M.Sc. Engg. Thesis

A TARGET MONITORING ALGORITHM IN DIRECTIONAL SENSOR NETWORKS CONSIDERING LIFETIME AND FAULT TOLERANCE

by
Suman Saha (1014052070 P)

Submitted to

Department of Computer Science & Engineering

(In partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering)

Department of Computer Science & Engineering

Bangladesh University of Engineering & Technology (BUET)

Dhaka 1000

July 13, 2019

Dedicated to my loving parents

Author's Contact

Suman Saha

Email: sumancsecu04@gmail.com

The thesis titled " A TARGET MONITORING ALGORITHM IN DIRECTIONAL SENSOR NETWORKS CONSIDERING LIFETIME AND FAULT TOLERANCE ", submitted by Suman Saha, Roll No. **1014052070 P**, Session October 2014, to the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Master of Science in Computer Science & Engineering and approved as to its style and contents. Examination held on July 13, 2019.

Board of Examiners

1. ______________________________

Dr. A.K.M. Ashikur Rahman — Chairman (Supervisor)
Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

2. ______________________________

Dr. Md. Mostofa Akbar — Member (Ex-Officio)
Head and Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

3. ______________________________

Dr. Md. Shohrab Hossain — Member
Associate Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

4. ______________________________

Dr. Sadia Sharmin — Member
Assistant Professor
Department of Computer Science & Engineering
Bangladesh University of Engineering & Technology, Dhaka.

5. ______________________________

Dr. Mohammad Rashedur Rahman — Member (External)
Professor
Department of Electrical and Computer Engineering
North South University, Dhaka.

Candidate's Declaration

This is hereby declared that the work titled "A TARGET MONITORING ALGORITHM IN DIRECTIONAL SENSOR NETWORKS CONSIDERING LIFETIME AND FAULT TOLERANCE", is the outcome of research carried out by me under the supervision of Dr. A.K.M. Ashikur Rahman, in the Department of Computer Science & Engineering, Bangladesh University of Engineering & Technology, Dhaka 1000. It is also declared that this thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Suman Saha

Candidate

Acknowledgment

First and foremost, I am grateful and expressing my gratefulness to Almighty God for providing me his spiritual blessing, patient, mental and physical strength throughout my research to complete this thesis successfully.

I would like to express my sincere gratitude to my supervisor, Prof. Dr. A.K.M. Ashikur Rahman, for his constant support, valuable advice, helpful criticism, and unending patience throughout this research. Without his guidance and motivation, it was impossible to accomplish this thesis.

I like to thank the honorable members of my thesis committee: Prof. Dr. Md. Mostofa Akbar, Dr. Md. Shohrab Hossain, Dr. Sadia Sharmin, and especially the external member Prof. Dr. Mohammad Rashedur Rahman, for their encouragements, insightful judgments, and valuable suggestions.

I am thankful to Abdullah Al Zishan (Graduate student, CSE-BUET). I sought help from him many times regarding LP formulation and simulation set-up of this research. I am also grateful to all respected teachers of CSE department for their kind cooperation.

Last but not the least, I remain immensely grateful to my beloved parents, lovely wife, and family members for their love, care, sacrifices and constant encouragements behind every success of mine.

Abstract

Target coverage by directional sensor networks (DSNs) is a fundamental problem where targets should be covered by at least one sensor. During coverage, prolonging network lifetime and providing fault tolerance are the major challenges. To tackle the first challenge, sensors are divided into multiple set covers each of which is capable of monitoring all the targets and schedule them in a round-robin fashion. Therefore, maximizing the number of set covers can lead to a longer network lifetime. Interestingly, the number of set covers is increased when the sensors are allowed to overlap within the set covers. Thus in one extreme, one can attain maximum lifetime by allowing the sensors to overlap boundlessly. Besides lifetime, fault tolerance is another important aspect which has been overlooked by most of the research works. Practically, sensor nodes are very much prone to failure and difficult to be replaced. If a particular sensor dies out, then all set covers containing the faulty sensor get affected. This indicates another extreme: increasing the sensor overlap within set covers decrease fault tolerance of the network. In this thesis, we address these two antithetical approaches and provide a solution to resolve this problem. We find the maximum possible number of set covers by activating a subset of sensors within a given upper bound on sensor overlap. A sensor can not participate within the set covers more than this bound value. To solve this problem, at first we formulate the problem as a Linear Programming (LP) problem which gives the optimal solution. As LP formulation is computationally expensive, we develop two approximation greedy algorithms which are solvable in polynomial time. We also investigate the performance of our proposed heuristics with respect to different performance metrices through extensive simulations.

Acronyms List

WSNs = Wireless Sensor Networks

DSNs = Directional Sensor Networks

FOV = Field of View

AOV = Angle of View

AOI = Area of Interest

PTZ = Pan, Tilt and Zoom

MCMS = Maximum Coverage using Minimum Sensors

MCRS = Maximum Coverage with Rotatable Sensors

BS = Base Station

GPS = Global Positioning System

TIS = Target in Sector

LP = Linear Programming

TOGH = Target Oriented Greedy Heuristic

SOGH = Sensor Oriented Greedy Heuristic

SCO = Set Cover Optimization

GI = Goodness Index

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In recent time, wireless sensor networks (WSNs) have attracted remarkable research interest due to its extensive usage in various applications such as environmental monitoring, traffic control, battlefield surveillance, tracking, smart spaces etc.[1] [2] [3] [4]. It is an auto-configured network which can be applied to measure critical physical parameters from the real world especially the hostile environment where the wired connections are not possible. Generally, a wireless sensor network is composed of a large number of inexpensive, low powered smart sensors nodes each of which is capable of self-controlling its field of view (FOV) to optimize the coverages of areas of interest (AOI). However, sensor nodes in WSNs are connected with each other via wireless links. Each sensor node has a sensing unit which is operated to collect the interesting data. Additionally, a wireless transceiver has incorporated with a sensor node which is employed to forward the captured data back to the base station (BS) called the sink node for further processing [5][6]. Usually, in WSNs, a sensor can be in one of three possible states:– active state, sleep state, and off state. In the active state, a sensor can transmit or receive the signal, and therefore energy is consumed [7][8]. On the other hand, the power of a sensor is not decayed in the sleep state. When total or entire power of a sensor gets decayed, it goes from current state to off state (or permanently turned off). Mostly sensor nodes expend majority of their power in data sensing, processing, and transmitting to perform tasks. Basically, there are two types of sensor in WSNs:– omnidirectional sensor, and directional sensor. Omni-directional sensors always have an omni angle (360^o) field of view and circular disk of sensing range. As opposed to, due to technical restrictions, directional sensors may have a limited field of view (i.e. finite section of sensing angle). Therefore, directional sensors cannot sense the entire circular disk. Currently, in directional sensor networks (DSNs), smart mechanically operated PTZ (Pan, Tilt, and Zoom) cameras are used as sensor nodes [9]. Users of PTZ camera can change the pan (Left, Right), tilt (Up and Down) and zoom using hardware or software devices to give the best coverage.

1.1 Background

Wireless sensor network constitutes the platform of extensive applications. The target coverage is a fundamental research issue in WSN. It measures the Quality of Service (QoS) that is how well the targets are being monitored by a set of randomly deployed sensors (directional)[10][11]. For example, in a video monitoring system of a shopping mall or any like others, some key positions like parking lot, entrances and exits etc. are required continuous coverage without any interruption to get the accurate data. Here, each target point in the sensor network must be within atleast one sensors' sensing range. In addition, if the monitoring targets have some specific requirements, then the targets also require to be monitored by more number of sensors for improving the sampling rate. In industrial environment, industrial wireless sensor network is a powerful monitoring tool for the industrial automation system [12]. Here, sensors are placed to monitor the industrial equipment as well as critical parameters to ensure regular operation. Therefore, industrial wireless sensor networks ensure continuous target coverage and connectivity between the sensor nodes with the base station.

Generally, the sensor networks can be defined as over-provisioned networks and under-provisioned networks [9]. A network is called over-provisioned network if given sensors are sufficient to cover all the targets; otherwise, the network is under-provisioned. Here, only over-provisioned networks have been considered. When a set of sensors are used to cover a particular set of targets, it is expected that some sensors share their common sensing region. In such scenarios, it is unnecessary and redundant to activate all the sensors at a time instead of activating a subset of them. For directional sensor network, Ai and Abuzeid [13] proposed Maximum Coverage with Minimum Sensors (MCMS) problem, where some randomly deployed

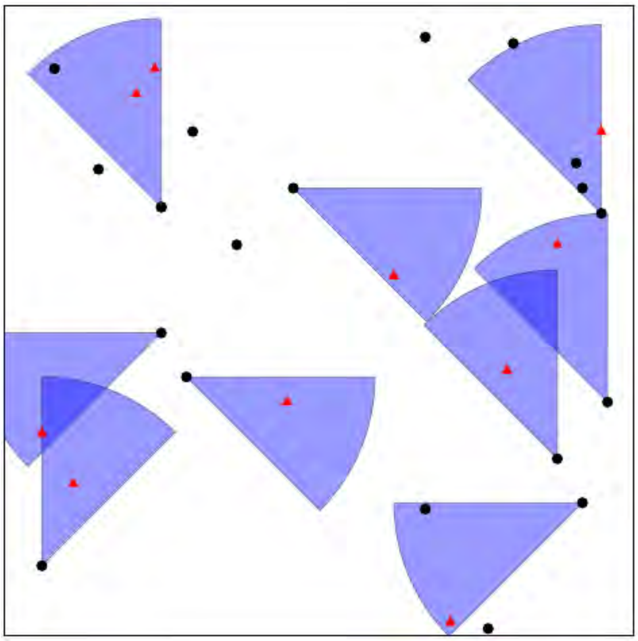


Figure 1.1: A random deployment scenario

targets should be given single coverage by activating minimum directional sensors. A situation may be a arises during coverage that some targets may not be possible to cover at a time by the randomly deployed sensors in the environment due to sensors' limited field of view. As a solution to this problem, Hsu et al. [14] proposed the Maximum Coverage with Rotatable Sensors (MCRS) problem where targets to be covered are maximized while minimizing the total rotated degrees. Usually, sensing coverage region of the sensor nodes may undergo from noise, co-channel obstructions, and multipath dispersion. Additionally, battery power or energy is the principal constraint of the sensor nodes. Since the sensor nodes are equipped with limited battery power (or energy), prolonging the network lifetime by effective utilization of sensors have been a significant research issue in DSNs [15][16]. Figure 1.1 shows a typical scenario of target coverage problem in a directional sensor networks (DSNs). Red triangles represent the targets and black circles indicate the sensors. Here, the targets are monitored by activating a subset of sensors instead of all. Only one pan of a sensor can be activated at a time.

Besides coverage, fault tolerance is another important aspect which is overlooked by most of the research works. Practically, a sensor node in DSNs may be damaged due to the failure of any hardware components, incorrect execution of software, or malicious attacks etc.[5] [17] [18]. Because of the sensor nodes are very much prone to failure and replacement or recharged of those sensor nodes are not possible, therefore targets are given more than one coverage. This type of problem is known as the k-coverage [19] problem which provides more fault tolerance. Sakib et al. [20] extend the basic k-coverage work for achieving a balanced k-coverage. However, both of these works are for directional sensors.

In one version of the target coverage problem, sensors are to be divided into multiple sets where each set completely covers all the targets. Since if all the sensors are kept active at a time then sensors would lose their energy very quickly. Furthermore, an optimum schedule is to be determined by assigning a time slot to each set during which the set will be activated which extends the network lifetime and save the sensors energy. These sets are called cover sets (or covers merely), and this type of problems are known as coverage scheduling problem [21][22]. Initially, all sensors are in the energy saving sleep state. When a sensor belongs to a running activated set cover, its state changes to active state and starts monitoring the targets. Other sensors which are not the member of the present working set go to in energy saving sleep state.

1.2 Motivations

In previous literature, there are two threads of work that deal with the coverage scheduling in DSNs. In one thread, pairwise disjoint set covers are generated [7][9]. Here, each set completely covers all the targets, and no sensor participates more than one set cover. Later these disjoint set covers are roundly activated one after another. This scheme provides maximum fault tolerance; if a particular sensor dies out then only the containing set will be affected - other cover sets can be scheduled without any alteration. In another thread, sensors are allowed to participate in any number of cover sets. Here, the main goal is to maximize the lifetime of the network. In a compelling work [23], Cardei et al. show that allowing sensor overlap within the covers can prolong the network lifetime than the pairwise disjoint scheme. Rossi et al. [24] extend the prior work for directional sensors.

Consider the Figure 1.2 which shows the effect on network lifetime due to allowing sensors overlapping within set covers. Here, we assume that available energy holds a sensor operated for 100 time unit. From the Figure 1.2, it is evident that allowing the sensors to overlap

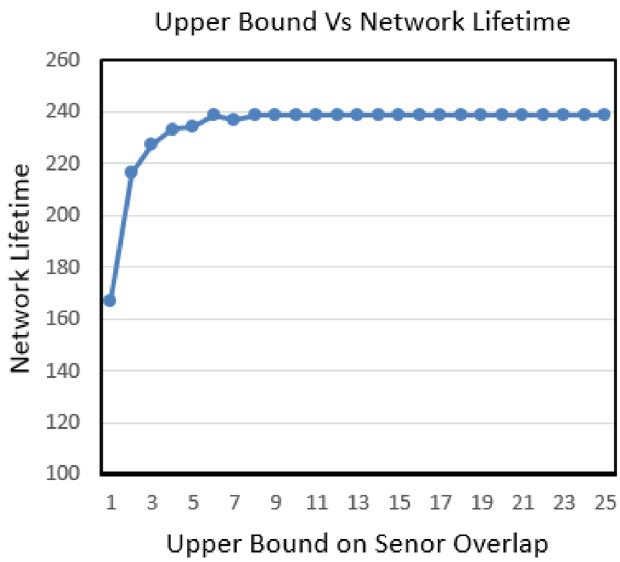


Figure 1.2: Measure the effect of sensor bounded overlapping over network lifetime

unboundedly will give us a maximum lifetime [23]. On the other hand, effect on network fault tolerance due to allowing sensors overlapping has been shown in Figure 1.3. Here, fault tolerance is calculated by the inverse of summing the membership count of the sensors within different set covers. It can be said from the Figure 1.3 that fault tolerance decreases as upper bound value is increased. That is, the network becomes more faulty as the bound value increases.

Novelty of this research:

The target coverage in directional sensor networks (DSNs) faces two major challenges:– prolonging network lifetime, and providing fault tolerance. Interestingly, many researchers have considered only lifetime maximization problem as the target coverage problem. On the other hand, there are few research works which ensure secure connectivity without interruption during coverage to make the system fault tolerant. But, none of these research works jointly consider the fault tolerance issue while extending the network lifetime. From the previous discussions, it can be concluded that disjoint sets provide maximum fault tolerance but under-perform in maximizing network lifetime. On the other hand, by allowing sensor overlap, we can attain a maximum lifetime by suppressing fault tolerance. However, our research work, in one sense, fit in the middle of these two antithetical approaches. The main objective of

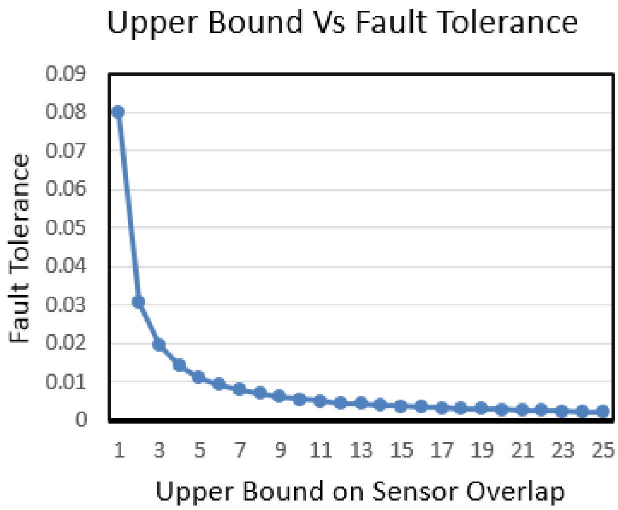


Figure 1.3: Measure the effect of sensor bounded overlapping over network fault tolerance

this research is to find out maximum possible number of set covers with a given upper bound on sensor overlap; no sensor can participate more than this given upper bound value. Here, sensors are directional. Therefore, this approach allows us to balance the network lifetime and fault tolerance.

In [25], Kamrul and Selim propose a similar problem, where, their goal is to maximize the number of set covers by minimizing the maximum membership count of the sensors. But they do not provide any mathematical formulation and sensors are omnidirectional whereas this research works deals with the directional sensors. To solve this problem, we develop a linear programming (LP) formulation followed by a two greedy approximation algorithms. LP formulation gives an optimal solution of the problem and greedy algorithms provide sub-optimal solutions. Additionally, we have introduced a performance metric called Goodness Index (GI) to determine the optimal bound value for which network performance is best by considering both network lifetime and fault tolerance.

1.3 Objectives of the Thesis

The main objectives of this research are as follows:

- Study the target coverage problem that generates maximum possible number of set covers by activating minimum number of sensors with given upper bound on sensor overlap.

- Construct an exact formulation of the problem as linear programming (LP).
- Design and implement practically implementable greedy heuristic for solving the problem approximately.
- Propose an efficient solution for balancing the lifetime and fault tolerance of a network.
- An empirical study of the performance of our proposed approaches used for target coverage.

1.4 Our Contributions

To achieve the objectives as mentioned earlier, we have developed LP formulation and two polynomial-time greedy heuristic for target coverage problem. The major contributions of this thesis are:

- We study the target coverage problem and find maximum possible number of the set covers using a subset of sensors within bounded overlapping such that each set completely covers all the targets.
- We develop an exact solution of the problem as linear programming (LP) to understand the optimal solution.
- As linear programming (LP) formulation is computationally expensive and not feasible for large problem instances, we propose a couple of polynomial time solvable greedy heuristics which provides maximum possible number of the set covers (approximately).
- We develop a directional sensor network consists of n number of sensors, m number of target points randomly deployed over the fixed deployment area of 100×100 units. We have implemented our algorithms using JDK 1.8, Eclipse IDE and LP was solved using the cplex optimizer.
- Finally, we measure the performance of the proposed heuristics. Also we perform sensitivity analysis concerning different performance criterion such as the number of generated set covers, network lifetime, and fault tolerance.

1.5 Organization of the Thesis

The rest of the thesis is organized in the following way:

In Chapter 2, we have discussed related works in the area of target coverage problem. To prolong the network lifetime, both disjoint and non-disjoint set covers approaches have been described here with their advantages and disadvantages. Finally concludes this chapter with a summary including some research directions.

Chapter 3 has started with some preliminaries and definitions used in our thesis. Linear programming (LP) formulation and proposed methodologies have been described in this chapter to solve the target coverage problem with i. presenting an exact formulation of the problem as LP and the pseudocode of the proposed heuristics for generating the maximum possible number of set covers within sensor' bounded overlapping ii. presenting a method to assign operational time to the set covers iii. introducing a mechanism for calculating the fault tolerance of the network.

