MEASURING AND IMPROVING THE QUALITY OF BARRIER COVERAGE

Thesis submitted in partial fulfilment
of the requirements for the degree
of

Master of Technology
in

Computer Science and Engineering
by

Nitin Awathare

13CS60R07

Under the supervision of

Prof. Arobinda Gupta

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur,
WB 721302, India

April 2015

Abstract

Coverage is a problem area in Wireless Sensor Networks that determines how well an area is monitored by the sensor network. One variation of the coverage problem is *Barrier Coverage.* The main aim of barrier coverage is to detect intruders as they penetrate the monitored area. A region is said to be k-barrier covered when every crossing path through the region is detected by at least k sensors.

Sensor nodes may fail due to several reasons such as environmental hazards, lack of power supply etc. The failure of sensors after a certain extent, deteriorates the degree of coverage provided by the sensor barriers and allows intruder penetration through paths undetected by the sensor network. Quality of barrier coverage is a measure of the degree of coverage provided by the sensor deployment. If the quality of coverage degrades below a threshold level, the network needs to be repaired. The degrade in the quality is due to formation of gaps in the barrier or decrease in the length of overlapping region in component of the deployment. Such components are termed as segments. The first problem addressed in the thesis thus focuses on determining the relationship between the quality of coverage and overlapping region. Later on effect of a sensor failure on the number of such overlapping region is discussed.

Improving the quality of coverage requires identifying the regions that needs to be repaired. The network can be repaired by either re-deploying all the sensors or moving sensors. Re-deploying of sensors is not always feasible. Instead the method of relocation of few sensors to the region needs to be repaired incurs less effort and is preferred for maintaining the coverage. This gave a scope to design algorithms to improve the quality of barrier coverage by relocating the sensors. The proposed algorithm gives methods for detecting the best suitable sensor in the network that should be relocated to the region that need to be repaired to improve the quality . Firstly, a centralised algorithm to improve the quality by moving the sensors horizontally is proposed. After that the algorithm to improve the quality by moving the sensors horizontally as well as vertically in the given deployment is proposed. Experimental results are presented to analyse the algorithms.

Chapter 1

Introduction

A sensor network consists of a collection of sensor nodes that can communicate among themselves. Each sensor node comprises of sensors which senses some physical parameters such as heat, light, temperature etc, a processing capability i.e a micro controller or CPU, memory to store the processed data, a radio transceiver to communicate with other nodes and a power source in the form of batteries and solar cells. A wireless sensor network (WSN) is formed by some sensor nodes that communicate among themselves to accomplish some distributed sensing tasks. Distributed sensing, ad-hoc deployment and adaptability to changing environment have made wireless sensors preferred over wired devices in many applications and increased its usage. Wireless sensor networks have been used in diverse fields such as in battlefield surveillance that includes monitoring and tracking of objects, industrial automation and process monitoring, military, traffic monitoring, air traffic control, monitoring of environmental conditions, medical device monitoring, entertainment, transportation etc.

There exists several research areas in wireless sensor networks. Sensor nodes in a wireless sensor network communicate with one or more sensors or base stations in the sensor field to produce information about the physical environment. These nodes are constrained in energy supply, processing, storage and communication bandwidth. The constraints pose challenges on design and management of WSN with main focus on resource management to increase network lifetime. Communication being a major source of energy consumption necessitates the routing techniques to be energy efficient [20]. With random deployment of sensors in a sensor field, clusters of sensor nodes with different densities can form over a small area. To capture all events and physical phenomena the field should be well covered. Moreover arbitrary sensor deployment and changing position of sensors may result in low communication among the nodes. Thus topology control is required to optimise the resources and provide good communication and connectivity

[21]. It characterises how well pairs of sensors are connected to the network. Moreover, as large amount of data is transmitted over the sensor network, data compression and aggregation are required. Data aggregation algorithms aims to gather and aggregate data at intermediate sensor nodes without loosing any information [7] . Aggregation removes redundant data and the processed data is passed to the next hop nearest to the base station. Data-compression techniques involve compressing the size of the data before transmission such that no information is lost and readings are retained. Reduced data size reduces the bandwidth required for transmitting data. To avoid collision during data transmission, global time synchronization allows data to be transmitted in a scheduled manner. Distributed, wireless sensor networks make extensive use of synchronized time to prevent redundant message delivery by recognising them as duplicate decisions made by different sensors of the same event [22], synchronising the wake up and sleep schedules etc. Deployment of sensors includes two important problems of wireless sensor network - coverage and localization. Localisation process refers to identifying the geographic position of a sensor node by virtue of the communication each sensor has with some sensor nodes whose locations are known using various localisation technologies. Localisation plays a vital role in WSN because in certain applications e.g. target tracking, if the location information is not available other related data cannot be derived. Existing localization methods includes global positioning systems (GPS) and proximity based localization [25]

In this thesis we focus on the problem of coverage in wireless sensor network. The coverage problem addresses the issue of how well a given region is covered by a set of sensors [8]. The problem is discussed in details below.

1.1 Coverage in Wireless Sensor Network

Coverage can be defined as how well an area is covered by a sensor network. A point is said to be covered if it falls in the sensing region of some sensors. Different variations of the coverage problem exist based on certain design parameters as follows:

- Region to be covered: Problem formulations can be made depending on whether the entire area or only discrete points or boundaries need to be covered.

- Sensor deployment schemes: This includes either probabilistic or deterministic deployment. In deterministic deployment the sensor locations are predefined and feasible for regions that are easily accessible. But in practical cases such as along long borders, sensors are basically air dropped and thus follows a probabilistic deployment.

- Objective: Based on the objective of whether to provide coverage maximising network lifetime or minimising number of sensors used for coverage, variations of coverage problem can be derived.

- Types of sensors: The problems can be classified based on static or mobile sensors used.

- Sensing characteristics: Coverage problems can be formulated by virtue of features such as sensing range and communication range. The sensing or communication range of all sensors may remain same or vary, accordingly posing different constraints in providing coverage.

Based on type of region to be covered coverage problem can be classified into area coverage and point coverage as shown in Figure 1.1

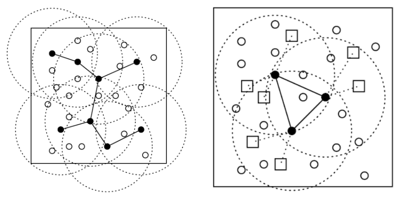
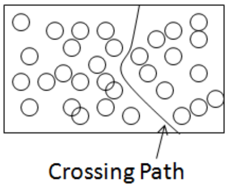


Figure 1.1: Different Coverage Types

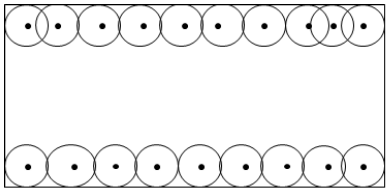
Area coverage requires an area to be covered such that each point in the area is covered by some sensor [24]. Figure 1.1(a) shows area coverage provided by the set of sensors denoted by black dots. The other sensors remain in a sleep state reducing energy consumption. Point or target coverage aims to cover a fixed number of targets with known locations. In perimeter coverage sensors are placed to cover the perimeter of the area to be monitored [23]. The main focus of this work is on barrier coverage which is discussed next section

1.2 Barrier Coverage

Barrier coverage deals with detection of intruders penetrating a protected area. The sensors form a barrier over the region. An intruder is an object that is to be detected by a sensor network as it crosses the barrier. As intruders should be detected before they cross the borders, full coverage to keep track of the trajectory at every location is not necessary in detecting intruders entering the region. Thus in barrier coverage the sensors form a barrier over the area such that any intruder penetrating the area is detected.



(a) Crossing Path



(b) 2-Barrier Coverage

Figure 1.2: Barrier Coverage

Figure 1.2(b) shows a deployment in which sensors are deployed across the length of a rectangular area such that an intruder penetrating from any of the parallel boundaries will be detected. Certain terminologies that are required to study barrier coverage in details is given below:

Definition 1. ***Crossing Line(Path)*** *[1]*
A line segment (or path) in a belt region is said to be a crossing line (or crossing path) if it crosses the complete width of the region. A crossing line is orthogonal if its length is equal to the belt's width w. For rectangular belts orthogonal crossing lines are parallel to the belts boundaries.

Figure 1.2(a) shows a crossing path.

Definition 2. ***k-Barrier Coverage*** *[1]*
A belt region with a sensor deployment is said to be k-barrier covered if and only if all crossing paths through the belt are at least k-covered by the sensors. In other words, the belt region is k-barrier covered if all crossing path intersects the sensing regions of at least k sensors.

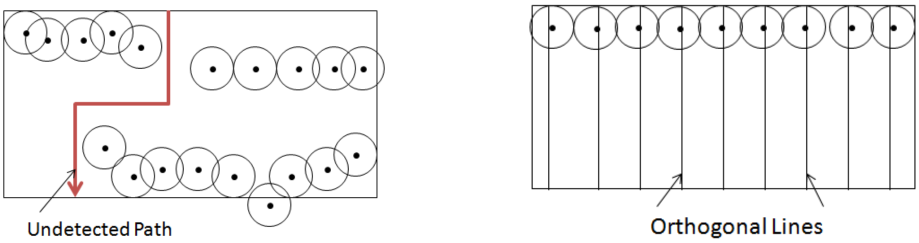
Figure 1.2(b) shows 2-barrier covered region where every crossing path is intersected by 2 sensors.

