$  model performs better for  $p_2 > 0.72$ . We check (not shown here) for several other values for  $p_2$  and find similar results.
5. We also note from Figure 8.2 (second graph) that, an increase in  $p_w$  increases the average search in both models and that both perform almost

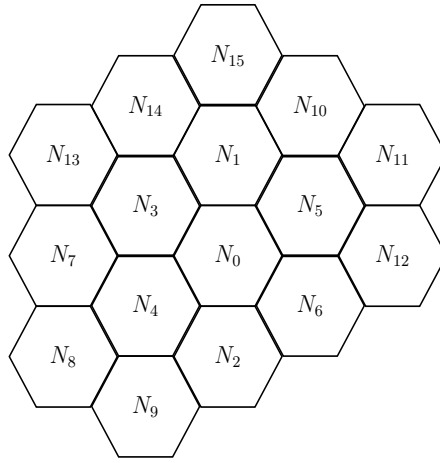


Figure 8.3: Nodes placed at the center regular hexagons

same for  $p_w = 0.05$ , and the  $Q$  model is better for larger  $p_w$ . This result is also supported by some other choices of  $p_w$ .

6. If we increase at least one of  $p_1$  and  $p_c$ , we note (third graph) that number of searches decreases for a success and the cardinality of the initial model will also be low. Impact of  $p_1$  and  $p_c$  are not similar; in both models  $p_c$  has more effects than  $p_1$ .
7. We observe that c.v. decreases till near 16 and after that it has a rise in all cases. Also c.v. is low in the range of 10 to 25 success and high for other values of success. That means that the average number of searches is more stable when the number of successes lies between 10 and 25.

## 8.5 Two Special Cases

In this section we consider two special cases and in each case there are only two possible models. So this is the case of hypothesis testing. Consider Figure 8.1, where events occur at  $N_4$  and  $N_5$  but the base station decides that event occurs at  $N_0$  or vice-versa. Another example is events occur at  $N_1$  and  $N_4$  but the base station decides that  $N_0$  and  $N_3$  or vice-versa.

### 8.5.1 Model $\{N_4, N_5\}$ vs Model $\{N_0\}$

Consider the first example, where we assume there are only two models, either events occur at  $N_4$  and  $N_5$  or at  $N_0$ . Now

$$P_{\{N_4, N_5\}}(Z_{N_0} = 1) = p_w(1 - p_2)^2 + p_c(1 - (1 - p_2)^2) = P_0 \text{ (say),}$$

$$\text{for } i = 4, 5; P_{\{N_4, N_5\}}(Z_{N_i} = 1) = p_w(1 - p_1) + p_cp_1 = P_1 \text{ (say),}$$

for  $N$  which belongs to exactly one of  $B(N_4)$  and  $B(N_5)$ ,  $N \neq N_4$  and  $N \neq N_5$ ;

$$P_{\{N_4, N_5\}}(Z_N = 1) = p_w(1 - p_2) + p_cp_2 = P_2 \text{ (say)}$$

and for  $N$  which does not belong to  $\{N_4, N_5\} \cup B(N_4) \cup B(N_5)$ ;  $P_{\{N_4, N_5\}}(Z_N = 1) = p_w$ . Again under model  $N_0$ ,

$$P_{\{N_0\}}(Z_{N_0} = 1) = p_w(1 - p_1) + p_cp_1 = P_1,$$

$$\text{for } i = 1, 2, 3, 4, 5, 6; P_{\{N_0\}}(Z_{N_i} = 1) = p_w(1 - p_2) + p_cp_2 = P_2$$

$$\text{and for other } N; P_{\{N_0\}}(Z_N = 1) = p_w.$$

Hence, if the base station receives data  $\{Z_N = z_N, \text{ for all } N \in R\}$  then the likelihood of the data under the model  $\{N_4, N_5\}$  is

$$\begin{aligned} L_{\{N_4, N_5\}} &= P_0^{z_0}(1 - P_0)^{1-z_0} P_1^{(z_4+z_5)}(1 - P_1)^{2-(z_4+z_5)} P_2^{(z_1+z_2+z_3+z_6+\dots+z_{12})} \times \\ &\quad (1 - P_2)^{10-(z_1+z_2+z_3+z_6+\dots+z_{12})} p_w^{(z_{13}+\dots+z_{n-1})} (1 - p_w)^{(n-13)-(z_{13}+\dots+z_{n-1})} \end{aligned}$$

and the likelihood under the model  $\{N_0\}$  is

$$\begin{aligned} L_{\{N_0\}} &= P_1^{z_0}(1 - P_1)^{1-z_0} P_2^{(z_1+\dots+z_6)}(1 - P_2)^{6-(z_1+\dots+z_6)} p_w^{(z_7+\dots+z_{n-1})} \times \\ &\quad (1 - p_w)^{(n-7)-(z_7+\dots+z_{n-1})}, \end{aligned}$$

where  $n$  is the total number of nodes.