Chapter 4 has presented the experimental results in a simulation environment. Here we compare the performance of our proposed heursitics with the optimal solution provided by linear programming (LP) formulation over different performance metrices through extensive simulations. Later we calculate an optimal bound value by considering both network lifetime and fault tolerance.

Finally, chapter 5 concludes this thesis and discuss some drawbacks and perspective works for further future research.

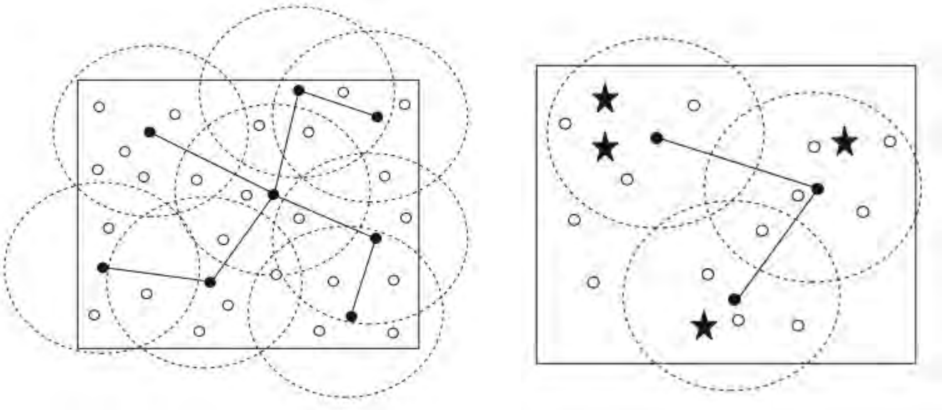
Chapter 2

Literature Review

Recently, coverage problem is an active research interest in wireless sensor netoworks (WSNs) since it reflects how well an area of interest (AOI) or a set interested targets can be monitored by randomly deployed sensors. The purpose of coverage problem is that each target point in the area of interest (AOI) must be inside in at least one sensors sensing range. Technique for solving the coverage problem depends upon the specification of the application. Generally, there are two types of coverage: area coverage and target coverage.

2.1 Area Coverage Problem

The main objective of area coverage in sensor network is to monitor a given area of interest (AOI) entirely by battery powered sensor nodes. In area coverage problem, the sensing field is divided into different subfields such that each subfield is monitored by a subset of sensors. Lifetime of such network can be increased by choosing an appropriate centralized or distributed protocol to select a subset of sensors [8]. However, the sensing activities are scheduled into different round such that before beginning of each round a sensor node decides whether it will be in active state or not. Figure 2.1 (a) illustrates an area coverage where given rectangular shaped area is covered by randomly deployed sensors. Circle represents sensors sensing range and black nodes indicate the active sensors.



(a) (b)

Figure 2.1: Types of coverage (a) area coverage (b) target coverage used from [26]

2.2 Target Coverage Problem

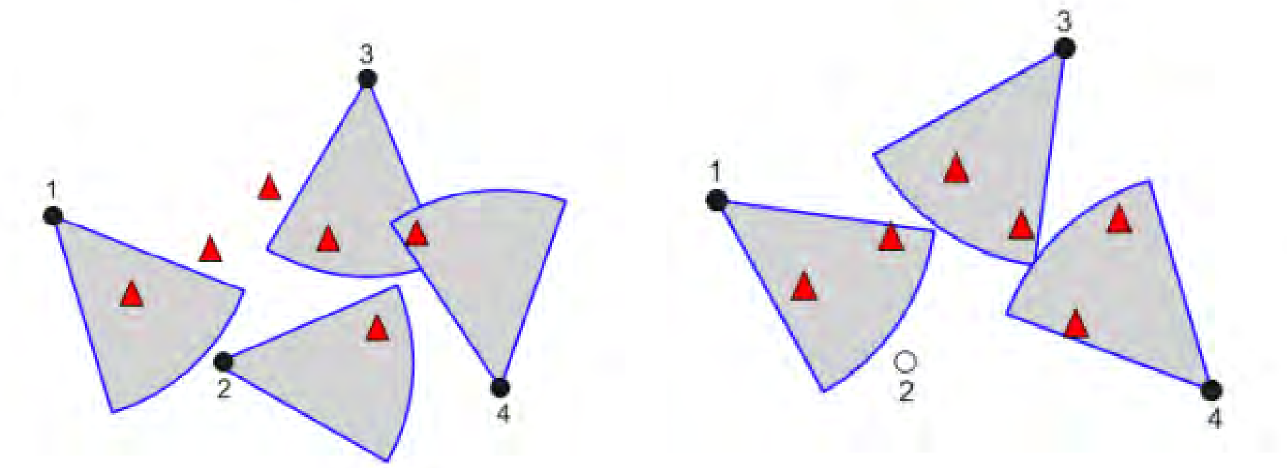
For target coverage problem, sensors monitor all the targets continuously until their battery gets depleted [27][28]. Since sensor nodes are equipped with limited battery power and cannot be easily replaced or recharged. Therefore, the primary goal of the target coverage problem is to prolong the network lifetime using the lowest resource consumption while providing the best coverage. Figure 2.1 (b) illustrates target coverage problem where given target points are covered by randomly deployed sensors. Generally, the target coverage problem can be classified as simple coverage, k-coverage, and Q-coverage. For simple coverage [13], each target should be covered by at least one sensor. On the other hand, when targets are given more than one coverage then it is called k-coverage [29] [30] [31]. In Q-coverage [32], targets have different coverage requirement. For example: given that a target vector $T = \{g_1, g_2, ..., g_m\}$ which is required to cover by Q = $\{k_1, k_2, ..., k_m\}$ number of sensors where k_i indicates minimum required number of sensors together monitor the target g_i. This type of problem can be defined as Q-coverage problem. The main difficulty for target coverage problem is how to efficiently monitor the finite set of targets while prolonging the network lifetime by conserve the sensors' energy usage. To overcome this difficulty, sensor activities are scheduled between active or sleep mode while satisfying coverage requirement in an energy efficient way. Only a subset of sensor are running to monitor the targets. All other sensors will be in energy saving sleep mode. However, in this research we focus only target coverage problem for directional sensors.

2.3 Related works on Target Coverage Problem

Over the last few years, lot of research studies have been done on several fundamental issues of wireless sensor networks(WSNs). Only some of them have been conducted with directional sensors. The main purposes of this researches in the field of WSN are the optimal placement of sensors and its orientation, energy efficient monitoring schedule for prolonging the network lifetime, and design a fault-tolerant network to assure secured connectivity of the network [9]. Among them, target coverage by activating minimum number sensors is the main attention. Compared to omnidirectional sensors, sensing region of a directional sensor is defined by both its position and orientation. So target coverage problem by directional sensors will be more challenging to manage rather than concentrating on omnidirectional sensors. Several comprehensive surveys on target coverage problem in WSNs can be found in [33][34][35][36]. Here, we have provided a brief description of some researches on target coverage.

2.3.1 Simple Coverage Problem

The first research work related on target coverage for directional sensor networks (DSNs) was introduced by Ai and Abuzeid in [13]. Here, authors proposed a Maximum Coverage with Minimum Sensors (MCMS) problem in which randomly deployed targets should be given single coverage by minimizing the number of sensors to be activated at any instant. To tackle this issue, they formulated the problem as an integer linear programming (ILP) which determines the optimal solution to this problem. They also developed practically-implementable Centralized Greedy Algorithm (CGA) and its relaxed version Distributed Greedy Algorithm (DGA) as an approximate solution of MCMS problem. Figure 2.2 shows two cases of covering six target nodes (red triangles) by using four sensors nodes (black circles). In Figure 2.2 (a), all the sensor nodes are active but covering only four target nodes. While rest of two target nodes are uncovered. But in Figure 2.2 (b), all the target nodes are covered by activating only three sensor nodes. Remaining sensor node is in energy saving sleep mode. Based on the formulation proposed by Ai and Abouzeid [13], Munishwar and Abu-Ghazaleh [37] proposed Centralized Forced Algorithm (CFA) and Distributed Forced Algorithm (DFA) ensuring the coverage maximization. Although, they did not concentrate on reducing the number of sensors usage.



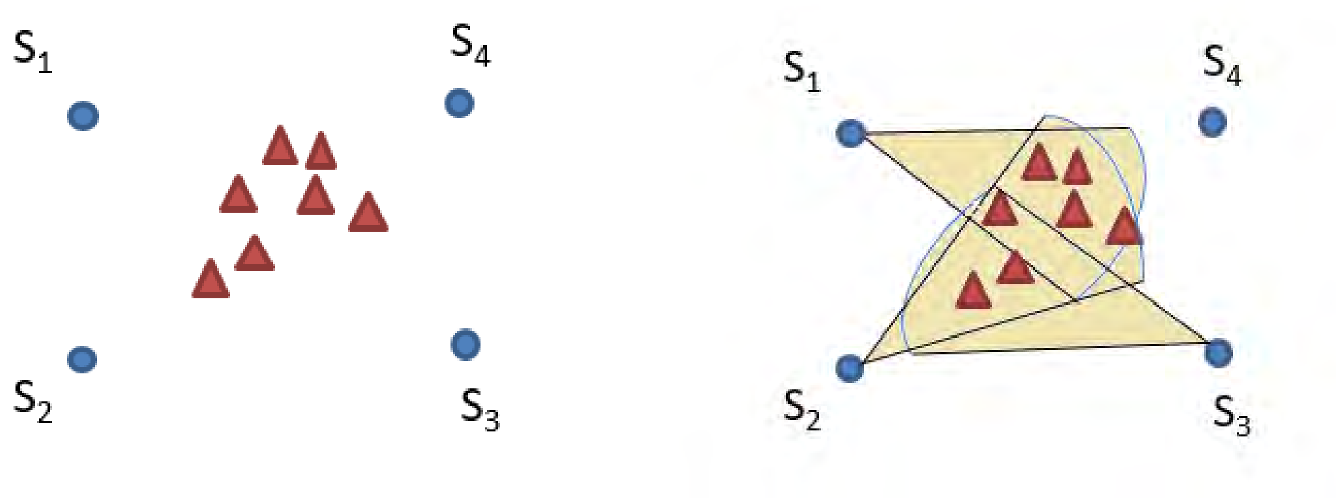
(a) (b)

Figure 2.2: A simple coverage scenario of six target nodes followed by four sensor nodes used from [13]

Due to limited angle of sensing direction of a directional sensor, it is likely that a directional sensor is unable to cover all the interesting targets even though targets are positioned in the sensing range of that sensor. Hsu et al. [14] proposed Maximum Coverage with Rotatable Sensors (MCRS) problem in which the coverage interms of the number of targets to be covered is maximized whereas the total rotated degrees are minimized. To solve this problem, they presented two centralized angle adjustment algorithms, namely the Maximal Rotatable Angles (MRA) scheme and the Maximum Coverage First (MCF) scheme.

2.3.2 k-Coverage Problem

Generally, sensor nodes in a wireless sensor network considerably depend on its limited battery life or energy. Therefore, the primary objective in designing a robust and secure network should be reduced power consumption for the system while maintaining its coverage reliability [38]. But the sensor nodes are very much prone to failure or may be disconnected due to the depletion of energy resources, devices failure, physical damage, transmission errors, incorrection execution of software, malicious attack, and so on [17] [39] [40]. From the real viewpoint, only single coverage of the given targets in a sensing region without redundancy would make such a network more vulnerable and detached at any times as sensors may be run out or die over time. Therefore besides coverage, fault tolerance mechanism is also needed when modeling the WSN. So that each target point of the monitoring area is covered by atleast k-number of sensors during coverage. This type of problem is known as k-coverage problem [19] [31] [30] [41] which improves fault tolerance capability of the network.

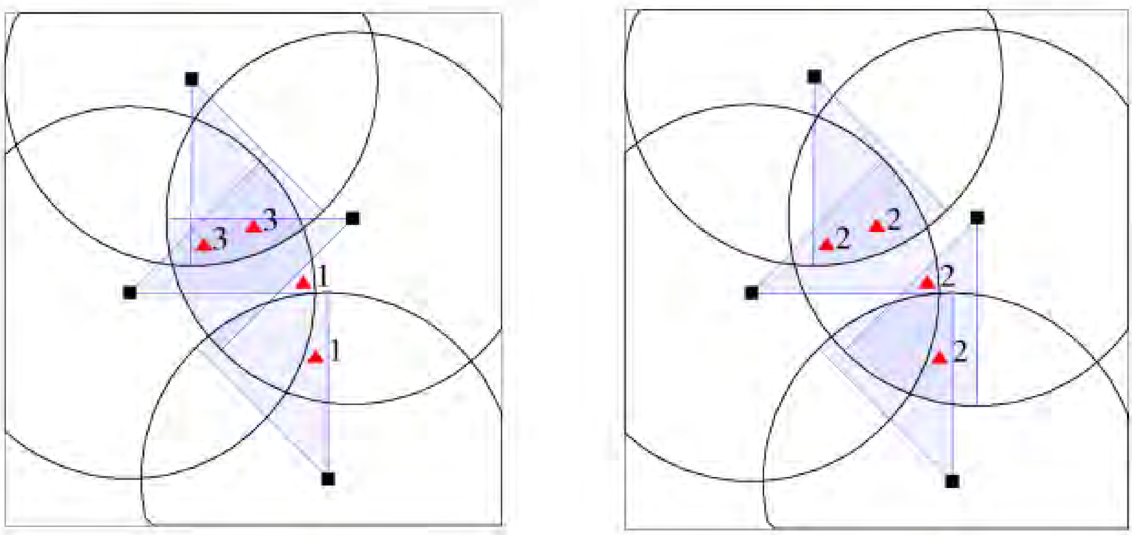


(a) (b)

Figure 2.3: A simple 2-coverage problem redrawn from [19]

Figure 2.3 illustrates a simple k-coverage problem (i.e. $k = 2$). In Figure 2.3(a), four sensor nodes are given to cover seven target nodes. Each sensor node has finite number of sensing angle. A probable solution for $k = 2$ is presented in Figure 2.3(b) where 3 sensors are enough to 2-cover all 7 targets used from [19]. To k-cover all the target nodes, Fusco and gupta [19] proposed selecting and orienting d-sensors for k coverage (SODkC) problem where minimum number of sensors have been selected and oriented them. Since k-coverage problem is NP-complete, authors designed greedy algorithm and its relaxed version distributed algorithm to k-cover at least half of the target nodes using at most M log(k$|C|$) sensors where $|C|$ is the highest number of target nodes covered by a sensor and M is the least number of sensor nodes needed to k-cover all the targets. However, they did not focus on balanced coverage of the targets. As a result during coverage in under provisioned network, some target are k-covered while other targets may be uncovered or single covered. Sakib et al. [20] addressed this problem and extend the basic k-coverage work for achieving a balanced k-coverage. To tackle this issue of coverage balancing, authors presented Integer Quadratic Programming (IQP) and Integer

Non Linear Programming (INLP) formulation as an exact solution of the problem. They also devised a faster Centralized Greedy k-Coverage Algorithm (CGkCA) as approximate solution of the problem. Both of these research works are for directional sensors. Figure 2.4 shows the coverage imbalance in under provisioned networks used from [20]. In Figure 2.4, there are four sensor nodes (black rectangles) and three target nodes (red triangles). However from Figure 2.4(a), two target nodes are 3-covered and others two are 1-covered. Clearly, the network is under provisioned for 3-coverage and therefore makes the coverage imbalanced. By changing the orientation of the sensors shown in Figure 2.4 (b), each of the target node is possible to cover by atleast two sensors which makes it more fault tolerant than the first approach shown in Figure 2.4(a).



(a) (b)

Figure 2.4: Imbalanced coverage in under-provisioned networks used from [20]

2.3.3 Q-Coverage Problem

The main challenging issue in wireless sensor network for target coverage problem is to obtain the best quality of coverage using minimum sensors. In some applications, coverage of different target nodes may need different sensing quality. For example: at the time of monitoring the hostile area, some critical targets like power plant may require more sensors support to provide secure coverage for getting the precise data. Targets with different quality of service (QoS) requirements can be defined as Q-coverage problem. It is the generalized version of k-Coverage and simple coverage problem. Let there are n number of sensors $S = \{s_1, s_2, ..., s_n\}$ to cover a set of m targets $G = \{g_1, g_2, ..., g_m\}$ randomly deployed in a sensing area. Given coverage vector $Q = (k_1, k_2, ..., k_m)$ indicates that each target g_i needs atleast k_i coverage. For Q-coverage, we need to find out a set of sensors that together monitor all the targets satisfying coverage vector requirements. When the value of all $k_i^{'}s$ are equal to an integer value k, then Q-coverage problem can be defined as k coverage problem. Moreover, if the requirements of k is 1 then the problem is modified to simple coverage problem. Gu et al. in [42] have formulated the optimization using linear programming techniques and proposed column generation based method to solve the Q-Coverage problem. In another research, Manju and Arun in [32] proposed a heuristic that satisfies Q-coverage requirements. Arivudainambi et al. in [43] proposed a energy efficient sensor scheduling mechanism for Q-coverage problem. Here, authors divide the sensors into multiple set covers such that each cover satisfies the coverage requirement which increases the network lifetime. Balaji et al. in [44] proposed a heuristic that identifies optimal sensors position for satisfying Q-Coverage constraints. However, all of their research works have been conducted with omnidirectional sensors.

2.3.4 Energy Efficient Target Coverage

Generally, a large number of sensor nodes are randomly scattered in a redundant way in the dense sensing field nearby to the targets and send the monitoring data to the base station (BS). The main requirement is that all the targets must be covered for as long as possible by at least one sensor considering that a sensor is capable to monitor all the target nodes in its sensing area. In a directional sensor network, it is expected that some sensors usually share common sensing fields. Thereby, activating all the sensors at the same time would misuse the sensors energy. An efficient technique for preserving the sensor's energy is to use the sensor nodes scheduling mechanism in which some sensors remain active to give sensing service, while the others will be in sleep state for saving their energy. Therefore, in one version of target coverage problem, sensors are to be partitioned into multiple cover sets where, each set completely covers all the targets. Later a scheduling mechanism is introduced where these set covers are activated one after another by assigning a time slot to each set cover. This types of problem are known as coverage scheduling or monitoring schedule problem [45] [46]. Since set covers can be either disjoint or non-disjoint. Slijepcevic et al.[47], Cardei et al. [48] and Diop et al. [11]

proposed centralized heuristics to solve the target coverage problem by arranging the sensors into disjoint subsets, where each set is capable to cover all the targets. Here, sensors are limited to engage in only disjoint subset, that is, a sensor can not become a member in more than one set cover. This type of problem is known as Maximum Disjoint Set Cover (MDSC) problem. It has been shown that MDSC was NP-complete thereby it was transformed to a maximum flow problem which was then formulated as mixed integer programming. But here authors addressed this problem only for the omnidirectional sensors. Ahn and Park [7] proposed Binary Integer programming formulation (BIP) and a new heuristic for maximum disjoint set cover (MDSC) problem similarly using for the omnidirectional sensors. In another research work, Munishwar and Abu-Ghazaleh [49] introduced target based heuristic for maximizing the coverage in which critical targets have been identified first and greedily select a sensor providing the maximum coverage including at least one critical target.