Based on the types of crossing paths that are detected by a deployment, barrier coverage can be classified as *strong* or *weak* barrier coverage [2].

Definition 3. ***Weak Barrier Coverage*** *[2]*
A rectangular belt is weak barrier covered if and only if all orthogonal lines passing through the belt are detected by the sensor deployment.

Definition 4. ***Strong Barrier Coverage*** *[2]*
A rectangular belt is strong barrier covered if and only if all crossing paths through the belt are detected by the sensor deployment.

Figure 1.3 (a) and Figure 1.3 (b) show examples of regions with weak and strong barrier coverage respectively.



(a) Weak barrier coverage (b) Strong barrier coverage

Figure 1.3: Different Types of Coverage

It can be concluded that if a region is strong k-barrier covered then, even if the location of sensors are known, intruders will not be able to find an uncovered path whereas in weak k-barrier coverage an uncovered crossing path can be detected if the sensor locations are known. Thus more number of sensors are required to provide strong barrier coverage compared to weak barrier coverage.

Definition 5. ***L-zone*** *[1]*
L-zone is a slice of the belt region of length L, whose two edges coincides with the parallel boundaries of the belt and the other two edges are orthogonal crossing lines separated by distance L.

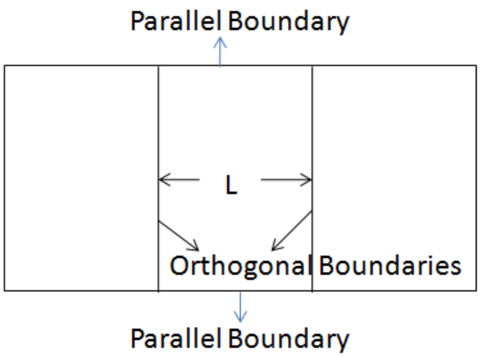


Figure 1.4: L-Zone.

Figure 1.4 shows a zone of length L of a rectangular belt. The orthogonal boundaries of the zone is parallel to the boundaries of the rectangular belt.

Definition 6. ***L-Local k-Barrier Coverage*** *[1]*
A belt region is said to be L-local k-barrier covered if every L-zone in the region is k-barrier covered for a positive number L. L-local barrier coverage guarantees that all crossing paths whose trajectory is confined within a length L of the belt region is being detected.

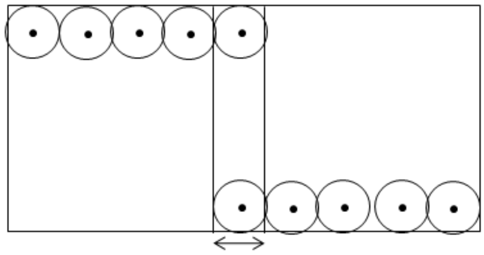


Figure 1.5: L-Local k-Barrier Coverage.

Figure 1.5 shows that every zone of length $L = 2r$ where r is the sensing range of individual sensor is 1-barrier covered.

Quality Q_k of k-coverage of a deployment is a metric to measure the amount of coverage being provided. It is defined in terms of L as the maximum L such that each zone of length L is k-barrier covered.

Definition 7. ***[Quality of k-barrier coverage,*** Q_k*] [1]: The quality of k-barrier coverage for sensor deployment is given by maximum L such that region is L-local k- barrier covered.*

Deployment in Figure 1.5 having Quality equal to L.

1.3 Motivation

Till now most work done in the area of quality of barrier coverage focus on defining metrics to measure the quality of network [1] or to determine whether a given belt region is k-barrier covered or not [2]. Improving the quality has not been achieved much with failed sensors handled by replacing them with new sensors. Thus there is scope of work to see if quality can be improved by relocating existing sensors. For relocating sensors, it is to be decided which sensors should be relocated where, to improve the quality with least cost. In case of failed sensors there may be situations in which replacing the failed sensor with some nearby redundant sensor, without disturbing the position of other sensors in the network can increase the quality of barrier coverage. On the other hand, for some deployments changing the orientation of a few sensors in the deployment may be easier and less costly than finding and moving a redundant. This gives scope to develop strategies to improve quality of barrier coverage which minimizing the number of sensors moved and also minimizing the average movement of individual sensors.

Intuitively a deployment having better quality implies that less effort in terms of movement will be required to improve the quality. Also more the desired quality, more effort is required to achieve the desired quality. This allows us to study relation between quality to be achieved and the effort required to achieve that quality.

1.4 Contribution of the Thesis

It has been experimentally observed that the effort required to improve quality of coverage increases as the quality to be achieved increases. We consider two parameters to measure the effort, average movement of sensor and average number of sensors moved. We have developed two algorithms to relocate sensors to achieve some desired quality of barrier coverage. Given a initial deployment the algorithm show which sensor is to be moved where to achieve the quality or indicate if it can not be achieved. Out of the two algorithms, the first one assume only horizontal movement of sensors and the second one assumes a sensor can move both horizontally and vertically.

1.5 Organization of the Thesis

The remaining chapters of the thesis are organised as follows:

- Chapter 2 discusses the existing work that have been done in the area of barrier coverage.
- Chapter 3 Relation of quality of the barrier coverage with critical region and sensing boundary region.
- Chapter 4 presents the algorithm to improve quality of barrier coverage by moving the sensors. Detailed simulation results are also shown.
- Chapter 5 gives an overall summary of the work done and discuss some scope of future work.

Chapter 2

Related Work

The concept of barrier coverage was broadly explained for the first time in [2]. The paper defined the k-barrier coverage problem as the problem of determining the number of k-disjoint paths in a region. It proposed two probabilistic barrier coverage concepts namely $-$*weak* and *strong* barrier coverage. Since then, there have been a significant amount of research done on the problem. The problem on barrier coverage can be classified based on the deployment models, the type of coverage provided, the objective of the coverage problem, types of sensors deployed and some other sensing characteristics. The work in each of the area has been explained in details below.

2.1 Deployment Models

The barrier coverage problem has been studied on various deployment models such as random and uniform deployment. In [2], the network model is described as a long, narrow rectangular belt over which sensors either have a random Poisson distribution with a rate n or are deployed uniformly. It has been proved [8] that if the value of n grows, for very large values, Poisson distribution of sensors with rate n is equivalent to random uniform distribution of n sensors where each sensor is equally likely to be distributed independent of other sensor locations. For deterministic deployment of sensors, the optimal configuration to achieve k-barrier coverage is to deploy k rows of sensors on the shortest path across the length of the belt region such that consecutive sensors sensing disks intersect each other. In real world, as deterministic deployment is not feasible in extreme locations, barrier coverage has been studied on random models in [9], [4]. Work in [9] includes the random deployment model where barrier coverage depends on the spatial distribution of sensor locations. It is considered that the random offset of each sensor from its target landing point follows a normal distribution which is addressed as *Line-based Normal Random Offset Distribution*

(LNRO). Simulation results depicted that LNRO model outperforms the Poisson Point process model thus concluding that sensor deployment strategies have direct impact on quality of the barrier coverage of wireless sensor networks. Another model of barrier coverage deployment includes deployment along a line randomly with random offsets which is a more realistic sensor placement model than the Poisson point process model [3]. The network model used in [4] is the 2-dimensional model where nodes are randomly distributed according to a Poisson point process of density λ.

2.2 Types of Coverage

The problem worked on in [9] can be portrayed as given the size of a service region, the number of sensors to be deployed, and a sensor distribution, what is the probability that the region is weakly k-barrier covered. This issue is addressed by finding a lower bound on the probability of weak k-barrier coverage with and without considering the border effect. Secondly, an algorithm is designed to check whether the region is weak k-barrier covered and if not what percentage is not covered. In contrast with [2] which gives the sufficient condition for barrier coverage, [9] proposes the necessary and sufficient condition for weak k-barrier coverage considering both the Toroidal (doesnot suffer from border effect) and the Euclidean model. In [2] the critical conditions do not depict the probabilistic measure of the weak barrier coverage. So this work studies the analytical aspects by giving the lower bound. An algorithm proposed to determine whether the area is weak barrier covered is based on the fact that weak coverage only depends on the coverage of sensors in the horizontal direction. Thus they showed that, the weak k-barrier coverage problem on a two-dimensional plane can be transformed to a one-dimensional k-coverage problem.

As weak barrier coverage guarantees only to detect intruders moving along congruent paths, to make the coverage scheme more robust, strong barrier coverage has been focussed on, that detects any crossing path across the region. The barrier construction methods for strong barrier coverage (whatever crossing paths the intruders chooses will be detected) is studied firstly in [4]. It has been shown in [4] that the critical conditions for strong barrier coverage is dependent on the width to length ratio of the belt region. If the width is greater than the logarithmic of the length of the region, then only there exists multiple disjoint barriers to provide strong barrier coverage. Below the critical width to length ratio, there is no strong barrier coverage. This multiple disjoint barriers provide redundancy making it tolerant to different detrimental conditions. To make efficient use of the redundant sensors, scheduling is required so that at any given moment there are only sufficient active sensors to cover the barrier. Reduction of cost, communication and delay in determining disjoint barriers is achieved by using the concept that the region is covered by strips. Each strip is divided into horizontal segments and vertical strips which independently calculates the barriers. The segment computes horizontal barriers and the vertical strip computes both horizontal and vertical barriers. The horizontal barriers of the segments combines with the vertical barriers of the vertical strips to form the continuous barrier. A distributed algorithm has been proposed to compute the disjoint barriers.