Therefore,  $L_{\{N_4, N_5\}} < L_{\{N_0\}}$  is equivalent to

$$P_0^{z_0}(1 - P_0)^{1-z_0} P_1^{(z_4+z_5)}(1 - P_1)^{2-(z_4+z_5)} P_2^{(z_7+\dots+z_{12})}(1 - P_2)^{6-(z_7+\dots+z_{12})} <$$

$$P_1^{z_0}(1 - P_1)^{1-z_0} P_2^{(z_4+z_5)}(1 - P_2)^{2-(z_4+z_5)} p_w^{(z_7+\dots+z_{12})}(1 - p_w)^{6-(z_7+\dots+z_{12})}.$$

which is equivalent to

$$P_0^{z_0}(1 - P_0)^{1-z_0} P_1^{(z_4+z_5-z_0)}(1 - P_1)^{1-(z_4+z_5-z_0)} P_2^{(z_7+\dots+z_{12})-(z_4+z_5)} \times \\ (1 - P_2)^{4-(z_7+\dots+z_{12})-(z_4+z_5)} < p_w^{(z_7+\dots+z_{12})}(1 - p_w)^{6-(z_7+\dots+z_{12})}.$$

Note that the above inequality does not depend on the  $z_N$  for  $N \notin \{N_i : i = 0, 1, \dots, 12\}$  hence we can consider only these 13 nodes. Now for all possible  $2^{13}$  choices of  $z_N$  we can calculate the above inequality and check whether that is true or not. We calculate the probability of all possible choices of  $z_N$  and then probability of selecting the model  $\{N_4, N_5\}$  when the model  $\{N_4, N_5\}$  is true. We also do the similar calculation when the model  $\{N_0\}$  is true (see Table 8.2).

### 8.5.2 Model $\{N_1, N_4\}$ vs Model $\{N_0, N_3\}$

In the second example, we consider that there are only two models, either events occur at  $N_1$  and  $N_4$  or at  $N_0$  and  $N_3$ . Now similar to the above situation one can see that when dataset is  $\{Z_N = z_N, \text{ for all } N \in R\}$  then the likelihood of the data under the model  $\{N_1, N_4\}$  is

$$L_{\{N_1, N_4\}} = P_0^{z_0+z_3}(1 - P_0)^{2-(z_0+z_3)} P_1^{(z_1+z_4)}(1 - P_1)^{2-(z_1+z_4)} \times \\ p_w^{z_6+z_{11}+z_{12}+z_{13}+z_{16}+\dots+z_{n-1}}(1 - p_w)^{(n-12)-(z_6+z_{11}+z_{12}+z_{13}+z_{16}+\dots+z_{n-1})} \times \\ P_2^{(z_2+z_5+z_7+z_8+z_9+z_{10}+z_{14}+z_{15})}(1 - P_2)^{8-(z_2+z_5+z_7+z_8+z_9+z_{10}+z_{14}+z_{15})}$$

and the likelihood of the data under the model  $\{N_0, N_3\}$  is

$$L_{\{N_0, N_3\}} = P_1^{z_0+z_3}(1 - P_1)^{2-(z_0+z_3)} P_0^{(z_1+z_4)}(1 - P_0)^{2-(z_1+z_4)} \times \\ P_2^{(z_2+z_5+z_7+z_6+z_{13}+z_{14})}(1 - P_2)^{6-(z_2+z_5+z_7+z_6+z_{13}+z_{14})} \times \\ p_w^{(z_8+\dots+z_{12}+z_{15}+z_{16}+\dots+z_{n-1})}(1 - p_w)^{(n-10)-(z_8+\dots+z_{12}+z_{15}+z_{16}+\dots+z_{n-1})}.$$



Now  $L_{\{N_1, N_4\}} < L_{\{N_0, N_3\}}$  is equivalent to

$$P_0^{z_0+z_3}(1-P_0)^{2-(z_0+z_3)}P_1^{(z_1+z_4)}(1-P_1)^{2-(z_1+z_4)}p_w^{z_6+z_{13}}(1-p_w)^{2-(z_6+z_{13})} \times \\ P_2^{(z_8+z_9+z_{10}+z_{15})}(1-P_2)^{4-(z_8+z_9+z_{10}+z_{15})} < \\ P_1^{z_0+z_3}(1-P_1)^{2-(z_0+z_3)}P_0^{(z_1+z_4)}(1-P_0)^{2-(z_1+z_4)}P_2^{(z_6+z_{13})}(1-P_2)^{2-(z_6+z_{13})} \times \\ p_w^{(z_8+z_9+z_{10}+z_{15})}(1-p_w)^{4-(z_8+z_9+z_{10}+z_{15})}.$$

Which is equivalent to

$$\left(\frac{P_0}{1-P_0}\right)^{(z_0+z_3-z_1-z_4)} \left(\frac{P_1}{1-P_1}\right)^{(z_1+z_4-z_0-z_3)} \times \\ P_2^{(z_8+z_9+z_{10}+z_{15}-z_6-z_{13})}(1-P_2)^{2-(z_8+z_9+z_{10}+z_{15}-z_6-z_{13})} < \\ p_w^{(z_8+z_9+z_{10}+z_{15})-z_6-z_{13}}(1-p_w)^{2-(z_8+z_9+z_{10}+z_{15}-z_6-z_{13})}.$$

Note that as in the above situation, here we have to consider  $\{N_i : i = 0, 1, \dots, 15\} \setminus \{N_{11}, N_{12}\}$  nodes only. Now for all possible  $2^{14}$  choice of  $z_N$  we calculate both sides of the above inequality and check whether that is true or not. Under model  $\{N_1, N_4\}$  we calculate the probability of all possible  $z_N$  and then probability of selecting the model  $\{N_1, N_4\}$  when the model  $\{N_1, N_4\}$  is true. We also do the similar calculation when the model  $\{N_0, N_3\}$  is true. We put them in the Table 8.2.

We also simulate these probabilities for two different situations for different choice of parameters. We consider a  $32 \times 32$  hexagonal grid and run the simulation 10000 times. Both the simulations and exact probability findings are done using Python-code. Results obtained from simulation are summarized in Table 8.2. The columns entitled ‘Simulated’ show the ratio of number of successes to 10000 and the columns entitled ‘Exact’ show the corresponding calculated probabilities.