Shahla et al. proposed Maximum Disjoint Set Covers (MDSC) problem for directional sensors [9] in which maximum number of pairwise disjoint set cover are generated using minimal number of sensors. Each set of sensors cover all the targets. To solve the MDSC problem, at-first, authors formulated the problem as an Integer Linear Programming (ILP) problem which provides the optimal solution. As calculation time of ILP formulation is so high, so that they also proposed Iterative Centralized Greedy Algorithm (ICGA), Iterative Centralized Forced Algorithm (ICFA) and Iterative Target-Oriented Algorithm (ITOA) to solve the problem approximately. All of these heuristics are practically implementable for determining pairwise disjoint set covers that individually monitors all the targets. The basic objective of coverage scheduling is to enhance the network lifetime and provide secure connectivity by designing a fault tolerant network. Larger the number of disjoint set covers will enhance the network lifetime as well as the fault tolerance capability of the network.

However, MDSC problem for omnidirectional sensors was further extended in [23], where sensors are not limited to participate in only one set cover, that is, a sensor could become members in more than one set cover. Therefore, sensors are divided into multiple non-disjoint set covers. This type of problem is known as Maximum Set Cover (MSC) problem. In [23], two algorithms have been introduced for solving the MSC problem: one is LP (linear programming)-MSC and another is Greedy-MSC. Authors proved that the Greedy-MSC has better time complexity than the LP-MSC. Moreover, Greedy-MSC improves the network lifetime longer than LP-MSC. Kim et al. in [50] proposed a scheduling mechanism for the MSC problem based on the branch and bound approach for extending the network lifetime. Cardei et. al showed that in [23], allowing sensor overlap within the covers can increase the network lifetime than the pairwise disjoint scheme.

Consider the example scenario shown in Figure 2.5. In Figure 2.5 (a), there are four sensor nodes $\{s_1, s_2, s_3, s_4\}$ with each sensor node has eight mutually disjoint pans or orientations $\{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8\}$ for monitoring three target points $\{g_1, g_2, g_3\}$. Coverage relationship between sensor, pan pairs and targets can be described as follows: $(s_1, p_1) = \{(g_1, g_2)\}$; (s_2, p_3) $= \{(g_1, g_3)\}$; $(s_3, p_6) = \{(g_2)\}$; $(s_3, p_7) = \{(g_3)\}$; $(s_4, p_4) = \{(g_1, g_2, g_3)\}$. Assume that all the sensor nodes are homogeneous interms of same sensing range and energy. In order to calculate the network lifetime, we also assume that each sensor can be running for 1 unit time. For the first approach, it is possible to organize the sensors into only two disjoint set covers and they are $C_1 = \{(s_4, p_4)\}$, $C_2 = \{(s_1, p_1), (s_3, p_7)\}$. If all the sensors are activated simultaneously to monitor the targets then the lifetime of network will be only 1 unit of time. By dividing the sensors into disjoint sets in [9], will result in a network of lifetime 2 unit of time.

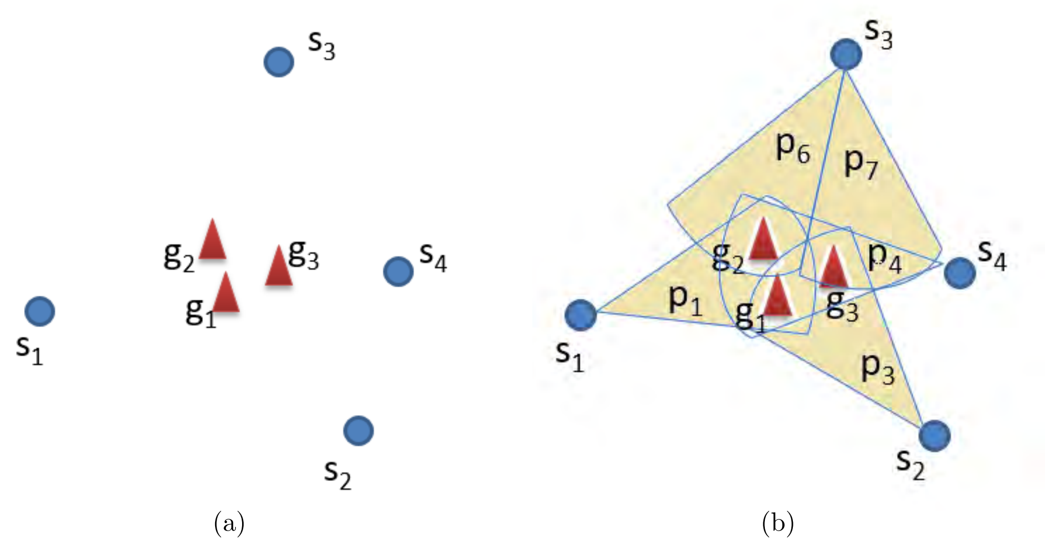


Figure 2.5: Example scenario of target coverage problem

On the other hand, if we allow sensor-overlap where a sensor can participate in atmost two set covers then it is possible to construct four set covers. These are $C_1 = \{(s_4, p_4)\}$; $C_2 = \{(s_1, p_1), (s_3, p_7)\}$; $C_3 = \{(s_1, p_1), (s_2, p_3)\}$, $C_4 = \{(s_2, p_3), (s_3, p_6)\}$. Assume that, the set covers are to be operated for different time intervals as follows: C_1 for 1 unit of time; C_2 for 0.5 unit of time; C_3 for 0.5 unit of time and C_4 for 0.5 unit of time. This will result in a network of lifetime 2.5 unit of time which is better than the disjoint set cover scheme.

The research works in [23][50] have been conducted only for omni-directional sensors. Han et al. [51] studied the target coverage problem for directional sensors and proposed Maximum Set covers for DSNs (MSCD) that finds the set covers covering all the targets and attains maximum network lifetime by allocating different activation time to each set cover. To solve this problem, authors proposed a greedy heuristic for finding a solution quicker than previous heuristics. However, none of these research works jointly consider the fault tolerance issue while extending the network lifetime.

2.4 Summary

In summary, there are two threads of work that deal with the coverage schedule. The disjoint set covers scheme in [9] gives us maximum fault tolerance. Since, if a sensor dies or runs out then only set cover containing this sensor will be affected. All other cover sets are yet fine for covering all target nodes. Moreover, k coverage of the target points can be achieved by activating k sets simultaneously at the same time. Therefore, fault tolerance of the network gets improved by generating more number of disjoint set covers. On the other hand, we can attain maximum lifetime by overlapping the sensors boundlessly within the set covers by suppressing the network fault tolerance. Since a sensor faultiness could affect all the set covers containing this sensor. In one sense, our research work will fit in the middle of these two antithetical approaches. We find out maximum possible number of set covers with a given upper bound on sensor overlap; no sensors are allowed to participate within set covers more than this given bound. By binding the sensors to overlap (i.e. allow sensor-overlap but not more than upper bound), this strategy improves the network lifetime than disjoint scheme as well as provides a fault tolerant network better than unbounded scheme. Kamrul and Selim in [25] proposed almost similar problem. Their goal is to maximize the number of set covers while minimizing the maximum membership bound of the sensors. But the problem is they did not concentrate on designing the mathematical formulation of the problem and their research work was conducted on omnidirectional sensors. Our work differs from [25] in the sense that the proposed research work deals with directional sensors. Besides, we also developed linear programming (LP) formulation of the problem followed by two greedy approximation algorithms.

Chapter 3

Proposed Solution Technique

In this chapter, we provide linear programming (LP) formulation of the problem that finds the maximum possible number of set covers within a given bound value on sensor overlap. In our experiment, this scheme is considered as an exact solution of the problem. As LP formulation is computationally expensive, so that we also proposed two heuristics which employ the greedy approach to form the set covers efficiently in polynomial time. Later, we present a procedure to calculate the operational time of the set covers and also describe a method which calculates network fault tolerance.

3.1 System Model and Definitions

In this section, we give the system overview for directional sensor and discuss TIS test. Also some definitions and assumptions used in our proposed heuristics have been defined here.

3.1.1 Directional Sensing Model

Directional sensing model controls the sensing coverage of a sensor. Based on the shape of the sensing region, the sensing model can be classified as omnidirectional sensing model and directional sensing model. Conventional sensor nodes are the omnidirectional sensor nodes which can sense 360 degrees at a time. On the other hand, directional sensors have a limited angle of sensing in a particular direction with fixed angle of view. Currently, almost all directional sensors are automatically operated Pan-Tilt-Zoom (PTZ) smart cameras. These

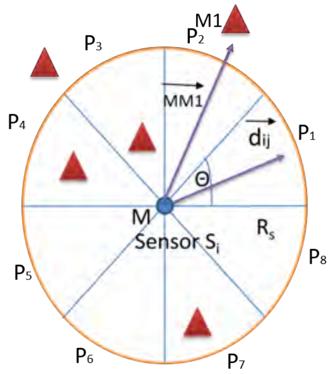


Figure 3.1: A directional sensor with eight discrete non-overlapping orientations

cameras are self-operative in three dimensions (i) Pan (vertical movement up and down), (ii) Tilt (Horizontal movement left and right), and (iii) Zoom. Normally, field of view (FOV) of these cameras can be fixed or self-adjustable. However, we have considered pan-only cameras in our experiment and ignore the effect of tilt and zoom.

Formally, in directional sensing model, a directional sensor can be represented by five parameters $(M, R_s, E, \overrightarrow{d_{ij}}, \theta)$ where M is the location of the sensor in two dimensional plane; R_s is the maximum sensing range beyond which a target can not be detected with reasonable accuracy; E is the energy or battery power of a sensor; $\overrightarrow{d_{ij}}$ is the unit vector which cuts the sensing sector into half representing the orientation of sensor s_i towards pan p_j; θ is the maximum sensing angle of the sensor called angle of view (AOV). Formally, θ and R_s defines the field of view (FOV) of a sensor. If field of view (FOV) of a sensor is defined as $\theta = \pi/4$ then a sensor has eight discrete non-overlapping pans. Figure 3.1 depicts all of these parameters. In our experiment, all the sensors are similar in terms of these parameters. We assume that every sensor is modeled as if it has finite number of non-overlapping pans and a sensor can choose only one pan at any instance. In Figure 3.1, $\overrightarrow{MM1}$ is the distance vector in the direction from sensor s_i located at M to a target g_t located at M1.

3.1.2 Target in Sector (TIS) Test

TIS test determines the relationship between a directional sensor with the targets. This test is performed to determine the set of targets covered by the sensor pan pairs within a particular sensing range [9][13][20]. Generally, a target is coverable by a sensor if it is located within a sensor's FOV and the distance between target and sensors is not greater than the sensor's maximum sensing range. In TIS test, following two conditions are tested:

i. $d(M, M1) \leq R_s$ where $d(M, M1)$ is the distance between location of sensor at M and target at M1; and R_s be the maximum sensing range of the sensor.

ii. $\delta_{it} \leq \frac{\theta}{2}$ where δ_{it} is the angle between a sensor's orientation $\overrightarrow{d_{ij}}$ of pan p_j and the distance vector $\overrightarrow{MM1}$. The angle δ_{it} is calculated by the equation as follows:

$$\delta_{it} = cos^{-1}\left(\frac{\overrightarrow{d_{ij}}.\overrightarrow{MM1}}{|\overrightarrow{d_{ij}}|.|\overrightarrow{MM1}|}\right)$$

Condition (i) indicates whether the target g_t is inside the sensor's maximum sensing range or not and condition (ii) indicates whether or not the target is located within the sensor's field of view. Generally, $\overrightarrow{d_{ij}}$ vector cuts the sensing sector into two portions. If the target is located in either of these two portions, then it must be coverable by the sensor s_i with pan p_j. That's why sensing angle δ_{it} is less than half of the sensor's field of view (FOV). If both conditions are met, then the result of TIS test is true; sensor s_i covers target g_t in pan p_j. Running this test over every pan p_j of sensor s_i and for all targets, we can easily construct a binary coverage matrix $A^m_{n\times q}$ of the network comprising of m targets and n sensors with q pans such that an element of this matrix can be computed as:

$$\alpha^t_{ij} = \begin{cases} 1, & \text{if sensor, pan pair}(s_i, p_j) \text{ covers target } g_t \\ 0, & \text{otherwise} \end{cases}$$

3.1.3 Related Definitions

Definition 1 : The set of targets covered by j^{th} orientation of i^{th} sensor is denoted by Φ_{ij}.

$$\Phi_{ij} = \{g_t \mid \alpha^t_{ij} = 1 \text{ i.e., } (s_i\ ,\ p_j) \text{ covers } g_t\}$$

Definition 2 : A set of sensor, pan pairs covering a target g_t is denoted by $\Phi^{-1}(g_t)$.

$$\Phi^{-1}(g_t) = \{(s_i, p_j)|\alpha^t_{ij} = 1\}$$

Definition 3 (Critical targets and sensors) : A target covered by minimum number of sensors is called critical targets and sensors covering the critical target are called critical sensors.

Definition 4 (Cardinality of target) : The number of sensor orientation pairs covering a target g_t is called cardinality of the target g_t and is denoted by D(g_t).

$$D(g_t) = \|\Phi^{-1}(g_t)\|$$

3.2 Problem Formulation

Given that a set of m targets, $G = \{g_1, g_2, g_3, ..., g_m\}$ and a set of n directional sensors, $S = \{s_1, s_2, s_3, ..., s_n\}$ each of which has q number of non-overlapping pans, $P = \{p_1, p_2, ..., p_q\}$ randomly deployed in a 2-D Cartesian coordinate system. Let, z be the given bound value on sensor overlap and sensor pan pair, (s_i, p_j) covers a subset of targets (i.e., $\Phi_{i,j} \subset G$). A set cover, C_k is a set of sensor, pan pairs such that sensor pan pairs together cover all the targets of G. Note that only one pan of a sensor can be activated at a time and there is atmost one pair for each sensor in a set cover. Now the problem is to find out the maximum possible number of set covers by activating a subset of sensors within the given bound value on sensor overlap such that sensor pan pairs in a set cover can be activated to cover all the targets. No sensor can become member of the set covers more than the given bound value z.

3.3 Proposed Solution

This section describes linear programming (LP) formulation of the problem with all necessary constraints. This formulation maximizes the number of set covers within sensors' bounded overlapping. As LP formulation is not feasible for large problem instances so that we also present two centralized greedy heuristics here that provide the solution approximately.

3.3.1 Optimum Solution of the Problem

A. LP Formulation

The set of parameters used in this formulation can be described as follows. n: the number of smart directional sensors; m: number of stationary targets; q: number of non overlapping pans available for each directional sensor; and z: upper bound on sensor overlap. Given that, a collection of all possible set covers C = $\{C_1, C_2, ..., C_r\}$ by n sensors each of which covers all the targets. x^l_{ij} is a selection variable that indicates whether sensor, pan pair $(s_i, p_j) \in C_l$ or not.

$$x^l_{ij} = \begin{cases} 1, & \text{if sensor, pan pair } (s_i, p_j) \in C_l \\ 0, & \text{otherwise} \end{cases}$$

Decision variables used in this formulation are also defined as follows.

$$y^l_t = \begin{cases} 1, & \text{if target } g_t \text{ is covered in } l^{th} \text{ set cover} \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_l = \begin{cases} 1, & \text{if } l^{th} \text{ set cover is selected} \\ 0, & \text{otherwise} \end{cases}$$

Through out this formulation, we use the following index variable:

- i ∈ {1, 2, . . ., n}
- j ∈ {1, 2, . . ., q}
- t ∈ {1, 2, . . ., m}
- l ∈ {1, 2, . . ., r}

Then goal of the LP formulation is to:

maximize:

$$\sum_{l=1}^{r} \gamma_l \tag{3.1}$$

subject to constraint:

$$\sum_{l=1}^{r} \sum_{j=1}^{q} x_{ij}^l \leq z; \quad \forall i \tag{3.2}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{q} [g_t \in \tau_{ij}] x_{ij}^l \geq 1; \quad \forall l, \forall t \tag{3.3}$$

$$\sum_{t=1}^{m} y_t^l \geq m; \quad \forall l \tag{3.4}$$

$$\sum_{j=1}^{q} x_{ij}^l \leq 1; \quad \forall l, \forall i \tag{3.5}$$

where:

$$x_{ij}^l = 0 \, or \, 1; \quad \forall l, \forall i, \forall j \tag{3.6}$$

$$y_t^l = 0 \, or \, 1; \quad \forall l, \forall t \tag{3.7}$$

The objective function of the LP formulation in (3.1) is to maximize the number of set covers. If $\gamma_l = 0$ then it means, the set cover, C_l is not selected. Constraint (3.2) assures that maximum membership count of a sensor cannot exceed the given upper bound on sensor overlap. Constraint (3.3) indicates that each target is covered by at least one direction of a sensor in a set cover. Constraint (3.4) provides the guarantee that each set completely covers all the targets. Constraint (3.5) imposes the condition that only one pan of a sensor can be activated at a time in a set cover. Constraint (3.6) and Constraint (3.7) state the binary integer requirements on the decision variables.

3.3.2 Centralized Greedy Heuristic

Although the LP formulation described in section 3.3.1 gives the optimal number of set covers, for large problem instances it does not scale well since the amount of time required for solving the mentioned problem of target coverage using LP formulation is computationally high. Therefore, we propose two greedy heuristics which are solvable in polynomial time that provide sub-optimal solutions.

Consider a network with a large number of directional sensors having a fixed number of pan randomly deployed in a sensing area adjacent to a set of targets. Targets are required to be continuously monitored by the sensors. The position of targets, sensors are known to the base station (BS). A sensor can choose only one pan or orientation at a time. Assume that, every sensor are equipped with GPS (Global positioning systems). Therefore, a sensor can calculate its coverage matrix and send it to the base station (BS). In over provisioned network, sensors are partitioned into different number of non-disjoint set covers while each set can monitor all the targets. The base station (BS) calculates a coverage scheduling where only one set cover can be active at a time. The schedule is sent to all the sensors. Based on the schedule, sensors have to determine when they need to be active. Only the sensors belong to a set cover are active at a time, and the rest of the sensors are in idle or sleep state. Normally, the base station executes all the algorithm and broadcasts the result to all sensor nodes. The set cover scheduling mechanism can be achieved as follows:

- Sensors send their location/position information to the Base Station (BS).
- For a given bound value on sensor overlap, BS executes algorithm to generate maximum number of set covers and send the information to all sensors. From the information, each sensor finds out the set covers where it belongs to.
- Every sensor makes plan itself for active/idle/sleep intervals.

To generate the set covers, base station (BS) executes one of the following two heuristics: (i) Sensor Oriented Greedy Heuristic (SOGH), and (ii) Target Oriented Greedy Heuristic (TOGH).

In both heuristics, set covers are formed individually by considering two issues. The first issue is to reduce the use of the same sensors in different set covers, and the next issue is to select those sensors first that maximize the coverage of uncovered targets.

