Connected Set Cover Partitioning Problem in Wireless Sensor Networks

a dissertation submitted in partial fulfilment of the requirements for the M.Tech. (Computer Science) degree of Indian Statistical Institute

By

Nargis Pervin

Roll No: CS0705

under the supervision of

Prof. Nabanita Das

Advanced Computing and Microelectronics Unit

INDIAN STATISTICAL INSTITUTE
203, Barrackpore Trunk Road
Kolkata - 700108

Indian Statistical Institute
Kolkata-700 108

CERTIFICATE

This is to certify that the thesis entitled **"Connected Set Cover Partitioning Problem in Wireless Sensor Networks"** is submitted in the partial fulfilment of the degree of M. Tech. in Computer Science at Indian Statistical Institute, Kolkata. It is fully adequate, in scope and quality as a dissertation for the required degree.

The thesis is a faithfully record of bona fide research work carried out by Nargis Pervin under my supervision and guidance. It is further certified that no part this thesis has been submitted to any other university or institute for the award of any degree or diploma.

Prof. Nabanita Das
(Supervisor)

Countersigned
(External Examiner)
Date:

Abstract

Wireless ad-hoc sensor network has recently emerged as an effective way of monitoring remote or inhospitable areas. In overdeployed sensor networks, one effective way to conserve energy is to keep only a small subset of sensor nodes active at a particular instant covering the whole area. Connectivity in any active subset is another important issue since the information gathered by the sensor nodes in any subset should be finally transferred to the sink node via multi-hop paths, in general. So, in this paper we pose the problem of finding maximum number of connected set covers that can guarantee the required coverage of a query region. Once the connected set covers are known, the sets may remain active in a round robin fashion to cover the region enhancing the life time of the network significantly. Here we have developed two centralized versions of the algorithms, a distributed one and finally compared the performances by simulation.

key words: wireless sensor network (WSN), set cover, connectivity, coverage, partitioning, one-hop neighbor, multi-hop paths

Thesis Roadmap

In **chapter 1**, we will give a brief overview of wireless sensor networks and its various applications in practical field. We will discuss the coverage with connectivity problem in sensor networks and a formal problem definition of connected set cover partitioning problem for which we are going to present a solution.

Next in **chapter 2**, we will give the literature review on coverage and coverage with connectivity problem along with few important results proposed in those articles.

Chapter 3 contains the algorithms we will propose along with its computational complexity study.

In **chapter 4**, we will give the performance study of the different algorithms and comparison amongst them. We will discuss the relations among the various network parameter viz. sensing range, transmission range, coverage of the network, number of sensors etc with the simulated results and plotted graphs.

Last chapter, **chapter 5** concludes the discussions of coverage and connectivity problem in sensor networks.

Contents

List of Figures

Acknowledgement

Acknowledgments are in place for all those people who have helped me during my stay at ISI.

First, with great pleasure I would like to express my heartfelt and sincere gratitude to my guide and adviser Professor Nabanita Das of Advanced Computing and Micro electronics Unit (ACMU), Indian Statistical Institute, Kolkata. She has always been there to listen to me, advise me and share her valuable ideas with me. I am grateful to her for providing me with a scope to work under her supervision.

I found myself in a wonderful and homely environment for work at the ACMU Lab where I could indulge in my dissertation problem. Special thanks go to everyone in the ACMU Lab.

I am thankful to my classmate Sanjay for inspiring me whenever I lost my spirits. He taught me to be persistent for accomplishing any goal.

Here in ISI while doing the coursework I got great friends like Kalikinkar, Pulak, Sandeep who have always helped me positively in my dissertation work. Without their substantial support and help it would have been really difficult for me to complete my work.

My classmates Aritra, Mrinmoy, Swarup, Subhabrata also deserve a special mention here for their constant support.

I would like to express my heartfelt gratitude to Somindu for all her advice and encouragement and her help and support in writing this thesis in an organized way. We have shared quite a few light-hearted moments together that helped ease out the work pressure.

I have studied with SreeVani for most of the time. I could share not only my academic problems but also my personal issues with her. We have enjoyed our time together to the fullest. My hostelmates SreeVani, Somindu, Debasmita, Preeti and Megha have made my stay at ISI memorable.

I am proud to mention that my schoolfriend, Debashis, though far away from me, has always motivated me to overcome any problem. Whenever I needed, I found his comforting advices and helping hand.

I would like to thank Mahiuddin, my fiance, who has always encouraged me with my work and listened to all my problems, however trivial, with amazing patience and interest. Without his unselfish help and support the path of my life would have been much more difficult.

Last, but not the least, I thank my family. I thank my parents, for educating me, for instilling in me aspects of both arts and sciences and for unconditional support and encouragement to pursue my interests. I am indebted to my brother, Saiful, who has given me moral support in everything I have done, listening to my complaints and frustrations, and believing in me.

Nargis Pervin

Declaration

No part of this thesis has been previously submitted by the author for a degree at any other university, and all the results contained within, unless otherwise stated are claimed as original. The results obtained in this thesis are all due to fresh work.

(Nargis Pervin)
Date: of July, 2009

Chapter 1

Introduction

1.1 A Brief Overview of Sensor Networks

Sensor networks are wireless networks comprised of a large number of sensor nodes, each of which is equipped with one or more sensors, processing elements, and a wireless transceiver within the size of several cube millimeters. The dimension of a sensor node are small enough to allow easy deployment of large number of sensor nodes into remote or inhospitable areas. Below is a picture of wireless sensor network consisting of a number of nodes.

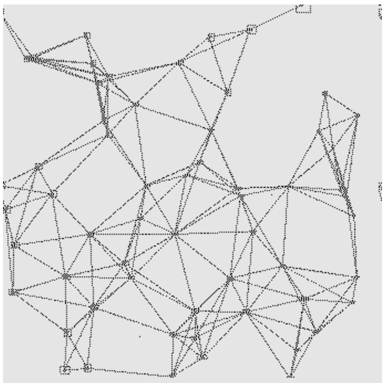


Figure 1.1: sensor network

Recently, there has been surge of interest in large-scale sensor networks. It is expected that the network will be deployed for various applications. For example, consider the millions of acres that are lost around the world, due to forest fires every year. In all fires early warning are critical in preventing small harmless brush fires from becoming monstrous infernos. By deploying specialized wireless sensor nodes is strategically selected high-risk areas, the detection time for such disaster can be drastically reduced, increasing the likelihood of success in early extinguishing efforts. Also since the nodes are self-configuring and do not need constant monitoring, the cost of such network is minimal compared to the huge losses incurred in a large blaze [6].