## Observations

1. We observe that for the above choices of parameters the success probabilities are good enough and simulated success ratio and exact calculated

Table 8.2: Simulated and exact probabilities of selecting a model when the corresponding model is true

Probabilities of selecting models $\{N_0\}$ and $\{N_4, N_5\}$ when the corresponding model is true							
$p_1$	$p_2$	$p_c$	$p_w$	$\{N_0\}$ (Simulated)	$\{N_0\}$ (Exact)	$\{N_4, N_5\}$ (Simulated)	$\{N_4, N_5\}$ (Exact)
0.95	0.70	0.99	0.2	0.9506	0.9514	0.9262	0.9264
0.95	0.70	0.90	0.1	0.9401	0.9392	0.9668	0.9657
0.95	0.65	0.99	0.1	0.9385	0.9404	0.9772	0.9779
0.95	0.65	0.90	0.1	0.9481	0.9463	0.9462	0.9362
0.95	0.65	0.90	0.1	0.9460	0.9463	0.9201	0.9204
0.95	0.65	0.85	0.1	0.9424	0.9416	0.9142	0.9146
0.80	0.70	0.90	0.1	0.9327	0.9524	0.9295	0.9327
0.80	0.70	0.85	0.1	0.9100	0.9134	0.9495	0.9484
0.80	0.65	0.99	0.2	0.9555	0.9527	0.9260	0.9263
0.80	0.65	0.99	0.1	0.9448	0.9477	0.9552	0.9531
0.80	0.65	0.90	0.1	0.9150	0.9120	0.9506	0.9502
0.80	0.65	0.85	0.1	0.9024	0.9025	0.9405	0.9395

Probabilities of selecting models $\{N_0, N_3\}$ and $\{N_1, N_4\}$ when the corresponding model is true							
$p_1$	$p_2$	$p_c$	$p_w$	$\{N_0, N_3\}$ (Sim)	$\{N_0, N_3\}$ (Exact)	$\{N_1, N_4\}$ (Sim)	$\{N_1, N_4\}$ (Exact)
0.99	0.70	0.95	0.2	0.8899	0.8914	0.9539	0.9559
0.99	0.70	0.95	0.1	0.9542	0.9554	0.9662	0.9671
0.95	0.75	0.95	0.1	0.9536	0.9523	0.9847	0.9814
0.95	0.65	0.85	0.1	0.9107	0.9129	0.9321	0.9331
0.95	0.65	0.95	0.1	0.9345	0.9327	0.9595	0.9583
0.95	0.75	0.95	0.2	0.8982	0.8978	0.9765	0.9739
0.88	0.70	0.95	0.1	0.9480	0.9474	0.9738	0.9744
0.88	0.65	0.95	0.1	0.9328	0.9349	0.9663	0.9636
0.88	0.65	0.85	0.2	0.8636	0.8629	0.8793	0.8811
0.88	0.65	0.85	0.1	0.9153	0.9180	0.9342	0.9333
0.88	0.70	0.85	0.1	0.9219	0.9520	0.9497	0.9519
0.88	0.65	0.85	0.1	0.9164	0.9149	0.9358	0.9362

success probabilities are very close for all the chosen values of parameter.

- Success probability decreases when  $p_w$  increases and success probability increases when any one of  $p_c, p_1, p_2$  increases, as desired.
- Probabilities of selection of the model  $\{N_1, N_4\}$  when the model  $\{N_1, N_4\}$  is true are larger than that of the model  $\{N_0, N_3\}$  when the model  $\{N_0, N_3\}$  is true for all the choices of parameters that we have made here.

## Conclusion and Future Work

This thesis dealt with two important aspects of WSNs: the coverage problem and the detection problem. We discussed three types of coverage problem:

1. Coverage on a continuous domain in a random deployment scenario,
2. Coverage using sensor relocation by an actuator,
3. Deterministic placement of sensors in a cylindrical grid.

We also discussed three different scenarios for the detection problem:

1. At most one event can occur at a specific point in ROI,
2. At most one event can occur but it may occur at any point in the ROI,
3. More than one event can occur at any points of the ROI.

### **Coverage Problem**

#### **Random deployment problem**

Considering the region of interest (ROI) as a continuous bounded set and sensor can detect an event in a disc centered at that sensor, we found that hexagonal placement of sensors is the optimal one. We also considered that sensors are deployed at random from air, so they may not be placed at the

target points. Hence some portion of the ROI may not be covered by any sensor. We used extra sensors to minimize the uncovered area. We developed two different strategies to reduce the uncovered area. The distance between the target point and the point where the corresponding sensor has been placed is a random variable. We considered those random variables to follow a i.i.d. uniform or normal distribution. For uniform we calculated the uncovered area of ROI. For both the distributions and strategies we simulated uncovered area computationally. We considered the region of interest as a three dimensional bounded region and developed two strategies to reduce the uncovered volume. We noticed that there are three different aspects of the minimal wastage problem:

1. One has to fix a partition and choose some vertices where the sensors will be targeted to deploy. Popular methods are square and hexagonal placement for two dimensions and we use face centered cube placement for three dimensions. Hexagonal placement is in some sense optimal when there are no extra sensors and no randomness in deployment.
2. The placement of sensors is random in many situations, one may model or fit a distribution from experimental data.
3. When there are extra sensors we developed two strategies of deployment for these extra sensors, but there may be many others. Search for an optimal one depending on different distributions and different methods for placement of sensors.