A. Sensor Oriented Greedy Heuristic (SOGH)

The basic idea behind the sensor oriented greedy heuristic (SOGH) is to greedily choose a sensor pan pair which provides maximum coverage (of uncovered targets) among least used sensors in previously formed set covers. The pseudocode of this heuristic has been given in Algorithm 1. The idea of this algorithm can be summarized as follows. In this algorithm, S holds currently available sensors for forming a new set cover C_k and variable $count_i$ indicates participation number of sensor s_i in different set covers thus far. It also indicates the priority of an available sensor, i.e., a sensor with least count value indicates highest priority sensor. This algorithm iteratively constructs the set covers from line 4 $\sim$ 33. At the beginning of each iteration, all the targets and available sensors are assigned to TARGETS and SENSORS set respectively (line 7 $\sim$ 8). At each step, at first it selects the sensors which cover at least one uncovered target(s) and create a set $S^{'}$ (line 11 $\sim$ 18). Next, sensors having minimum count value are selected from $S^{'}$ and make another set $S^{''}$ (line 22 $\sim$ 28). After that, it selects the sensor pan pair from $S^{''}$ which maximizes coverage of the uncovered targets (line # 29). Ties are broken arbitrarily. The newly selected sensor pan pair is included in the current set cover C_k (line # 30). Next, all the covered targets by newly selected sensor pan pair are removed from TARGETS (line # 31) and the newly selected sensor has been removed from SENSORS (line # 32). Steps 9 $\sim$ 33 continue until TARGETS set is empty. When all the targets are covered, a new set cover C_k has been formed.

Once a set cover C_k has been obtained, the newly created set cover is optimized (line # 34) to remove any redundant sensors from the set cover which did not contribute during coverage. The algorithm used for optimization at this step is provided in Algorithm 2 in details. Next, priority of each sensor in optimized set cover has been updated (# 36). When priority of a sensor reaches to the given bound z, it has been removed from S and will not be available for further set cover formation (line 37 $\sim$ 39). This algorithm terminates when no sensors are available to cover at least one uncovered target in TARGETS. Finally, algorithm returns the collection of set covers $C = \{C_1, C_2, ..., C_k\}$.

Algorithm 1 Sensor Oriented Greedy Heuristic

Input: S = set of sensors; G = set of targets; P = set of discrete pans; z = upper bound value on sensor overlap

Output: Collection of set covers C = $\{C_1, C_2, ..., C_k\}$

```
function MaximizetheNumberOfSetCovers(S, G, z)
    set count_i ← 0; 1 ≤ i ≤ n
    k = 0
    while (S ≠ ϕ) do
        k = k + 1
        C_k = ϕ
        TARGETS = G
        SENSORS = S
        while (TARGETS ≠ ϕ) do
            S‘ = ϕ
            for each sensor s_i ∈ SENSORS do
                for pan p_j ∈ P do
                    if (| Φ_ij ∩ TARGETS | ≥ 1) then
                        S‘ = S‘ ∪ {s_i}
                        break
                    end if
                end for
            end for
            if (S‘ =ϕ) then
                goto marker
            end if
            mincount = min_{s_i∈S′,1≤i≤n} count_i
            S“ = ϕ
            for each s_j ∈ S‘ do
                if (count_j == mincount) then
                    S“ = S“ ∪ { s_i}
                end if
            end for
            (s_i, p_j) ← arg max_{s_i∈S“,1≤j≤q} Φ_ij∩ TARGETS
            C_k = C_k ∪ {(s_i, p_j)}
            TARGETS = TARGETS \ {Φ_ij}
            SENSORS = SENSORS \ {s_i}
        end while
        C_k = SetCoverOptimization (C_k, G)
        for each sensor s_i ∈ C_k do
            count_i=count_i + 1
            if (count_i == z) then
                S= S \ {s_i}
            end if
        end for
    end while
    marker: return the collection of set covers C = {C_1, C_2, ......, C_k}
```

Let us analyze the time complexity of SOGH. The algorithm has two while loops. For each iteration of the inner while loop (line # 9), algorithm needs to perform two major tasks. First task (line 11 $\sim$ 18) takes $(m + 1)nq$ operations. As we need to create a coverage matrix by performing TIS test which runs for n sensors in each q pan over m targets and then pick the sensors that cover at least one uncovered target. For the second task, (line # 29), again algorithm executes $(m+1)nq$ operations. Since, inner while loop runs over all the targets which requires at most m iterations. So the cost of the inner while loop is $(2(m+1)mnq)$. Besides, set cover optimization takes $(m+1)mnq$ operations in its calculation which is discussed later. Thus the total cost of the inner while loop becomes $(2(m + 1)mnq) + (m + 1)mnq = 3(m + 1)mnq$. The outer while loop (line # 4) runs over all sensors at most nz times. Therefore, the overall cost of this algorithm is $nz(3(m + 1)mnq)$. Thus the worst case time complexity of sensor oriented greedy heuristics (SOGH) is $O((m+1)mn^2qz)$ which eventually becomes $O(m^2n^2qz)$.

Set Cover Optimization:

Once the coverage requirement is fulfilled, set covers are needed to be optimized to eliminate the redundant sensors within set cover. A problem may arises during set cover formation in both algorithms 1 and 3 is that it does not require activating all the sensors for coverage.This is because set cover formation in both algorithms is a step by step procedure of adding sensors having least count value. Least usage sensor is prioritized first for selection. Therefore, a sensor may not contribute to the coverage after set cover formation (rest of the sensors may already cover all the targets) could be dropped.

Pseudocode for set cover optimization is shown in Algorithm 2. By optimizing the newly obtained set cover, the proposed algorithm minimizes the usages of sensors where possible. Initially, it creates an empty set cover C_k^* where all the sensors in C_k are assigned and the set cover C_k is made empty. For each iteration, it selects a sensor pan pair from C_k^* that covers maximum number of uncovered targets. The selected sensor pan pair is included in the set cover C_k, and targets covered by the newly selected sensor pan pair are removed from G. When G is empty (i.e., no more uncovered targets), the newly optimized set cover C_k has been formed and all the redundant sensors within set cover are eliminated. The run time of this algorithm is $O(m^2nq)$. Because the while loop (line # 3) runs for m iterations in worst case (i.e., one target covered per iteration). The major step in while loop is (line # 4) which requires TIS testing and needs mnq operations. Thus worst case time complexity becomes $O(m^2nq)$.

Algorithm 2 Set Cover Optimization

Input: C_k, a set cover consists of sensor pan pairs covering all the targets; G = set of targets; P = set of discrete pans

Output: optimized set covers, C_k

```
function SetCoverOptimization(C_k, G)
    C_k^* = C_k, C_k = φ
    while G ≠ φ do
        (s_i, p_j) ← arg max_{(s_i,p_j)∈C_k} Φ_ij ∩ G
        C_k = C_k ∪ (s_i, p_j), G = G − Φ_ij
        C_k^* = C_k^* \ {(s_i, p_j)}
    end while
    return C_k
```

B. Target Oriented Greedy Heuristic

Target Oriented Heuristic (TOGH) differs from SOGH in a sense that, it considers targets first while selecting a sensor. In other words, it determines critical target(s) first (i.e. the target(s) with least coverage) and selects sensor(s) that cover critical target(s). Pseudocode of this heuristic is shown in Algorithm 3. Working procedure of this scheme can be explained as follows. In Algorithm 3, variable $count_i$ indicates priority of sensor s_i. S is the set of currently available sensors. At each iteration, at first it finds the critical targets and makes a set $G^{'}$ (line # 15). Then it selects the sensor that covers at least one critical target and makes a set S (line 16 $\sim$ 20). Since more than one sensor may cover critical targets, so sensors having minimum count value have been selected from $S^{'}$ to create another set $S^{''}$ (line 23 $\sim$ 27). If two sensors have the same priority, then the sensor pan pair with highest contribution (i.e., covering maximum uncovered targets) from $S^{''}$ (line # 29) is selected (line # 28). Ties are broken by randomly choose one of them. Once the sensor pan pair is selected, it has been added to the current set cover C_k (line # 29). All the additional targets covered by the sensor pan pair are removed from the TARGETS (line # 30) and newly select sensor is excluded from SENSORS (line # 31). The above procedure (line 9 $\sim$ 33) is continued until all of the targets are covered. When a set cover is created, the newly created set cover is optimized using Algorithm 3 like SOGH. After that, the priority of each sensor in the optimized set cover are updated (line # 36). If count value of a sensor is equal to the bound value (z), it is removed from the set of available sensors S (line 37 $\sim$ 39).

Algorithm 3 Target Oriented Greedy Heuristic

Input: S = set of sensors; G = set of targets; P = set of discrete pans; z = upper bound value of sensor overlap

Output: Collection of set covers C= $\{C_1, C_2, ..., C_k\}$

```
function MaximizetheNumberOfSetCovers(S,T, z)
    set count_i ← 0; 1 ≤ i ≤ n
    k = 0
    while S ≠ φ do
        k = k + 1
        C_k = φ
        TARGETS = G
        SENSORS = S
        while (TARGETS ≠ φ) do
            find D_min = min D(g_t): g_t ∈ G
            if |D_min|=0 then
                goto marker
            else
                S‘ = φ
                find the targets having cardinality equals D_min and create set G‘
                for each g_t ∈ G‘ do
                    for each (s_i, p_j) ∈ Φ^{-1}(g_t) do
                        S‘ =S‘ ∪ {(s_i, p_j)}
                    end for
                end for
                mincount = min_{s_i ∈ S‘, 1≤i≤n} count_i
                S“ = φ
                for each s_j ∈ S‘ do
                    if (count_j == mincount) then
                        S“ = S“ ∪ { s_i}
                    end if
                end for
                (s_i, p_j) ← arg max_{s_i ∈ S“, 1≤j≤q} Φ_ij ∩ TARGETS
                C_k = C_k ∪ {(s_i, p_j)}
                SENSORS = SENSORS \ {s_i}
                TARGETS =TARGETS \ {Φ_ij}
            end if
        end while
        C_k = SetCoverOptimization (C_k, G)
        for each sensor s_i ∈ C_k do
            count_i = count_i + 1
            if (count_i == z)
                S = S \ { s_i }
            end if
        end for
    end while
    marker: return the collection of set covers C = {C_1, C_2, ......, C_k}
```

However, when no sensors are available to cover at least one uncovered target in TARGETS, algorithm terminates and return the collection of set covers $C = \{C_1, C_2, ..., C_k\}$.

Let's investigate the time complexity of TOGH. Like SOGH, TOGH also has two while loops. Inner while (line # 9) loop consists of three major tasks: firstly it takes mnq operations for finding the targets with minimum cardinality (line # 15) as it runs over all the targets (m) for each sensor pan pair (i.e., $n \times q$). The second task is to find the sensor pan pairs covering the critical targets which also takes mnq operations (line 16 $\sim$ 20). And for the third task, it takes $mnq + nq$ operations to find out a sensor pan pair which covers maximum uncovered targets (line # 28). Since inner while loop (line # 9) is bounded by the number of targets (m) therefore the cost of inner while loop is ($m(3m + 1)nq$) which is eventually O(m^2nq). Since set cover optimization requires O(m^2nq) operations (line # 34) and outer while loop (line # 4) is bounded by n number of sensors and runs at most nz times. Thus the overall time complexity of TOGH in the worst case is $O(m^2n^2qz)$.

3.4 Coverage Scheduling

Both of the heuristics SOGH and TOGH generate multiple set covers $C_1, C_2, \ldots, C_k$ each of which is capable of monitoring all the targets. Therefore, these set covers can be activated one at a time and only the sensors present in the activated set cover monitor the targets. As the sensor energy is consumed only when the sensor is turned on (we ignore sleep energy), a scheduling scheme can be applied where set covers are switched from one cover to another in a round-robin fashion such that only a subset of sensors consume energy at a time. Needless to say that this scheme extends the network lifetime. For example, let we have three set covers C_1, C_2 and C_3 and operational time (i.e., or activation time) of these three set covers are 1.00, 1.00 and 0.5 unit of time respectively. Two strategies could be used during coverage schedule. In the first strategy, each set covers is assigned a fixed time slot for execution in a cyclic way, e.g. if the time slot is 0.5 unit of time then C_1 runs for 0.5 unit of time, C_2 runs for 0.5 unit of time, C_3 runs for 0.5 unit of time, then again C_1 runs for 0.5 unit of time, and C_2 runs for 0.5 unit of time. In the second strategy, set covers are executed in a fixed number of rounds, e.g., if number of rounds is three (03), then at each round C_1 runs for 0.333 unit of time, C_2 runs for 0.333 unit of time, and C_3 runs for 0.166 unit of time.

Algorithm 4 Lifetime Calculation

Input: a set of k set covers, C= $C_1, C_2, ..., C_k$; z = upper bound value on sensor overlap; E = initial energy of a sensor

Output: activation time of k set covers t = { $t_1, t_2, ..., t_k$}

```
1:  function SetCoverActivationTime(C, z, E)
2:      set l_i ← E ; 1 ≤ i ≤ n
3:      set t_i ← 0 ;1 ≤ i ≤ k
4:      flag[i] = 0; 1 ≤ i ≤ k
5:      find a set cover C_i such that flag[i] = 0; 1≤i ≤k
6:      if no such set cover C_i is found then go to step 28
7:      minlife = E, duration = 0
8:      for each sensor s_j ∈ C_i do
9:          if (l_j < minlife) then
10:             minlife =l_j
11:         endif
12:     endfor
13:     if (minlife ≤0) then
14:         flag[i] = 1
15:         go to step 4
16:     else
17:         if (minlife ≤ (E/z)) then
18:             duration = minlife
19:         else
20:             duration = (E/z)
21:             t_i = t_i + duration
22:         end if
23:         for each sensor s_j ∈ C_i do
24:             l_j= l_j - duration
25:         end for
26:     end if
27:     go to step 5
28:     LT_z = Σ_{i=1}^{k} t_i
29:     return activation time of k set covers t = { t_1, t_2, ...., t_k}
```

Algorithm 4 is used to compute the activation time (i.e. operational time) of each set cover. The algorithm works as follows. Let us assume a homogeneous system where each sensor has same maximum usable energy that can run each sensor for a maximum of E time unit. Suppose, l_i indicates the remaining energy of sensor s_i and t_i is the activation time of a set cover C_i. $flag[i]$ is a Boolean variable indicating whether a set cover C_i can be considered for assigning further activation time or not. For example, if $flag[i] = 1$ then the set cover C_i will not be considered for assigning further activation time. At each iteration the algorithm finds a set cover C_i with $flag[i] = 0$ (line # 5). Next, minimum remaining lifetime of all the sensors in a set cover C_i is determined (line 8 $\sim$ 12). Let it be $minlife$. If $minlife$ is 0 then the set cover C_i can not be used further; consequently $flag[i]$ is set to 1 and the algorithm resumes from step 5 to check another set cover (line 13 $\sim$ 15). Otherwise, it calculates the minimum of $minlife$ and E/z (line 17 $\sim$ 21) and sets this minimum value to *duration*. Note that the maximum assigned time duration for a set cover in a single iteration is E/z. After that, activation time of the set cover C_i (line # 22) and the remaining energy (in time unit) of each sensor in the set cover C_i are updated (line 23 $\sim$ 25). Iteration stops when $flag[i]$ is 1 for all set covers in which case no more set covers could be chosen for activation. The total lifetime of the network is calculated by summing up the activation time of all set covers (line # 28).

3.5 Fault Tolerance Calculation

In non-disjoint set cover scheme, sensors are allowed to participate in more than one set covers. The more a sensor participate in set covers the network becomes more error-prone.

Algorithm 5 Calculation of Network Fault Tolerance

Input: a set of k set covers, $C = \{C_1, C_2, ..., C_k\}$; and S = set of sensors
Output: fault tolerance, F_z

function FaultToleranceCalculation(C, S)
    for each sensor $s_i \in$ S **do**
        $count_i = 0$
    end for
    for each setcover $C_i \in$ C **do**
        for each sensor $s_j \in C_i$ **do**
            $count_j = count_j + 1$
        end for
    end for
    $F_z = 1/(\sum_{i=1}^{n} count_i)$
    return F_z

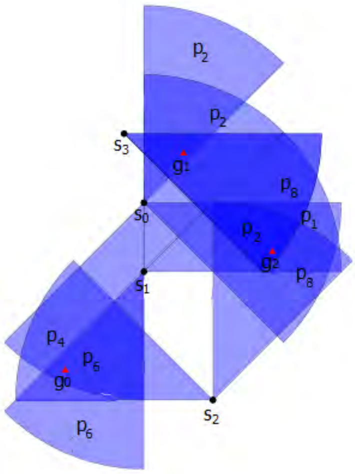


Figure 3.2: A simple scenario with four sensor nodes and three target nodes

Algorithm 5 shows how to calculate the network fault tolerance. The algorithm calculates the membership count of each sensors within set covers (line 5 $\sim$ 9). At the end, the network fault tolerance is calculated by summing the inverse of membership count of each sensors (line # 10).

3.6 Analysis of Proposed Scheme

Consider a sample scenario shown in Figure 3.2. There are four sensor nodes s_1, s_2, s_3, and s_4 which are randomly deployed in the network to cover three target nodes g_1, g_2, and g_3. Each sensor has eight non overlapping sensing pans $p_1, p_2, p_3, p_4, p_5, p_6, p_7$, and p_8. Maximum sensing range of a sensor is set to 20 unit. From Figure 3.2, coverage relationship between sensors and targets can be described as follows: $(s_0, p_6) = \{g_0\}$; $(s_0, p_2) = \{g_1\}$; $(s_0, p_8) = \{g_2\}$; $(s_1, p_6) = \{g_0\}$; $(s_1, p_2) = \{g_1\}$; $(s_1, p_1) = \{g_2\}$; $(s_2, p_4) = \{g_0\}$; $(s_2, p_2) = \{g_2\}$; $(s_3, p_8) = \{g_1, g_2\}$. Also sensor pan pairs cover a particular target can be represented as follows. $\Phi^{-1}(g_0) = \{(s_0, p_6), (s_1, p_6), (s_2, p_4)$; $\Phi^{-1}(g_1) = \{(s_0, p_2), (s_1, p_2), (s_3, p_8)\}$; and $\Phi^{-1}(g_2) = \{(s_0, p_1), (s_2, p_2), (s_3, p_8)\}$. Normally, two strategies are used to generate the set covers. In one strategy:– disjoint set covers (i.e., $z = 1$) are generated where sensors are not allowed to participate more than one set cover. In another strategy:– non-disjoint set cover (i.e., z>1) are generated where sensors can become member of more than one set cover. Now let us discuss the execution of LP, SOGH, and TOGH for the above scenario of Figure 3.2.