2.3 Lifetime Maximisation Problems

Reducing energy consumption being a critical issue in increasing the network lifetime, several scheduling algorithms have been developed which takes advantage of redundancy and activate only a set of sensors. A localised algorithm i.e *Localised Barrier Coverage Protocol (LBCP)* has been proposed in [10]. *Random Independent Sleeping (RIS)* algorithm is another sleep wake up schedule given in [2]. Performance analysis showed that $LBCP$ provides better network lifetime, while providing global barrier coverage most of the time, outperforming RIS. The work in [11] also proposes one of the decentralised algorithms to check for k-barrier coverage taking into account the energy consumption of the sensors. Reduction in energy consumption is achieved by the determination of maximum number of disjoint set of sensors so that each set provides k-barrier coverage with the minimum number of sensors. The set of sensors can be activated in turn to reduce the energy consumption and achieve load balance. Three mechanisms, called *Basic, Backtracking, and Branch* are proposed for constructing more number of disjoint sets of sensors that satisfy the requirement of k-barrier coverage. In this approach the network model is divided into grids such that each grid consists of k sensors whose sensing region covers the entire grid in which it is located.

The work in [12] considers cases when lifetime of individual nodes are not equal. Optimal coverage algorithms are provided that work both for non-disk sensing regions and heterogeneous lifetime of sensors. For the homogeneous lifetime (when all sensors have equal lifetime) the optimal sleep wakeup schedule *Stint* has been proposed that minimizes the number of sensor switches (turning on/off). A second scheduling algorithm *Prahari* have been proposed for individual sensors having heterogeneous lifetime. Another variant of the lifetime maximisation is when sensors have adjustable sensing range [13]. Multiple sensing power level model is used here. The problem formulated is, to determine what sensing power level assignment and sleep-wake-up schedule can be used to make the network last beyond the lifetime of an individual sensor node so that the network lifetime is maximized. Each node is divided into some virtual sub-nodes according to the available sensing ranges.

2.4 Quality of Barrier Coverage

The coverage algorithms discussed in [5] checks whether the network provides the desired quality and gives a 0/1 reply. Most of the work does not give a knowledge of how much coverage the network provides. Thus in [1], the quality of barrier coverage has been studied based on the concept of L-local barrier coverage. The authors concluded after the concept of barrier coverage came in [2], that it is not feasible to determine locally by individual sensors whether a given region is k-barrier covered. This lead to the concept of L-local barrier coverage in [10]. Based on the concept that all crossing paths are mostly along shorter distances, local barrier coverage guarantees the detection of all movements whose trajectory is confined to a slice of the belt region. It poses that when L is equal to the length of the entire deployment region, L-local barrier coverage is equivalent to global barrier coverage. Work in [1] defines a metric to measure quality. The quality of k-barrier coverage in a deployment is the maximum L such that all zones of length L in the deployment is k-barrier covered. If the value of Q_k is less than the desired value then [1] described a distributed algorithm to identify the weak zones and proposed methods to repair the deployment.

2.5 Barrier Coverage With Mobile Sensors

All the works discussed so far are restricted to stationary sensors. When sensors are randomly deployed, much more sensors are required to achieve barrier coverage than deterministic deployment using stationary sensors. To address this issue, mobile sensors are being used which can be relocated after the initial deployment. The problem of computing the most suitable position of relocated sensors while minimizing the moving energy consumption, called as *minimum-energy barrier-coverage (MEBC)* problem was formulated in [15]. Both a centralised *CBarrier* and a distributed *DBarrier* algorithm have been proposed to solve the MEBC problem. The distributed algorithm mentioned uses the virtual force model concept where the sensors adjust their positions according to the total repulsive and attractive forces. Both of these coverage algorithms provide 1-barrier coverage only.

The coverage problem for the mobile sensors can also be constrained on the number of sensors m present actually which is less than the number n needed to be deployed for full coverage [16]. The sensor scarcity case is handled by the dynamic sensor patrolling scheme. The problem is formulated as a dynamic programming problem where the movement strategy of all sensors should be decided in each time slot dynamically. This aims at maximizing the intruder detection probability, based on the current locations of sensors and intruder arrival information collected in the past time slots. The first algorithm *Periodic Monitoring Scheduling (PMS)*, considers monitoring m out of n points in each time interval where the points are the optimal sensor locations pre-calculated according to the existing work on deterministic deployment [2]. PMS proves the best strategy is to let the sensors be stationary at n points if movement is not known, thus concluding the importance of intruder arrival information for the strategy to work. The second algorithm *Coordinated Sensor Patrolling (CSP)* decides each sensors current movement strategy based on correction of intruder arrival time to improve average intruder detection probability.

The work in [17] focuses on the problem of moving mobile sensors to the perimeter of a simple polygon in order to detect intruders from either entering its interior or exiting from it. This addresses the issue, given that the mobile sensors have detected the existence of a crossing path, how to reposition the sensors most efficiently within a specified region so as to repair the existing security hole and thereby prevent intruders. It defines the min-max (minimizing the maximum movement of a sensor) and min-sum (minimizing the sum of movement of all sensor) problem and gives polynomial time approximation schemes for the min-sum problem. [6] contributes bounds for barrier coverage under sensor mobility. When a total number of m mobile sensors are deployed in a rectangular area of dimension $l \times w$, if all sensors have a sensing range of r, it is shown that a maximum number of $\frac{2mr}{l}$ sets can be formed along with strategy which gives the minimum of the maximum (minimax) moving distance among all sensors. In [16], barrier coverage is achieved when the number of sensors present n, are less than the number needed to guarantee full barrier coverage m. Thus the movement of the sensors affected by the intruder arrival probability, is modelled dynamically.

2.6 Objective of the Barrier Coverage Problem

In [14], the relocation problem to construct k-barrier coverage with minimum energy consumption is concentrated. An approximation algorithm *Approximate to Horizontal Grid Barrier (AHGB)* has been proposed to obtain 1-barrier coverage with the minimum movement of the sensors. Based on it, the divide and conquer algorithm is proposed to obtain k-barrier coverage. The network model used is partitioned into grids and the *AHGB* algorithms works in two steps on the grid. It first finds out the row of the grid which can form the horizontal barrier, based on the location of the nearby sensors. Secondly it determines the optimal movement strategy of the sensors so that minimum distance is traversed. The divide and conquer algorithm divides the grid into subregions and finds horizontal and vertical barrier cover for each subregion to provide strong k-barrier coverage. By dividing the large strip into small subregions, the message delay, communication overhead, and computation cost can be reduced to a great extent. The position information of a sensor node only needs to be transmitted within the small subregion where the node is located, resulting in smaller delay and communication overhead compared with the whole network.

2.7 Other Sensing Characteristics

In omni-directional sensing model, as the sensing region is considered to be a disc region, energy wastage occurs. To minimise energy consumption, research work has been carried out on directional sensing model where sensors are orientated in such a way that the sensing area covers just a limited sector of the boundary length. The work in [18] gives a polynomial time algorithm to determine the sensors and their orientations, needed to find strong barrier coverage for 1-dimensional plane with the minimum number of sensors. It introduces the concept of virtual node to study strong barrier coverage of a 2-dimensional region. The main aim of this virtual node is to capture the geographical relationships among directional sensors and deployment region boundaries and records all the intersecting sensing regions among neighbouring sensors. Researches have also focussed on a variant of strong barrier coverage with directional sensors such as radar, audio or video sensors. In order to provide barrier coverage with directional sensors, the two main aspects are to 1) select a subset of directional sensors and 2) determine the orientations of the selected sensors. The directional sensing models can be classified into *overlapping* and *non-overlapping* model. In non-overlapping model, a directional sensor has mutually disjoint sensing sectors. In [19], directional coverage graph to model barrier coverage with directional sensors was proposed. On the basis of this, an Integer Linear Programming formulation of the barrier coverage model denoted by *Maximum Directional Sensor Barrier Problem* (MDSBP) was proposed whose objective was to find the maximum number of disjoint directional sensor barriers.

The paper [25] includes camera sensor networks. The difference between camera and scalar sensors are that cameras from different positions can form quite different views of the object. So combining the sensing range of the cameras across the field does not necessarily form an effective camera barrier since the important aspect of the object may not be captured. The angle between the object's direction and the camera's viewing direction measure the quality of sensing.

Chapter 3

Relationship between Quality of the Barrier Coverage with Different Parameters

Sensors deployed in a region may fail due to various reasons. Though sensor failure can be tolerated with extra sensors in the deployment that do not form a part of the barrier, after certain limit, with increase in number of failures, the quality of barrier coverage decreases. Decrease in quality is because of formation of critical regions with length less than the desired quality or formation of non k-covered sensing boundary regions. Intuitively, a critical region is the slice of region to be covered such that if we extend the size of that slice on both sides simultaneously then it allows an intruder to cross the boundary without being detected by k-sensors, but if we extend the region on either side one at a time then the region will still be barrier covered. Intuitively, it seems that with better quality less effort is required for repairing the network providing coverage. In this chapter we will study the effect of different parameters on the quality of barrier coverage.