Wireless sensor networks provide a viable alternative to several existing technologies. For example, large buildings contain hundreds of environmental sensors that are wired to a central air conditioning and ventilation system. The significant wiring costs limit the complexity of current environmental controls and the reconfigurability of these systems. However replacing the hardwired monitoring units with ad-hoc wireless sensors can improve the quality and energy efficiency of the environmental system while allowing almost unlimited reconfiguration and customization in the future. In many instances, the savings in the wiring costs can alone justify the use of the wireless sensor networks.

Due to small dimension of the sensor nodes the energy supply is very limited. In addition, it is usually hard to recharge the battery after deployment, either because the number of sensor nodes deployed is too large or because the deployment area is hostile. Once deployed, a sensor network is expected to keep working for several weeks or months without any maintainence or recharging. Therefore, energy consumption becomes the essential requirement in WSNs.

We will illustrate the fact with example. Habitat monitoring may require continuous operation for months, and monitoring civil structures (e.g., bridges) requires an operational lifetime of several years. Recent research has found that significant energy savings can be achieved by dynamic management of node duty cycles in sensor networks with high node density. In this approach, some nodes are scheduled to sleep (or enter a power-saving mode) while the remaining active nodes provide continuous service. A fundamental problem is to minimize the number of nodes that remain active while still achieving acceptable quality of service for applications. In particular, maintaining sufficient sensing coverage and network connectivity with the active nodes is a critical requirement in sensor networks.

1.2 Coverage and Connectivity Problem in Wireless Sensor Networks

We will consider two important characteristics, viz. coverage and connectivity of a sensor network. A piece of area in the deployment region is *covered* if every point of that area is within the sensing range (S) of some sensor nodes. On the other hand two sensor nodes are said to be *connected* if one node is within the transmission range (T) of another node. We say that the network is *connected* if any active node can communicate with any other active node (possibly using other active nodes are relays).

Here follows some definitions used in this thesis.

Definition 1.2.1. A sensor node i located at a point p' is said to *cover* a point p in the query region, if the Euclidean distance between p and p' is less than S, the sensing range of i.

Definition 1.2.2. Given a sensor network consisting of a set of sensors $\mathcal{I}$, the *communication graph* for the sensor network is the undirected graph with $\mathcal{I}$ as the set of vertices and an edge between any two sensors if the Euclidean distance between the two is less than T, the transmission range. The *communication graph induced* by a set of sensors $\mathcal{M} \subseteq \mathcal{I}$ is the subgraph of the communication graph involving only the vertices in $\mathcal{M}$ and the corresponding edges.

Definition 1.2.3. Consider a sensor network consisting of a set $\mathcal{I}$ of n sensors and a query region Q. A set of sensors $N \subseteq \mathcal{I}$ is said to be a *connected* 1-*cover* for the query region if the following two conditions hold:

- Each point p in Q is covered by at least one sensor from N
- The communication graph induced by N is connected

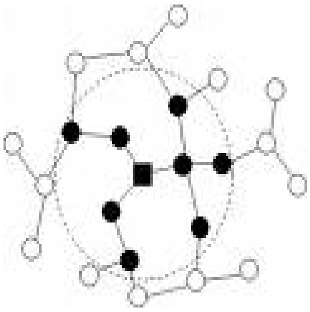


Figure 1.2: communication graph

Formally we can define the coverage and the connectivity problem as follows:
Connected Set Cover Problem: Given n sensor nodes distributed over a query region, the Connected Set Cover Problem is to find a *connected 1-cover* of smallest size. This problem is NP-hard [9].

Any solution to this problem results a connected node subset $\mathcal{M} \subseteq \mathcal{I}$. However, it may leave a large number of sensors in the set $\mathcal{I} \backslash \mathcal{M}$ unused. In this thesis we pose the following problem.
Connected Set Cover Partitioning Problem: The Connected Set Cover Partitioning Problem is to partition the sensors into connected 1-covers such that the number of covers is maximized where the communication graph induced by the sensors in each cover is connected.

Here is a network with 10 nodes placed randomly in a 5×5 query region $\mathcal{Q}$. We consider the transmission radius and the sensing radius for all the sensors are same and equal to 3 units. Here from the figure we can see that two connected set cover partitions are $\{1, 2, 4, 6, 9\}$ and $\{3, 4, 5, 8, 10\}$. Both the covers give the full coverage for the query region $\mathcal{Q}$.

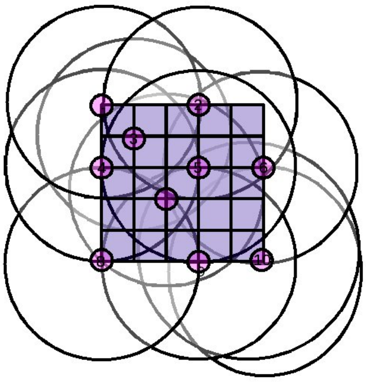


Figure 1.3: connected set cover partitions

1.3 Problem Statement

Let us assume that the whole area is divided into elementary areas. In an over deployed region there will be lots of elementary areas covered by more than one sensor node. To improve the longevity of a network and conserve battery power, rather than using all the sensor nodes to monitor an event, we can activate only a group of sensor nodes in rounds, so that the battery life of a sensor is not wasted on areas that are covered by some other sensor nodes. So our aim is to partition the set of sensor nodes into mutually exclusive set covers and only one such cover will be activated at a time to monitor the area of interest. Also the sensor nodes in each cover should be connected so that they can send their information to the sink node. Our aim is to maximize the number of such covers. If we have K such covers then activating only one cover at a time will increase the life time of the network by K times.

1.4 Summary

In this chapter we have given a brief overview of wireless sensor networks (WSNs) and its applications in different practical fields. Sensor networks consist of large number of sensor nodes which are very small in dimension. Due to small dimension of sensor nodes we have very limited battery power and also the deployment of the sensor nodes is very easy. Since the deterministic placement of sensor nodes is impractical ([9]), the sensor nodes are deployed in cluster. To conserve the power of the battery and thus increase the longevity of the network, we select mutually exclusive sets of sensor nodes and activate only one set of sensor nodes at a particular instant. The communication graph induced by the sensor nodes of a partition should be connected so that they can collectively send the information to a central node, called the sink node.