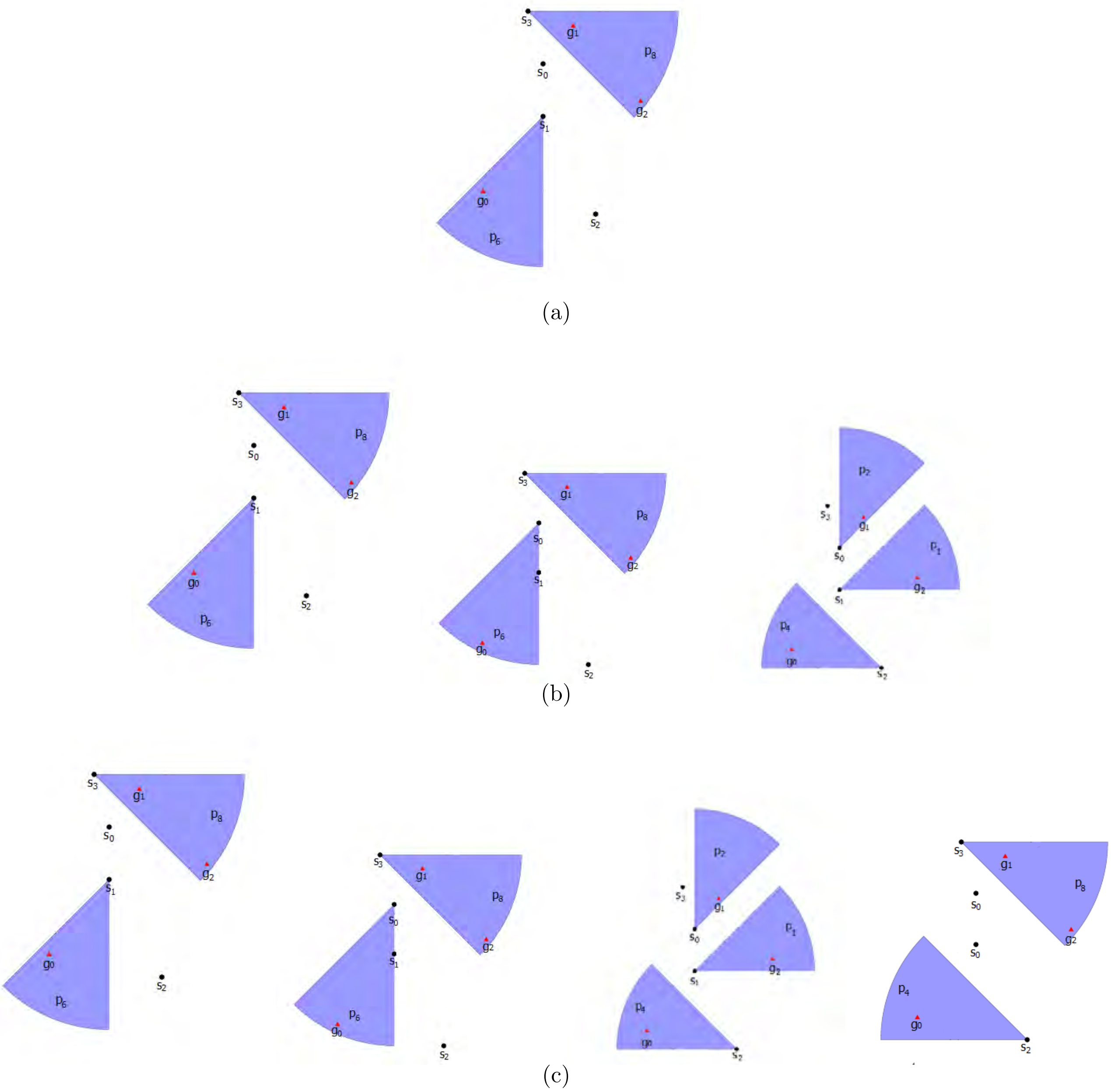


Figure 3.3: Maximum possible set covers generated by LP when upper bound value (a) $z = 1$ (b) $z = 2$ (c) $z = 3$

LP: As we know that, LP formulation provides maximum possible set covers within a given bound value (z). So output from LP is considered to be the optimum solution of the problem. However when we set $z = 1$ for the above scenario of Figure 3.2, only one disjoint set cover,

LP Formulations	
Upper Bound Value (z)	Possible Set Covers
$z = 1$	$C_1 = \{(s_1, p_6), (s_3, p_8)\}$
$z = 2$	$C_1 = \{(s_1, p_6), (s_3, p_8)\}$; $C_2 = \{(s_0, p_6), (s_3, p_8)\}$; $C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$
$z = 3$	$C_1 = \{(s_1, p_6), (s_3, p_8)\}$; $C_2 = \{(s_0, p_6), (s_3, p_8)\}$; $C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$; $C_4 = \{(s_2, p_4), (s_3, p_8)\}$

Table 3.1: Output of LP formulation for scenario of Figure 3.2

$C_1 = \{(s_1, p_6), (s_3, p_8)\}$ has been generated. Since due to sensors limited sensing capability, remaining sensors are not enough to cover all targets. Next for $z = 2$, three non-disjoint set covers are generated. They are:– $C_1 = \{(s_1, p_6), (s_3, p_8)\}$; $C_2 = \{(s_0, p_6), (s_3, p_8)\}$; and $C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$. As we increase the bound value from $z = 2$ to $z = 3$, then LP provides one more set cover. Therefore set covers are:– $C_1 = \{(s_1, p_6), (s_3, p_8)\}$; $C_2 = \{(s_0, p_6), (s_3, p_8)\}$; $C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$; and $C_4 = \{(s_2, p_4), (s_3, p_8)$. Output of LP formulation is summarized in Table 3.1. Also resultant set covers generated by LP for the above scenario of Figure 3.2 have been displayed in Figure 3.3.

Sensor Oriented Greedy Heuristic (SOGH): In SOGH, atfirst it finds out the least usage sensors within previously formed set covers and then selects a sensor pan pair which provides maximum coverage (of uncovered targets). If two sensor pan pairs have the same coverage, then randomly select one from them. Set cover formation using SOGH for the scenario shown in Figure 3.3 can be described as follows. Here, we describe the execution of SOGH for bound value upto $z = 2$.

A. Possible Set covers for $z = 1$:

i. Iteration 1 : $k = 1$ and S = $\{s_0, s_1, s_2, s_3\}$

- SENSORS = $\{s_0, s_1, s_2, s_3\}$ and TARGETS =$\{g_0, g_1, g_2\}$. All the sensors in SENSORS have equal priority. Among them, most contributing sensor pan pair is (s_3, p_8).

Therefore, it has been selected. Targets g_1 and g_2 have been removed from TARGETS and sensor s_3 has been removed from SENSORS respectively.

- SENSORS $=\{s_0,\ s_1,\ s_2\}$ and TARGETS $=\{g_0\}$. Sensors in SENSORS have equal priority and sensor pan pairs covered targets g_0 are:– $(s_0, p_6), (s_1, p_6)$, and (s_2, p_4). Among them, (s_2, p_4) has been selected by heuristic. Target g_0 has been removed from TARGETS and sensor s_2 has been removed from the SENSORS.
- SENSORS $=\{s_0,\ s_1\}$ and TARGETS $=\{\ \}$. Since TARGETS set is empty, so a new set cover, $C_1 = \{(s_2, p_4), (s_3, p_8)\}$ has been formed. After executing of SetCoverOptimization algorithm on set cover C_1, C_1 keeps remain same since there is no redundant sensor. Priority of each sensor in C_1 has been updated. Sensors s_3 and s_2 have been excluded from S and starts a new iteration.

ii. Iteration 2 : $k = 2$ and S $= \{s_0,\ s_1\}$

- SENSORS $=\{s_0,\ s_1\}$ and TARGETS $=\{g_0,\ g_1,\ g_2\}$. All the sensors have same priority and $(s_1, p_1), (s_1, p_2), (s_1, p_6), (s_0, p_2), (s_0, p_6), (s_0, p_8)$ sensor pan pairs provide same coverage. Among them, (s_0, p_8) has been selected. Target g_2 and sensor s_0 have been removed from TARGETS and SENSORS respectively.
- SENSORS $=\{\ s_1\}$ and TARGETS $=\{\ g_0,\ g_1\}$. There is only one sensor and sensor pan pair (s_1, p_6) has been selected to cover target g_0. Therefore, target g_0 and sensor s_1 have been removed from TARGETS and SENSORS respectively.
- SENSORS $=\{\}$ and TARGETS $=\{g_1\}$. Since the SENSORS set is empty but yet some targets in TARGETS. Therefore formation of new set cover, C_2 has been failed.

B. Possible Set covers for $z = 2$:

i. Iteration 1 : $k = 1$ and S $= \{s_0, s_1,\ s_2,\ s_3\}$

- SENSORS $=\{s_0, s_1,\ s_2,\ s_3\}$ and TARGETS $=\{g_0,\ g_1,\ g_2\}$. All the sensors in SENSORS have equal priority. Most contributing sensor pan pair is $(s_3,\ p_8)$. So, (s_3, p_8) has been selected. Targets g_1 and g_2 have been removed from TARGETS and sensor s_3 has been removed from SENSORS respectively.

- SENSORS = $\{s_0,\ s_1,\ s_2\}$ and TARGETS = $\{g_0\}$. Sensors have equal priority and sensor pan pairs covered targets g_0 are:– $(s_0, p_6), (s_1, p_6), and (s_2, p_4)$. Among them, (s_1, p_6) has been selected by the heuristic. Target g_0 has been removed from TARGETS and s_1 has been removed from the SENSORS respectively.
- SENSORS = $\{s_0,\ s_2\}$ and TARGETS = $\{\ \}$. Since TARGETS set is empty, so a new set cover $C_1 = \{(s_1, p_6), (s_3, p_8)\}$ has formed. After executing of SetCoverOptimization algorithm on set cover C_1, C_1 keeps remain same since there are no redundant sensors. Priority of each sensor in C_1 has been updated.

ii. Iteration 2 : $k = 2$ and S = $\{s_0, s_1,\ s_2, s_3\}$

- SENSORS = $\{s_0, s_1,\ s_2,\ s_3\}$ and TARGETS = $\{g_0,\ g_1,\ g_2\ \}$. Highest priority sensors are:– s_0 and s_2. Among these sensors, sensor pan pairs provide maximum coverage are:– $(s_2, p_2), (s_2, p_4), (s_0, p_2), (s_0, p_6), and (s_0, p_8)$. Since, (s_0, p_8) has been selected by heuristic. Target g_2 and sensor s_0 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_1,\ s_2,\ s_3\}$ and TARGETS = $\{g_0,\ g_1\}$. Sensor s_2 has higher priority than other sensors in SENSORS. So heuristic selects sensor s_2 with pan p_4 to cover targets g_0. Therefore, target g_0 and sensor s_2 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_1,\ s_3\}$ and TARGETS = $\{g_1\}$. Both sensors have equal priority and heuristic selects (s_3, p_8) to cover target g_1. Next, target g_1 has been removed from TARGETS and sensor s_3 has been removed from SENSORS.
- SENSORS =$\{s_1\}$ and TARGETS = $\{\ \}$. Since TARGETS set is empty, so a new set cover $C_2 = \{(s_0, p_8), (s_2, p_4), (s_3, p_8)\}$ has been formed. Here, sensor pan pair (s_0, p_8) is redundant in present of (s_3, p_8) which has been identified by the execution of SetCoverOptimization algorithm. So after optimization, $C_2 = \{(s_2, p_4), (s_3, p_8)\}$. Next, priority of each sensor in C_2 has been updated. As count value of sensor s_3 equal to bound value z, therefore s_3 has been removed from S.

iii. Iteration 3 : $k = 3$ and S = $\{s_0, s_1,\ s_2\}$

- SENSORS = $\{s_0, s_1,\ s_2\}$ and TARGETS = $\{\ g_0,\ g_1,\ g_2\ \}$. Sensor s_0 has higher priority than the others. So heuristic selects s_0 sensor with its pan p_8 to cover target g_2 . Target g_2 and sensor s_0 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_1,\ s_2\}$ and TARGETS = $\{g_0,\ g_1\}$. Both sensors have equal priority. Within them, $(s_1, p_2), (s_1, p_6), (s_2, p_4)$ sensor pan pairs provide equal coverage. However, heuristic selects sensor pan pair (s_2, p_4). Target g_0 and sensor s_2 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_1\}$ and TARGETS = $\{g_1\}$. Only one sensor in SENSORS. So heuristic selects s_1 and active its pan p_2 to cover target g_1. Next, target g_1 has been removed from TARGETS and sensor s_1 has been removed from SENSORS.
- SENSORS =$\{\}$ and TARGETS = $\{\ \}$. Since TARGETS set is empty, so new set cover $C_3 = \{(s_0, p_8), (s_2, p_4), (s_1, p_2)\}$ has been formed. Since, there is no redundant sensor in C_3, so new set cover C_3 remains same after optimization. Next, priority of each sensor in C_3 has been updated. Since s_1 and s_2 have been removed from S, because count value of both sensors reach to upper bound value z.

iv. Iteration 4 : $k = 4$ and S = $\{s_0\}$

- SENSORS = $\{s_0\}$ and TARGETS = $\{g_0,\ g_1,\ g_2\ \}$. There is only one sensor s_0 and it has been selected by heuristic with its pan p_8 and covers target g_2. Therefore, target g_2 and sensor s_0 have been removed from TARGETS and SENSORS respectively.
- SENSORS =$\{\}$ and TARGETS = $\{\ g_0,\ g_1\}$. Since SENSORS set is empty but still there are some uncovered targets, so formation of new set cover, C_4 has been failed.

Therefore, SOGH is able to generate one disjoint set cover, $C = \{(s_2, p_4), (s_3, p_8)\}$ for bound value $z = 1$ and three non-disjoint set covers,$C = \{\{(s_1, p_6), (s_3, p_8)\}, \{(s_2, p_4), (s_3, p_8)\}, \{(s_0, p_8), (s_2, p_4), (s_1, p_2)\}\}$ for bound value $z = 2$.

Target Oriented Greedy Heuristic (TOGH): In TOGH, atfirst it finds out the critical targets and then selects sensors that cover the critical targets. These sensors are called critical sensors. If more one critical sensor are found, then TOGH selects sensor(s) having highest priority. If two sensors have same priority, then it chooses a sensor, pan pair which provides maximum coverage (of uncovered targets). Execution of TOGH for the scenario of Figure 3.2 can be described as follows. Like SOGH, here we have explained the working procedure of TOGH for bound value up to $z = 2$.

A. Possible Set covers for $z = 1$:

i. Iteration 1 : $k = 1$ and S = $\{s_0, s_1, s_2, s_3\}$

- SENSORS = $\{s_0, s_1, s_2, s_3\}$ and TARGETS = $\{g_0, g_1, g_2\}$. There are two critical targets are:– g_0, and g_1. Sensors cover the critical targets are:– s_0, s_1, s_2, and s_3. Since all the sensors have equal priority. Among them, (s_3, p_8) provides maximum coverage. Therefore, it has been selected. After that, targets g_0 and g_2 have been removed from TARGETS and sensor s_3 has been removed from SENSORS respectively.
- SENSORS = $\{s_0, s_1, s_2\}$ and TARGETS = $\{g_0\}$. Critical target is g_0 and sensors cover the critical target are:– s_0, and s_1. All the sensors have same priority. Among them, heuristic selects (s_2, p_4). Therefore, target g_0 and sensor s_2 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_0, s_1\}$ and TARGETS =$\{ \}$. Since TARGETS set is empty, so a new set cover $C_1 = \{(s_2, p_4), (s_3, p_8)\}$ has formed. After executing of SetCoverOptimization algorithm on set cover C_1, C_1 keeps remain same as there are no redundant sensors. Priority of each sensor in C_1 has been updated. Sensors s_3 and s_2 have been excluded from S and starts a new iteration.

ii. Iteration 2 : $k = 2$ and S = $\{s_0, s_1\}$

- SENSORS = $\{s_0, s_1\}$ and TARGETS = $\{g_0, g_1, g_2\}$. Critical targets are:– g_0, g_1, and g_2. Sensors cover the critical targets are:– s_1, and s_0. Since both sensors have same priority and same contribution, therefore heuristic selects (s_0, p_8). Next, target g_2 and sensor s_0 have been removed from TARGETS and SENSORS respectively.

- SENSORS = $\{s_1\}$ and TARGETS = $\{ g_0, g_1\}$. Critical targets are:– g_0 and g_1. Sensor covers the critical targets is s_1. There is only one sensor and heuristic selects (s_1, p_2) to cover target g_1. Target g_1 and sensor s_1 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{\}$ and TARGETS = $\{g_1\}$. Since the SENSORS set is empty but yet a target is uncovered in TARGETS. Therefore, formation of new set cover, C_2 has been failed.

B. Possible Set covers for $z = 2$:

i. Iteration 1 : $k = 1$ and S = $\{s_0, s_1, s_2, s_3\}$

- SENSORS = $\{s_0, s_1, s_2, s_3\}$ and TARGETS = $\{g_0, g_1, g_2\}$. There are two critical targets:– g_0, and g_1. Sensors cover critical targets are:– s_1, s_0, s_2, and s_3. All the sensors have equal priority. Among them, (s_3, p_8) provides maximum coverage. Therefore, it has been selected. After that, targets g_0 and g_2 have been removed from TARGETS and sensor s_3 has been removed from SENSORS respectively.
- SENSORS = $\{s_0, s_1, s_2\}$ and TARGETS =$\{g_0\}$. Critical target is g_0 and sensors cover the critical target are:– s_1, and s_0. All sensors have same equal priority. Among them, heuristic selects (s_0, p_6). Therefore, target g_0 and sensor s_0 have been removed from TARGETS and SENSORS respectively.
- SENSORS =$\{s_1, s_2\}$ and TARGETS =$\{ \}$. Since TARGETS set is empty, so a new set cover $C_1 = \{(s_0, p_6), (s_3, p_8)\}$ has formed. After executing of SetCoverOptimization algorithm on set cover C_1, C_1 keeps remains same as there are no redundant sensors. Priority of each sensor in C_1 has been updated.

ii. Iteration 2 : $k = 2$ and S = $\{s_0, s_1, s_2, s_3\}$

- SENSORS = $\{s_0, s_1, s_2, s_3\}$ and TARGETS = $\{ g_0, g_1, g_2 \}$. Critical targets are:– g_0, and g_1. Sensors cover the critical targets are:– s_0, s_2, s_1, and s_3. Sensors s_1 has higher priority than others. However, (s_1, p_2) sensor pan pair has been selected by heuristic. Therefore, target g_1 and sensor s_1 have been removed from TARGETS and SENSORS respectively.