3.1 Critical Region and Sensing Boundary Region

The quality of barrier coverage is mainly decided by whether the deployment contain non k-covered sensing boundary region or a critical region with length less than the desired quality Q^*. Therefore we will first discuss what exactly these two types of regions are and then how do these relate to quality of barrier coverage.

Definition 8. ***Middle line**[1] : The middle line of a belt is the curve that is parallel to, at the middle between, the belts two parallel boundaries. Its two ends are referred to as the left and the right endpoints.*

Definition 9. ***Coordinate of an orthogonal line,*** *V_l [1] : For an orthogonal line l, V_l is the length of the middle line from the middle lines left endpoint to its intersection with l.*

Definition 10. ***Leftmost/rightmost orthogonal line of a sensing region*** *[1]:*

A sensing region's leftmost orthogonal line is the orthogonal line with smallest coordinate (V value) that has intersection with the sensing region. A sensing re- gion's rightmost orthogonal line is similarly defined. The leftmost and rightmost orthogonal lines of a sensor node a's sensing region are denoted by $ll(a)$ and $rl(a)$, respectively.

Definition 11. ***Segment:*** *A Segment consists of a list of sensors such that sensing region of every single sensor intersects with at least one other sensor's sensing region in the list.*

Definition 12. ***Length of segment****: It is the maximum distance from the leftmost orthogonal line of the sensing region of a sensor in the segment, which is not intersecting with any other sensor on its left, to the rightmost orthogonal line of the sensing region of a sensor which is not intersecting with any other sensor on its right in the segment.*

Definition 13. ***Zone from node a to node b,*** *$Zn(a, b)$[1]: If two nodes a and b are such that $V_{ll}(a) \geq V_{rl}(b)$, then we denote by $Zn(a, b)$ the zone from $ll(a)$ to $rl(b)$ and simply refer to it as the zone from a to b. If $V_{ll}(a) > V_{rl}(b)$, then $Zn(a, b)$ does not exist, in which case we write $Zn(a, b) = \phi$.*

Definition 14. ***Critical k-barrier covered zone*** *[1]: For two sensor nodes a and b such that Zn(a,b) != ϕ, Zn(a,b) is said to be critical k-barrier covered zone if Zn(a,b) is k-barrier covered and if we extend it on one side then it will still remain k-barrier covered, but if extend it on both sides simultaneously then it results in non k-barrier covered region.*

In Figure 3.2 region bounded by $ll(a)$ and $rl(b)$ is critical 1- barrier covered zone.

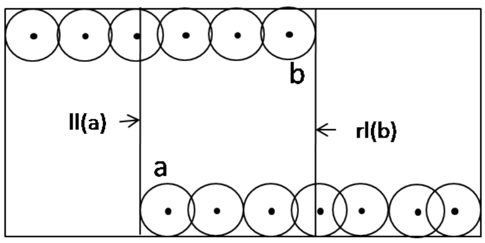


Figure 3.1

Figure 3.2: Critical 1-barrier covered zone

Definition 15. ***Sensing-boundary region****[1] : Let D be the set of all sensor's left and right orthogonal lines, together with the belts left and right orthogonal boundaries. Let $l_0, l_1, ..., l_n$ be all the lines in D as ordered from left to right, i.e., $V_{l_0} < V_{l_1} < \quad < V_{l_n}$. (l_0 and l_n are the belt's left and right orthogonal boundaries, respectively.) We define $Zn(l_{i-1}, l_i\ )$ as a sensing-boundary region (0 $< i < n$).*

Lemma 3.1.1. *For any non negative constant L, if length of every critical k-barrier covered zone (region) is greater than L and there doesn't exist any non k-covered sensing boundary region, then the quality of barrier coverage is greater than or equal to L.*

Proof. Suppose this proposition is False. This statement is false means there exist L, $L \in Z^+$ for which length of every critical k-barrier covered zone is greater than L, but the quality of barrier coverage is less than L. There can be three cases. 1) slice of the belt region which is considered for quality measurement will not contain any critical region 2) the slice of belt region contain part of critical region 3) the slice of belt region contain complete critical region

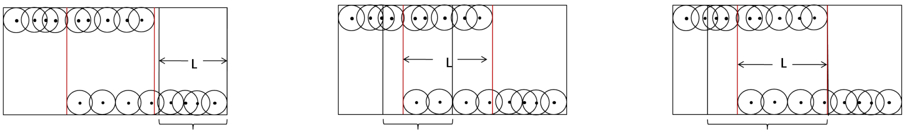
Case 1: If the slice of belt region does not contain any critical region and sensing boundary region, then the region is k-barrier covered. Now let us consider the region for quality measurement which doesn't contain any critical region and sensing boundary region, which is the same as a deployment which provides full coverage. Since it similar to deployment which provides full coverage, quality of the barrier coverage will always be greater than L provided length of belt region is greater than L. The scenario is shown in Figure 3.3 (a).

Case 2: If we consider a slice of region which contains part of some critical region then by definition of critical region, if we extend the critical region on one side, then it will still remain k-covered and as we know every critical region is having length greater than L, hence extended region is also having length greater than L. The scenario is shown in Figure 3.3 (b).

Case 3: If we consider a slice of region which contains some complete critical region then as we know the length of every critical region is greater than L, therefore the region that we are considering for quality measurement is also having length greater than L. The scenario is shown in Figure 3.3 (c).

As we can see in all three cases we are able to get quality of deployment of grater than or equal to L. Hence quality of coverage can't be less than L. □

Lemma 3.1.2. *To increase the number of critical regions, at least two sensors need to be fail. If only one sensor fails, then the number of critical regions either remain the same or it may decrease.*



(a) (b) (c)

Figure 3.3: Relation between quality of barrier coverage and critical region

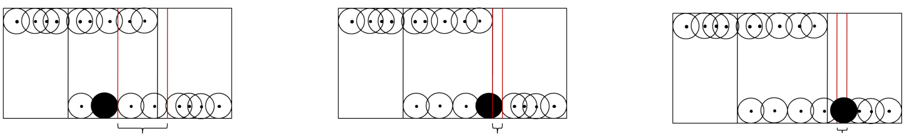
Proof. Let us consider the case of single sensor failure. There are mainly three possible cases where a sensor may fail. 1) Sensor completely contributing to a critical region 2) Sensor partially contributing to a critical region 3) Sensor not part of a critical region.

Case 1: If a sensor completely contributing to the critical region fails then it results in a division of the segment where the failed sensor was present into two segments out of which one will be a redundant segment and the other will form a critical region with smaller length than what it was previously as shown in Figure 3.4 (a), but doesn't form any new critical region

Case 2: If a sensor contributing partially in some critical region fails then also it results in formation of two segments out of which one will be redundant, but the second one is not intersecting with any other segment which was previously part of the critical region. This results in the formation of a gap between segment which is called a sensing boundary region. So failure of a sensor which is contributing partially in the critical region decreases the number of critical regions and increases the number of sensing boundary regions by 1. The scenario is shown in Figure 3.4 (b).

Case 3: If a sensor that is not a part of any critical region fails then it may or may not result into formation of sensing boundary region, but it will have no effect on number of critical regions in the deployment. The scenario is shown in Figure 3.4 (c). Formation of sensing boundary region after single sensor failure will depend on whether the failed sensor is redundant or not.

From the above three cases it can be shown that the number of critical region will either remain same or decrease with single sensor failure. □

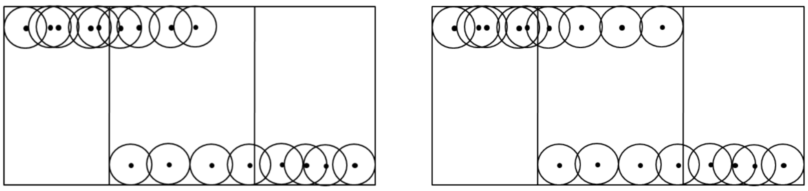


(a) (b) (c)

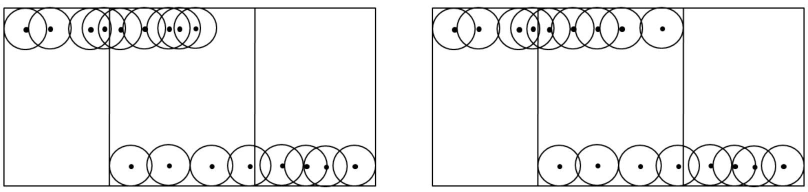
Figure 3.4: Relation between failure of a sensor and Critical regions

If there are two deployments i and j with initial quality Q_k^i and Q_k^j respectively, average number of sensors moved and average movement of sensor have no relation even if to achieve same desired quality Q^*. Let us discuss this with two cases. The first one is with same number of sensors but varies in the density of sensors towards critical region and second one is vice versa.