In future, one may try to find theoretical results for normal distribution and will consider the minimal wastage problem for higher dimensions to find optimal placement of sensors. We will consider ROI as a square grid in two dimensions and with other interesting distributions. Here we consider only two strategies but there may be other strategies which may be better for some specific distributions. One may try to classify them with respect to uncovered volume for different distributions and different type of partitions. One may develop algorithms for actuator(s), using extra sensors with different strategies.

### Deterministic placement of sensors in a cylindrical grid

We considered the region of interest as a cylindrical grid. Events can occur at a vertex. A sensor may be placed at a vertex and can detect an event at adjacent vertices. We solved the coverage problem in the cylindrical grid using graph theory. We found  $\gamma(P_m \square C_n)$ , for  $m = 2, 3$  and  $4$ ,  $n \geq 3$ . Minimum dominating sets corresponding to the above mentioned graphs are also constructed. Finally, we give bounds on  $\gamma(P_5 \square C_n)$ ,  $n \geq 3$ .

In [75], Nandi et. al. found the exact values of  $\gamma(P_5 \square C_n)$ ,  $n \geq 3$  as well as minimum dominating sets for  $P_5 \square C_n$ ,  $n \geq 3$ . Moreover, they found the bounds of  $\gamma(P_m \square C_n)$ , for  $m \geq 6$  and  $n \geq 3$ . In future one may find the exact values of  $\gamma(P_m \square C_n)$ ,  $n \geq 3$  as well as minimum dominating sets for  $P_m \square C_n$ , for  $m \geq 6$  and  $n \geq 3$ .

There is no work on random deployment as well as the use of an actuator for the cylindrical grid graph. This area is totally open for research.

### Placement problem using actuators

After deployment of sensors, sensors are relocated to cover the ROI. We consider the static sensors only. An actuator will relocate the sensors to cover the ROI. We developed three algorithms for the actuator in a sensor node deployment scenario, under two different conditions. We simulated and analyzed the simulation results. We also deduced some theoretical results and compared with the simulation results. We observed that if an actuator placed the sensors using the third algorithm, the number of empty nodes after the placement becomes almost zero. We also observed that number of empty nodes after deployment is approximately normally distributed for large grids and for repetition probability greater than 0.2 and error probability less than 0.25. We observed the traversed length of the actuator and number of empty nodes decreases as error probability decreases or repetition probability increases.

In future, one may develop other randomized algorithms, and compare them in a similar way. One may also impose more conditions on the actuator or can relax some conditions, e.g., one may assume that the actuator may carry more than one sensor or there is more than one actuator in practice.

## Detection problem

### **At most one event can occur at a specific point in ROI**

After covering the ROI, sensors are used to detect one or more events. At first, we considered the problem of fault detection in WSN when the ROI is a rectangular grid with square cells. We discussed how to address both noise-related measurement error and sensor fault simultaneously in fault detection, where the sensors are placed at the centers of square cells of the ROI and an event may occur at a specific square of the grid. We also considered the ROI partitioned into regular hexagonal cells and did the same analysis. We proposed fault detection schemes that explicitly introduce the error probability into the optimal event detection process. We developed the schemes under classical hypothesis testing and a Bayes test. We identified and analyzed all the situations in which these tests are effective and cases where they are not applicable.

We observed that type I and type II errors decrease when detection probabilities and probability of sending correct information of occurrence of the event increase and errors increase when the probability of sending wrong information of normal situation increases. When probability of occurrence of the event increases, type I errors increase but type II errors decrease. If detection probability is low then type I error is close to probability of occurrence of the event. If the probability of occurrence of the event is close to 0.5 then type I error is close to probability of occurrence of the event, which means that there is no use of sensors; in that case, we have to use sensors with high detection probability.

In future, one may do the following:

1. Develop schemes to find which particular square is the event square.
2. Develop schemes to find and isolate dead and faulty sensors, i.e., the sensors which are sending false information to the base station.
3. We may assume that sensors can detect different types of events; thus, the response of sensors may not be simply binary.
4. We may assume that sensors can measure distance, direction, speed,

humidity, wind speed, soil makeup, temperature, etc. and send the measurement of continuous type variables.

### **At most one event can occur but it may occur at any point in ROI**

We considered the problem of fault detection in wireless sensor network (WSN), where the ROI is partitioned into regular hexagons with the event occurring at only one hexagon but not at a specific hexagon. We proposed fault detection schemes that explicitly introduce the error probabilities into the optimal event detection process. We developed the schemes under the consideration of a model selection technique, multiple model selection technique and Bayesian model averaging method. The different error probabilities are calculated by means of simulation. We noticed that the same analysis can be carried out when ROI is partitioned into squares as well.

In the model selection approach, we select the model with higher likelihood. In the classical Neyman-Pearson hypothesis test, a model is selected if its likelihood is greater than some constant times the likelihood of the other. This constant is fixed before the test depending on the size of the test. In model selection approach, the constant is 1, leaving no choice for the size of the test. On the other hand, we cannot apply the classical Neyman-Pearson test with more than two models to be considered for selection. The principle of hypothesis testing places a large confidence in the null hypothesis and does not reject it unless there is strong evidence against it. This safeguard of null hypothesis cannot be ensured in the model selection approach of detection. The multiple model selection approach of detection provides some safeguard in this regard.

In future, the principle of model selection can be extended to the situation when there are two or more event hexagons and the objective is to detect the event hexagons. One may also assume that the sensors can detect different types of events. That is, response of sensors may not be binary; sensors can measure continuous type variables and report them to the base station. One needs a different formulation of the problem.