- SENSORS =$\{s_0, s_2, s_3\}$ and TARGETS = $\{g_0, g_2\}$. Critical targets is g_0 and sensors cover the critical targets are:– s_0, and s_2. Sensor s_2 has higher priority than s_0. Therefore, heuristic selects (s_2, p_4). Target g_0 and sensor s_2 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_0, s_3\}$ and TARGETS = $\{g_2\}$. Critical target is g_2 and sensors cover critical target are:– s_0, and s_3. Both sensors have same priority. Among them, heuristic selects (s_3, p_8). Therefore, targets g_2 and sensor s_3 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_0\}$ and TARGETS = $\{\}$. Since TARGETS set is empty, so new set cover $C_2 = \{(s_1, p_2), (s_2, p_4), (s_3, p_8)\}$ has been formed. Here, sensor pan pair (s_1, p_2) is redundant in present of (s_3, p_8) which identified by the execution of Set-CoverOptimization algorithm. Therefore, (s_1, p_2) has been excluded from C_2. So after optimization: C_2= $\{(s_2, p_4), (s_3, p_8)\}$. Next, priority of each sensor in C_2 has been updated. Since count value of sensor s_3 is equal to bound value z, therefore s_3 has been removed from S.

iii. Iteration 3 : $k = 3$ and S = $\{s_0, s_1, s_2\}$

- SENSORS = $\{s_0, s_1, s_2\}$ and TARGETS = $\{g_0, g_1, g_2\}$. Critical targets is g_1. Sensors cover the critical target are:– s_0, and s_1. Sensor s_1 has higher priority than s_0. Therefore ,(s_1, p_2) sensor pan pair has been selected by heuristic. Target g_1 and sensor s_1 have been removed from TARGETS and SENSORS respectively.
- SENSORS = $\{s_0, s_2\}$ and TARGETS = $\{g_0, g_2\}$. Critical targets are:– g_0 and g_2. Sensors cover the critical targets are:– s_0, and s_2. Both sensor have same priority. Therefore, heuristic selects (s_2, p_2). Target g_2 and sensor s_2 have been remove from TARGETS and SENSORS.
- SENSORS = $\{s_0\}$ and TARGETS = $\{g_0\}$. Critical target is g_0 and sensor covers critical target is s_0. Therefore, it has been selected and active its pan p_6 to cover

target g_0. Targets g_0 and sensor s_0 have been removed from TARGETS and SENSORS respectively.

- SENSORS = { } and TARGETS = {}. Since TARGETS set is empty, so new set cover $C_3 = \{(s_1, p_2), (s_2, p_2), (s_0, p_6)\}$ has been formed. Since, there is no redundent sensors so C_3 remains same. Next, priority of each sensor in C_3 has been updated. As count value of both sensor both s_0 and s_2 are equal to bound value z, therefore they have been removed from S.

iv. Iteration 4 : $k = 4$ and S = $\{s_1\}$

- SENSORS = $\{s_1\}$ and TARGETS = $\{g_0,\ g_1,\ g_2\}$. Here, there critical targets are:– g_0, g_1, and g_2. Sensor covers the critical targets is s_1. Therefore, heuristic selects (s_1, p_1). Target g_2 and sensor s_1 have been removed from TARGETS and SENSORS respectively.
- SENSORS ={} and TARGETS =$\{g_0,\ g_1\}$. SENSORS list is empty so formation of new set cover, C_4 has been failed.

Therefore, TOGH heuristic is also capable to generate one disjoint set cover, $C = \{(s_2, p_4), (s_3, p_8)\}$ for bound value $z = 1$ and three non-disjoint set covers, $C = \{\{(s_0, p_6), (s_3, p_8)\}, \{(s_2, p_4), (s_3, p_8)\}, \{(s_1, p_2), (s_2, p_2), (s_0, p_6)\}\}$ for bound value $z = 2$.

Cover Activation Time and Network Fault Tolerance: Given that, a collection of set covers, C = $\{(s_3, p_8), (s_2, p_4)\}$ for bound value $z = 1$ and a collection of set covers, C = $\{\{(s_3, p_8), (s_2, p_4)\}, \{(s_3, p_8), (s_0, p_6)\}, \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}\}$ for bound value $z = 2$. Assume that, the set covers have been generated either by SOGH or TOGH. We also assume that initial energy of a sensor is set to $E = 100$ unit and maximum assigned activation time to a set cover at a single iteration is (E/z). For a particular bound value z, execution of SetCover-ActivationTime algorithm (i.e.,Algorithm #4) over the given set covers has been described in Table 3.2. Also, calculation of network fault tolerance using algorithm # 5 has been described in Table 3.3.

z	i	C_i	minlife	Flag[i]	(E/z)	duration	Activation time, $t_i = t_i$ + duration	Remaining sensors energy
1	1	$C_1 = \{(s_3, p_8), (s_2, p_4)\}$	100	0	100	100	$t_1 = 100$	$l_2 = 0$, $l_3 = 0$
1	1	$C_1 = \{(s_3, p_8), (s_2, p_4)\}$	0	1	-	-	-	-
2	1	$C_1 = \{(s_3, p_8), (s_2, p_4)\}$	100	0	50	50	$t_1 = 50$	$l_2 = 50$, $l_3 = 50$
	2	$C_2 = \{(s_3, p_8), (s_0, p_6)\}$	50	0	50	50	$t_2 = 50$	$l_0 = 50$, $l_3 = 0$
	3	$C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$	50	0	50	50	$t_3 = 50$	$l_0 = 0$, $l_1 = 50$, $l_2 = 0$
	1	$C_1 = \{(s_3, p_8), (s_2, p_4)\}$	0	1	-	-	-	-
	2	$C_2 = \{(s_3, p_8), (s_0, p_6)\}$	0	1	-	-	-	-
	3	$C_3 = \{(s_0, p_2), (s_1, p_1), (s_2, p_4)\}$	0	1	-	-	-	-

Table 3.2: Calculation of set covers' activation time

3.7 Summary

In this chapter, target coverage algorithms considering both network lifetime and fault tolerance have been presented and discussed in details. LP formulation gives the optimal solution whereas greedy heuristics provide the approximate solution. In our algorithms, z expresses the upper

Bound Value (z)	Sensor (s_i)	membership count of s_i ($count_i$)	total membership count ($\sum_{i=1}^{n} count_i$)	fault tolerance = $1/(\sum_{i=1}^{n} count_i)$
$z = 1$	s_0 s_1 s_2 s_3	0 0 1 1	2	0.5
$z = 2$	s_0 s_1 s_2 s_3	2 1 2 2	7	0.1428

Table 3.3: Calculation of network fault tolerance

bound value on sensor overlap. Its value could be gradually increasing. Based on this value (z), set covers are generated either by SOGH or TOGH. Besides, Algorithm # 4 has been used to assign the activation time (i.e., operational time) to the set covers and Algorithm # 5 describes how to calculate the network fault tolerance. However, z is a tunable parameter and it is determined by network analyst. When $z = 1$, only disjoint set covers are considered. As the value of z is increased, possibility of generating set covers also increases.

Chapter 4

Experimental Results

This chapter shows the experimental results in a simulation environment. At first, we discuss the experimental setup and then compare the performance of the solutions provided by LP and other proposed schemes through extensive simulations. The number of set covers, network lifetime, and fault tolerance defined in section 4.2 have been used as the performance metrics of evaluation. Hence, at first we will show results of our proposed heuristics. Then we will analyze our experimental results and determine an optimal bound value on sensor overlap considering both lifetime and fault tolerance of a network.

4.1 Simulation Environment

Consider a stationary network in a 2-D plane. Assume that there are m number of targets randomly deployed in the simulation network followed by uniformly distributed n number of sensors to cover the targets. The position of sensors and targets are generated by uniform random distribution as a point (x, y) in a 2-D grid. Each sensor is modeled as directional sensor with FOV, $\theta = \pi/4$, therefore, a sensor can sense in one of eight non-overlapping pans. Initially, battery power and maximum sensing range of each sensor is assumed to be E time unit and R_s meter respectively in the network. In our simulation, we tune the following parameters.

- n: number of sensors which is varied from 10 to 80 to understand the impact of sensors' density over network performance.

- m: number of coverable targets which is fixed to 10.
- z: upper bound value on sensor overlap which is varied from 1 to 15 to observe the impact of sensor overlap in network performance.
- R_s: maximum sensing range of a sensor. R_s is also varied from $15m$ to $25m$ with fixed number of discrete pans $q = 8$.

Since the simulation environment is a static network and not any kind of protocols have been applied here. Only the proposed algorithms have been implemented. Therefore, rather than using any real simulator, we have worked directly with the raw code for evaluating the performance of our algorithms. The real simulator could be used if the simulation environment is a distributed network and some existing protocols are used in simulations. However, for experimental study, all the proposed heuristics have been programmed here in Java programming language using JDK 1.7 in Eclipse IDE and LP formulation has been solved using cplex optimizer library [52].

Scenario Generation: We simulated a fixed stationary network consisting of randomly deployed sensors and targets in a sensing field of $100m \times 100m$. Initial battery power, E of each sensor is set to 100 time unit (i.e., battery is charged to continue operation of a sensor for 100 time unit). Number of sensors, n is varied ranging from 10 to 80 with an increment of 10 from one scenario to another while keeping the number of targets, m fixed at 10. Here, new sensors were combined with the sensors from the previous scenario in a way such that the former scenario is a subset of the new one. This assures a consistent evaluation of the result of the extended group of sensors by preserving all the features of the former scenario and simply making it better in terms of coverage. Maximum sensing range, R_s is also varied from 15 to 25 with an increment of 2 at each step. The number of pan of each sensor is kept constant, $q = 8$. We also vary the upper bound value of sensor overlap from $z = 1$ to 15 with an increment of 1 to understand the effect of sensor overlap on network performance. For a particular value of n, m, R_s , and z, a simulation network has been generated. LP formulation and the other proposed algorithms are simulated over this network. We have created 5 instances for each scenario over which the simulations have been conducted and results are reported as an average of these 5 instances. Parameters used in the simulation are summarized in Table 4.1.

Parameter	Value
Number of targets, m	10
Number of sensors, n	$10 \sim 80$
Sensing range, R_s	$15m \sim 25m$
Sensing field	$100m \times 100m$
Upper bound value, z	$1 \sim 15$
Initial energy, E	100 unit

Table 4.1: List of simulation parameters

4.2 Performance Metrices

In our experiments, three metrics were used to analyze the performance of the proposed heuristics. They are:– i) number of set covers, ii) network lifetime, and iii) fault tolerance. These performance metrics are defined below.

4.2.1 Number of Set Covers

The number of set covers is an indicator of how many sensor sets capable of monitoring all the targets are discovered by a heuristic. It is always advantageous to create more and more set covers because then the heuristic gets more options to choose from. The bounding parameter z plays a major role here. When $z = 1$, only disjoint set covers are found and this is the minimum possible number of set covers. The number of set covers (i.e., non-disjoint) increases when z is increased. Thus allowing the sensors to overlap unboundedly one can avail the maximum possible number of set covers. Consequently, the number of set covers, $k \propto z$.

4.2.2 Network Lifetime

The network life time is a metric that indicates how long the network remains operative. Longer network lifetime is always expected. As the proposed heuristics partition the sensors into different set covers to schedule them in a round robin fashion for monitoring the targets, consequently the lifetime of the network depends on the lifetime of the individual set covers. Here is how network lifetime is calculated. Assume that all sensors initially have similar energy level. Additionally, assume that available power holds a sensor to run for t time unit. If all the sensors are activated at the same time to cover all the targets, then the lifetime of the network will become $L = t$. Since, sensors are partitioned into different set covers which are scheduled in different time slots, partitioning extends the network lifetime. Therefore, maximizing the number of set covers also maximizes the network lifetime. Since number of set covers, $k \propto z$, so the lifetime, $L \propto z$. If there are k disjoint set covers and each set is being active for t time unit, then the network lifetime becomes, $L = kt$. On the other hand, if there are k overlapping set covers and activation time for a set cover C_i is t_i then network lifetime becomes, $L = \sum_{i=1}^{k} t_i$.

4.2.3 Fault Tolerance

In reality, sensors within a set cover may become faulty or damaged over the time. For disjoint set covers, when a sensor malfunctions only the cover set containing the faulty sensor will be no longer available for providing coverage. Sensors in all other set covers still might continue to monitor the targets. Therefore, increasing number of disjoint set covers will provide maximum fault tolerance in such case. The scenario is different when we allow overlaps within set covers:– the fault tolerance of the network gets suppressed. Since when a sensor becomes faulty or it dies out, all the set covers containing the faulty sensor get affected which notably decrease the network performance. Therefore, the network becomes more faulty as we increase the overlapping upper bound z. Hence, fault tolerance, $F \propto \frac{1}{z}$. Assume that, n sensors are deployed to cover m targets and k non-disjoint set covers have been generated using the overlap upper bound z. And if membership count of a sensor s_i is m_i then the fault tolerance of the network can be formally calculated as:

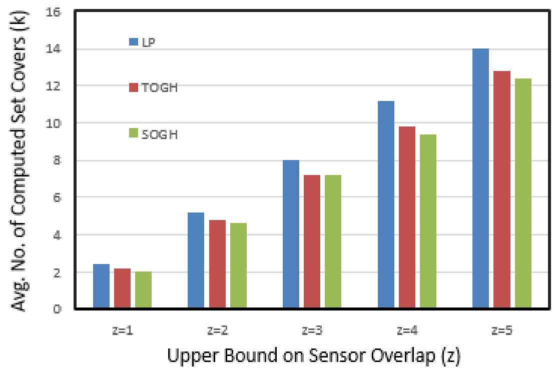
$$F = \frac{1}{\sum_{i=1}^{n} m_i}$$

4.3 Experimental Results with Analysis

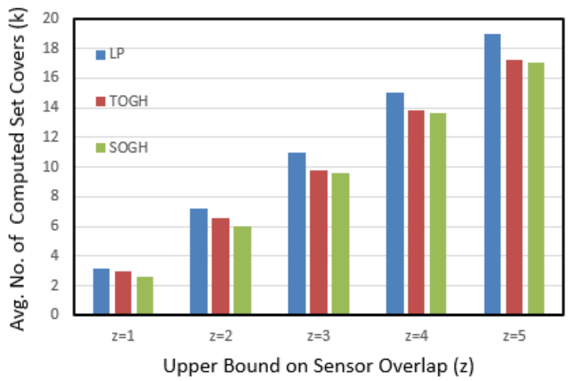
In this section, we will show the experimental result with performance comparison graphs of each experiment. Experimental results are presented for the evaluation of three performance metrices described in previous section 4.2. Simulation results with analysis are discussed below.

4.3.1 Number of Set Covers Evaluation

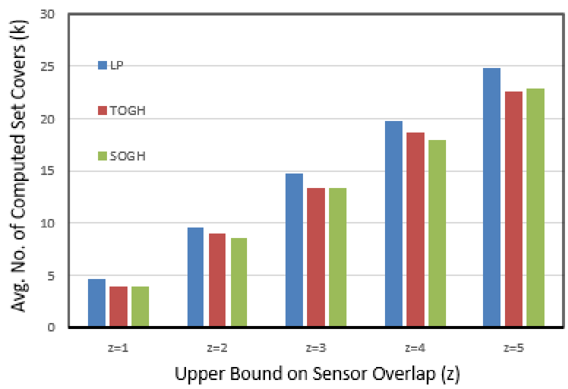
In this experiment, we compare the average number computed set covers (k) generated by TOGH and SOGH with the optimal number of set covers given by LP. To do this comparison, we generate maximum possible set covers using LP and other two heuristics TOGH, and SOGH



(a)



(b)



(c)

Figure 4.1: Average number of computed set covers by varying bound value on sensor overlap for number of sensors (a) $n = 15$ (b) $n = 20$ (c) $n = 25$

by keeping number of targets fixed at $m = 10$; $q = 8$; and $R_s = 25$ unit. We vary the upper bound value, z from 1 to 5 with an increment of 1.

Figure 4.1(a), (b), and (c) demonstrate the simulation results for number of sensors $n = 15$, $n = 20$ and $n = 25$ respectively. In Figure 4.1, X-axis denotes the upper bound value (z) (i.e., the allowable sensor overlap), and Y-axis denotes the average number of computed set covers (k) of 5 random instances. From all subfigures of Figure 4.1, it is evident that the number of set covers is increased as the upper bound value on sensor overlapping is increased. However, LP formulation maximizes the number of set covers and outperforms other two heuristics TOGH and SOGH. On the other hand, solutions provided by TOGH and SOGH are very much close to LP. Among TOGH and SOGH, TOGH provides a better result than SOGH in most cases and SOGH gives the worst performance. Although, LP formulation provides the best result but the computational time of LP is much higher compared to greedy heuristics TOGH and SOGH. In Table 4.2, we compare the running time between LP, SOGH, and TOGH for the same scenarios. This comparison has been done for $n = 25$, $m = 10$, $q = 8$, $z = 1$, and sensing range is varied ranges from 15 to 25 with an increment of 2. From the Table 4.2, it can be seen that running time of LP is much higher in compared to other heuristics SOGH and TOGH and is increased with the problem size. Whereas the proposed greedy heuristics TOGH and SOGH are solvable in polynomial time and much more scalable than LP. Based on this Table 4.2, the approximation ratio of the proposed heuristics has been shown in Table 4.3. Approximation ratio of an algorithm is the ratio between the calculated solution by the algorithm and the actual optimal value. Mathematically, this ratio can be computed as follows.

$$\rho = \max \{\tfrac{C^*}{C}, \tfrac{C}{C^*}\}$$

where C^* indicates the actual optimal solution and C represents the calculated solution. The value of ρ indicates the diversion of the calculated solution from the optimal value. When it is equal to 1, it means the obtained solution is optimal. The more it is greater than 1 indicates more it has deviated from the optimal value. From Table 4.3, average approximation ratio for SOGH and TOGH are 1.15 and 1.10 respectively.

