CHAPTER 2

SENSOR SCHEDULING FOR COVERAGE PROBLEM

2.1 Introduction

Coverage is one of the fundamental issues that arises in sensor networks, in addition to localization, tracking, and deployment. Due to the large variety of sensors and their applications, coverage is subject to a wide range of interpretations. In general, coverage can be considered as the measure of quality of service of a sensor network (Meguerdichian et al., 2001).

Dense deployment of sensor nodes in a wireless sensor network complicates the organization and scheduling of sensor nodes. Random deployment of sensor nodes usually involves a higher number of sensor nodes. Figure 2.1 shows a region where nodes are randomly deployed. The locations, where sensor nodes are deployed, is marked in red. A sensor node with sensing range $s_r$ is said to cover a circular region of radius $s_r$ around it. When a target is covered, it may be covered by exactly one sensor node or multiple sensor nodes. If each object that has to be monitored in the sensor field is within distance $s_r$ from at least one sensor node, then the network is said to provide complete coverage. Figure 2.2 shows a sample deployment where 5 sensor nodes \{S_1, S_2, S_3, S_4, S_5\} monitor 3 targets \{T_1, T_2, T_3\}. All the targets are completely covered in the given deployment. The coverage requirement depends on the application. Some applications require complete coverage at all times, whereas the coverage requirement can slightly be compromised for some other applications (Jain and Liang, 2005). The time during which all the sensors can sense or communicate is limited because of irreplaceable batteries.

In deterministic deployment, coverage can be maximized as a result of optimal placement of sensor nodes. But when sensor nodes are randomly deployed, few objects
in the region may be densely covered and few may be sparsely covered. Connectivity is another important issue which is directly related to coverage problem (Costa and Guedes, 2010). Two sensor nodes are said to be connected if they both lie within the communication range of one another. Figure 2.3 shows a deployment where there are three sensor nodes \( \{S_1, S_2, S_3\} \) to monitor two targets \( \{T_1, T_2\} \). Each sensor node has
Figure 2.3: Connected Coverage

A sensing range $s_r$ and a communication range $c_r$. For direct base station connected simple coverage to be satisfied, it is sufficient if $S_1$ and $S_3$ are activated. But for connected coverage, in addition to $S_1$ and $S_3$, $S_2$ should also be activated since $S_2$ connects $S_1$ and $S_3$.

Many researchers have studied coverage problem and solutions have been proposed. Some of these address coverage problem for random deployment, and some for deterministic deployment. Network connectivity has also been looked at in some works. To ensure successful data transmission, network connectivity is essential. These algorithms also aim at maximizing the network lifetime by minimizing energy consumption (Chen and Koutsoukos, 2007). Connectivity helps to send the results back to the user and a higher level of coverage ensures accuracy of monitoring. A higher level of connectivity is required for some other applications when $M$ paths should exist within the nodes that are active, to guarantee the correctness of the information collected and also to ensure the reachability of the information at the base station. This leads to the requirement of $M$-connected coverage schemes.
Energy will be wasted if all the sensor nodes are active at a time. It is therefore necessary to identify subsets of sensor nodes which will satisfy the coverage requirement and schedule them accordingly. Though failure of one or more sensor nodes in densely populated sensor regions may not cause drastic decrease in coverage; such occurrence in sparse sensor regions will reduce coverage and affect connectivity (Zhang et al., 2008). The failure of one node should not affect the performance of network and hence high density of sensor nodes is preferred for random deployment. The choice of a good scheduling scheme is also critical for an efficient sensor network.

### 2.1.1 Types of Coverage Problems

Coverage problem can be classified based on the region/object to be monitored (Liu, 2007) as,

- **Area Coverage Problem**: The area coverage problem aims at monitoring/gathering information about an entire region.

- **Target Coverage Problem**: The target coverage problem concerns about monitoring a set of specific locations in the region.

Area coverage and target coverage problems can further be categorized based on connectivity requirement as,

- **Direct Base Station Connected Coverage**: Each sensor cover should satisfy the coverage requirement; and there may or may not exist a connected path among these nodes.

- **Connected Coverage**: Each sensor cover should satisfy coverage requirement and at least one path should exist among all the nodes that are in the sensor cover.

Connected Coverage can be classified based on the level of connectivity required as,

- **1-Connected Coverage**: The least complete connectivity requirement can be termed as 1-connected coverage. At least one path should exist among all the nodes in the sensor cover.

- **M-Connected Coverage**: Each node in a sensor cover should be connected to at least \( M \) other nodes in the same sensor cover.
The coverage requirement can be categorized as,

- **Simple Coverage (also known as 1-Coverage)**: Each target should be monitored by at least one sensor node.

- **k-Coverage**: Each target needs to be monitored by at least \( k \) sensor nodes, where \( k \) is a predefined integer constant, \( k \)-coverage problem arises. Simple coverage problem is a special case where \( k = 1 \).

- **Q-Coverage**: \( T = \{T_1, T_2, \ldots, T_n\} \) should be monitored by \( Q = \{q_1, q_2, \ldots, q_n\} \) sensor nodes such that target \( T_j \) is monitored by at least \( q_j \) number of sensor nodes, where \( n \) is the number of targets and \( 1 \leq j \leq n \).

The formal definitions are given in Section 2.2.

### 2.1.2 Importance

Sensor coverage is important while evaluating the effectiveness of a wireless sensor network. A lower coverage level (simple coverage) is enough for environmental or habitat monitoring (Yick et al., 2008) or applications like home security (Li and Gao, 2008). Higher degree of coverage (\( k \)-coverage) will be required for some applications like target tracking to track the targets accurately (Yick et al., 2008), or if sensors work in a hostile environment such as battle fields or chemically polluted areas (Li and Gao, 2008). More reliable results are produced for higher degree of coverage which requires multiple sensor nodes to monitor the region/targets.

In some cases, for the same application, the coverage requirement may vary. For example, for forest fire detections, the coverage level may be low in rainy seasons, but high in dry seasons (Li and Gao, 2008). An example of Q-coverage is a video surveillance system deployed for monitoring hostile territorial area where some sensitive targets like a nuclear plant may need more sensors cooperate to ensure source redundancy for precise data (Gu et al., 2007). Both sensing and communication has to be considered in some cases. In such applications, the nodes that are turned on should be connected to ensure proper data transmission. In case of simple connected coverage, where it is enough to have the nodes 1-connected, the network will be disconnected even if a single node fails. Hence it is important to have sufficient connectivity along with sufficient coverage. This paved the way for studying \( M \)-connected coverage problem where at
At least \( M \)-connectedness should exist within the nodes that are turned on, to guarantee the correctness of the information collected and also to ensure the reachability of the information at other nodes including the base station.

### 2.1.3 Network Lifetime and Energy Consumption

One of the most important factors to consider while developing a coverage scheme is energy. Battery of sensor nodes is mostly irreplaceable and cannot be recharged. This necessitates the need for a mechanism which conserves energy to prolong the network lifetime. A sensor network should have reasonably high number of sensor nodes for a random deployment, else the network may not satisfy coverage requirement or will not be able to survive for a longer time.