### More than one event can occur at any points of ROI

Detection of an event in the ROI in the general case is a challenging problem. Finding theoretical results in general situation is hard. One has to simulate different situations or study special situations. We introduced the sensor fault probabilities into the optimal event detection process. We apply a near-optimal model selection approach and a new method to find a solution of the problem. We also discuss two interesting situations. We simulated different situations with different parameters. One may consider the situation when the sensing radii are larger and more sensors can detect the event hexagon but with different probabilities. Classify the nodes with respect to the probability of detecting the event at a node, which may as well depend on the distance from the particular node. Suppose that the sensors in the  $i$ -th class detect the event hexagon with probability  $p_i, i = 1, 2, 3, \dots$ . The theoretical analysis is similar but having more probability terms.

In future, the challenge is to detect events in the general situation more efficiently. Also we have to develop a strategy for the base station when sensor can measure one or more continuous variables not merely a binary one.

Coverage and detection are two important area of WSNs but there are many other aspects of WSNs. One closely related area is data reliability of Unattended WSNs. While the network is unattended, a mobile adversary can migrate between compromised set of sensors and inject fraudulent data or erase data [51, 80, 81, 83, 84]. In future one may combine this with fault detection.

In this thesis, we presented our results from the last four years (2010-2014) on the analysis of coverage and detection problem in WSNs or WSANs. We identified long-standing open problems as well as contemporary results from the literature, and provided answers to six research problems. During the course of this thesis, we also studied several related problems on WSN, and presented some open problems for future research. We believe that the plethora of problems arising from the WSN will continue to amaze and motivate the community for years to come.



# Bibliography

- [1] D. P. Agrawal, M. Lu, T. C. Keener, M. Dong and V. Kumar. “Exploiting the use of WSNs for Environmental Monitoring”. *EM Magazine*, 27-33, 2004.
- [2] D. P. Agrawal. “Designing Wireless Sensor Networks: from theory to applications”. *Central European Journal of Computer Science*, 1(1): 2-18, 2011.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. “Wireless Sensor Networks: A Survey”. *Computer Networks*, 38(4): 393-422, 2002.
- [4] X. Bai, S. Kumary, D. Xuan, Z. Yun and T. H. Lai. “Deploying wireless sensors to achieve both coverage and connectivity”. In *Proc. 7th Int’l Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Florence, Italy*, 131-142, 2006.
- [5] M. A. Batalin and G. S. Sukhatme. “The analysis of an efficient algorithm for robot coverage and exploration based on sensor network deployment”. In *Proc. IEEE Int’l Conference on Robotics and Automation (ICRA), Barcelona, Spain*, 3489-3496, 2005.
- [6] L. Booth, J. Bruck, M. Franceschetti and R. Meester. “Covering algorithms, continuum percolation and the geometry of wireless networks”. *Annals of Appl. Probability*, 13(2): 722-741, 2003.

## BIBLIOGRAPHY

---

- [7] N. Bulusu, J. Heidemann and D. Estrin, “GPS-less Low Cost Outdoor Localization for Very Small Devices”. *IEEE Personal Communications*, 7(5): 28-34, 2000.
- [8] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao. “Habitat Monitoring: Application Driver for Wireless Communications Technology”. *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.
- [9] M. Chaudhary A. K. Pujari. “Q-Coverage Problem in Wireless Sensor Networks”. *Proceeding ICDCN '09 Proceedings of the 10th International Conference on Distributed Computing and Networking*; 325 - 330, 2009.
- [10] T. Y. Chang and W. E. Clark. “The Domination numbers of the  $5 \times n$  and  $6 \times n$  grid graphs”. *J. Graph Theory*, 17: 81-107, 1993.
- [11] J. Chen, S. Kher and A. Somani. “Distributed Fault Detection of Wireless Sensor Networks”. *DIWANS '06 Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, ACM press, New York, 65-72, 2006.
- [12] K. Chintalapudi and R. Govidan. “Localized Edge Detection in Sensor Fields”. *Proc. IEEE Int'l Workshop Sensor Network Protocols and Applications*, 273-291, 2003.
- [13] B. N. Clark, C. J. Colbourn and D. S. Johnson. “Unit disk graphs”. *Discrete Math.*, 86: 165-177, 1990.
- [14] H. Cohn. “New upper bounds on sphere packings II”. *Geometry and Topology*, 6: 329-353, 2002.
- [15] H. Cohn and N. Elkies. “New upper bounds on sphere packings I”. *Annals of Mathematics*, 157: 689-714, 2003.
- [16] C. Collins and K. Stephenson. “A circle packing algorithm”. *Comput. Geom.*, 25(3): 233-256, 2003.
- [17] J. H. Conway and N. J. A. Sloane. “Sphere packings, Lattices and Groups”. Third edition, Springer-Verlag, 1999.