Chapter 2

Related Work

Coverage and connectivity are two important performance metrics in sensor networks. Sensing coverage gives the quality of service provided by a network and connectivity guarantees that the accumulated information of an event are sent to the central node for taking immediate necessary actions. In the next two sections we are going to discuss the already existing literature on both coverage problem and coverage with connectivity problem of a network.

2.1 Literature on Coverage Problem

Recently, there has been a lot of research work done to address the coverage problem in sensor networks. In [3], Huang et al. have formulated the coverage problem as a decision problem. Their goal is to determine whether every point in the service area of the sensor network is covered by at least K sensors, where K is a predefined value.

In [5], Meguerdichian et al. defined worst and best case coverage problems, which are to identify regions of low and high observability, respectively. They have used Voronoi diagram and Delaunay triangulation to solve the problem.

In [11] the authors have given a different definition of coverage of a sensor network and have formulated an exact mathematical expression for expected k-coverage in the presence of border-effects. The concept of border-effect is that a sensor placed near the border of the deployment region will cover less area than the sensor placed at the center since all its disk-shaped sensory region will be within the deployment area. As the authors defined the coverage, a region is said to be k-covered if every location within it is covered by at least k sensors. k-coverage is defined to be the size of the k-covered region after a number of sensors have been randomly placed. The results they have given can be used to determine the related parameters for a desired network coverage. Also the result can be used to determine the minimal number of sensor nodes required for a desired coverage.

In [2] authors have solved the problem of coverage optimization of a sensor network by introducing weighted sensing topology and the basic maximal independent node-set. They have given a heuristic for solving the problem while considering the per-node location information known. They have proved the following important result. Let SG(V,E_s) and CG(V,E_c) be graphs of a WSN, with the sensing radius S and transmission radius T. Then when $T \geq 2S$, SG(V,E_s) is a subgraph of CG(V,E_c); when $S \geq 2T$, CG(V,E_c) is a subgraph of SG(V,E_s). Furthermore, CG(V,E_c) and SG(V,E_s) are identical when $T = 2S$.

M. P. Sing and M. M. Gore [8] have proposed a solution on the coverage problem of sensor network where they have given a solution to determine whether every point in the sensor network area is covered by communication range of the sensor nodes or not. They have assumed that the sensing range of every sensor is same as the communication range of the network. Now the problem is how to deploy the sensor nodes in a specified area a so that the area will be fully covered to give good quality of service and increase the fault tolerance of the network.

In [9], for a given constant K the authors have designed a heuristic to select mutually exclusive K sets of sensor nodes where the members of each of those sets together completely cover the region of interest. Each set is active for the same amount of interval, and only one set is active at a particular time. The authors have presented a heuristic approach to solve the *Set K-Cover Problem.* They have solved the problem by finding the list of the areas covered by each sensor node. Next they have given the heuristic solution so called Most Constrained - Minimally Constraining Covering Heuristic where the main idea is to minimize the coverage of the sparsely covered areas within a cover.

However, finding the list of the areas covered by a sensor node is not practical, unless we know the geographical location of the nodes. Also connectedness among the nodes in the same cover has not been considered.

In [1] the authors have designed three approximation algorithms for the Set K-Cover Problem, where the objective is to partition the sensors into covers such that the number of such covers is maximized. They have given a randomized algorithm with an expected approximation factor of $(1-\frac{1}{e})$. In this algorithm the sensors simply generate a random number and assign themselves to one of the K covers. They have also proposed a distributed greedy algorithm with a deterministic approximation ratio and one centralized greedy algorithm with deterministic $(1 - \frac{1}{e})$ approximation ratio. In the distributed greedy algorithm, each sensor assigns on itself to the cover in which the the intersection is minimum with the areas covered by the sensor and the areas covered by the cover so far. This reduces the redundant sensors in a cover. On the other hand the centralized greedy algorithm is similar except the fact that the area in the intersection is weighted on the basis of how likely it is to be covered by some other sensor in future in the assignment process. The simulations indicate that the longevity of a network increases with the amount of overlap among the sensors.

2.2 Literature on Connected Coverage Problem

Himanshu Gupta et al. [12] have considered the coverage problem of sensor network while maintaining the connectedness among the sensors in the cover. In particular, they have considered the problem of selecting a minimum size connected K-Cover, which is defined as a set of sensors M such that each point in the sensor network is covered by at least K different sensors and the communication graph induced by the set M is connected so that they can collectively transmit data to a central node. Here K is a configurable parameter and when the network has higher probability of failure, larger value of K can be used. They have suggested a centralized approximation algorithm and a distributed algorithm for the problem. In this article the authors have shown that in a sensor network wherein each sensor has the uniform sensing and transmission radius of S and T respectively where $T \geq 2S$, K-Coverage implies K-connectivity in any closed area. They have claimed that in a sensor network with uniform sensing and transmission range of S and T, where $T \geq 2S$, the coverage graph of a set of sensors M is a subgraph of the communication graph of M. Also they have stated that if a set of nodes M is a sensor 1-cover, then M's coverage graph is connected and if $T \geq 2S$, if a set of nodes is sensor K-cover, then M is also a K-connected set.

In [4] the authors have dealt with the problem of scheduling sensor nodes to save energy and meet both constraints of sensing and network connectivity without accurate per-node location information. They have given a scheduling scheme where the sensing coverage is above a given requirement, all the active sensor nodes are connected among themselves, and each active sensor node knows at least one shortest route to the sink node. They have used random scheduling for network coverage and then turn on extra sensor nodes, if necessary to achieve network connectivity. The algorithm is fully distributed. The authors have considered *stationary* sensor networks in a two dimensional field and the sensor nodes are randomly and independently deployed in a field. They have also assumed that a sensor's transmission range is fixed and totally independent of its sensing range. They have presented some analytical results for the relationship among node density, scheduling parameters, coverage quality.

Shakkotai et al. [7] have considered an unreliable wireless sensor grid network where the sensor nodes are placed in a square region and have proposed a sufficient and necessary condition for the random grid network to cover the unit square region and also ensured the connectivity among the active sensor nodes.They have shown that the necessary and sufficient conditions for the random grid network to cover the unit square region as well as to ensure that the active nodes are connected are of the form

$$p(n)r^2(n) \sim \frac{log(n)}{n},$$

where $p(n)$ is the node success probability and $r(n)$ is the transmission radius and if the node success probability $p(n)$ is very small, the connectivity does not imply coverage.