**Direct Base Station Connected Coverage**

In this model, we consider targets in an inaccessible region to be monitored. The sensor nodes are randomly deployed. The base station obtains the sensor node locations using some localization method. The base station then computes a schedule and communicates it to the sensor nodes. Network lifetime is defined as the time interval between the instant at which the network starts functioning and the instant at which the targets cannot be simple\(/
\frac{k}{Q}\) covered. The following two parameters are considered for direct base station connected coverage:

- Remaining energy of each sensor node
- Sensing range (prespecified and fixed)

**\( M \)-Connected Coverage**

Network lifetime is defined as the time interval between the instant at which the network starts functioning and the instant at which the targets cannot be simple\(/
\frac{k}{Q}\) covered or \( M \)-connected coverage is not satisfied. The following three parameters are considered for \( M \)-connected coverage:

- Remaining energy of each sensor node
• Sensing range (prespecified and fixed)
• Communication range (prespecified and fixed)

2.1.4 Sensor Scheduling: A Method for Network Lifetime Enhancement

Energy efficiency is a major concern since it directly influences network lifetime. In most cases, wireless sensor networks are expected to operate for a long time in spite of the fact that devices have limited battery. As the battery of the sensor nodes cannot be recharged or replaced easily, it is important to make use of the available energy effectively and efficiently. Activating all the sensor nodes at the same time leads to multiple nodes monitoring the same target, thereby reducing the network lifetime. Thus, energy conservation is a critical issue in sensor networks.

Generally random deployment includes a large population of sensor nodes and scheduling is a frequently used method to conserve energy. Only a minimum number of sensor nodes are activated to satisfy the coverage requirement and the remaining nodes are set to sleep for conserving energy. Hence these scheduling schemes prolong the lifetime of the sensor network. Apart from prolonging network lifetime, it also avoids frequent communication collisions and redundant messages in a sensor network with dense activated nodes (Chen and Koutsoukos, 2007). If more nodes are left to sleep, the WSN may be disconnected. This will affect data communication and transmission. Compared to direct base station connected coverage, some extra nodes might have to be active to keep the network connected for satisfying connected coverage.

The algorithms designed for sensor scheduling can either be centralized or distributed. Since distributed/localized algorithms run on more nodes throughout the network, it may be more complex (Mulligan and Ammari, 2010). Raman and Chebrolu (2008) makes it clear why a centralized design need not always be looked down upon. Lack of fault tolerance and lack of scalability are the key concerns for avoiding a centralized design. The former does not raise any major threat since most sensor network deployments have a sink which is a single point of failure, and the latter too is not a big concern since the sink nodes will generally have far greater CPU and memory capacity.
2.2 Problem Definition

2.2.1 Direct Base Station Connected Coverage

Let us assume \( m \) sensor nodes \( \{S_1, S_2, \ldots, S_m\} \) randomly deployed to cover the area \( R \) with \( n \) targets \( \{T_1, T_2, \ldots, T_n\} \). Here, each sensor node can directly communicate with the base station. The base station obtains the location of the sensor nodes through some localization scheme and constructs a sensor-target matrix. Each sensor node has an initial energy \( E_0 \) and a sensing radius, \( s_r \). A sensor node \( S_i, 1 \leq i \leq m \), is said to cover a target \( T_j, 1 \leq j \leq n \), if the distance \( d(S_i, T_j) \) between \( S_i \) and \( T_j \) is less than \( s_r \). The sensor-target coverage matrix is defined as,

\[
ST_{ij} = \begin{cases} 
1 & \text{if } S_i \text{ monitors } T_j \\
0 & \text{otherwise} \end{cases}
\]  

(2.1)

where \( i = 1, 2, \ldots, m \) and \( j = 1, 2, \ldots, n \)

This matrix assists the base station in deciding the schedule.

1-Coverage Scheduling

Definition 1: Given an energy constrained wireless sensor network with \( m \) randomly placed sensor nodes and \( n \) targets, schedule the sensor nodes such that all the targets are continuously monitored and the network lifetime is maximized. In other words, given a set of sensor nodes \( S = \{S_1, S_2, \ldots, S_m\} \) with battery power \( b = \{b_1, b_2, \ldots, b_m\} \), energy consumption rate \( e_i \) for \( S_i \) and a target set \( T = \{T_1, T_2, \ldots, T_n\} \), find a schedule \( \{C_1, \ldots, C_Y\} \) for time tick \( \{t_1, \ldots, t_Y\} \) such that for all ticks, each target is monitored by at least one of the sensor nodes and the network lifetime \( \sum_{P=1}^{Y} t_P \) is maximized.

\( k \)-Coverage Scheduling

Definition 2: Given a set of sensor nodes \( S = \{S_1, S_2, \ldots, S_m\} \) with battery power \( b = \{b_1, b_2, \ldots, b_m\} \), energy consumption rate \( e_i \) for \( S_i \) and a target set \( T = \{T_1, T_2, \ldots, T_n\} \),
generate a schedule \( \{C_1, \ldots, C_Y\} \), for \( \{t_1, \ldots, t_Y\} \), such that \( T = \{T_1, T_2, \ldots, T_n\} \) is covered by at least \( k \) sensor nodes, \( 1 \leq k \leq m \) and the network lifetime \( \sum_{P=1}^{Y} t_P \) is maximized.

**Q-Coverage Scheduling**

Definition 3: Given a set of sensor nodes \( S = \{S_1, S_2, \ldots, S_m\} \) with battery power \( b = \{b_1, b_2, \ldots, b_m\} \), energy consumption rate \( e_i \) for \( S_i \) and a target set \( T = \{T_1, T_2, \ldots, T_n\} \), generate a schedule \( \{C_1, \ldots, C_Y\} \), for \( \{t_1, \ldots, t_Y\} \), such that \( T = \{T_1, T_2, \ldots, T_n\} \) is covered by at least \( Q = \{q_1, q_2, \ldots, q_n\} \) sensor nodes, where each target \( T_j \), \( 1 \leq j \leq n \), is covered by at least \( q_j \) sensor nodes at any time and the network lifetime \( \sum_{P=1}^{Y} t_P \) is maximized. 1-coverage and \( k \)-coverage are special cases of \( Q \)-coverage where all \( q_j = 1 \) and \( q_j = k \) respectively.

**Upper bound**

The upper bound is the maximum achievable network lifetime for a particular configuration and as stated by Gu et al. (2007) and Chaudhary and Pujari (2009), the upper bound is calculated as,

\[
 u = \min_j \left[ \sum_{i} ST_{ij} * b_i \right] \right] \quad \text{(2.2)}
\]

where \( e_i \) is the energy consumption rate of \( S_i \). For \( k \)-coverage, \( q_j = k \), \( j = 1, 2, \ldots, n \).

**2.2.2 \( M \)-Connected Coverage**

Let us assume \( m \) sensor nodes \( \{S_1, S_2, \ldots, S_m\} \) randomly deployed to cover \( n \) targets \( \{T_1, T_2, \ldots, T_n\} \) in a region. Each sensor node has an initial energy \( E_0 \), sensing radius \( s_r \) and communication radius \( c_r \). A sensor node \( S_i \), \( 1 \leq i \leq m \), is said to cover a target
$T_j$, $1 \leq j \leq n$, if the distance $d(S_i, T_j)$ between $S_i$ and $T_j$ is less than $s_r$. Two sensor nodes are said to be connected if one sensor node lies within the communication range of the other. Here two matrices are computed: coverage matrix and connectivity matrix.