$Case1$: $S_i = S_j$ and $Q^i_k = Q^j_k$ but density of sensors varies. As we can see in Figure 3.5, the deployment in Figure 3.5 (a) have less sensor density towards critical region while the deployment in Figure 3.5 (b) have more sensor density towards critical region. Since the initial quality of both the deployments is same, the distance that sensors need to cover in both deployments will also same. As we are assuming number of sensors in both the deployments is same and the distance that sensors need to cover in both deployments will also be the same therefore the average movement of sensors will be same, but the number of sensors moved in each deployment will be different. It will be more for the deployment with less sensor density towards critical region and less for deployment with more sensor density. Hence we can conclude that in this case for this two deployments, the average movement of sensor remain same, but the number of sensor moved varies.



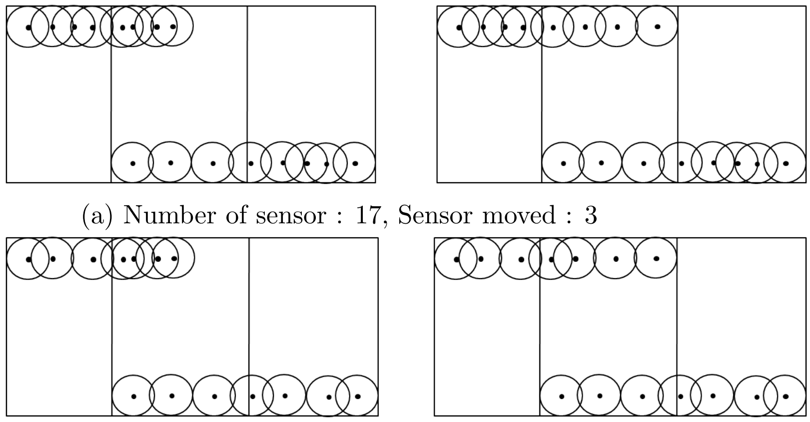
(a) Number of sensor : 17, Sensor moved : 3



(b) Number of sensor : 17, Sensor moved : 2

Figure 3.5: Relation between average movement of each sensor and number of sensors moved

$Case2$: $S_i \neq S_j$ and $Q^i_k = Q^j_k$ having same density towards critical region as shown in Figure 3.6. Since desired quality Q^* that needs to achieve is same and both the deployments are not having any sensing boundary region, additional distance that sensors need to move is same in both the deployments. As we have assumed that the number of sensors are different in both cases, the average movement of a sensor will be more for the deployment with less number of sensors and less for deployment with more number of sensors. Density of sensors towards the critical region is same, therefore number of sensor moved will be same. Hence we can conclude that average movement of sensor is different while number of sensors moved is same in this case.



(a) Number of sensor : 17, Sensor moved : 3

(b) Number of sensor : 14, Sensor moved : 3

Figure 3.6: Relation between average movement of each sensor and number of sensors moved with different number of sensors

From the above two cases it can be easily seen that for some deployments the average movement of each sensor varies and the number of sensors moved remain same and for some other deployments it will be vice-versa. In other words average movement of each sensor and number of sensors moved varies independently.

If the desired quality Q^* is greater than L, where L is maximum length of segment, then Merging is prefered over moving the sensors to improve the quality of barrier coverage to achieve the desired quality Q^*. This happens because for higher value of the desired quality, increase in the length of overlapping region to achieve desired quality is not possible.

Consider that L is the length of the longest segment in the deployment, also consider that the desired quality Q^* that we have to achieve is just greater than L. As discussed earlier weak regions are of two types, non k-covered sensing boundary region and critical region having length less that Q^*. For critical region having desired quality less that Q^* it is easy to say that merging is more beneficial than moving the sensor because maximum overlapping region between two segment can be L as we have already assumed L is maximum length of longest segment in the deployment therefore it is not possible to form critical region with length greater than L.

For non k-covered sensing boundary region we have two ways either we can move the segment to make them overlap but as we have seen earlier it is not possible to overlap the segments such that overlapping region between them will exceeds L. So moving is not possible

From the above explanation it can be seen that for higher quality, let say greater than L, merging the segment is prefered over moving the sensors to increase overlapping region between segments.

Chapter 4

Barrier Coverage Maintenance with Optimized Movement

Deployment of sensors in the belt region is random, and hence most of the time the quality of barrier coverage is very low (-1) as sensors form clusters and leave the large gaps between two clusters. As deployment is random, more number of sensors are usually deployed than actually required to provide the coverage to a given region. Also as we have more number of sensors there may be scope to improve the quality by moving the sensors closer to each other. While moving the sensors to improve the quality our aim is to optimize the number of sensors moved as well as average movement of each sensor. In this section we will discuss two movement strategies out of which one will consider only horizontal movement of sensors while the second one will consider both horizontal as well as vertical movement of sensors.

Improving the quality of deployment can be divided into following steps:

- Calculating the initial quality Q of the given deployment
- Identifying weak zones if initial quality Q is less than desired quality Q^*
- Apply some movement strategies to repair weak zones and improve the quality of coverage (ideally to the desired quality Q^*)

As explained in the previous chapter the quality of barrier coverage is limited by the smallest critical region if there does not exist any non k-covered sensing boundary region. So calculating initial quality means finding any non k-covered sensing boundary region, else we need to find the smallest critical region. The initial quality is compared with the desired quality and if it is less than the desired quality, then the weak zones are to found. Weak zones include non k-covered sensing boundary region and critical region having length less than Q^*. We will discuss some movement algorithms to repair these weak zones. Before discussing the movement algorithms some necessary concepts are discussed in the next section.

4.1 Basic Structure

Consider that the sensing regions of two sensors s_1 and s_2 overlap as shown in Figure 4.1. From the figure it is clear that we can move sensors till the length of the line joining the centres of the two sensors is less than or equal to the diameter of the sensing regions of a sensor. In the first movement algorithm we will consider movement along horizontal direction only as it will result in increasing the length of the segment more than whatever be achieved by movement in vertical direction.

(a) (b)

Figure 4.1: Movement of sensor

As shown in Figure 4.1 (a), a and A is the difference between X-coordinate of the two sensors before and after movement respectively. Value of A can be calculated using Equation 4.1 which is calculated by applying pythagoras theorem on the right angled triangle formed as shown in 4.1 (b). b and B is the difference between Y-coordinate of two sensor before and after movement respectively. values of b and B doesn't change as movement is along horizontal direction only.

$$A = \sqrt{d^2 - B^2} \tag{4.1}$$

The distance moved D by the sensor from its initial location can be calculated as follows.

$$D = A - a \tag{4.2}$$

If there is a sequence of n sensors in the segment, the maximum length that the segment can cover can be obtained just by applying summation on Equation 4.2. If $a_{i,j}$, $A_{i,j}$ is the distance between the sensors S_i and S_j

along horizontal direction before and after movement respectively, then the maximum length, *maxLength*, that the segment can be extended can be calculated by Equation 4.3

$$\text{maxLength}= \sum_{i=2}^{n} \text{A}_{i,i-1} + \text{d} \quad (4.3)$$

4.2 Movement Strategies

In this section centralized algorithms to improve the quality of barrier coverage are proposed. The algorithms work on the assumption that the number of sensors deployed are sufficient to provide full coverage. The position of each sensor in the rectangular belt region is known and a sensor s_i is assumed to be located at (x_i, y_i) with top left corner of the belt region as origin. Each sensor covers a sensing disc of radius r. A sensor s_i intersects another sensor s_j if euclidean distance between their centers is less than or equal to $2r$. For a sensor s_i the leftmost and the rightmost orthogonal lines are represented as $ll(s_i)$ and $rl(s_i)$ respectively.

4.2.1 Movement of Sensors only in Horizontal Direction

The algorithm will check overlapping of every segment with rightmost segment on its left. If two segments overlap then it calculates the length of the overlapping region between them. If length of overlapping region is less than the desired quality Q^*, then the overlapping region is increased by moving the sensors along horizontal direction. Movement of the sensor start from the end of the segment participating in the critical region. If the two segments do not overlap then also the algorithm moves the sensors along the horizontal direction such that segments will merge to form one segment or segments will overlap such that the length of the overlapping region between the segments will be Q^*. Note here that the merging of the segments is not intentional, i.e., there is a chance of merging the segments while moving the sensors along horizontal direction. Before actually moving the sensor, algorithm will check whether it is possible to extend the segment. To do so it calculates the difference between the maximum distance that the segment can cover and the actual distance that it is covering now. If our aim is to achieve the quality indicated by marked region as shown in Figure 4.3 then we have to move the sensor nodes in the segments such that it will not create any non 1-covered sensing boundary region. Same movement algorithm can be applied to the deployment shown in Figure 4.2 only difference is in the formulae to calculate maximum distance that the segment can cover.

Notations	Purpose Used For
$actDist_i$	actual distance covered by segment i
$actDist_j$	actual distance covered by segment j
$Xstart_i$	x coordinate of leftmost sensor in segment i
$Xstart_j$	x coordinate of leftmost sensor in segment j
$Xend_i$	x coordinate of rightmost sensor in segment i
$Xend_j$	x coordinate of rightmost sensor in segment j
Num_i	Number of sensors in segment i
Num_j	Number of sensors in segment j
$maxDist_i$	max distance covered by segment i
$maxDist_j$	max distance covered by segment j
L	length of overlapping region between segment i and j

Table 4.1: Variables used in the algorithm 1

The main advantage of this movement algorithm is that, it is simple to implement and hence less computational overhead. Also it has some disadvantages. Even if the deployment is having sufficient number of sensors to provide full barrier coverage, it is not possible to improve the quality of coverage.