- [18] J. Cortes, S. Martinez, T. Karatas and F. Bullo. "Coverage control for mobile sensing networks". *IEEE Trans. Robotics and Automation*, 20(2):243-255, 2004.
- [19] J. Cortes, S. Martinez, T. Karatas and F. Bullo. "Coverage control for mobile sensing networks". *IEEE Trans. Robotics and Automation*, 20(2): 243-255, 2004.
- [20] G. K. Das, S. Das, S. C. Nandy and B. P. Sinha. "Efficient algorithm for placing a given number of base stations to cover a convex region". *Journal of Parallel and Distributed Computing*, 66: 1353-1358, 2006.
- [21] P. Delsarte. "Bounds for unrestricted codes, by linear programming". *Philips Res. Rep.*, 27: 272-289, 1972.
- [22] M. H. El-Zahar, S. M. Khamis and Kh. M. Nazzal. "On the domination number of the Cartesian product of the cycle of length  $n$  and any graph". *Discrete Appl. Math.*, 155: 515-522, 2007.
- [23] M. H. El-Zahar and C. M. Pareek. "Domination number of products of graphs". *Ars Combin.*, 31: 223-227, 1991.
- [24] R. Falcon, X. Li, and A. Nayak. "Carrier-Based Focused Coverage Formation in Wireless Sensor and Robot Networks". *IEEE Trans. on Automatic Control*. 56(10): 2406–2417, 2011.
- [25] C. Farah, F. Schwaner, A. Abedi and M. Worboys. "A Distributed Homology algorithm to Detect Topological Events via Wireless Sensor Networks". *Wireless Sensor Systems, IET*, 1(3): 151-160, 2011.
- [26] R. J. Faudree and R. H. Schelp. "The domination number for the product of graphs". *Congr. Numer.*, 79: 29-33, 1990.
- [27] A. Filippou, D. A. Karras and R. C. Papademetriou. "Coverage Problem for Sensor Networks: An Overview of Solution Strategies". *Proc. 17th Telecommunication Forum (TELFOR), Serbia, Belgrade*, 134-136, 2009.
- [28] G. Fletcher, X. Li, A. Nayak and I. Stojmenovic. "Randomized Robot-assisted Relocation of Sensors for Coverage Repair in Wireless Sensor

## BIBLIOGRAPHY

---

- Networks”. *Proc. IEEE 72nd Vehicular Technology Conference (VTC), Ottawa, Canada*, 1-5, 2010.
- [29] G. Fletcher, X. Li, A. Nayak, and I. Stojmenovic. “Randomized Robot-assisted Relocation of Sensors for Coverage Repair in Wireless Sensor Networks”. *IEEE Comm. Lett.*, 2010.
- [30] A. Gallais, J. Carle, D. Simplot-Ryl and I. Stojmenovic. “Localized Sensor Area Coverage with Low Communication Overhead”. *IEEE Transactions on mobile Computing*, 7(5): 661-672, 2008.
- [31] J. K. Ghosh, M. Delampady and T. Samanta. “An introduction to Bayesian Analysis, Theory and Methods”, Springer, 2011.
- [32] T. Hales. The Kepler conjecture. *arXiv:math.MG/9811078.*, 1998.
- [33] F. Harary. *Graph Theory, Addison Wesley, Reading Mass*, 1969.
- [34] B. L. Hartnell and D. F. Rall. “On Vizing’s conjecture”. *Congr. Numer.*, 82: 87-96, 1991.
- [35] A. Heppes and J. B. M. Melissen. Covering a rectangle with equal circles. *Period. Math. Hungar.*, 34: 65-81, 1997.
- [36] J. A. Hoeting, D. Madigan, A. E. Raftery and C. T. Volinsky. “Bayesian Model Averaging: A Tutorial”. *Statistical Science*, 14(4): 382-417, 1999.
- [37] S. Hore, A. Dewanji and A. Chatterjee. “Design Issues Related to Allocation of Experimental Units with Known Covariates into Two Treatment Groups”. Accepted JSPI, 2015.
- [38] A. Howard and S. Poduri. “Potential Field Methods for Mobile Sensor Network Deployment”. In *N. Bulusu and S. Jha, Wireless Sensor Networks A System Perspective, Artech House, London*, 136, 2005.
- [39] C. F. Huang and Y. C. Tseng. “The Coverage Problem in a Wireless Sensor Network”. *Mobile Networks and Applications*, 10(4): 519-528, 2005.
- [40] C. F. Huang and Y. C. Tseng. “The Coverage Problem in a Wireless Sensor Network”. *WSNA '03*, 2003.

- 
- [41] M. S. Jacobson and L. F. Kinch. “On the domination of the products of graphs I”. *Ars Combin.*, 18: 33-44, 1983.
- [42] O. Khatib. “Real time obstacle avoidance for manipulators and mobile robots”. *International Journal of Robotics Research*, 5(1): 90-98, 1986.
- [43] S. Klavzar and N. Seifter. “Dominating Cartesian products of Cycles”. *Discrete Appl. Math.* 59: 129-136, 1995.
- [44] E. Kranakis, D. Krizanc, O. M. Ponce, L. Narayanan, J. Opatrny and S. Shende. “Expected Sum and Maximum of Displacement of Random Sensors for Coverage of a Domain”. *Proceedings of the 25th ACM symposium on Parallelism in algorithms and architectures, New York, USA*, 73-82, 2013.
- [45] B. Krishnamachari and S. Iyengar. “Distributed Bayesian Algorithms for fault-tolerant Event Region Detection in Wireless Sensor Network”. *IEEE Trans. on Computers*, 53(3): 241-250, 2004.
- [46] S. Li, S. Chen, B. Liu, Y. Li and Y. Liang. “Decentralized kinematic control of a class of collaborative redundant manipulators via recurrent neural networks”. *Neurocomputing*, 91: 1-10, 2012.
- [47] S. Li, H. Cui, Y. Li, B. Li and Y. Lou. “Decentralized Control of Collaborative Redundant Manipulators with Partial Command Coverage via Locally Connected Recurrent Neural Networks”. *Neural Computing and Applications*, 23: 1051-1060, 2012.
- [48] X. Li, H. Frey, N. Santoro and I. Stojmenovic. “Localized sensor self deployment with coverage guarantee”. *ACM SIGMOBILE Mobile Computing and Communication Review*, 12(2): 50-52, 2008.
- [49] X. Li, H. Frey, N. Santoro and I. Stojmenovic. “Localized sensor self deployment for guaranteed circular coverage radius maximization”. In *Proc. IEEE Int'l Conference on Communications (ICC)*, 2009.
- [50] X. Li, G. Fletcher, A. Nayak, and I. Stojmenovic. “Randomized carrier based sensor relocation in wireless sensor and robot networks”. *Ad Hoc Networks*. 11: 1951-1962, 2013.