The coverage matrix is defined as,

$$ST_{ij} = \begin{cases} 
1 & \text{if } S_i \text{ monitors } T_j \\
0 & \text{otherwise}
\end{cases} \quad (2.3)$$

where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$

The connectivity matrix is defined as,

$$CM_{iz} = \begin{cases} 
1 & \text{if } S_i \text{ and } S_z \text{ are connected and } i \neq z \\
0 & \text{otherwise}
\end{cases} \quad (2.4)$$

where $i = 1, 2, \ldots, m$ and $z = 1, 2, \ldots, m$

**M-Connected 1-Coverage Scheduling**

Definition 1: Given $m$ sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$ with battery power $b = \{b_1, b_2, \ldots, b_m\}$, energy consumption rate $e_i$ for $S_i$ and $n$ targets $T = \{T_1, T_2, \ldots, T_n\}$, find a schedule $\{C_1, \ldots, C_Y\}$ for time tick $\{t_1, \ldots, t_Y\}$ such that for all ticks,

1. each target is covered by at least one of the sensor nodes
2. each sensor node in $C_P$ is connected to at least $M$ other nodes in $C_P$ where $1 \leq M \leq m$
3. network lifetime $\sum_{p=1}^{Y} t_p$ is maximized

**M-Connected k-Coverage Scheduling**

Definition 2: Given a set of sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$ with battery power $b = \{b_1, b_2, \ldots, b_m\}$, energy consumption rate $e_i$ for $S_i$ and a target set $T = \{T_1, T_2, \ldots, T_n\}$, generate a schedule $\{C_1, \ldots, C_Y\}$, for $\{t_1, \ldots, t_Y\}$, such that for all ticks,
1. each target is covered by at least $k$ sensor nodes, $1 \leq k \leq m$
2. each sensor node in $C_P$ is connected to at least $M$ other nodes in $C_P$ where $1 \leq M \leq m$
3. network lifetime $\sum_{p=1}^{Y} t_P$ is maximized

$M$-Connected Q-Coverage Scheduling

Definition 3: Given a set of sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$ with battery power $b = \{b_1, b_2, \ldots, b_m\}$, energy consumption rate $e_i$ for $S_i$ and a target set $T = \{T_1, T_2, \ldots, T_n\}$, generate a schedule $\{C_1, \ldots, C_Y\}$, for $\{t_1, \ldots, t_Y\}$, such that for all ticks,
1. $T = \{T_1, T_2, \ldots, T_n\}$ is covered by at least $Q = \{q_1, q_2, \ldots, q_n\}$ sensor nodes, where each target $T_j$, $1 \leq j \leq n$, is covered by at least $q_j$ sensor nodes
2. each sensor node in $C_P$ is connected to at least $M$ other nodes in $C_P$ where $1 \leq M \leq m$
3. network lifetime $\sum_{p=1}^{Y} t_P$ is maximized

2.3 Related Work

Slijepcevic and Potkonjak (2001) propose a heuristic that selects mutually exclusive sets of sensor nodes, where the members of each of those sets together completely cover the monitored area. The intervals of activity are the same for all sets, and only one of the sets is active at any time. Cardei and Du (2005) present Maximum Covers using Mixed Integer Programming (MC-MIP) for simple coverage problem, to extend the sensor network operational time which outperformed the heuristic proposed by Slijepcevic and Potkonjak (2001). The sensors are organized into a maximal number of disjoint set covers that are activated successively. Cardei et al. (2005) design Linear Programming-Maximum Set Covers (LP-MSC) and Greedy-Maximum Set Covers (Greedy-MSC) to solve simple coverage problem. Since Greedy-MSC has a lower running time compared to LP-MSC, it is more suitable for larger networks. A comparison on the network
lifetime achieved using Greedy-MSC and the theoretical upper bound shows that for higher number of sensors, Greedy-MSC could not achieve the upper bound.

Huang and Tseng (2003) present polynomial-time algorithms where the goal is to determine whether every point in the service area of the sensor network is covered by at least \( k \) sensors, where \( k \) is a predefined integer value. This paper has not given any experimental results explicitly and also not compared with any of the existing schemes. It only highlights the principle and methods used. The result is a generalization of some earlier results where only \( k = 1 \) is assumed. Based on Helly’s Theorem and the geometric properties of Reuleaux triangle, Ammari and Das (2010a) show how to achieve energy-efficient \( k \)-coverage of a region of interest. Ammari and Das (2010b) derive the minimum sensor spatial density to ensure \( k \)-coverage of a 3D space based on the geometric properties of Reuleaux tetrahedron. Yen et al. (2006) formulate a mathematical expression for expected \( k \)-coverage taking into account the border effects. In this paper, they try to maximize degree of coverage of a sensor network. They provide a formula for estimating the degree of coverage and verified through experiments. The authors claim that it depends only on the desired expected coverage ratio, and not on the number of sensors or the sensory range.

Zhang et al. (2007) propose a framework for At Most \( k \) Coverage Problem (AM \( k \)-Coverage). A centralized algorithm is proposed to divide dense wireless sensor network into coverage subsets based on GA (Genetic Algorithm), which is a quasi-parallel method. The degree of coverage is flexible in this framework. The result comparison with MC-MIP (Cardei and Du, 2005) shows that the heuristic proposed by Zhang et al. (2007) achieved more cover sets than MC-MIP (Cardei and Du, 2005). An algorithm was devised by Hefeeda and Bagheri (2007) for computing near-optimal hitting sets where an optimal hitting set corresponds to an optimal solution for \( k \)-coverage. The obtained network lifetime is less than the optimal network lifetime by at most a logarithmic factor. Li and Gao (2008) develop Perimeter Coverage Level Greedy Selection (PCL-GS) and Perimeter Coverage Level Greedy Selection Algorithm (PCL-GSA) to solve the \( k \)-coverage problem. GS deals with the case where sensors have fixed sensing range and sensors are divided into disjoint cover sets. GSA deals with the case where sensors can adjust their sensing range and sensors are divided into non-disjoint cover...
sets. Our approach resembles GS where the sensing range is fixed. Experimental results of GS show that the network lifetime can be 90% of the ideal network lifetime.

**Q-Coverage problem** is addressed by Gu et al. (2007) and Chaudhary and Pujari (2009). A method is proposed by Gu et al. (2007) based on column generation, where each column corresponds to a feasible solution. The column with the steepest ascent in lifetime has to be identified, and based on that a search for the maximum lifetime solution will be iteratively performed. An initial solution is generated through a random selection algorithm. This column based approach achieved better network lifetime compared to LP-MSC (Cardei et al., 2005). Chaudhary and Pujari (2009) present a greedy heuristic, High Energy and Small Lifetime (HESL), to generate Q-covers by prioritizing sensors in terms of the residual battery life. HESL yields network lifetime close to the optimal network lifetime for less cover activation time and smaller values of Q. Our aim is to maximize the network lifetime so as to achieve the optimal network lifetime in all cases.