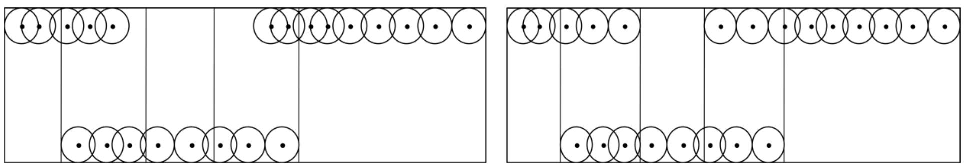


Figure 4.2: Quality improvement by considering only horizontal direction.

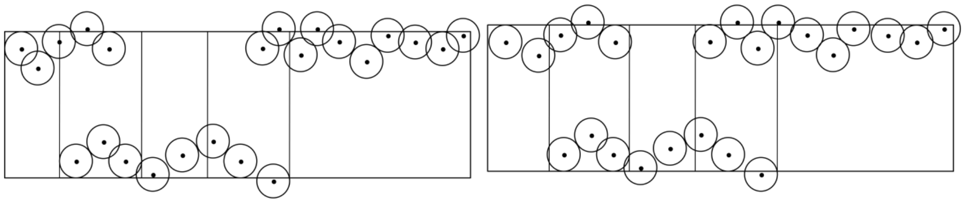


Figure 4.3: Quality improvement in random deployment by considering only horizontal direction.

Algorithm 1 Algorithm for improving the quality using movement in horizontal direction

Data: List of segment
Result: Deployment with desired quality Q^*
for *i := 1 to count* **do**
  for *j := 1 to i-1* **do**
    if segment i overlap with segment j **then**
      if *region is less than desired quality* Q^* **then**
        $actDist_i = Xstart_i - Xend_i$ + 2*r
        $actDist_j = Xstart_j - Xend_j$ + 2*r
        $maxDist_i = Num_i$ *2*r
        $maxDist_j = Num_j$ *2*r
        L = $Xend_i - Xstart_j$
        if $(maxDist_i - actDist_i + maxDist_j - actDist_j) > Q^* - L$ **then**
          $dist = Q^* - L$
          if $maxDist_i - actDist_i > dist$ **then**
            Move sensors in segment i starting from right end
            Update new location of sensor
          else
            if $maxDist_j - actDist_j > dist$ **then**
              Move sensors in segment j starting from left end
              Update new location of sensor
            else
              Move sensors in segment i and segment j
              Update new location of sensor
            end
          end
          break
        end
      end
    end
  end
  if segment i do not overlap with any of the segment **then**
    overlap or merge segment i with nearest segment on left
  end
end

Figure 4.4: Algorithm for improving the quality using movement in horizontal direction

The notations used in the movement algorithm are in Table 1.1. It is also important to note here that the algorithm is assuming left and right boundaries of the belt region as one of the segment which can not be extended. So if the first segment does not cover the boundary of belt region then the algorithm will move the sensors so that the sensing disc of one of the sensor, probably the first sensor of the segment, intersects with belt boundary. The pseudo code of the algorithm is given in Figure 4.4

4.2.2 Movement of Sensors by Considering Both the Directions

From the movement algorithm explained above it can be easily observe that even if there are sufficient number of sensors available, it may not be possible to improve the quality of barrier coverage of the belt region. This give us scope to move in both directions to repair weak regions. The basic idea of the algorithm can be divided into two steps

- Form the segment.
- Apply movement strategy.

The basic idea of segment formation algorithm is to find the connected component in the given random deployment and find the longest sequence of sensors along horizontal direction such that sensing disc of consecutive sensors intersect with each other. The segment can be formed by applying any algorithm (like DFS, BFS) which is used to find connected component in a graph. It is important to note that the left and right boundaries of belt region are treated as rigid segments, rigid in the sense that those can not be extended.

Once the segments are formed we will apply the movement algorithm on the formed segments. First step in the algorithm is to sort all the segment in increasing order of their starting points. The basic idea of the movement algorithm is to form the different states and at the end the state with minimum average movement of sensors is selected as a final state. Each state will contain number of segments and transition from one state to another will represent movement or merging of the rightmost segment in the state with one of the segment having starting point greater than this. Movement and merging of segments are intended to improve quality. The algorithm will backtrack if it reaches to the state where rightmost segment is right boundary of belt region or if it is not possible to move or merge the segment with any of the segment having staring point greater this. In case the algorithm reaches to the state where rightmost segment is right boundary of belt region, it calculate the average movement of sensor and report it as a local minimum value of average movement of each sensor and backtrack in search of the global minimum value of average movement of each sensor. Before moving or merging the segment, algorithm check for overlapping region between two segments. If the length of this region is more than the desired quality Q^*, then it just includes the segment and go on to the next state with increment in the value of average movement of each sensors by zero.

Now the question is how actually movement and merging will take place. Movement as well as merging is assumes to be feasible if it doesn't result in relocation of entire segment. The algorithm will try to move the sensor using horizontal direction first as explained earlier. If it is not possible to move the sensors in the segment to achieve desired quality without relocating entire segment, then it finds the number of sensors in the segment such that difference between maximum distance that can be covered by those sensors and actual distance covered by those sensors is greater than the extra distance that need to be covered. It take that many number of sensors in the segment, arranges them in a straight line and then moves the sensors along horizontal direction as explained earlier. Note that sensors selected from the end of the segment participate in the weak region.

Now while merging, the first step is to find the closest pair of sensors. Move one sensor such that both will belong to the same segment. Find the closest sensor from the other segment to the sensor which has just moved and repeat the same till two segment merge completely. The pseudo code of algorithm is given in Figure 4.7.

Let us try to understand this with the example given in Figure 4.8. If we apply the algorithm on the initial deployment, i.e. root of the tree, then algorithm traverse the tree in depth first manner. Whenever it reaches leaf node which includes right boundary of belt region, it records average movement of each sensor. The ultimate aim of the algorithm is to reach the state with average movement of each sensor equal to 3.1 as shown in Figure 4.8. At each stage there are two possibilities, first one is to merge the segment with the subsequent segment and second one is to move the sensors in the segment to increase the length of overlapping region with the subsequent segment. Therefore there will be two child nodes in the tree for each subsequent segments. So number of child node for each nodes can be given by twice the number of subsequent segments for a given rightmost segment in the state.

The main advantage of this algorithm is that it provide full coverage provided sufficient number of sensors are available.

Algorithm 2 Algorithm to improve quality by considering both direction

Data: List of segment

Result: Deployment with desired quality Q^*

Function `ImproveQuality(`*n, listIncludeParam, listRemainingParam, moveValue ,moveNum, minValue, minNum*`):`

```
for i := 1 to n do
    listInclude ← listIncludeParam;
    listRemaining ← listRemainingParam
    moveNumLocal =moveNum
    remove i-1 segments from listRemaining
    move P and Q
    moveValue=moveValue+moveSensors(listInclude,
    listRemaining, moveNum )
    if listRemaining is empty then
        minValue = min (moveValue, minValue)
        minNum = moveNum
    else
        recursively call the same function
        ImproveQuality(sizeof(listRemaining),listInclude,listRemaining,
        moveValue, moveNum, minValue, minNum)
    end
    listInclude ← listIncludeParam;
    listRemaining ← listRemainingParam
    remove i-1 segments from listRemaining
    moveNum = moveNumLocal merge P and Q
    moveValue=moveValue+mergeSegment(listInclude,listRemaining,
    moveNum )
    if listRemaining is empty then
        minValue = min (moveValue, minValue)
        minNum = moveNum
    else
        recursively call the same function
        ImproveQuality(sizeof(listRemaining),listInclude,
        listRemaining, moveValue, moveNum, minValue, minNum )
    end
end
```

Figure 4.5: Algorithm to improve quality by considering both directions

Algorithm 3 Algorithm to move sensors in the segment

```
double Function MoveSensor(listInclude, listRemaining, moveNum):
    L = Xend_P - Xstart_Q
    if L < Q* then
        Compute the actual distance that segments are covering
        actDist_P = Xstart_P − Xend_P + 2*r
        actDist_Q = Xstart_Q − Xend_Q + 2*r
        Calculate the maximum distance maxDist_P and maxDist_Q that the
        segments can cover
        if (maxDist_P − actDist_P + maxDist_Q − actDist_Q) > Q* − L then
            dist = Q* − L
            if maxDist_P - actDist_P > dist then
                Move sensors in segment P starting from right end
                Update new location of sensor
                calculate average movement of sensors moved
            else
                if maxDist_Q - actDist_Q > dist then
                    Move sensors in segment Q starting from left end
                    Update new location of sensor
                    calculate average movement of sensors moved
                else
                    Move sensors in segment P and segment Q
                    Update new location of sensor calculate average movement
                    of sensors moved and update moveNum..
                end
            end
        else
            if dist < 2 * r * (numP + numQ) − actDist_p − actDist_Q then
                arrange required sensors in segment P and Q in straight line
                move that many sensors along horizontally to improve quality
                calculate average movement of sensors moved
            else
                movement not possible, set moveNum and avgMove to infinity
            end
        end
    end
    return avgMove
```