4.3.2 Network Lifetime Evaluation

In Figure 4.2 , 4.3, and 4.4, we present the network lifetime (L) which is proportionate to the number of set covers (k). Results are shown as an average of 5 random instances. Figure 4.2 shows the network lifetime in terms of varying the number of sensors n between 10 to 80 with

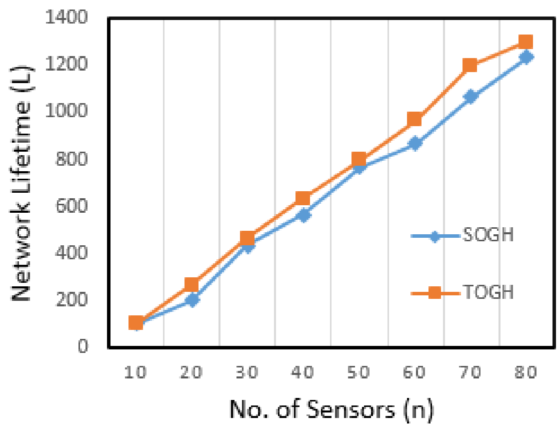
Sensing range (R_s)	LP		SOGH		TOGH	
	Run time (ms)	Avg. no. of computed set covers (k)	Run time (ms)	Avg. no. of computed set covers (k)	Run time (ms)	Avg. no. of computed set covers (k)
15	18638.28	2.4	5.58	2.2	17.37	2.2
17	19917.40	3	6.12	2.8	17.65	2.8
19	26182.38	3.4	6.61	3	19.03	3
21	31630.61	4	7.22	3.4	19.73	3.4
23	32616.84	4	7.53	3.4	21.69	4
25	42298.99	4.6	7.58	3.6	23.75	4

Table 4.2: Runtime of LP, SOGH and TOGH

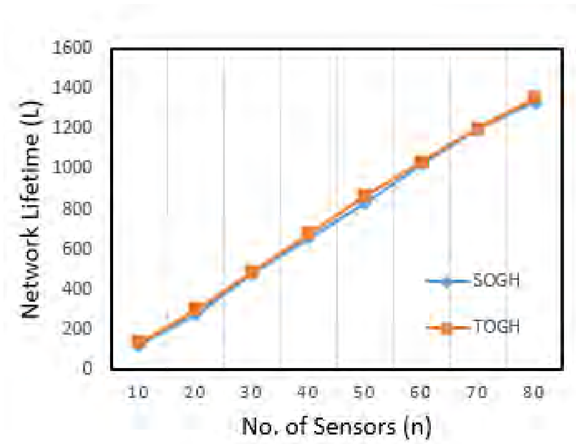
Sensing range (R_s)	Approximation ratio (SOGH)	Approximation ratio (TOGH)
15	1.09	1.09
17	1.07	1.07
19	1.13	1.13
21	1.17	1.17
23	1.17	1
25	1.27	1.15
Average	1.15	1.10

Table 4.3: Approximation ratio of SOGH and TOGH for disjoint sets approach

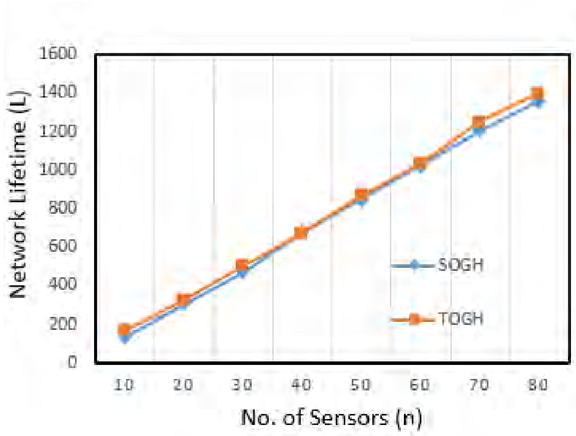
an increment of 10 while keeping $m = 10$; $R_s = 25$ unit; $q = 8$. Figure 4.2 (a), (b), and (c) show the simulation results for bound value $z = 1$, $z = 3$ and $z = 5$ respectively. Initially battery power of each sensor set to 100 unit. Clearly, lifetime of the network (L) increases with the number of sensors (n). This is because when more sensors participate to cover the targets, probability of generating more set covers also increases which enhance the network lifetime. On the other hand, network lifetime with respect to varying sensors sensing range are shown in Figure 4.3. Here, we vary sensing range R_s from 15 to 25 with an increment of 2 while keeping $n = 25$; $m = 10$; $q = 8$. Simulation results for bound value $z = 1$, $z = 3$, and $z = 5$ are displayed in Figure 4.3 (a), (b) and (c) respectively. Like Figure 4.2, network lifetime is also increased when we increase the sensors' sensing range as the coverage region of a sensor is enhanced by increasing sensors sensing range. In Figure 4.4, we plot the network lifetime with respect to overlapping bound z of the sensors within set covers. Figure 4.4 (a), (b), and (c) show the simulation result for $n = 15$, $n = 20$ and $n = 25$ sensor nodes respectively while keeping the number of targets at $m = 10$, sensing range at, $R_s = 25$ unit and $q = 8$.



(a)



(b)



(c)

Figure 4.2: Network lifetime with respect to varying number of sensors for bound value (a) $z = 1$ (b) $z = 3$ (c) $z = 5$

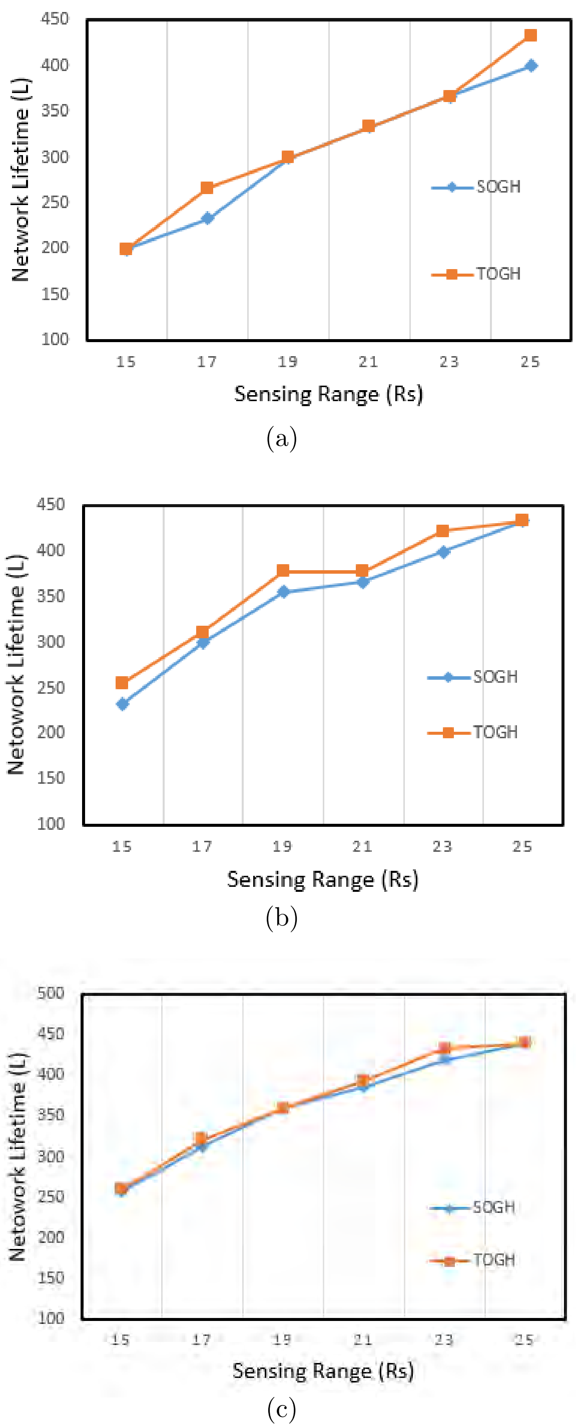
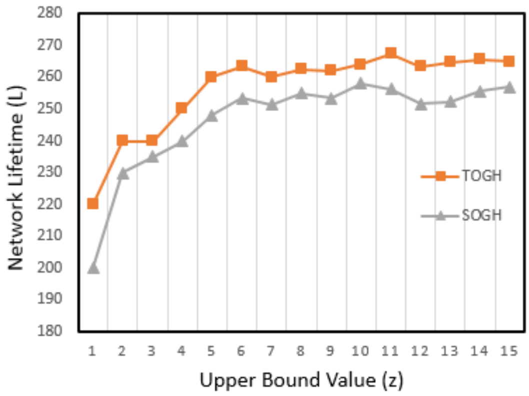
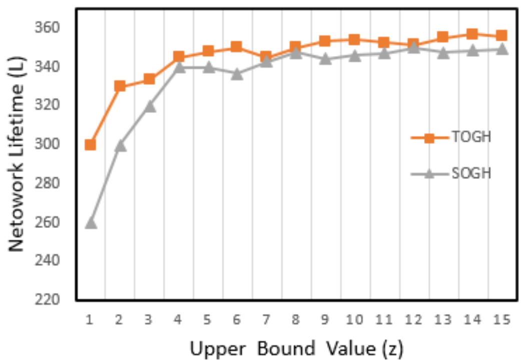


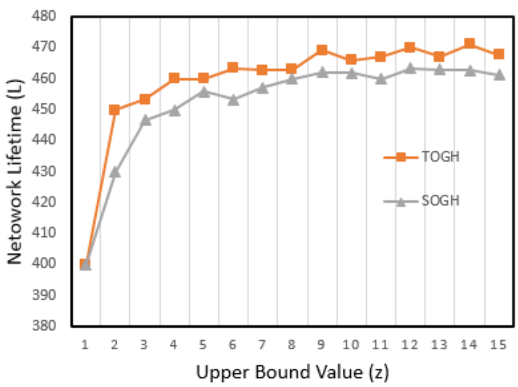
Figure 4.3: Network lifetime with respect to varying sensing range for bound value (a) $z = 1$ (b) $z = 3$ (c) $z = 5$



(a)



(b)



(c)

Figure 4.4: Network lifetime with respect to bound value on sensor overlap for number of sensors (a) $n = 15$ (b) $n = 20$ (c) $n = 25$

Upper bound value is varied from 1 to 10 with an increment of 1. From the Figure 4.4, it is evident that disjoint set covers (i.e., $z = 1$) provide the least network lifetime and lifetime of the network increases as the bound value z is increased. Therefore, increasing sensor overlap within set covers enhances the network lifetime. From Figure 4.2 , 4.3, and 4.4, it is observed that performance of both heuristics TOGH and SOGH are very close to each other. However, TOGH has better network lifetime than SOGH in most of the cases as TOGH generates more number of set covers than SOGH.

4.3.3 Network Fault Tolerance Evaluation

In Figure 4.5, 4.6, and 4.7, we plot the network fault tolerance (F) w.r.t. number of sensors, sensing range and upper bound on sensor overlap (z) respectively. From the Figures it is clearly evident that fault tolerance is inversely proportional to the number of overlapping set covers. For instance, in Figure 4.5, we vary the number of sensors, n from 10 to 80 with an increment of 10 while in Figure 4.6, sensing range is varied ranges from 15 to 25 with an increment of 2. In both cases we set $m = 10$; $R_s = 25$; $q = 8$. Figure 4.5 (a), (b) and (c) show the network fault tolerance w.r.t. number of sensors for bound value $z = 1$, $z = 3$ and $z = 5$ respectively while 4.6 (a), (b) and (c) show the network fault tolerance w.r.t. sensors sensing range for bound value $z = 1$, $z = 3$ and $z = 5$ respectively. In both Figures 4.5 and 4.6, network fault tolerance decreases when either the number of sensors or sensors' sensing range is increased. This is because, if we increase number of sensors or sensors' sensing range, the number of participating sensors within the set covers also gets increased. With more sensor overlaps the fault tolerance gets lowered. Figure 4.7 shows the network fault tolerance (F) when the upper bound value on sensor overlap (z) is varied. z was varied from 1 to 15 with the increment of 1 at each step while keeping $n = 25$; $m = 10$; $q = 8$. Figure 4.7 (a), (b), and (c) show the simulation result for $n = 15$, $n = 20$ and $n = 25$ sensor nodes. Clearly, fault tolerance decreases when (z) is increased. Disjoint set cover provides maximum fault tolerance, and allowing more sensor overlap within the set covers, the network becomes more error-prone. Another thing, from the Figure 4.5, 4.6, and 4.7, it is evident that fault tolerance of sensor oriented heuristics (SOGH) is higher than target oriented heuristics (TOGH) in most scenarios. As TOGH generates more number of (overlapping) set covers than SOGH, naturally the fault tolerance value gets lowered.

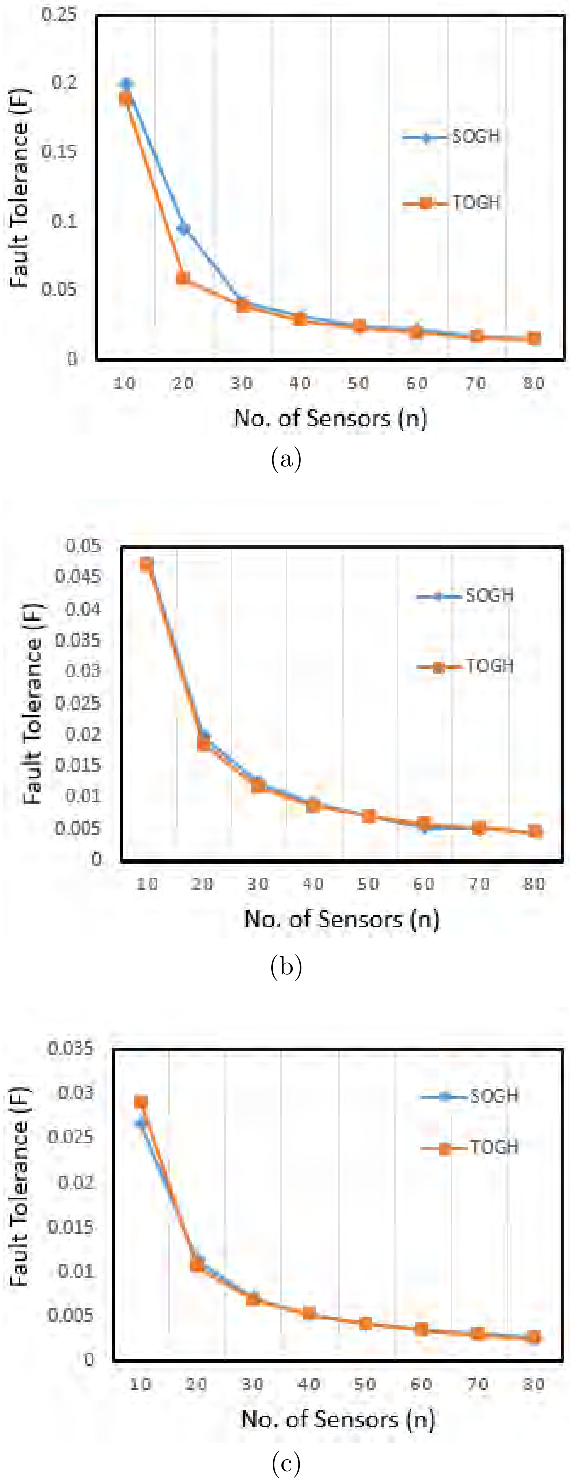
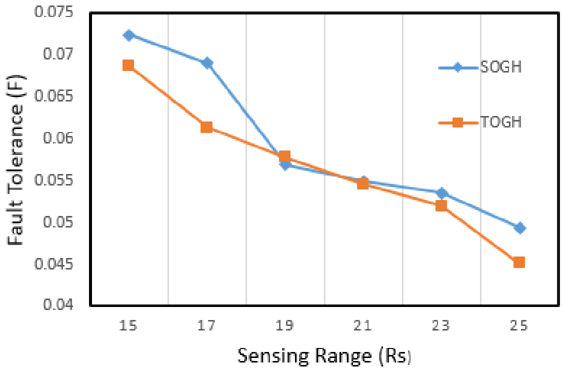
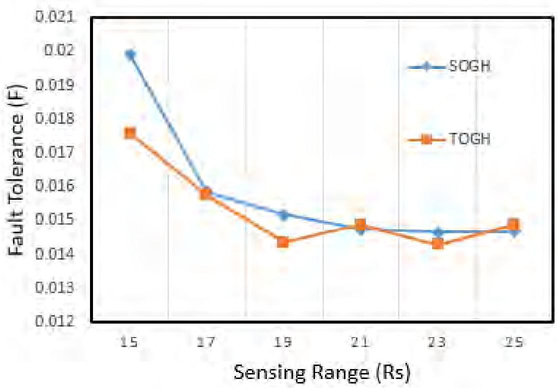


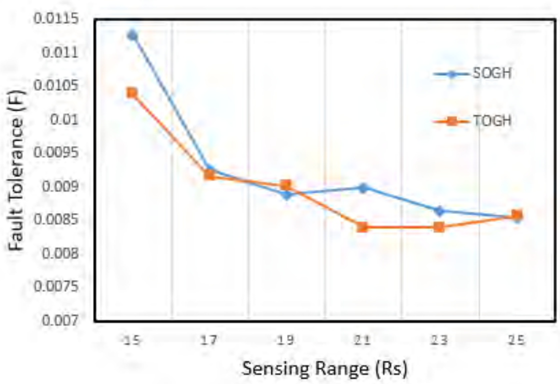
Figure 4.5: Network fault tolerance with respect to varying number of sensors for bound value (a) $z = 1$ (b) $z = 3$ (c) $z = 5$



(a)



(b)



(c)

Figure 4.6: Network fault tolerance with respect to varying sensing range for bound value (a) $z = 1$ (b) $z = 3$ (c) $z = 5$

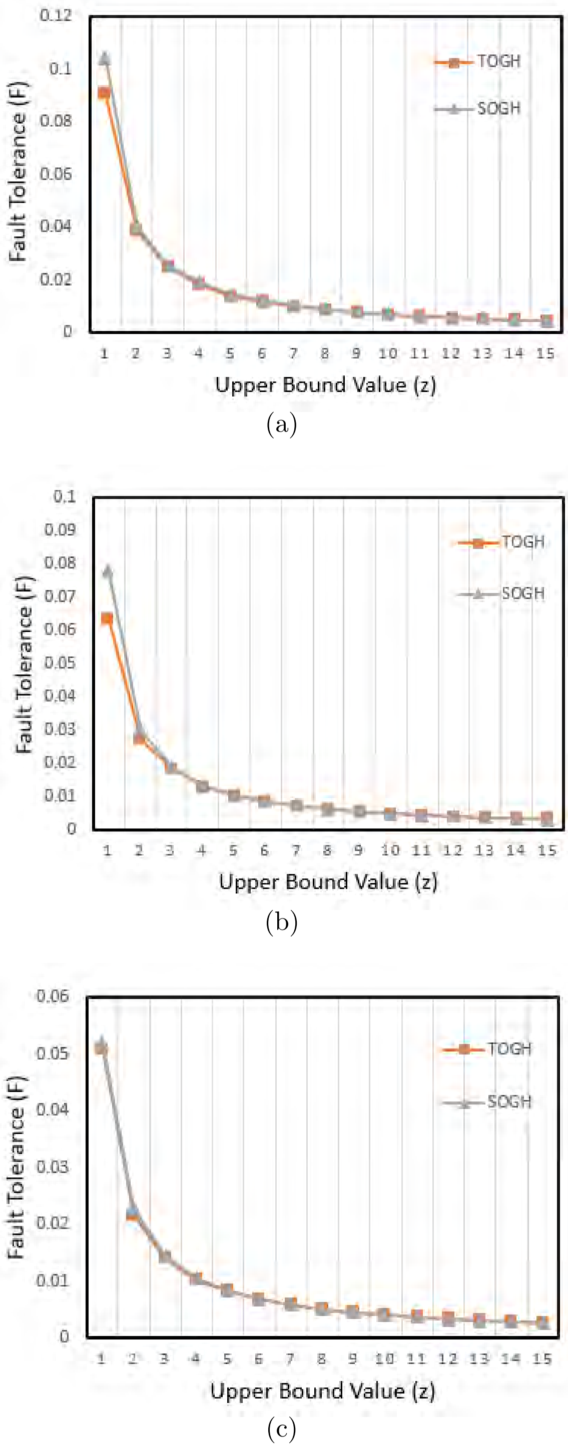


Figure 4.7: Network fault tolerance with respect to bounded sensor overlap for number of sensor nodes (a) $n = 15$ (b) $n = 20$ (c) $n = 25$

4.3.4 Determination of Optimal Bound Value

To determine the optimal upper bound (z^*) on sensor overlap, we have introduced a metric called Goodness Index, GI, which is a function of lifetime and fault tolerance of a network. It denotes how good a network is when both lifetime and fault tolerance is considered together. The particular value of z for which GI is maximum is defined as optimal upper bound (z^*). Goodness Index (GI) is defined as follows:

$$GI = \alpha L_z + (1\text{-}\alpha) F_z$$

Here,

- The fault tolerance index F_z denotes normalized value of the fault tolerance of the network at upper bound z. Recall that, fault tolerance is calculated from membership count of a sensor within the set covers. For instance, if membership count of a sensor s_i is m_i then the fault tolerance $F = \frac{1}{\sum_{i=1}^{n} m_i}$. And then all fault tolerance values are normalized and a lower value of F_z index indicates the network is more error prone.
- L_z denotes normalized value of lifetime index of the network at bound value z.
- α is a tunable parameter between 0 to 1. When $\alpha = 1$, only lifetime is considered; when $\alpha = 0$, only fault tolerance is considered. But for $\alpha = 0.5$, both lifetime and fault tolerance is considered in equal proportion to evaluate GI.