## BIBLIOGRAPHY

---

- [51] X. Li, X. Liang, R. Lu, S. He, J. Chen and X. Shen. “Toward Reliable Actor Services in Wireless Sensor and Actor Networks”. *Eight IEEE Int. Conference on Mobile Ad-Hoc and Sensor Systems*, 2011.
- [52] S. Li, B. Liu, B. Chen and Y. Lou. “Neural network based mobile phone localization using Bluetooth connectivity”. *Neural Computing and Applications*, 23(3-4): 667-675, 2013.
- [53] S. Li, Y. Li and Z. Wang. “A class of finite-time dual neural networks for solving quadratic programming problems and its k-winners-take-all application”. *Neural Networks* 39: 27-39, 2013.
- [54] S. Li, B. Liu and Y. Li. “Selective Positive-Negative Feedback Produces the Winner-Take-All Competition in Recurrent Neural Networks”. *IEEE Transactions on Neural Networks and Learning Systems*, 24(2): 301-309, 2013.
- [55] D. Li, H. Liu, X. Lu, W. Chen and H. Du. “Target Q-Coverage Problem with Bounded Service Delay in Directional Sensor Networks”. *International Journal of Distributed Sensor Networks*, Volume 2012, Article ID 386093, 2012.
- [56] D. Li, H. Liu, X. Lu, W. Chen and H. Du. “Target Q-Coverage Problem with Bounded Service Delay in Directional Sensor Networks”. *International Journal of Distributed Sensor Networks*, Volume 2012, Article ID 386093, 2012.
- [57] X. Li, A. Nayak, D. Simplot-Ryl and I. Stojmenovic. “Sensor Placement in Sensor and Actuator Networks”. *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, A. Nayak and I. Stojmenovic (eds.), John Wiley & Sons, Hoboken, New Jersey, Chapter 10: 263-294, 2010.
- [58] S. Li and F. Qin. “A dynamic neural network approach for solving non-linear inequalities defined on a graph and its application to distributed, routing-free, range-free localization of WSNs”. *Neurocomputing*, 117: 72-80, 2013.

- [59] S. Li, Z. Wang and Y. Li. “Using Laplacian Eigenmap as Heuristic Information to Solve Nonlinear Constraints Defined on a Graph and Its Application in Distributed Range-Free Localization of Wireless Sensor Networks”. *Neural Process Letters*, 37: 411-424, 2013.
- [60] X. Li, S. Yan, C. Xu., A. Nayak and I. Stojmenovic. “Localized delay bounded and energy-efficient data aggregation in wireless sensor and actor networks”. *Wireless Communications and Mobile Computing* 11(12): 1603-1617, 2011.
- [61] H. Liu, A. Nayak and I. Stojmenovic. “Sensor Area Coverage”. *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, A. Nayak and I. Stojmenovic (eds.), John Wiley & Sons, Hoboken, New Jersey, Chapter 3: 75-93, 2010.
- [62] H. Liu, A. Nayak and I. Stojmenovic. “Application, Models, Problems, and solution Strategies”. *Wireless Sensor and Actuator Networks: Algorithms and Protocols for Scalable Coordination and Data Communication*, John Wiley & Sons, 2010.
- [63] X. Luo, M. Dong and Y. Huang. “On Distributed Fault-Tolerant Detection in Wireless Sensor Networks”. *IEEE Trans. on Computers*, 55(1): 58-70, 2006.
- [64] D. Madigan and A. E. Raftery. “Model selection and accounting for model uncertainty in graphical models using Occam’s window”. *J. Amer. Statistical Assoc.* 89: 1535-1546, 1994.
- [65] D. Madigan and J. York. “Bayesian graphical models for discrete data”. *Internat. Statist. Rev.*, 63: 215-232, 1995.
- [66] F. Martincic and L. Schwiebert. “Introduction to Wireless Sensor Networking”. *Handbook of Sensor Networks: Algorithms and Architectures* (Chapter 1, pp. 1-40), Ivan Stojmenovic (ed), John Wiley & Sons, Hoboken, New Jersey, 1-40, 2005.