In [10] the authors have presented the design and analysis of novel protocols that can dynamically configure a network to achieve guaranteed degrees of coverage and connectivity. They have presented a Coverage Configuration Protocol (CCP) that can provide different degrees of coverage requested by applications. This flexibility allows the network to self-configure for a wide range of applications and environments. Also they have provided a geometric analysis of the relationship between coverage and connectivity. Apart from that they have proposed a probabilistic coverage model and extend CCP to provide probabilistic coverage guarantees.

2.3 Summary

An effective approach for energy conservation in wireless sensor networks is scheduling sleep intervals for extraneous nodes while the remaining nodes stay active to provide continuous service. For the sensor network to operate successfully, the active nodes must maintain both sensing coverage and network connectivity. In this chapter we have discussed about coverage as well as connectivity problem in sensor networks. Many authors have given different definitions for coverage problem. But the main idea is to conserve energy and increase the life time of a network. If the network is overdeployed many sensor nodes will cover the same area. We can activate those sensors in different time intervals so that power can be conserved to increase life time. Also connectivity of a network is very important to send the accumulated information to a central node. So we concentrate on the problem of partitioning the given set of sensor nodes into maximum number of partition, where each partition is connected and covers the query region. Each partition will be made active in round robin fashion to extend the lifetime of the network.

Chapter 3

Connected Set Cover Partitioning Problem

3.1 Motivation

With the proposed cover finding algorithm, the coverage quality of the network can be guaranteed while the number of partitions are not predefined. We also have considered the connectedness of sensor nodes in the partitions at the same time. To operate successfully the network must be connected so that the sensor nodes can report the detected events to the sink node. Therefore, in addition to the coverage of the sensor network, the network must remain connected. We are here hence motivated to enhance both coverage quality and connectivity of the network at the same time.

3.2 Network Model

There are many ways to organize the communication architecture of a sensor network. A sensor network could be in a hierarchical structure where each sensor communicates with a local cluster head and the cluster head communicates directly with the sink node. Alternatively, it could be in a flat communication structure as well, where each sensor has essentially the same role and relies on other sensor to relay its messages to the sink node via multi-hop radio communication. Here we assume the flat communication architecture, where the joint problem of sensing coverage and network connectivity arises.

We consider two important performance metrics viz. coverage and connectivity of a network. We say that the network is connected if the active nodes can communicate with any other active node directly or using other nodes as relays. We assume that all the sensor nodes in the network has the same sensing range (S) and transmission range (T) and not necessarily $S = T$. Although we have simulate results for all the three cases when $T \geq S$, when $T = S$, and when the transmission $T \leq S$, we will be focused on the case when $T \leq S$. Radio transceivers are required for message communications, to set the transmission range of the sensor node and it is very much power consuming, where as the sensors are important for the sensing coverage of the sensor node, which needs very low power compared to transceivers. So it is preferable to keep the transmission range much lower than the sensing range of the sensor nodes. Also if $T \geq S$ only coverage of the network can guarantee the connectedness of the network and needs no further investigations, but the reverse is not true.

3.3 Algorithms for connected set cover partitioning

In this section we propose some heuristic approaches to maximize the number of connected set cover partition for a given network. The input to these algorithms are a set of sensor nodes $\mathcal{I}$ and a query region $\mathcal{Q}$ comprised of a set of elementary areas,$\mathcal{Q} : \{a_1, a_2, ..., a_m\}$. Each of $\mathcal{I}$ is represented by a subset of $\mathcal{Q}$ (the set of elementary areas covered by the sensor). Output is a set $\mathcal{S}$ of connected set covers $\{\mathcal{C}\}$ comprising of disjoint subsets of $\mathcal{I}$ and we assume that for all the sensor nodes, the sensing range S and transmission range T are uniform. First we will present two centralized versions of the algorithm and then a distributed version of the algorithm.

3.3.1 Centralized Algorithm 1

The first approach for finding the maximum number of connected set covers is a greedy approach and we start with an arbitrary sensor node and grow the cover around that node. At each step we add a sensor node from the remaining set of sensor nodes to the cover that is connected to that cover set such that the increment in the area covered is maximized. We grow the cover till the required coverage is achieved or $\mathcal{I}$ is exhausted. Once we get a connected cover, we try to make a different cover from rest of the sensor nodes. We define $\mathcal{S}$ to be the set of connected set covers. The algorithm is given below.

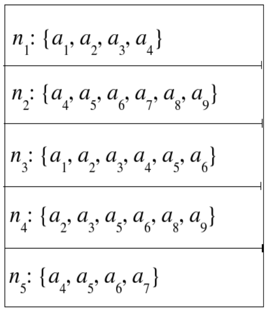
Algorithm 1

Input: Set of sensor nodes $\mathcal{I}$, a query region $\mathcal{Q}$, and covers of each node $i \ \ \forall i \in \mathcal{I}$
Output: Set of connected set covers $\mathcal{S}$

```
Initialize $\mathcal{S} = \emptyset$
Choose an arbitrary sensor node $v \in \mathcal{I}$
$\mathcal{C} \leftarrow \{v\}$
$\mathcal{I} \leftarrow \mathcal{I} \backslash v$
while $\mathcal{C}$ does not cover $\mathcal{Q}$ or $\mathcal{I} \neq \emptyset$
    Choose a sensor node $u \in \mathcal{I}$ such that $u$ is connected to $\mathcal{C}$
    and maximizes the coverage
```

```
7            $\mathcal{C} \leftarrow \mathcal{C} \cup \{u\}$
8            $\mathcal{I} \leftarrow \mathcal{I} \setminus \{u\}$
9        endwhile
10       if $\mathcal{C}$ covers $\mathcal{Q}$
11          then $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{C}\}$
12       if $\mathcal{I} \neq \emptyset$
13          then goto 2
14       return $\mathcal{S}$
```

Example 1: Let us consider a network with 5 sensor nodes $\{n_1, n_2, ..., n_5\}$, covering a set of elementary areas $\{a_1, a_2, ..., a_9\}$. The elementary areas covered by each node is shown in Fig.1, and the communication graph is shown in Fig.2.



n_1: $\{a_1, a_2, a_3, a_4\}$
n_2: $\{a_4, a_5, a_6, a_7, a_8, a_9\}$
n_3: $\{a_1, a_2, a_3, a_4, a_5, a_6\}$
n_4: $\{a_2, a_3, a_5, a_6, a_8, a_9\}$
n_5: $\{a_4, a_5, a_6, a_7\}$

Fig.1

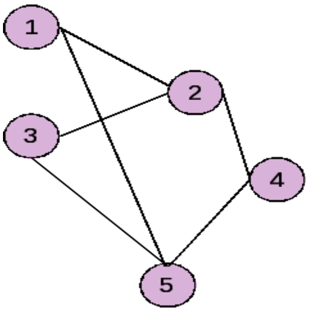


Fig.2

Without loss of generality let us assume that n_1 is chosen as the first node. Then Algorithm 1 results the set of partitions $\{\{n_1, n_2\}, \{n_3, n_4, n_5\}\}$.