Zhou et al. (2004) present a centralized approximation algorithm and a distributed version of the algorithm to solve connected k-coverage problem. The distributed priority algorithm is more efficient in applications where the query is executed for less than a few hundred times. For longer running queries, the distributed greedy algorithm is more efficient. Lu et al. (2005) generalize the sleep/active mode by adjusting sensing range to maximize the total number of rounds and presents a distributed heuristic. A more generic connectivity condition that can be used even when the transmission range is less than twice the sensing range is considered. It deals with the case of scheduling sensors’ activity by self-configuring sensing ranges, in the environment where both discrete target coverage and network connectivity are satisfied. Gupta et al. (2003) design and analyze algorithms for self-organization of a sensor network into an optimal logical topology in response to a query. A distributed version of the approximation algorithm that is run by the sensors in the network and results in a self-organization of the network into a topology involving a near-optimal number of sensors is also designed.

Zhao and Gurusamy (2008) consider the Connected Target Coverage (CTC) problem with the objective of maximizing the network lifetime by scheduling sensors into multiple sets, each of which can maintain both target coverage and connectivity among
all the active sensors and the sink. A faster heuristic algorithm based on the approximation algorithm called Communication Weighted Greedy Cover (CWGC) algorithm is designed and a distributed implementation of the heuristic algorithm is presented.

Ammari and Das (2008) compute the minimum sensor spatial density necessary for complete $k$-coverage of a sensor field. A tighter bound on network connectivity of $k$-covered WSNs, where the radius of the communication disks of sensors only needs to be at least equal to the radius of their sensing disks, is also derived.

Heinzelman et al. (2000) propose a method, Low-Energy Adaptive Clustering Hierarchy (LEACH) which is based on clustering and also points out that using a direct communication protocol or MTE routing will not be optimal. Direct communication will require a large amount of transmission power from each node if the base station is far away from the nodes. In the case of minimum-transmission-energy (MTE) routing, the nodes closest to the base station will be used to route a large number of data messages to the base station. Thus these nodes will die out quickly, causing the energy required to get the remaining data to the base station to increase and more nodes to die. With the use of clusters, LEACH is able to achieve large reduction in energy dissipation. Local computation in each cluster reduce the amount of data that must be transmitted to the base station.

Bai et al. (2008) investigate the problem of finding an optimal deployment pattern that achieves four connectivity and full coverage. A Diamond pattern, which can be viewed as a series of evolving patterns is proposed.

Most of the works related to target coverage focus on direct communication or MTE routing, where the nodes closest to the base station is largely used. Unlike all of these, we focus on identifying a minimum-sized cluster which is $M$-connected and meets simple/$k/Q$-coverage requirement. Once this cluster is decided, routing information to the base station is done as in LEACH.
2.4 Proposed Method

2.4.1 Direct Base Station Connected Coverage

We propose a weight-based multi-stage method for determining the cover sets. It includes the following main steps:
1. Weight Assignment
2. Cover Formation
3. Cover Optimization
4. Cover Activation and Energy Reduction

Algorithm 1 shows the proposed method. Each run $(r)$ is one iteration set to check the network lifetime. The first run finds the network lifetime using the priority of the sensor nodes. If the obtained network lifetime is less than the theoretical upper bound, it tunes the weight component to search for a better solution. This tuning is done till the network lifetime equals the upper bound or it exceeds the maximum number of times tuning can be done$(\max_{\text{run}})$.

Weight Assignment

Weight assignment is performed to decide the priority of sensor nodes. The more the weight of a sensor node, the higher the priority of the sensor node. Cover sets are decided based on this priority.

Weights are calculated for each sensor node by the base station by considering three factors:
1. Weight due to the remaining energy $(w_1)$
2. Weight due to covered targets $(w_2)$
3. Weight due to peers $(w_3)$

1. Weight due to the remaining energy

Each sensor node assigns a weight to itself which is equivalent to the remaining battery power of the sensor node. For each node $S_i$ in the optimized cover set, the
Algorithm 1 Proposed Method

1: Input: ST, B
2: Initialize k/Q, max_run
3: for r = 1 to max_run do
4: for iteration = 1 to \( \sum_{i=1}^{m} b_i \) do
5: if cover possibility exists then
6: Calculate weight due to remaining energy (according to Equation 2.5)
7: Determine cover based on priority (Algorithm 2)
8: Optimize cover (Algorithm 3)
9: Activate optimized cover and reduce battery power (Algorithm 4)
10: else
11: break
12: end if
13: end for
14: Calculate network lifetime (nlife)
15: if nlife \(< u \) then
16: Tune weight deciding component and/or parameter (according to Equation 2.9)
17: else
18: break
19: end if
20: end for

Weight assigned by itself decrements by the rate of energy consumption.

\[
 w_1 = b_i \quad (2.5)
\]

2. Weight due to covered targets

All sensor nodes are assigned weights based on the targets it cover and is calculated as given below:

\[
w_{st_{ij}} = \frac{ST_{ij}}{\sum_{i=1}^{m} ST_{ij}} \quad (2.6)
\]

The sum of weight due to covered targets of a sensor node \( i \) is given by,

\[
w_{t_i} = \sum_{j=1}^{n} w_{st_{ij}} \quad (2.7)
\]
\[ w_2 = wt_i \]  

(2.8)

3. Weight due to peers

Weight assigned to a sensor node by other neighboring nodes within the communication radius constitute weight due to peer sensor nodes \((w_3)\). This component is not considered in this section as we consider only direct base station connected model.

Total Weight

The total weight of a sensor node is a function of the weight due to the targets it cover and the weight due to the remaining energy.

We define \[ w = w_1^\beta + w_2^\gamma + w_3^\delta \]  

(2.9)

where \(-1 \leq \beta \leq 1\), \(-1 \leq \gamma \leq 1\) and \(-1 \leq \delta \leq 1\)

Nodes with different coverage degree may coexist in a network. Though the initial battery power of all the nodes in the network might be the same, it may vary in accordance with the cover activation.

The weights are recalculated for all the nodes at each time instant if,
1. Weight due to the remaining energy changes: It happens due to reduction in battery power for nodes which were in the previous cover.
2. Node turns off due to no battery power: The targets which were monitored by sensor nodes that turn off will reassign weights to all other sensor nodes monitoring it.

This weight recalculation might trigger a priority change and subsequently a new cover might be generated at the next time instant.

Cover Formation

A cover can be generated in different ways if the network has nodes which make all the targets \(k/Q\) covered. The proposed approach uses a priority based method. In the order of priority, if any new sensor node contributes to \(k/Q\) coverage requirement, it will be
added to the cover set. In general, a sensor node $S_i$ can be added to a cover set $Cov_S$ if and only if

1. for simple coverage problem: $Cov_S \cup \{S_i\}$ covers any new target
2. for $k$-coverage problem: $Cov_S \cup \{S_i\}$ contributes to $k$-coverage requirement
3. for $Q$-coverage problem: $Cov_S \cup \{S_i\}$ contributes to $Q$-coverage requirement

Algorithm 2 describes cover formation.

Algorithm 2 Cover Formation

1: Input: Sorted $S$ in descending order of $w$
2: Output: $Cov_S$
3: Initialize $Cov_S = \phi$
4: for $i = 1$ to $m$ do
5: \ \ \ \ if $S_i$ contributes to coverage then
6: \ \ \ \ \ \ \ \ $Cov_S = Cov_S \cup \{S_i\}$
7: \ \ \ \ end if
8: \ \ \ \ if coverage requirement met then
9: \ \ \ \ \ \ \ \ break;
10: \ \ \ \ end if
11: end for

Cover Optimization

Once the coverage requirement is met, the obtained cover set is optimized. By optimizing the generated cover, the proposed scheme attempts to minimize the energy usage.