## BIBLIOGRAPHY

---

- [67] S. Meguerdichian and M. Potkonjak. “Low power 0/1 coverage and scheduling techniques in sensor networks”. *Tech. Report 030001, UCLA*, 2003.
- [68] A. M. Mood, F. A. Graybill and D. C. Boes. “Introduction to the theory of Statistics”. McGraw-Hill, Singapore, 1974.
- [69] H. Mousavi, A. Nayyeri, N. Yazdani and C. Lucas. “Energy conserving movement assisted deployment of Ad hoc sensor networks”. *IEEE Comm. Lett.*, 10(4): 269-271, 2006.
- [70] M. Nandi, A. Dewanji, B. Roy and A. Baksi. “Different aspects of Fault Detection in Wireless Sensor Networks”. Communicated (pre-print).
- [71] M. Nandi and X. Li. “Coverage problem of sensor networks”. *Ad Hoc and Sensor Wireless Networks*, 13(1-2): 105-120, 2011.
- [72] M. Nandi, B. Roy, A. Nayak and S. sarkar. “Decision Theoretic Approach for Fault Detection in Wireless sensor Networks”. *International Journal of Parallel, Emergent and Distributed Systems*, 2014. <http://dx.doi.org/10.1080/17445760.2014.910514>.
- [73] M. Nandi, B. Roy, A. Dewanji and S. Sarkar. “Model Selection Approach for Distributed Fault Detection in Wireless Sensor Networks”. *International Journal of Distributed Sensor networks*, Volume 2014, Article ID 148234, 2014.
- [74] M. Nandi, S. Parui and A. Adhikary. “The domination numbers of cylindrical grid graphs”. *Applied Mathematics and Computation*, 217: 4879-4889, 2011.
- [75] M. Nandi, S. Parui and A. Adhikary. “Bounds on the size of the minimum dominating sets of some cylindrical grid graphs”. *International Journal of Combinatorics*, Volume 2014, Article Id. 348359, 2014.
- [76] M. Nandi and S. Sarkar. “On Coverage Problem with application in WSNs”. Communicated (pre-print).



- [77] A. Nayak and I. Stojmenovic. "Wireless Sensor and Actuator Networks, Algorithms and Protocols for Scalable Coordination and Data Communication". John Wiley & Sons, 2010.
- [78] K. Ni, N. Ramanathan, M. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen and M. Srivastava. "Sensor Network Data Fault Type". *ACM Transactions in Sensor Networks*, 5(3): 1-29, 2009.
- [79] R. Nowak and U. Mitra. "Boundary Estimation in Sensor Networks: Theory and Methods". In *Proc. 2nd International Workshop on Information Processing in Sensor Networks, Palo Alto, CA*, 1-16, 2003.
- [80] R. Di Pietro, L. V. Mancini, C. Soriente, A. Spognardi and G. Tsudik. "Data Security in Unattended Wireless Sensor Networks". *IEEE Transactions on Computers*, Vol. 58, No. 11, 2009.
- [81] R. Di Pietro, C. Soriente, A. Spognardi and G. Tsudik. "Intrusion-resilient integrity in data-centric unattended WSNs". *Pervasive and Mobile Computing*, 7: 495-508, 2011.
- [82] G. J. Pottie and W. J. Kaiser. "Wireless Integrated Network Sensors". *Communications of the ACM*, 43(5): 51-58, 2000.
- [83] S.K.V.L. Reddy, S. Ruj and A. nayak. "Distributed data survivability schemes in mobile Unattended Wireless Sensor Networks". *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012.
- [84] S.K.V.L. Reddy, S. Ruj and A. nayak. "Data Authentication Scheme for Unattended Wireless Sensor Networks against a Mobile Adversary". *WCNC*, 1836-1841, 2013.
- [85] S. Seetharaman and A. A. Minai. "Self organized scheduling of node activity in large scale sensor networks". <http://rave.ohiolink.edu/etdc/view?acc,num=ucin1092939502>
- [86] A. B. Sharma, L. Golubchik and R. Govindan. "Sensor Faults: Detection Methods and Prevalence in Real-World dataset". *ACM Journal on Sensor Networks*, Vol: 6, Number 3, Article 23, June 2010.

## BIBLIOGRAPHY

---

- [87] V. de Silva and R. Ghrist. “Coverage in Sensor Networks via Persistent Homology”. *Algebraic & Geometric Topology*, 7: 339-358, 2007.
- [88] V. de Silva and R. Ghrist. “Coordinate-Free Coverage in Sensor Networks with controlled boundaries via Homology”. *Intl. Journal of Robotics Research*, 2007.
- [89] V. de Silva, R. Ghrist and Abubakr Muhammad. “Blind Swarms for Coverage in 2-D”. *Proceedings of Robotics: Science and Systems*, 2005.
- [90] S. Slijepcevic, M. Potkonjak. “Power efficient organization of wireless sensor networks”. *Proceedings of IEEE International Conference on Communications*, 2: 472-476, 2001.
- [91] I. Stojmenovic. “A General Framework for Broadcasting in Static to Highly Mobile Wireless Ad hoc, Sensor, Robot and Vehicular Networks”. *IEEE 18th International Conference on Parallel and Distributed Systems, Singapore*, 2012: 8-13.
- [92] T. Tiarnai and Z. Gasper. “Covering a square by equal circles”. *Elementary Math.*, 50: 167-170, 1995.
- [93] J. N. Tsitsiklis. “Decentralized Detection”. *Advances in Statistical Signal Processing*, 2:297-344, 1993.
- [94] P. K. Varshney. “Distributed Detection and Data Fusion”. Springer-Verlag, New York, 1991.
- [95] V. G. Vizing. “The Cartesian product of graphs”. *Vychisl Sistemy* 9: 30-43, 1963.
- [96] R. Williams. “The geometrical foundation of natural structure”. A Source Book of Design, Dover Pub. Inc., New York, 1979.
- [97] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4): 621-655, 2008.

- [98] M. Yun, D. Bragg, A. Arora and H. Choi. “Battle Event Detection using Sensor Networks and Distributed Query Processing”. *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference*, 750-755, 2011.
- [99] Z. Zhou, S. Das and H. Gupta. “Connected k-coverage problem in sensor networks”. *Proc. Int. Conf. Computer Communications and Networks (ICCCN)*, 373-378, 2004.
- [100] Z. Zhou, S. Das, H. Gupta. “Connected K-Coverage Problem”. *Proceedings of the International Conference on Computer Communications and Networks (IC3N)*, 2004.