Complexity Analysis

Here we will discuss the worst case complexity analysis of the above algorithm. Let the total number of sensor nodes deployed be n, the number of partitions achieved be l and $n_1, n_2, ..., n_l$ be the number of nodes in each partition. Then clearly, $\sum_{i=1}^{l} n_i \leq n$. Let for the i^{th} partition k nodes have already been chosen. To add the next node in the partition we need to check the connectedness from each of the k nodes with each of the remaining $(n - k)$ nodes. The union of the coverage of the candidate node and the the coverage of the partition can be done in constant time. Let the time complexity to form the i^{th} partition be $T(n_i)$. Then we can write,

$$\begin{aligned}
T(n_i) &= \sum_{k=1}^{n_i}[k(n-k)+O(1)] \\
&= n\sum_{k=1}^{n_i} k - \sum_{k=1}^{n_i} k^2 + cn_i \\
&= \frac{nn_i(n_i+1)}{2} - \frac{n_i(n_i+1)(2n_i+1)}{6} + cn_i \\
&= \frac{n({n_i}^2+n_i)}{2} - \frac{(2{n_i}^3+3{n_i}^2+n_i)}{6} + cn_i
\end{aligned}$$

The total time complexity is the sum over all $T(n_i)$.

$$\begin{aligned}
T(n) &= \sum_{i=1}^{l} T(n_i) \\
&= \sum_{i=1}^{l}\frac{n({n_i}^2+n_i)}{2} - \sum_{i=1}^{l}\frac{{n_i}^3}{3} - \sum_{i=1}^{l}\frac{{n_i}^2}{2} - \sum_{i=1}^{l}\frac{n_i}{6} + \sum_{i=1}^{l} cn_i \\
&< \frac{n}{2}\sum_{i=1}^{l}({n_i}^2+n_i) + \sum_{i=1}^{l} cn_i \\
&\leq \frac{n}{2}(\frac{\sum_{i=1}^{l} n_i}{l})^2 + n^2 + cn \quad [\text{since } \sum_{i=1}^{l} n_i \leq n] \\
&= \frac{n}{2}\cdot\frac{n^2}{l} + n^2 + cn \\
&= \frac{n^3}{2l} + n^2 + cn \\
&= O(n^3)
\end{aligned}$$

So the time complexity is $O(n^3)$.

So far, we have considered full coverage of $\mathcal{Q}$, i.e., each and every elementary area of $\mathcal{Q}$ is covered by at least one sensor $s_j \in \mathcal{C}_i$, $\forall \mathcal{C}_i \in \mathcal{S}$. However, depending on the application, we can relax the coverage restriction, such that any partition $\mathcal{C}_i$ is acceptable, if it can cover at least say $p\%$ of the elementary areas of $\mathcal{Q}$.

3.3.2 Centralized Algorithm 2

In the second algorithm, we start from an optimistic consideration. Here first we find the elementary area a, which is covered by minimum number of sensor nodes, say m. Then m is a natural upper bound on the number of possible partitions with full coverage. We put those m sensor nodes in m partitions. To attain the required coverage we grow the partitions around these sensor nodes till it is possible to improve the coverage. Each time we add only one sensor node to the cover which is connected to at least one sensor node in the cover. Here also, in each step the increment in coverage is maximized. We grow each cover till the required coverage is achieved or it is not possible to grow the cover any more. Define $\mathcal{S}$ to be connected set covers. The algorithm is as follows:

Algorithm 2
Input: Set of sensor nodes $\mathcal{I}$, coverage of each node i, $\forall i \in \mathcal{I}$
Output: Set of connected set covers $\mathcal{S}$

```
1     Find the minimally covered area $a$ covered by $m$ nodes $\mathcal{L}[i]$ $i = 1, 2, ..., m$
2     for $i = 1$ to $m$
3         Initialize $\mathcal{S}[i] \leftarrow \mathcal{L}[i]$
4     endfor
5     $\mathcal{I} \leftarrow \mathcal{I} \backslash \{L\}$
6     while $\mathcal{I} \neq \emptyset$
7             for $i = 1$ to $m$
8                 if $\mathcal{S}[i]$ has attained the required coverage
9                    then goto 7.
10                endif
11                Choose a sensor node $u \in \mathcal{I}$ such that $u$ is connected to $\mathcal{S}[i]$
                  and maximizes coverage
12                $\mathcal{S}[i] \leftarrow \mathcal{S}[i] \cup \{u\}$
13                $\mathcal{I} \leftarrow \mathcal{I} \backslash \{u\}$
14            endfor
15    endwhile
16    return $\mathcal{S}$
```

Example 2: For the network described in *Example 1*, according to *Algorithm 2* the minimally covered area is a_1 and it is covered by 2 nodes n_1 and n_3. Then n_1, n_3 will be put in different partitions. n_2 is connected to n_3, so n_2 can be added to cover 1. Next n_3 in cover 2 adds n_4 and n_5 to cover 2. The algorithm outputs the partition $\{\{n_1, n_4, n_5\},\{n_2, n_3\}\}$.

Complexity Analysis

Herein we will discuss the worst case time complexity of the above algorithm which is similar as that of algorithm 1. First finding the minimally covered area in the query region will take a constant amount of time. As we have done in the previous algorithm here also we have to check the connectedness between the remaining nodes and the nodes in the partition, which is of $O(n^3)$. So the asymptotic behavior of both the algorithms will be same. But in this algorithm the complexity depends on the initial number of partitions, if the number of partitions are more in number, then the complexity will reduce because many sensors are already placed in different partitions and need not to be checked further. But in the worst case the running time will be the same for both the algorithms, i.e., $O(n^3)$.