It should be noted that this is the second phase of redundancy elimination, the first one being at the cover formation. A problem that arises with the cover formed at the cover formation phase is that it might still have nodes that need not be active to cover all the targets. This is possible because it is a step by step addition till all the targets are covered. A node can thus be dropped for not contributing to coverage at the time of cover formation or for not contributing to coverage after cover formation. The nodes in the cover set are subject to optimization using least priority first approach. This method of elimination prevents the higher priority nodes being discarded at the initial stages of optimization itself. The least priority node in the cover set cannot be eliminated from the cover set as it satisfies the $k/Q$ coverage requirement. Elimination starts from the last but one node as per increasing priority. A node $S_i \in Cov_S, 1 \leq i \leq length(Cov_S)$, represented as $S_i, Cov_S$ will not be added to the optimized cover set $Opt.Cov_S$ if
Cov\_S − \{S_i,Cov\_S\} meets k/Q coverage requirement. Cover optimization is discussed in Algorithm 3.

**Algorithm 3 Cover Optimization**

1: Input: Cov\_S  
3: Initialize Opt.Cov\_S = φ  
4: for \(i = \text{length}(Cov\_S)\) down to 1 do  
5: if Cov\_S − \{S_i,Cov\_S\} meets k/Q coverage requirement then  
6: Ignore S_i.Cov\_S  
7: Cov\_S = Cov\_S − \{S_i,Cov\_S\}  
8: else  
10: end if  
11: end for  

**Cover Activation and Energy Reduction**

The sensor nodes in the optimized cover are activated. The total energy that each node consumes should not fall beyond the minimum usable energy, \(E_{\text{min}}\). When the battery power reaches \(E_{\text{min}}\), the node becomes inactive and will not be able to monitor any more targets further. A node changes its state from active to inactive when the remaining battery power is lower than the minimum usable energy. As the battery power is drained when a node is active, the weight assigned by the node to itself reduces. The network terminates when no cover can further be formed. The detailed steps are shown in Algorithm 4.

As we assume that the number of sensors deployed in the area is greater than the optimum number required to monitor the targets, determining the sensor covers and switching from one cover to another in a scheduled manner such that only minimum number of sensor nodes remain active at any time instant is supposed to improve network lifetime.

To summarize, the proposed method is a multi-stage process where the first stage considers \(w_1\) is the only weight deciding factor. The cover sets are computed based on the descending order of weight. The network lifetime is computed and compared with the theoretical upper bound. If both are same, those are the best possible cover sets, else
Algorithm 4 Cover Activation and Energy Reduction

1: Input: Opt.Cov_S
2: for i = 1 to length(Opt.Cov_S) do
3:     S_i.state = true
4:     decrement b_i
5:     if b_i ≤ E_min then
6:         for j = 1 to n do
7:             M_{ij} = 0
8:         end for
9:     end if
10: end for

w_2 is considered for deciding the weight and the process is repeated. If the best network lifetime is still not achieved, a third phase includes tuning of parameters β and γ.

2.4.2 $M$-Connected Coverage

Cover Formation

Initially, a cover is computed without looking into connectivity. There may be different ways to generate sensor covers if the network has nodes which make all the targets $1/k/Q$ covered. We use a priority based method to compute the covers. The priority of sensor nodes is calculated based on the remaining battery power. The more the remaining battery power of a sensor node, the higher the priority of the sensor node. In the order of priority, if any new sensor node contributes to $1/k/Q$ coverage requirement, it will be added to the cover set. In general, a sensor node $S_i$ can be added to a cover set $Cov_S$ if and only if

1. for simple coverage problem: $Cov_S \cup \{S_i\}$ covers any new target
2. for k-coverage problem: $Cov_S \cup \{S_i\}$ contributes to k-coverage requirement
3. for Q-coverage problem: $Cov_S \cup \{S_i\}$ contributes to Q-coverage requirement

The detailed algorithm is discussed in Algorithm 5.

Cover Optimization

The cover may have nodes which need not be active for coverage condition to be satisfied. These nodes will be eliminated at this phase. The last node in $Cov_S$ will be
Algorithm 5 Cover Formation

1: Input: Sorted $S$ in descending order of battery power
2: Output: $Cov\_S$
3: Initialize $Cov\_S = \phi$
4: for $i = 1$ to $m$ do
5: if $S_i$ contributes to coverage then
6: $Cov\_S = Cov\_S \cup \{S_i\}$
7: end if
8: if coverage requirement met then
9: break;
10: end if
11: end for

The one which completes the $1/k/Q$ coverage requirement. Hence it will not be eliminated. Elimination check starts from the last but one node in $Cov\_S$. It continues for all the other nodes in $Cov\_S$ in the least priority first order. This curbs the possibility of higher priority nodes being eliminated at this stage. A node $S_i \in Cov\_S$, $1 \leq i \leq \text{length}(Cov\_S)$, represented as $S_i.Cov\_S$ will not be added to the optimized cover set $Opt.Cov\_S$ if $Cov\_S - \{S_i.Cov\_S\}$ meets $1/k/Q$ coverage requirement. The cover optimization algorithm is given in Algorithm 6.

Algorithm 6 Cover Optimization

1: Input: $Cov\_S$
3: Initialize $Opt.Cov\_S = \phi$
4: for $i = \text{length}(Cov\_S)$ down to 1 do
5: if $Cov\_S - \{S_i.Cov\_S\}$ meets $1/k/Q$ coverage requirement then
6: $Cov\_S = Cov\_S - \{S_i.Cov\_S\}$
7: else
9: end if
10: end for

$M$-Connected Cover Formation

The optimized cover is checked for $M$-connectivity. The connectivity matrix helps in finding out whether the nodes are $M$-connected or not. If the nodes are not $M$-connected, the nodes which got eliminated at the cover optimization phase and the nodes which did not form a part of $Cov\_S$ with battery power more than the minimum usable energy will also be considered. Since any node can play a vital role in making
an $M$-connected cover, we add these remaining nodes one by one to the optimized cover and check for all possible $M$-connected subsets at each new addition. If any $M$-connected subset meets coverage requirement, this subset will be the $M$-connected cover. Algorithm 7 shows the formation of $M$-connected cover.