Next we describe how we have normalized the values. Consider Table 4.4 which shows the network lifetime and fault tolerance of a sample network scenario for different upper bounded on sensor overlap. The value of network lifetime and fault tolerance at column # 2 and 3 have been normalized between 0.5 to 1 and results are shown at column # 4 and 5 respectively. Here, the values are normalized by applying the following min-max normalization equation.

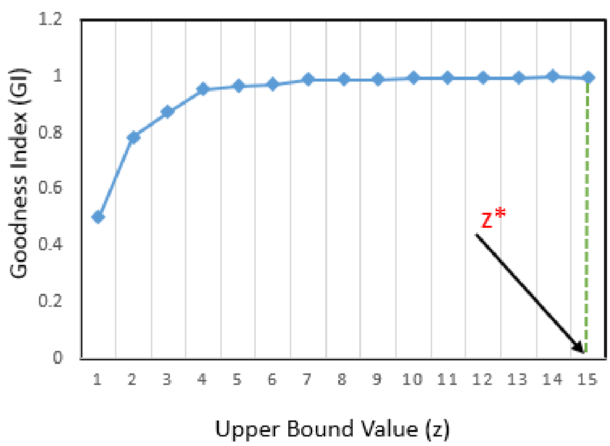
$$x' = \frac{x - min_Y}{max_Y - min_Y}(new_{max} - new_{min}) + new_{min}$$

Where, min_Y and max_Y are the minimum and maximum value of an attribute Y. This formulation converts the value, x of an attribute Y to x' in the limit $[new_{min}, new_{max}]$.

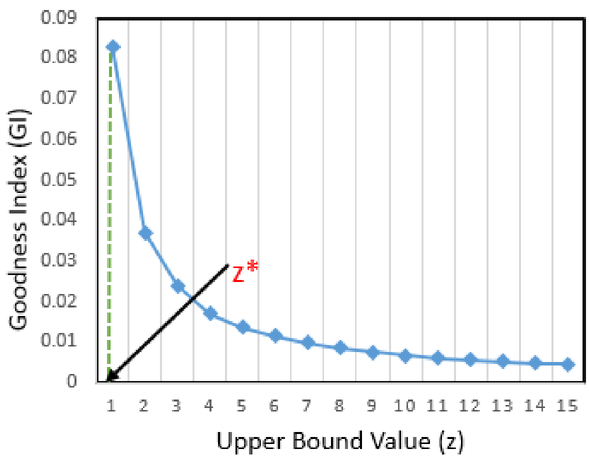
Bound value (z)	Lifetime	Fault Tolerance	Normalized Lifetime (L_z)	Normalized Fault Tolerance (F_z)
1	225	0.082921245	0.5	1.0
2	256.25	0.037087912	0.784090909	0.714392882
3	266.25	0.023990544	0.875	0.63277714
4	275	0.016951588	0.954545455	0.588914154
5	276.25	0.013562312	0.965909091	0.567794015
6	277	0.011499494	0.972727273	0.554939643
7	279	0.009712885	0.990909091	0.543806458
8	279	0.008493301	0.990909091	0.536206666
9	279	0.00753325	0.990909091	0.530224144
10	279.5	0.00672801	0.995454545	0.525206322
11	279.5	0.006097674	0.995454545	0.521278411
12	279.50	.005610307	0.995454545	0.518241401
13	279.5	0.00522225	0.995454545	0.51582324
14	280	0.004790717	1.0	0.513134156
15	279.5	0.00447434	0.995454545	0.511162665

Table 4.4: Sample network lifetime and fault tolerance for bounded overlapping

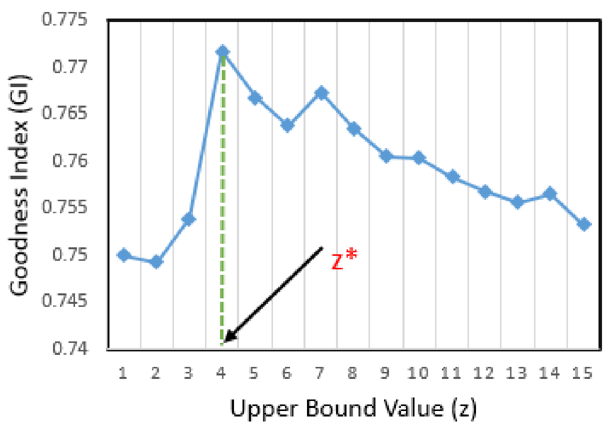
Based on this Table 4.4, we draw three plots of goodness index (GI) with respect to the upper bound on sensor overlap (z) as shown in Figure 4.8. X-axis indicates the upper bound value z and Y-axis represents the goodness index (GI). From the plot in Figure 4.8 (a), it is evident that, when we set $\alpha = 1$, only lifetime will be considered and goodness index (GI) increases as z is increased. In that case, largest value of z indicates the optimal bound value (z^*). In Figure 4.8 (b), we set $\alpha = 0$, i.e. only fault tolerance is considered and GI decreases as z is increased. In that case, disjoint scheme (i.e., $z = 1$) indicates the optimal bound value (z^*). Finally in Figure 4.8 (c), we set $\alpha = 0.5$, i.e., both the lifetime and fault tolerance are considered together. In this case, some upper bound between 1 to ∞ becomes the optimal bound value (z^*).



(a)



(b)



(c)

Figure 4.8: Measure the goodness index for bounded sensor overlap when (a) $\alpha = 1$ (b) $\alpha = 0$ (c) $\alpha = 0.5$

4.4 Summary and Comments

From the above discussions, it can be stated that disjoint scheme (i.e., $z = 1$) gives us maximum fault tolerance and increasing the sensor overlap within set covers will increase the faultiness of the network. As a result, network becomes bad interms of fault tolerance for unbounded sensor overlapping. On the other hand, in another approach: allowing the sensors to overlap unboundedly ($z = \infty$) will provide maximum network lifetime. Therefore, there is a trade-off between prolonging the network lifetime and providing fault tolerance by increasing sensors overlapping within set covers. Hence, it can be concluded that during target coverage we should walk in the middle, neither disjoint scheme nor unbounded overlap will give us the best network performance.

Chapter 5

Conclusion and Future Works

Wireless sensor networks (WSNs) have received a lot of attention during the past several years due to its enormous contributions in various day-to-day applications. Due to directional sensors sensing characteristics, coverage issue in directional sensor networks has taken newly attraction which indicates how well the area of interest (AOI) is monitored. As directional sensor nodes are equipped with limited battery power and difficult to replace or renew their battery power or energy, therefore keeping all the sensors active at a time would drain their power quickly during coverage. Sometimes, a sensor node failures or malfunctions may cause total coverage failure. So considering these factors, the target coverage problem is yet a challenging task, which attempts to settle between maintaining the coverage of a set of targets with the required level of coverage reliability and maximizing the network lifetime.

5.1 Summary

In this thesis, we have studied the target coverage problem using directional sensors in WSNs and constructed energy-efficient target monitoring algorithms considering both lifetime and fault tolerance of a network. At first, we have formulated an exact solution of the problem as a linear programming (LP) formulation by adding necessary constraints to minimize power consumption using least number of sensors. Then we have developed two greedy heuristics to find the maximum possible set covers each of which individually covers all the targets.

Additionally, we have developed:– i) a method to schedule the set covers such that the network can avail maximum lifetime by considering sensor membership value and residual energy, and ii) a method for measuring network fault tolerance. Performance of the proposed algorithms have been demonstrated by conducting extensive simulations. During simulation, we have used three performance metrics to examine the performance of our algorithms. Simulation results reveal the fact that there is an inevitable trade-off between prolonging the network lifetime and providing fault tolerance by increasing sensor overlaps within set covers. As a solution of this trade-off, we have used a performance metric called goodness index which helps in determining the optimal bound on sensor overlap by considering both network lifetime and fault tolerance.

5.2 Future Works

There are some scopes for future studies on this thesis. They are listed as follows:

i. In our study, sensor and target nodes are considered to be static with known location. So, further experiment can be done over mobile nodes as extension of our work.

ii. Next, in our experiment, we have considered only pan camera. Therefore, we may conduct the experiment over a sensor with two other dimensions in real sensor network.

iii. In near future, we will perform more simulations using different grid sizes and varying sensing ranges.

Bibliography

[1] J.-M. Gil and Y. H. Han. A target coverage scheduling scheme based on genetic algorithms in directional sensor networks. *Sensors*, 11(2):1886–1906, 2011.

[2] J. Willson, Z. Zhang, W. Wu, and D. Z. Du. Fault-tolerant coverage with maximum lifetime in wireless sensor networks. In *IEEE Conference on Computer Communications (INFOCOM), Kowloon, Hong Kong, April 26 - May 1*, 2015.

[3] S. Mini, S. K. Udgata, and S. L. Sabat. Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal*, 14(3):636 – 644, 2014.

[4] M. A. Attar, S. Yarahmadian, and S. Samavi. Coverage estimation in heterogenous floorplan visual sensor networks. In *IEEE SENSORS 2013, Baltimore, Maryland, USA, November 3 - 6*, 2013.

[5] S. C. Misra, I. Woungang, and S. Misra. *Guide to Wireless Sensor Networks*. Springer, 2009.

[6] O. Banimelhem, M. Naserllah, and A. Abu-Hantash. An efficient coverage in wireless sensor networks using fuzzy logic-based control for the mobile node movement. In *2017 Advances in Wireless and Optical Communications (RTUWO), Riga, Latvia, November 2 - 3*, 2017.

[7] N. Ahn and S. Park. A new mathematical formulation and a heuristic for the maximum disjoint set covers problem to improve the lifetime of the wireless sensor network. *Ad-Hoc and Sensor Wireless Networks*, 13(1):209–225, 2011.

[8] D. Bajaj and M. Choudhary. Maximum coverage heuristics (mch) for target coverage problem in wireless sensor network. In *IEEE International Advance Computing Conference, At ITM University, Gurgaon, India, February 21 - 22*, 2014.

[9] S. Farzana, K. A. Papry, A. Rahman, and R. Rab. Maximally pair-wise disjoint set covers for directional sensors in visual sensor networks. In *Wireless Days (WD), Toulouse, France, March 23-25*, 2016.

[10] S. Biswas, R. Das, and P. Chatterjee. Energy-efficient connected target coverage in multi-hop wireless sensor networks. *Industry Interactive Innovations in Science, Engineering and Technology*, 11(1):411–421, 2017.

[11] B. Diop, D. Diongue, and O. Thiare. Target coverage management in wireless sensor networks. In *2014 IEEE Conference on Wireless Sensors (ICWiSE), Subang, Malaysia , October, 26-28*, 2014.

[12] G. Han, L. Liu, J. Jiang, L. Shu, and G. Hancke. Analysis of energy-efficient connected target coverage algorithms for industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 13(1):135–143, 2017.

[13] J. Ai and A. A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *Journal of Combinatorial Optimization*, 11(1):21–41, 2006.

[14] Y. C. Hsu, Y. T. Chen, and C. K. Liang. The coverage problem in directional sensor networks with rotatable sensors. In *8th International Conference on Ubiquitous Intelligence and Computing, UIC, Banff, Canada, September 2-4*, 2011.

[15] X. Z. Zhang and Y. F. Li. Simulation of coverage problem research in wireless sensor networks based on energy saving. In *2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China, March 23-25*, 2012.

[16] C. W. Yu, K.-M. Yu, W.-T. So, C.-T. Lin, and C. Y. Lin. Target coverage in wireless sensor networks. In *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks, Beijing, China, December 16-18*, 2011.

[17] K. Rajeswari and S. Neduncheliy. Cluster based fault tolerance using genetic algorithm in wireless sensor network. In *International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, February 25-26*, 2016.

[18] D. Dash. Energy efficient fault tolerant barrier coverage using limited mobility wireless sensors. In *3rd International Conference on Recent Advances in Information Technology (RAIT), Dhanbad, India, March 3-5*, 2016.

[19] G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), Rome, Italy, June 22-26*, 2009.

[20] S. M. B. Malek, M. M. Sadik, and A. Rahman. On balanced k-coverage in visual sensor networks. *Journal of Network and Computer Applications*, 72(1):72–86, 2016.

[21] K. B. Amir, M. N. A. Shaon, and M. A. Matin. A new energy-efficient scheduling scheme for multiple target coverage in wireless sensor networks. In *2011 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, November 9-11*, 2011.

[22] D. P. Singh, R. H. Goudar, and S. Rao. Coverage lifetime improvement through efficient deployment of sensor nodes in wsn. *International Journal of System Assurance Engineering and Management*, 7(2):201–212, 2014.

[23] M. Cardei and J. Wu. Energy-efficient coverage problems in wireless ad hoc sensor networks. *Computer Communications*, 29(4):413–420, 2006.

[24] A. Rossi, A. Singh, and M. Sevaux. Lifetime maximization in wireless directional sensor network. *European Journal of Operational Research*, 231(1):229–241, 2013.

[25] K. Islam and S. G. Akl. A localized algorithm for target monitoring in wireless sensor networks. In *International Conference on Ad-Hoc, Mobile and Wireless Networks, Murcia, Spain, September 22-25*, 2009.

[26] M. Cardei. *Coverage Problems in Sensor Networks*. Springer, 2013.

[27] S. Tomovic and I. Radusinovic. Energy efficient target coverage in partially deployed software defined wireless sensor network. In *Cognitive Radio Oriented Wireless Networks 11th International Conference, CROWNCOM , Grenoble, France, May 30 - June 1*, 2016.

[28] T. Shimokawa and A. Fujiwara. Centralized algorithms for the connected target coverage in wireless sensor networks. In *2012 Third International Conference on Networking and Computing, Okinawa, Japan , December 5 - 7*. IEEE, 2012.

[29] J. Yu, Y. Chen, and B. Huang. On connected target k-coverage in heterogeneous wireless sensor networks. In *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI), Beijing, China, October 22-23*, 2015.

[30] J. Yu, X. Deng, D. Yu, G. Wang, and X. Gu. Connected k-coverage working sets construction algorithm in wireless sensor networks. *AEU - International Journal of Electronics and Communications*, 67(11):937 – 946, 2013.

[31] H. M. Ammari and S. K. Das. Coverage, connectivity, and fault tolerance measures of wireless sensor networks. In *8th International Conference on Stabilization, Safety, and Security of Distributed Systems, Dallas, TX, USA, November 17 - 19*, 2006.

[32] M. Chaudhary and A. K. Pujari. Q-coverage problem in wireless sensor networks. In *10th International Conference on Distributed Computing and Networking, ICDCN 2009, Hyderabad, India, January 3-6*. Springer, 2009.

[33] B. Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys (CSUR)*, 43(4):Article No. 32, 2011.

[34] A. Sangwan and R. P. Singh. Survey on coverage problems in wireless sensor networks. *Wireless Personal Communications*, 80(4):1475–1500, 2015.

[35] D. P. Singh and B. Pant. An approach to solve the target coverage problem by efficient deployment and scheduling of sensor nodes in wsn. *International Journal of System Assurance Engineering and Management*, 8(2):493 – 514, 2016.

[36] N. Gupta, N. Kumar, and S. Jain. Coverage problem in wireless sensor networks: A survey. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, October 3-5*, 2016.

[37] V. P. Munishwar and N. Abu-Ghazaleh. Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 9(4):Article 45, 2013.

[38] D. Bein, W. W. Bein, and S. Malladi. Fault tolerant coverage model for sensor networks. In *5th International Conference on Computational Science, Atlanta, GA, USA, May 22-25*, pages 535 – 542, 2005.

[39] W. M. Elsayed, S. F. Sabbeh, and A. M. Riad. A distributed fault tolerance mechanism for self-maintenance of clusters in wireless sensor networks. *Arabian Journal for Science and Engineering*, 43(12):6891–6907, 2018.

[40] H. Liu, A. Nayak, and I. Stojmenovi. *Fault-Tolerant Algorithms/Protocols in Wireless Sensor Networks*. Springer, London, 2009.

[41] D. Li, J. Cao, M. Liu, and Y. Zheng. K-connected target coverage problem in wireless sensor networks. In *First International Conference, COCOA, Xian, China, August 14-16*, 2007.

[42] Y. Gu, H. Liu, and B. Zhao. Target coverage with qos requirements in wireless sensor networks. In *2007 International Conference on Intelligent Pervasive Computing, Jeju City, South Korea, October 11-13*, 2007.

[43] D. Arivudainambi, S. Balaji, R. Pavithra, and R. N. Shakthivel. Energy efficient sensor scheduling for q-coverage problem. In *2017 IEEE 22nd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Lund, Sweden, June 19-21*, 2017.

[44] S. Balaji, T. Priyanka, and M. Anitha. Optimal deployement of sensors in 3d - terrain with q-coverage constraints. In *IEEE SENSORS, New Delhi, India, October, 28-31*, 2018.

[45] G. Calinescu and R. B. Ellis. Monitoring schedules for randomly deployed sensor networks. In *Proceedings of the DIALM-POMC Joint Workshop on Foundations of Mobile Computing, Toronto, Canada, August 18-21*, 2008.

[46] J. Jia, C. Dong, X. He, D. Li, and Y. Yu. Sensor scheduling for target coverage in directional sensor networks. *International Journal of Distributed Sensor Networks*, 13(6):1 – 12, 2017.

[47] S. Slijepcevic and M. Potkonjak. Power efficient organization of wireless sensor networks. *IEEE International Conference on Communications*, 2(2):472 – 476, 2001.

[48] M. Cardei and D. Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3):333 – 340, 2005.

[49] V. P. Munishwar and N. Abu-Ghazaleh. Target-oriented coverage maximization in visual sensor networks. In *9th MOBIWAC ACM International Symposium on Mobility Management and Wireless Access MOBIWAC, Miami Beach, FL, USA, October 31-November 04*, 2011.

[50] Y. Kim, H. Lee, Y. Han, and Y. Jeong. A branch and bound algorithm for extending the lifetime of wireless sensor networks. In *IEEE 70th Vehicular Technology Conference Fall , Anchorage, AK, USA, September 20-23*, 2009.

[51] Y.-H. Han, C.-M. Kim, and J.-M. Gil. A greedy algorithm for target coverage scheduling in directional sensor networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, 1:96–106, 2010.

[52] LP solver. `https://www.ibm.com/analytics/cplex-optimizer`. [Online; accessed 22-January-2019].