3.3.3 Centralized Algorithm 3

We have tried to improve the above *Algorithm 2* by reconsidering the partitions which did not attain the required coverage. The simple solution to this problem is, if possible, merge those partitions to improve the coverage. Here also we have tried to merge the partitions starting with the one which gives the least coverage. We will try to merge the worst covered partition with the one for which merging will improve the maximum possible coverage provided at least one of the sensor node in the later partition is connected to one or more sensor nodes in the other partition. If that partition attain the required coverage then we have increased the number of covers by one. Otherwise we will continue the merging process with the partitions which could not attain the required coverage. We will give only the rearrangement algorithm below. $\mathcal{L} = \{L_1, L_2, ..., L_t\}$ be the list of t partitions which did not attain the required coverage (th) and $\mathcal{P}$ be a list of m partitions which can give the required coverage.

Algorithm 3
Input: Set of sensor nodes $\mathcal{I}$, coverage of each node i, $\forall i \in \mathcal{I}$
Output: Set of connected set covers $\mathcal{P}$

```
1     if $t = 1$ and $\mathcal{P} \neq \emptyset$
2     Merge $\mathcal{L}_k$ with one $\mathcal{L}_i \in \mathcal{P}$ for which $|\mathcal{L}_i|$ is maximum
3     return $\mathcal{P}$
4     else
5         Find partition $\mathcal{L}_k$ with $n_k$ sensor nodes, having the minimum coverage
6         for $j = 1$ to $t$, $t \neq k$
7             for $i = 1$ to $n_k$
8                 if $\mathcal{L}_k[i]$ is connected to at least one sensor in $\mathcal{L}_j$
9                   $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{L}_j$
10                  goto 1
11                end if
12            end for
13        end for
14        for all $\mathcal{L}_i \in \mathcal{C}$
15            Find $\mathcal{L}_i$ for which $|COV(\mathcal{L}_i \cup \mathcal{L}_k)|$ is maximum
16        end for
17        Merge $\mathcal{L}_i$ with $\mathcal{L}_k$
18        if$COV(\mathcal{L}_i \cup \mathcal{L}_k) \geq th$
19          $\mathcal{P} \leftarrow \mathcal{P} \cup \{\mathcal{L}_i \cup \mathcal{L}_k\}$
20          $t \leftarrow t - 1$
21        end if
22    if $t = 1$
```

```
        goto 1
    end if
```

Complexity Analysis

Let k be the number of unsuccessful partitions which could not attain the required coverage. Let n_i be the number of sensor nodes in i^{th} partition. If the sensor nodes in any partition is not connected to any other partition then it will not be checked further. In the worst case the partition with minimum coverage is connected to at least one sensor in other partition. In that case number of comparison will be maximum. Below we will consider the later case.

Let p^{th} partition gives the minimum coverage. Then according to the algorithm we will try to merge this partition with one of the remaining partitions. Then n_p sensors are compared with all n_j sensors, $1 \leq j \leq k-1$ to check connectivity.
So number of comparisons for connectivity checking = $\sum_{j=1,j\neq p}^{k} n_j \cdot n_k$
To find the union of coverage, it will take constant amount of time. So for only one merging it will take $\sum_{j=1,j\neq p}^{k} n_j \cdot n_k$ + O(1).
Let k^{th} partition is merged with, say, l^{th} partition and m^{th} partition gives the minimum coverage after merging. So merging will take

$$\sum_{j=1,j\neq m,l,p}^{k} n_j \cdot n_m + (n_p + n_l) \cdot n_m \quad = \quad \sum_{j=1,j\neq m}^{k} n_j \cdot n_m$$

If at the last t^{th} partition is merged to l^{th} partition then total computation complexity will be

$$\begin{aligned}
T(n) &= \sum_{j=1,j\neq p}^{k} n_j \cdot n_p + \sum_{j=1,j\neq m}^{k} n_j \cdot n_m + ... + \sum_{j=1,j\neq t}^{k} n_j \cdot n_t \\
&= (\sum_{j=1}^{k} n_j - n_p) \cdot n_p + (\sum_{j=1}^{k} n_j - n_m) \cdot n_m + ... + (\sum_{j=1}^{k} n_j - n_t) \cdot n_t \\
&= \sum_{j=1}^{k} n_j (n_p + n_m + ... + n_t) - ({n_p}^2 + {n_m}^2 + ... + {n_t}^2) \\
&< \sum_{j=1}^{k} n_j \cdot (n_p + n_m + ... + n_t) \\
&\leq n \cdot (n_p + n_m + ... + n_t) \\
&\leq n \cdot n \\
&= n^2
\end{aligned}$$

So the complexity of merging will be

$$\begin{aligned} T(n) &= n^2 + k \cdot O(1) \\ &= O(n^2) \end{aligned}$$

3.3.4 Distributed Algorithm

Since the sensor networks are self-organized, to accumulate the global information about the whole network to a single node, sink, will incur much overhead in terms of communication and time. Hence distributed algorithms are needed, where nodes can compute based on their local information only.

Algorithm Details

In the proposed distributed algorithm, we assume that each sensor node has the information about its 1-hop neighbors (adjacent nodes in the communication graph) only the node-id's and their coverage. Let the number of nodes deployed be n. Then $t \cdot n$ number of leaders are selected randomly, $0 < t < 1$. Each leader node L_i initiates to grow a cover $\mathcal{C}(L_i)$ around it. At each iteration, all the nodes of a cover $\mathcal{C}(L_i)$ selects a 1-hop neighbor that maximizes the coverage of $\mathcal{C}(L_i)$ and forwards the information to L_i. Among all these nodes L_i selects the one resulting maximum coverage and broadcasts it. Each leader terminates when its partition results the required coverage(th), or no connected neighbor of the partition is left to be included. Finally successful leaders forward their partition to the sink.

Following three types of messages are broadcast in the network for communication.

- *selected message*: If a node v is selected in any partition, it broadcasts the message to its 1-hop neighbors and v is deleted from all of its neighbors' *List*.
- *include(L_i, k) message* : Node k is to be included in partition $\mathcal{C}(L_i)$.
- *max-nbr message*: If a node-k gives the maximum coverage at any step, k's 1-hop neighbor already selected in a partition $\mathcal{C}(L_i)$ sends this message to its parent node in $\mathcal{C}(L_i)$.