**Algorithm 7 M-Connected Cover Formation**

1: Input: $S$ in descending order of battery power, $\text{Cov}_{S}, \text{Opt.Cov}_{S}$
2: Output: $M_{\text{Connected.Cover}}$
3: Initialize $M_{\text{Connected.Cover}} = \emptyset$; flag $= 0$
4: $\text{Total}_{S} = \text{Opt.Cov}_{S} \cup \{\text{Cov}_{S} - \text{Opt.Cov}_{S}\} \cup \{S - \text{Cov}_{S}\}$
5: $\text{Rem.Nodes} = \text{Total}_{S} - \text{Opt.Cov}_{S}$
6: $\text{ToCheck} = \text{Opt.Cov}$
7: if $\text{ToCheck}$ $M$-connected then
8: $M_{\text{Connected.Cover}} = \text{ToCheck}$
9: else
10: for $i = 1$ to $\text{length}($Rem_Nodes$)$ do
11: $\text{ToCheck} = \text{ToCheck} \cup \text{Rem.Nodes}(i)$
12: if $\text{ToCheck}$ $M$-connected then
13: $M_{\text{Connected.Cover}} = \text{ToCheck}$
14: break;
15: else
16: $\text{ToCheck1} = \text{ToCheck}$
17: $j = 1$
18: while $\text{ToCheck1} \neq \emptyset$ do
19: $M_{\text{Connected.Subset}} = \text{ToCheck1}(j) \cup \text{Nodes in ToCheck satisfying M-connectivity starting with ToCheck1}(j)$
20: if $M_{\text{Connected.Subset}}$ meets coverage requirement then
21: $M_{\text{Connected.Cover}} = M_{\text{Connected.Subset}}$
22: flag $= 1$
23: break;
24: end if
25: $\text{ToCheck1} = \text{ToCheck1} - M_{\text{Connected.Subset}}$
26: end while
27: end if
28: if flag $= 1$ then
29: break;
30: end if
31: end for
32: end if

**M-Connected Cover Optimization**

A node $S_i \in M_{\text{Connected.Cover}}, 1 \leq i \leq \text{length}(M_{\text{Connected.Cover}})$, represented as $S_i M_{\text{Connected.Cover}}$ will not be added to the optimized cover set
Opt.$M_{\text{Conn Cov}}$ if $M_{\text{Connected Cover}} - \{S_i, M_{\text{Connected Cover}}\}$ is M-connected and meets $1/k/Q$ coverage requirement. M-connected cover optimization algorithm is given in Algorithm 8.

<table>
<thead>
<tr>
<th>Algorithm 8 M-Connected Cover Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Input: $M_{\text{Connected Cover}}$</td>
</tr>
<tr>
<td>2: Output: Opt.$M_{\text{Conn Cov}}$</td>
</tr>
<tr>
<td>3: Initialize Opt.$M_{\text{Conn Cov}} = \phi$</td>
</tr>
<tr>
<td>4: for $i = \text{length}(M_{\text{Connected Cover}})$ down to 1 do</td>
</tr>
<tr>
<td>5: if $M_{\text{Connected Cover}} - {S_i, M_{\text{Connected Cover}}}$ meets $k/Q$ coverage requirement and is M-connected then</td>
</tr>
<tr>
<td>6: Ignore $S_i, M_{\text{Connected Cover}}$</td>
</tr>
<tr>
<td>7: $M_{\text{Connected Cover}} = M_{\text{Connected Cover}} - {S_i, M_{\text{Connected Cover}}}$</td>
</tr>
<tr>
<td>8: else</td>
</tr>
<tr>
<td>9: Opt.$M_{\text{Conn Cov}} = $Opt.$M_{\text{Conn Cov}} \cup {S_i, M_{\text{Connected Cover}}}$</td>
</tr>
<tr>
<td>10: end if</td>
</tr>
<tr>
<td>11: end for</td>
</tr>
</tbody>
</table>

### 2.5 Results and Discussion

#### 2.5.1 Direct Base Station Connected Coverage

We consider a 500m×500m region with the number of sensors varying from 300-600 to monitor 25 targets. Sensing range of each sensor node is fixed as 75m. Initial battery power of each node is set to 100 units. $E_{\text{min}}$ is 0 and $e$ is 1 unit of energy per unit of time. Results are noted for various $k$ values, ranging from 1-5 and for $Q$ ranging from 1-3, 1-4 and 1-5. Results are reported as an average of 25 experiments for each case.

**Comparison of upper bound and network lifetime without cover optimization**

The theoretical upper bound is calculated initially. Experiments are conducted with remaining battery power as the only priority deciding factor. Figure 2.4 shows the experimental results for simple coverage problem. Figure 2.5 and Figure 2.6 show the upper bound and the network lifetime without cover optimization for different values of $k$ and $Q$ respectively.
Figure 2.4: Comparison of upper bound (U.B), network lifetime without cover optimization (W.C.O), network lifetime with cover optimization (C.O) using only $w_1$ and network lifetime using proposed approach (P.A) for simple coverage problem

Comparison of network lifetime without cover optimization and with cover optimization

There is a scope of improved network lifetime with cover optimization. Figure 2.7 and Figure 2.8 show that optimizing the generated cover yields much better results. Network lifetime cannot be further improved so as to achieve the upper bound because remaining battery power is the only criteria that decided the cover formation.

Comparison of upper bound and network lifetime using proposed approach

Since the upper bound is known, the parameter $\alpha$ can be varied to check for the best network lifetime. The selection of tuning parameter is crucial. This specifies how the weight components can be tuned to search for improved network lifetime. The upper bound is met for certain instances. For those instances where the upper bound is not
Figure 2.5: Comparison of upper bound and network lifetime without cover optimization for $k$-coverage problem

Figure 2.6: Comparison of upper bound and network lifetime without cover optimization for $Q$-coverage problem
Figure 2.7: Comparison of network lifetime without cover optimization and with cover optimization for $k$-coverage problem

Figure 2.8: Comparison of network lifetime without cover optimization and with cover optimization for $Q$-coverage problem
Figure 2.9: Comparison of upper bound and network lifetime using proposed method for $k$-coverage problem.

Figure 2.10: Comparison of upper bound and network lifetime using proposed method for $Q$-coverage problem.

41
Figure 2.11: Impact of varying number of nodes & varying $k$ for $k$-coverage problem achieved, adaptive tuning of weight deciding parameter is done. It is evident that there exists some instances where the sensor nodes that cover least number of targets should be given priority and the nodes which cover large number of targets should be kept in reserve for use in later cycles. An inverse of target assigned weight helps to achieve the upper bound. Figure 2.9 and Figure 2.10 show that this adaptive tuning mechanism achieved the optimal network lifetime for all the experimented cases. The theoretical upper bound and the experimental output using the proposed heuristic scheme are represented using bold face in Table 2.1 and Table 2.2.

**Impact of varying number of nodes, varying $k/Q$**

The network is modeled as both sparse and dense to study the impact of varying number of nodes. The number of sensor nodes deployed in the region is varied between 300-600. Figure 2.11 and Figure 2.12 show that for higher number of sensor nodes, we experience an increase in the network lifetime. It also shows that higher $k/Q$ value decreases the network lifetime. The higher the coverage requirement, the lesser the network lifetime, because there may not be enough number of sensor nodes to satisfy
the coverage requirement.