Figure 4.6: Algorithm to move sensors in the segment

Algorithm 4 Algorithm to merge the segments

```
double Function MergeSegment(listInclude, listRemaining, moveNum):
    P := last segment in listInclude
    Q := first segment in listRemaining
    avgMove := average movement of sensor

    find the closest pair of sensors between segments P and Q
    move one sensor from among closest pair from P to Q
    for i := 1 to sizeof(P)-1 do
        find the closest sensor from P to the sensor just moved to Q
        if P and Q can be merge then
            calculate avgMove
            return avgMove
        end
    end
    if not Possible to merge P to Q then
        go back to initial configuration of P and Q
        move one sensor from among closest pair from Q to P
        for i := 1 to sizeof(Q)-1 do
            find the closest sensor from Q to the sensor just moved to P
            if P and Q can be merge then
                calculate avgMove
                return avgMove
            end
        end
    end
    if not Possible to merge P to Q then
        set avgMove to infinity
        return avgMove
    end
```

Figure 4.7: Algorithm to merge two segments

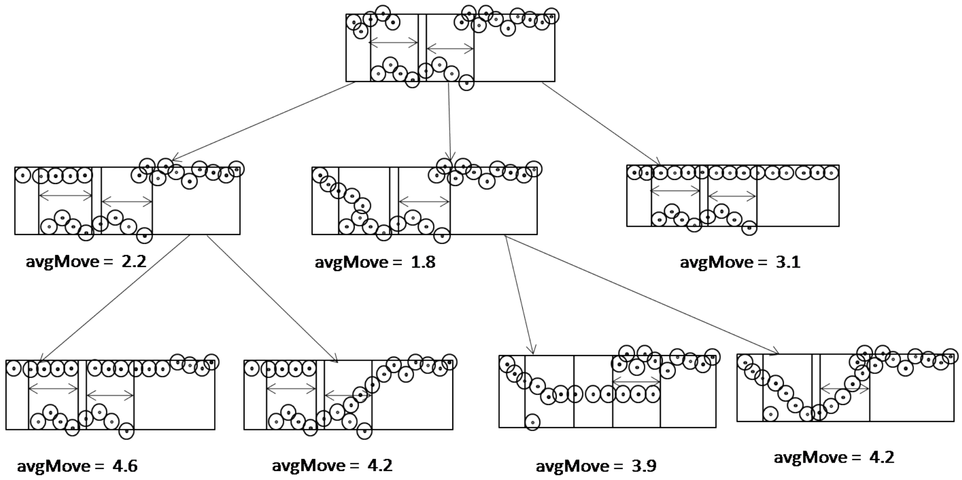


Figure 4.8: Example : Quality improvement by considering both directions

Notations	Purpose Used For
$actDist_i$	actual distance covered by segment i
$actDist_j$	actual distance covered by segment j
$Xstart_i$	x coordinate of leftmost sensor in segment i
$Xstart_j$	x coordinate of leftmost sensor in segment j
$Xend_i$	x coordinate of rightmost sensor in segment i
$Xend_j$	x coordinate of rightmost sensor in segment j
Num_i	Number of sensors in segment i
Num_j	Number of sensors in segment j
$maxDist_i$	max distance covered by segment i
$maxDist_j$	max distance covered by segment j
L	length of overlapping region between segment i and j
$listRemaining$	list of segment yet to process
$countInclude$	number of segments in listInclude
$countRemaining$	number of segments in listRemaining
P	last segment in listInclude
Q	first segment in listRemaining
$avgMove$	average movement of sensor
$moveNum$	average number of sensors moved

Table 4.2: Variables used in the algorithm 2, algorithm 3 and algorithm 4

The notations used in Algorithm 2, Algorithm 3,Algorithm 4 are listed in Table 1.2. The Algorithm 1 given in the Figure 4.5 will run in exponential time is not efficient. The main reason behind high complexity of algorithm is that it requires to take decisions whether to move the segment or to merge the segment. As the desired quality increases, merging the segments is prefered over moving the segments and in that case we don't need to bother about movement, hence the only option remaining is merging the segments. If we have only one option then we can apply dynamic programming approach to develop polynomial time algorithm. The basic idea of the algorithm is to apply the merge algorithm given in Figure 4.7 to the segment which will merge the segment with every segment having starting point less than this segment and select the segment causes minimum average movement of each sensor. Repeat the same procedure till we reach right boundary of belt region which will also act as rigid segment as explained earlier. For this purpose we are maintaining one array of size n, where n is number of segments in the deployment. Each entry of the array represents minimum value of average movement of each sensor required to reach that segment, hence last entry in the array represents the minimum value of average movement of sensor required to reach right boundary of belt region. The pseudo code of algorithm is in Figure 4.9.

Algorithm 5 Move sensors in the segment

Data: List of segment
Result: Deployment with desired quality Q^*
n:= number of segments in the list double moveAvgArr[n]
double moveNum[n]
for *i := 0 to n-1* **do**
    moveAvgArr[i] = INF
end
for *i := 1 to n-1* **do**
    for *j := 0 to i-1* **do**
        if *moveAvgArr[j] != INF* **then**
            merge segment seg_i and seg_j
            moveValue = mergeSegment(seg_i, seg_j, moveNum)
            if *moveAvgArr[j]+ moveValue < moveAvgArr[i]* **then**
                moveAvgArr[i] = moveAvgArr[j]+ moveValue
                moveNum[i] = moveNum[j] + moveNum
            end
        end
    end
end

Figure 4.9: Improve quality of barrier coverage in polynomial time

Notations	Purpose Used For
n	number of segment in the list given as input to algorithm
moveAvgArr[n]	array containing average movement of sensor
moveNum[n]	array containing average number of sensor moved
seg_i	i^{th} segment in the list
seg_i	j^{th} segment in the list
moveNum	variable containing average number of sensor moved

Table 4.3: Notations used in the algorithm 5

The notations used in Algorithm 5 are listed in Table 1.3. The simulation result of each policy with explanation is given in the next section. The result is analysed against different values of quality and average movement of each sensor node and average number of sensor moved is obtained.

4.3 Simulation Result

In this section the experimental results for the algorithms proposed in the previous section are shown.

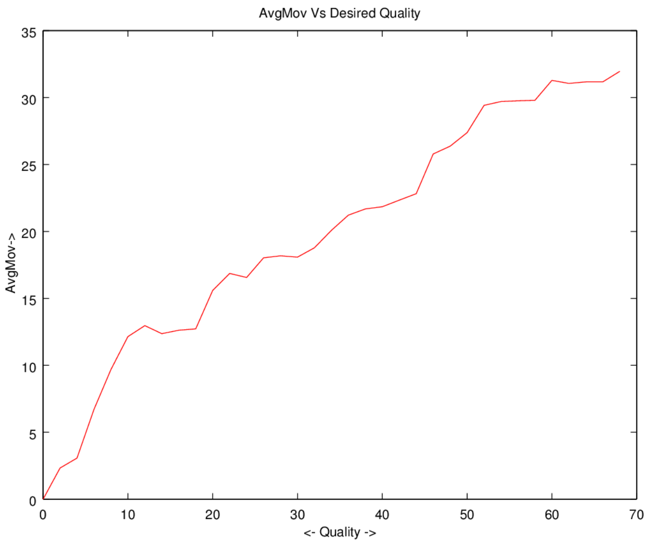


Figure 4.10: Movement Algorithm 1: Variation of average movement of each sensor with the desired quality

4.3.1 Quality improvement by considering only horizontal direction

The performance of the proposed algorithm to improve quality of barrier coverage by considering only horizontal direction, Algorithm 1 in Figure 4.4 is evaluated by simulation. All the simulations have been done for 1-barrier coverage. The input consists of N number of randomly deployed sensors whose value is 1000 in this case. The belt region is assumed to have length l = 500 units and width $w = 10$ units where r is sensing radius of each sensor. The value of r is kept fixed at 2 units. Thus the minimum number of sensor required to form barrier of sensors is $l/2r$. The output parameters which have been plotted include the average number of sensors moved and average movement of each sensor.

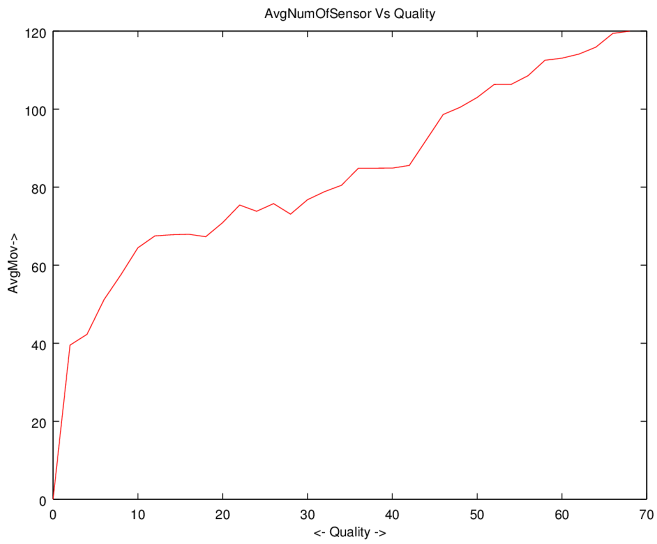


Figure 4.11: Movement Algorithm 1: Variation of average number of sensors moved with the desired quality

The algorithm is assumes that the sensors are deployed randomly and the number of sensors deployed is more than the minimum number of sensors required to provide barrier. If initially barrier is not providing full barrier coverage then there exists some clusters in deployment which we have to extend to increase length of coverage, ultimately resulting in improving the quality. These randomly deployed sensors are having some weak regions and steps in Algorithm 1 are followed to repair these weak regions. each of the data points plotted is average of 200 experiments. The results are shown bellow.