Algorithm 4
Input: 1-hop neighbor list of each node-i: *List*, Coverage Threshold: th, Leader percent: t, coverage of each 1-hop neighbor $COV(\{i\})$
Output: Partitions $\mathcal{C}(L_i)$ from leaders L_i at sink node

For each node-i,

```
Step 1   Initialize $select \leftarrow 0$, $L \leftarrow 0$
Step 2   Generate a random number $r$
         if $r < t$, $select \leftarrow 1$, $L \leftarrow 1$, $\mathcal{C}(i) \leftarrow \{i\}$, $L_i \leftarrow i$, $\mathcal{C}(L_i) \leftarrow \{i\}$, $COV(\mathcal{C}(L_i)) \leftarrow COV(\{i\})$
         Broadcast selected message endif
```

```
Step 3    if $select = 0$ and $L = 0$ wait and listen to messages from 1-hop neighbors endif
Step 4    if receives selected message from $j \in List$, $List \leftarrow List \backslash \{j\}$
Step 5    if receives $include(L_i, k)$ message from node-$j$
            if $k = i$, $parent(i) \leftarrow j$, $select \leftarrow 1$, broadcast selected message
            endif
              if $k \in List$, $List \leftarrow List \backslash \{k\}$
                if $select = 1$ and be in the same cover; forward include message to $k$; endif
                Find max-nbr(i) $\in List$ that maximizes $COV(\mathcal{C}(L_i))$;
                  if $List = \emptyset$, max-nbr-i$= -1$; endif
                Forward max-nbr message to $parent(i) \neq \emptyset$
              endif
            endif
          endif
Step 6    if   receives max-nbr message from $j$, forward to $parent(i) \neq \emptyset$
            if   $parent(i) = \emptyset$, on receiving max-nbr messages from $j$,  $\forall j \in \mathcal{C}$,
              if   max-nbr(j)$= -1$  $\forall j \in \mathcal{C}$, terminate
              else   find node $MAX$ from $\{$max-nbr(j)$, \forall j \in \mathcal{C}\}$ that maximizes $COV(\mathcal{C}(L_i))$
                $\mathcal{C} \leftarrow \mathcal{C} \cup MAX$, $COV(\mathcal{C}(L_i)) \leftarrow COV(\mathcal{C}(L_i)) \cup COV(\{k\})$

                  if   $|COV| \geq th$, send $\mathcal{C}$ to sink and terminate
                  else   broadcast $include(L_i, MAX)$ to $j$,  $\forall j \in \mathcal{C}$
                  endif
              endif
            endif
          endif
Step 7    return $\mathcal{C}(L_i)$
```

Proof of Correctness

In order to prove that the above algorithm is correct we need to prove that each partition is a connected disjoint set covering $\mathcal{Q}$ and the process terminates.

From Step 5 it is clear that the algorithm will terminate certainly since every time a node is selected in any partition it is deleted from its 1-hop neighbors' *Lists* and is no more considered in further investigations. The process continues till a partition attains the required coverage or the *List* of the nodes included in a partition is empty. Also for the same reason the partition will be disjoint. Connectedness within the partitions is obvious from the fact that in each iteration, only a node adjacent to any node in the partition is included in it. Finally, the successful leaders achieving a partition with desired coverage reports at the sink.

3.4 Summary

In this chapter we have proposed 3 centralized version of the algorithm to find connected set cover partitions and one distributed version of the algorithm. In

Algorithm 1, we have chosen one arbitrary sensor node and the partition is grown around that node to achieve the required coverage. The worst case complexity of the algorithm is O(n^3). *Algorithm 2* starts with maximum number of partitions for 100% coverage, where initialization of each partition is done by the sensor nodes covering the minimally covered area. Though the worst case complexity is O(n^3), it depends on the number of initial partitions. *Algorithm 3* improves the number of partitions achieved by *Algorithm 2* by merging the unsuccessful partitions. Next, we have proposed a distributed version of the algorithm assuming that each sensor node has the information of its one-hop neighbors only.

Chapter 4

Performance Evaluation

In this chapter we are going to discuss the results of our simulation that we ran to compare the performance of the proposed algorithms described in the previous chapter. We ran our algorithms on randomly generated sensor networks wherein a certain number of sensor nodes are placed randomly in an area of 100×100 unit square. Each sensor has a uniform sensing radius and transmission radius. Below we present the comparison of the various algorithms discussed in the previous chapter viz. three centralized algorithms *Algorithm 1*, *Algorithm 2*, *Algorithm 3* and the distributed algorithm.

4.1 Simulation Model

We consider stationary sensor networks in a two dimensional field and assume that the sensors are deployed randomly on a 100×100 unit square grid where a sensor is allowed only at grid points, and the unit cells of the grid are considered to be the elementary areas. It is assumed that each sensor covers an $S \times S$ sub-grid around it and is adjacent to all nodes within a $T \times T$ sub-grid around it. The sub-grids are in fact the maximum squares inscribed within the circles with radius $s/\sqrt{2}$ and $T/\sqrt{2}$ respectively.

Conventionally, for a sensor node with sensing range S, it is assumed that, a circular region of radius S around it is actually covered. Similarly, a node with transmission range T is connected to all sensors lying within a circle of radius T around it. Here, its is assumed that instead of a circular region, the area of influence is limited within a maximum square inscribed in the circle of radius $s/\sqrt{2}$ and $T/\sqrt{2}$ respectively. Hence, by knowing the location of a sensor in the grid, its connectivity and coverage in terms of elementary grid size can be determined easily.

4.2 Simulation Results

For the simulation study n sensor nodes are distributed randomly over a 100×100 grid with given values of T and S, algorithms are executed for $p\%$ coverage of the whole grid. Results are reported by varying n, T, S and p respectively. For each point on the graphs, in fact 20 random distributions of sensors are considered and the average value is plotted.

Fig.s 3-5 show the variation of number of partitions varying the number of nodes n from 100 to 900 for 100%, 90% and 80% coverage requirements respectively. All three algorithms show comparable performance. The variation of number of partitions with p, the percentage coverage is shown in Fig. 6. It shows that in case of distributed algorithms the number of partitions remain almost constant, whereas for both algorithms 1 and 2 it decreases with increase in p, which is self explanatory. Fig.s 7-8 show the variation of number of partitions with sensing range and transmission range respectively for $n = 500$ and $p = 90\%$.