![Figure 2.12: Impact of varying number of nodes & varying Q for Q-coverage problem](image)

**Comparison of Greedy-MSC, HESL and proposed approach**

Table 2.3 shows the comparison of network lifetime using Greedy-MSC and proposed approach. The proposed approach consistently achieves the best network lifetime. Table 2.4 shows the comparison of network lifetime using HESL and proposed approach. The number of sensor nodes is fixed as 450 and the number of targets is varied between 200-400. The proposed approach outperforms HESL in all the experimented cases.
Table 2.1: Network Lifetime for $k$-coverage problem

<table>
<thead>
<tr>
<th>N.S</th>
<th>N.T</th>
<th>k=1</th>
<th>k=2</th>
<th>k=3</th>
<th>k=4</th>
<th>k=5</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>25</td>
<td>828</td>
<td>795.56</td>
<td>827.96</td>
<td>828</td>
<td>412</td>
</tr>
<tr>
<td>350</td>
<td>25</td>
<td>996</td>
<td>963.48</td>
<td>995.96</td>
<td>996</td>
<td>498</td>
</tr>
<tr>
<td>400</td>
<td>25</td>
<td>1128</td>
<td>1071.3</td>
<td>1128</td>
<td>1128</td>
<td>564</td>
</tr>
<tr>
<td>450</td>
<td>25</td>
<td>1260</td>
<td>1227.6</td>
<td>1260</td>
<td>1260</td>
<td>630</td>
</tr>
<tr>
<td>500</td>
<td>25</td>
<td>1448</td>
<td>1375.6</td>
<td>1447.9</td>
<td>1448</td>
<td>724</td>
</tr>
<tr>
<td>550</td>
<td>25</td>
<td>1724</td>
<td>1627.2</td>
<td>1723.9</td>
<td>1724</td>
<td>862</td>
</tr>
<tr>
<td>600</td>
<td>25</td>
<td>1836</td>
<td>1727</td>
<td>1835.9</td>
<td>1836</td>
<td>918</td>
</tr>
</tbody>
</table>

1 Number of sensor nodes
2 Number of targets
3 Upper bound
4 Network lifetime with remaining battery power as the only priority deciding factor
5 Without cover optimization
6 With cover optimization
7 Network lifetime using proposed approach
Table 2.2: Network Lifetime for \( Q \)-coverage problem

<table>
<thead>
<tr>
<th>( Q = 1-3 )</th>
<th>( Q = 1-4 )</th>
<th>( Q = 1-5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>R.B.P (^1)</td>
<td>R.B.P (^1)</td>
<td>R.B.P (^1)</td>
</tr>
<tr>
<td>W.C.O (^2)</td>
<td>C.O (^6)</td>
<td>P.A (^7)</td>
</tr>
<tr>
<td>U.B (^3)</td>
<td>W.C.O (^2)</td>
<td>C.O (^6)</td>
</tr>
<tr>
<td>N.T (^2)</td>
<td>U.B (^3)</td>
<td>P.A (^7)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300 25 350.4</td>
<td>345 350.4 350.4</td>
<td>258.92 258.92 258.92</td>
</tr>
<tr>
<td>350 25 425.04</td>
<td>418.2 424.96 425.04</td>
<td>306.32 306.32 359.32</td>
</tr>
<tr>
<td>400 25 470.44</td>
<td>460 470.44 470.44</td>
<td>353.08 356.88 394.72</td>
</tr>
<tr>
<td>450 25 515.44</td>
<td>503.52 543</td>
<td>393.96 393.96 393.96</td>
</tr>
<tr>
<td>500 25 618.44</td>
<td>608.28 618.44</td>
<td>451.84 459.56 459.56</td>
</tr>
<tr>
<td>550 25 716.44</td>
<td>707 716.44 716.44</td>
<td>518.76 529.88 529.88</td>
</tr>
<tr>
<td>600 25 724.44</td>
<td>718.24 724.44</td>
<td>582.12 584.56 584.56</td>
</tr>
</tbody>
</table>

1. Number of sensor nodes
2. Number of targets
3. Upper bound
4. Network lifetime with remaining battery power as the only priority deciding factor
5. Without cover optimization
6. With cover optimization
7. Network lifetime using proposed approach
Table 2.3: Comparison of network lifetime using Greedy-MSC (G-MSC) and proposed approach for $k$-coverage problem

<table>
<thead>
<tr>
<th>N.S</th>
<th>k = 1</th>
<th>k = 2</th>
<th>k = 3</th>
<th>k = 4</th>
<th>k = 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>828</td>
<td>817.8</td>
<td>828</td>
<td>412</td>
<td>400.24</td>
</tr>
<tr>
<td>350</td>
<td>996</td>
<td>992.44</td>
<td>996</td>
<td>498</td>
<td>488.68</td>
</tr>
<tr>
<td>400</td>
<td>1128</td>
<td>1113</td>
<td>1128</td>
<td>564</td>
<td>548.56</td>
</tr>
<tr>
<td>450</td>
<td>1260</td>
<td>1244.1</td>
<td>1260</td>
<td>630</td>
<td>618.48</td>
</tr>
<tr>
<td>500</td>
<td>1448</td>
<td>1432.5</td>
<td>1448</td>
<td>724</td>
<td>708.6</td>
</tr>
<tr>
<td>550</td>
<td>1724</td>
<td>1713.7</td>
<td>1724</td>
<td>862</td>
<td>850.32</td>
</tr>
<tr>
<td>600</td>
<td>1836</td>
<td>1815.3</td>
<td>1836</td>
<td>918</td>
<td>903.24</td>
</tr>
</tbody>
</table>
Table 2.4: Comparison of network lifetime using HESL and proposed approach for Q-coverage problem

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>U.B</td>
<td>HESL</td>
<td>P.A</td>
<td></td>
<td>U.B</td>
<td>HESL</td>
</tr>
<tr>
<td>200</td>
<td>327.64</td>
<td>325.96</td>
<td>327.64</td>
<td></td>
<td>245.64</td>
<td>243.92</td>
</tr>
<tr>
<td>250</td>
<td>301</td>
<td>301</td>
<td>301</td>
<td></td>
<td>243.64</td>
<td>241.76</td>
</tr>
<tr>
<td>300</td>
<td>205.32</td>
<td>203.68</td>
<td>205.32</td>
<td></td>
<td>231.28</td>
<td>231</td>
</tr>
<tr>
<td>350</td>
<td>289.76</td>
<td>287.76</td>
<td>289.76</td>
<td></td>
<td>231.64</td>
<td>231</td>
</tr>
<tr>
<td>400</td>
<td>317</td>
<td>312.16</td>
<td>317</td>
<td></td>
<td>223.88</td>
<td>221.68</td>
</tr>
</tbody>
</table>

Time Complexity Analysis

We compare the execution time of Greedy-MSC, HESL, and the proposed algorithm. Execution time highly depends on the machine in which the algorithm is executed. Let the execution time of a demo code be $t_x_1$ units of time and the execution time of the algorithm which has to be evaluated be $t_x_2$ units of time. `$t_x_2/t_x_1$` is constant for all machines where the demo code and the algorithm is executed. We report the time complexity values in Table 2.5 based on this to avoid machine dependence.

Greedy-MSC vs Proposed Approach

The execution time is observed separately for three different cases; (a) when both methods achieve the same network lifetime (b) when the proposed algorithm outperforms Greedy-MSC at the first run itself and (c) when the proposed algorithm performs better than Greedy-MSC after tuning weight deciding component. We present the execution time for all the above three categories in the time complexity comparison of Greedy-MSC and proposed algorithm in Table 2.5. The execution time of the proposed approach is less when compared to Greedy-MSC in cases where both methods achieved the optimal network lifetime. But in certain cases where Greedy-MSC cannot achieve
the optimal network lifetime, the proposed approach can achieve the best result in the first run itself with a slight increase in the execution time. This increase is because there are more number of cover sets to be formed as compared to G-MSC. But in cases where the optimal network lifetime was not achieved at the first run of the proposed approach, the weights are recalculated and the cover sets are recomputed. This increases the execution time significantly, but the optimal network lifetime is achieved.