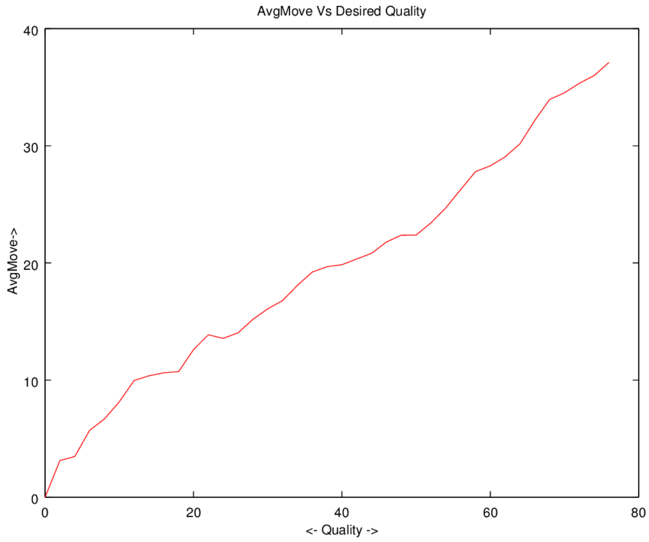


Figure 4.12: Variation of average movement of each sensor with the desired quality

Figure 4.10 shows the maximum quality that can be achieved and average movement of sensor to achieved the desired quality Q^*. In the graph a value of $x = 10$ indicates the desired quality that need to be achieved is 10 and value of $y = 2.4$ indicates the average movement of sensor against respective value of x is 2.4. As the value of desired quality increases the average movement of each sensor required to repair weak region also increases.

Figure 4.11 shows the maximum quality that the algorithm can achieve and average number of sensors moved to achieved desired quality Q^*. As the value of desired quality increases the average number of sensors moved required to repair weak region also increases.

If same movement algorithm is applied to the deployment having all the sensors aligned in the same straight line, the algorithm is able to achieve larger quality with slightly larger value for two parameters that we are comparing. The simulation results are shown in Figure 4.12 and Figure 4.13 . As we can see from the graph, though have run the experiment for 1000 sensors, which is sufficient to provide full barrier coverage to the belt region of length 500 units, we are able to achieve quality of 76 only.

4.3.2 Quality improvement by considering both directions

The performance of the proposed algorithm to improve quality of barrier coverage by considering both horizontal and vertical directions, Algorithm

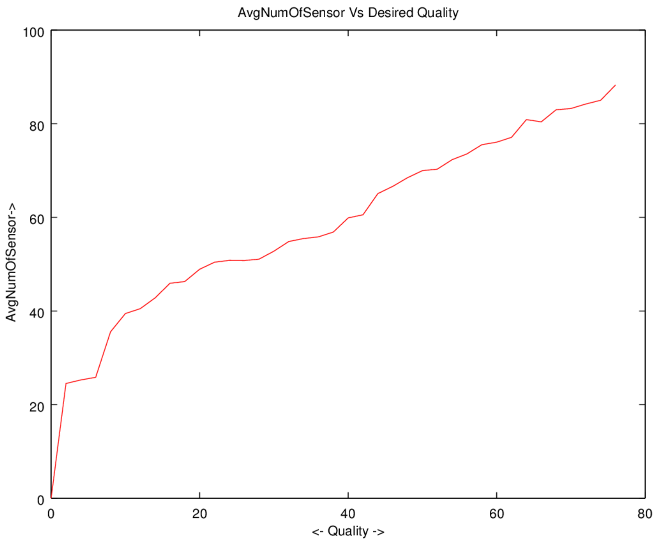


Figure 4.13: Variation of average movement of each sensor with the desired quality

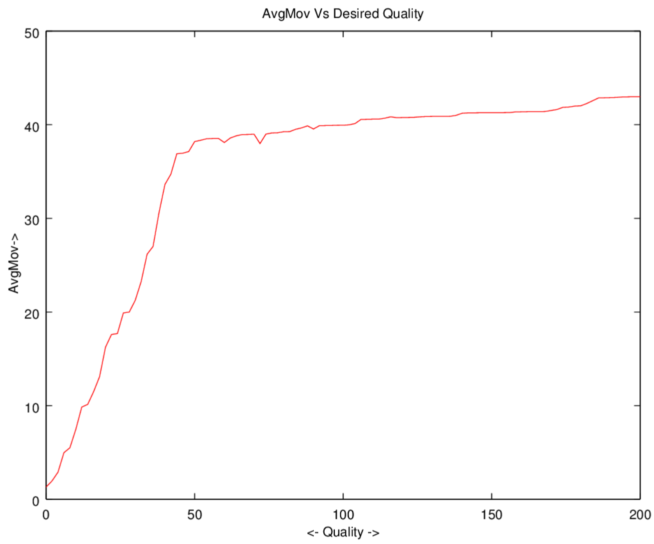


Figure 4.14: Movement Algorithm 2: Variation of average movement of each sensor with the desired quality

2 in Figure 4.5 is evaluated by simulation. All the simulations have been done for 1-barrier coverage. The input consists of N number of randomly deployed sensors whose value is 1000 in this case. The belt region is assumed to have length $l = 500$ units and width $w = 10$ units where r is sensing radius

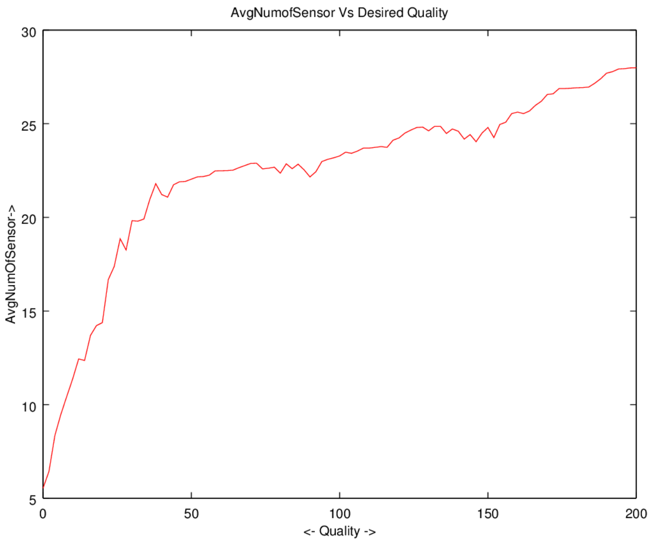


Figure 4.15: Movement Algorithm 2: Variation of average number of sensors moved with the desired quality

of each sensor. The value of r is kept fixed at 2 units. Thus the minimum number of sensor required to form barrier of sensors is $l/2r$ The output parameters which have been plotted include the average number of sensors moved and average movement of each sensor.

Figure 4.14 shows the maximum quality that can be achieved and average movement of sensors to achieved the desired quality Q^*. In the graph a value of $x = 10$ indicates the desired quality that need to be achieved is 10 and value of $y = 2.4$ indicates average movement of sensor against respective value of x is 2.4. As the value of desired quality increases the average movement of sensor required to repair weak region also increases very fast initially, but after some value of the desired quality increase in the average movement of each sensor slows down remaining almost the same as merging is preferred over moving for large value of the desired quality.

Figure 4.15 shows maximum quality that can be achieved and average number of sensors moved to achieved desired quality Q^*. As the value of the desired quality increases the value of average number of sensors moved required to repair weak region also increases very fast initially, but after some value of the desired quality, increase in the average number of sensors moved slows down remaining almost the same as merging is preferred over moving for lerge value of the desired quality.

Chapter 5

Conclusion and Future Work

The first part of this thesis focuses on the relation of quality of the barrier coverage with critical region. It also explain the effect of a sensor failure on the number of critical regions and hence on the quality of barrier coverage. Then the relation between average movement of each sensor and number of sensors moved to achieve the desired quality Q^* is explained. At the end situation where merging of segments prefered over moving the sensors in the segment to increase the quality of barrier coverage is discussed.

The second part of the thesis consists of design schemes to improve the quality of barrier coverage by moving the existing sensors. Algorithms are designed to repair the weak regions in a deployment to improve the quality of barrier coverage. The first algorithm gives the method to improve the quality of barrier coverage by moving the sensors horizontally. The second algorithm gives the method to improve the quality of barrier coverage by moving the sensors horizontally as well as vertically. Since the complexity of second algorithm is very high, a polynomial time algorithm to improve the quality of barrier coverage, with some restriction on the value of desired quality, is proposed. The performance of first and second algorithm is evaluated through simulations. The results show that with increase the desired quality, both average movement of each sensors and average number of sensors moved increases.

The extension of the work that is possible in future includes:

- A polynomial time algorithm design to improve the quality of barrier coverage which will work for all value of the desired quality.

- Design a distributed algorithm to improve the quality of barrier coverage.