**HESL vs Proposed Approach**

There are cases where HESL and proposed method achieve the same network lifetime and cases where proposed method achieved better network lifetime than HESL. These two categories are shown in the time complexity comparison of Greedy-MSC and proposed algorithm in Table 2.5. The time for execution of HESL and proposed algorithm are the same when both achieves the same network lifetime. In cases where proposed approach achieved better results, the time taken for execution is high because the covers are recomputed. HESL studies the influence of cover activation time over network lifetime. The more time a cover remains active, less will be the execution time since the next cover will be computed only after that greater amount of time.

The proposed approach performs slightly better in terms of execution time where the results are comparable between Greedy-MSC. The proposed approach requires more execution time when it is able to increase the network lifetime which cannot be achieved.

Table 2.5: Time Complexity of Greedy-MSC, HESL and proposed algorithm

<table>
<thead>
<tr>
<th>Alg</th>
<th>T.C</th>
<th>N.L</th>
</tr>
</thead>
<tbody>
<tr>
<td>G-MSC</td>
<td>5.929</td>
<td>1200</td>
</tr>
<tr>
<td>P.A</td>
<td>5.209</td>
<td>1200</td>
</tr>
<tr>
<td>G-MSC</td>
<td>5.006</td>
<td>1160</td>
</tr>
<tr>
<td>P.A</td>
<td>5.310</td>
<td>1200</td>
</tr>
<tr>
<td>G-MSC</td>
<td>3.921</td>
<td>960</td>
</tr>
<tr>
<td>P.A</td>
<td>6.102</td>
<td>1000</td>
</tr>
<tr>
<td>HESL</td>
<td>7.426</td>
<td>433</td>
</tr>
<tr>
<td>P.A</td>
<td>7.398</td>
<td>433</td>
</tr>
<tr>
<td>HESL</td>
<td>13.263</td>
<td>425</td>
</tr>
<tr>
<td>P.A</td>
<td>12.063</td>
<td>433</td>
</tr>
</tbody>
</table>

1 when G-MSC and P.A achieved upper bound  
2 when P.A outperformed G-MSC at first run  
3 when P.A outperformed G-MSC after tuning weight deciding component  
4 when HESL and P.A achieved upper bound  
5 when P.A outperformed HESL after tuning weight deciding component  
6 Algorithm  
7 Time complexity  
8 Network Lifetime
by the other two existing approaches. The objective of this work is to maximize the network lifetime and moreover the scheduling is decided off-line. Thus an increase in execution time is of little concern particularly when the network lifetime can be increased.

2.5.2 $M$-Connected Coverage

We consider a $200m \times 200m$ region for experiments. The number of sensor nodes is varied between 150-250 to monitor 25 targets. The sensing range is fixed as 40m and the communication range is 80m. Experiments are carried out for simple coverage, $k$ values 2, 3, 4 and $Q$ values ranging from 1-3, 1-4 and 1-5. $M$ can take an integer value between 1 and 3.

Impact of varying $k/Q$

When $k/Q$ requirement increases, the number of nodes that need to be active increases and since the number of nodes is large in each cover, there is a large possibility of the nodes being connected. This will leave the network lifetime unaffected when connectivity is also considered.

Impact of varying $M$

For simple coverage problem, when $M$ increases, a slight decrease in network lifetime is observed (Figure 2.13). This is because only very few nodes need to be active for satisfying the coverage requirement. For making them $M$-connected, some other nodes will have to be active, bringing down the overall network lifetime. In case of higher $k/Q$ coverage requirement, since more number of nodes need to be active, there are chances that these nodes will be $M$-connected as well. Figure 2.14 and Figure 2.15 show the network lifetime for $M$-connected $k$-coverage and $M$-connected $Q$-coverage respectively.
Figure 2.13: Network Lifetime for $M$-Connected simple coverage problem

Figure 2.14: Network Lifetime for $M$-Connected $k$-coverage problem
Impact of varying number of nodes

When average network lifetime is considered, for higher number of sensor nodes, network lifetime may or may not increase. The location of targets, location of sensors and the $k/Q$ values contribute to determining the network lifetime. When given a region with more sensor nodes, it need not be necessary that the network lifetime will be high. If the region has more idle sensor nodes, there are chances that the network lifetime may drop compared to a region with less number of sensor nodes where most of them are non-idle.

Comparison with CWGC

Communication Weighted Greedy Cover (CWGC) (Zhao and Gurusamy, 2008) uses a greedy method to select the set of source nodes (called source set) that cover the targets and it couples the communication cost and the selection of source sets. Though the method is for multi-hop communication where a sensor cannot reach the sink node
directly, and many of the assumption do not match our model, we perform a comparison based on the operational time of a sensor cover in CWGC. In CWGC, each cover operates for a fixed time duration, unless some sensors in the cover will die before the end of the time duration due to the lack of energy. This might not give the optimal
Figure 2.18: Comparison of CWGC and proposed approach for $M$-Connected $k$-coverage problem where $M = 2$

network lifetime for some cases. We have shown this with an example in Figure 2.16 for $k$-Coverage problem with $k = 2$. The region has 3 sensor nodes $S_1$, $S_2$, $S_3$ with battery power 100 units and energy consumption rate 1 unit, and two targets $T_1$ and $T_2$. Each sensor node is able to monitor both the targets and all the three sensor nodes are connected. Let the initial cover formed be $\{S_1, S_2\}$. With CWGC, the same cover will be active till at least one node dies. So, the cover will be active for 100 units of time, yielding a network lifetime of 100 units. With our proposed method, if covers are computed for each time instant, based on priority of sensor node(battery power), the following will be the sensor covers $\{\{S_1, S_2\}, \{S_3, S_1\}, \{S_2, S_3\}, \ldots\}$. This will give a network lifetime of 150 units. Figure 2.17 shows a comparison of our proposed approach with CWGC for simple coverage problem. Figure 2.18 and Figure 2.19 show a comparison of our proposed approach with CWGC for $k$-coverage and $Q$-coverage problems respectively together with $M$-Connectivity.
Figure 2.19: Comparison of CWGC and proposed approach for $M$-Connected Q-coverage problem where $M = 2$

2.6 Conclusion

In dense networks, where the number of sensor nodes are large, the possible combinations of cover sets are large and choosing the best cover sequence among those is a difficult task. Cover optimization naturally extends lifetime for some cases where the optimal solution cannot be achieved. There are special cases where nodes that monitor more number of targets should be reserved for later use. The results obtained show that in such cases, the adaptive tuning helps to achieve the optimal network lifetime. Our proposed approach finds a cover sequence such that the network lifetime matches the theoretical upper bound for all experimented simulations.

Sensitive applications of wireless sensor networks require high level of connectivity as well as coverage. We propose a method to schedule the sensor nodes such that only minimum number of sensor nodes will be active, satisfying connectivity and coverage requirement. This leads to a higher network lifetime. The need for such a method arises when all the targets need not be monitored with the same proximity and when $M$-connectedness should exist within the nodes that are turned on, to guarantee the
correctness of the information collected and also to ensure the reachability of the in-
formation at other nodes including base station. We observe that the introduction of
connectivity does not affect the network lifetime to a greater extent. The proposed
method performs better than CWGC.