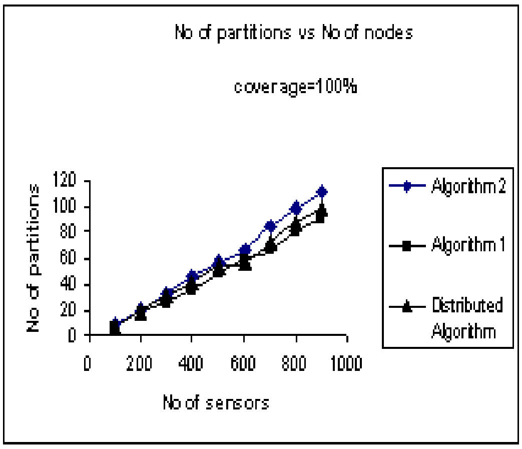


Fig. 3

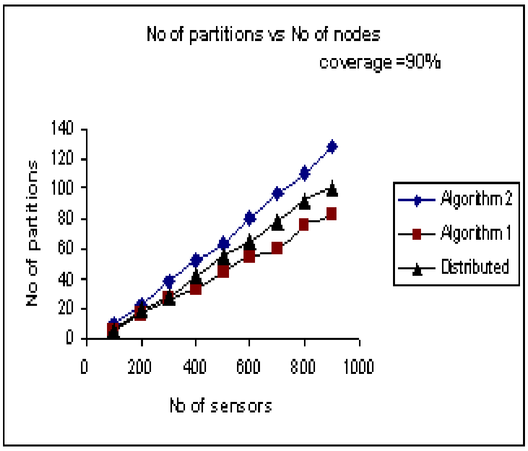


Fig. 4

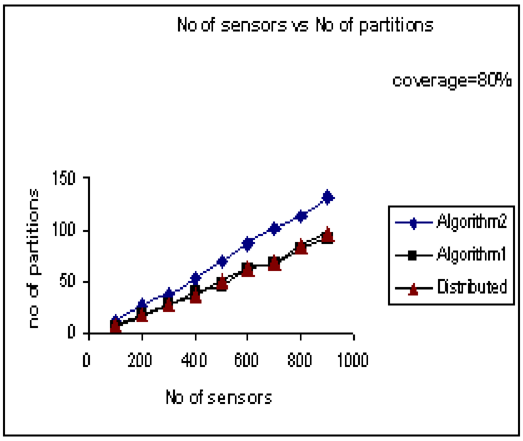


Fig. 5

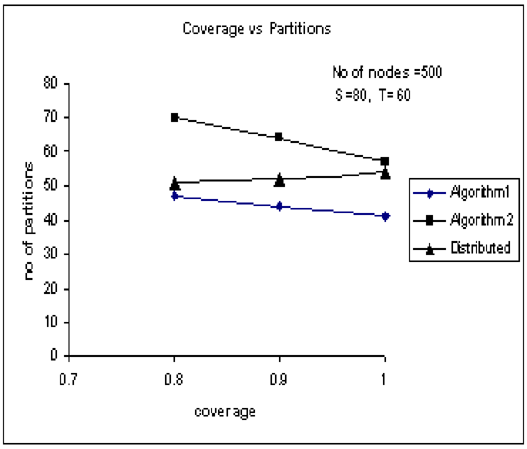


Fig. 6

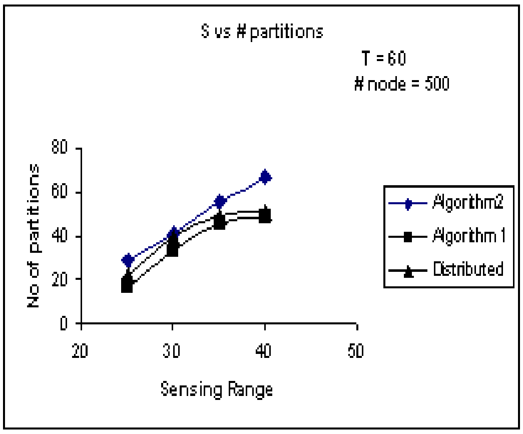


Fig. 7

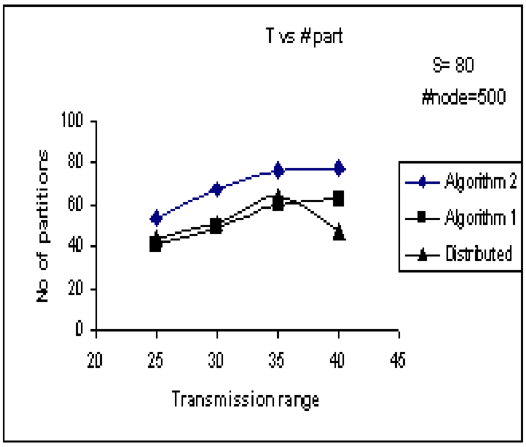


Fig. 8

Similarly, we have simulated three algorithms with $S = 50$ and $T = 40$. Fig.s 9-11 show the variation of number of partitions varying the number of nodes n from 100,150,..,300 for 100%, 90% and 80% coverage requirements respectively. The variation of number of partitions wit the percentage coverage is shown in Fig. 12. Fig.s 13 - 16 show the variation of number of partitions with sensing range and transmission range respectively for $n = 200$ and $p = 90\%$. Here, leader selection threshold is taken as 0.95.

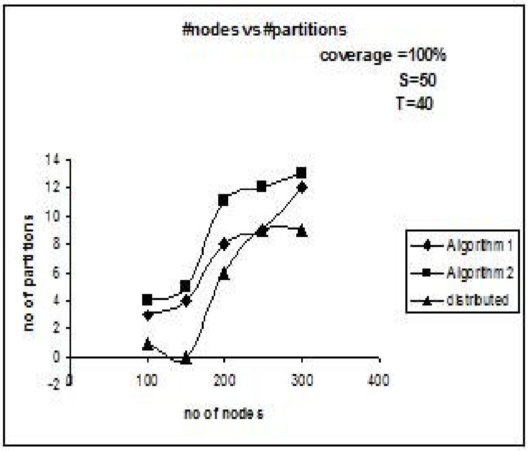


Fig. 9

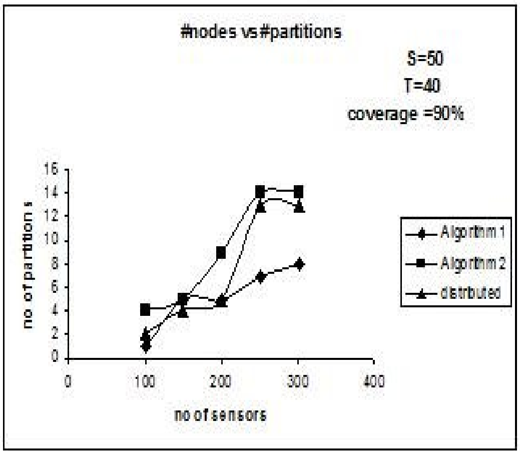


Fig. 10

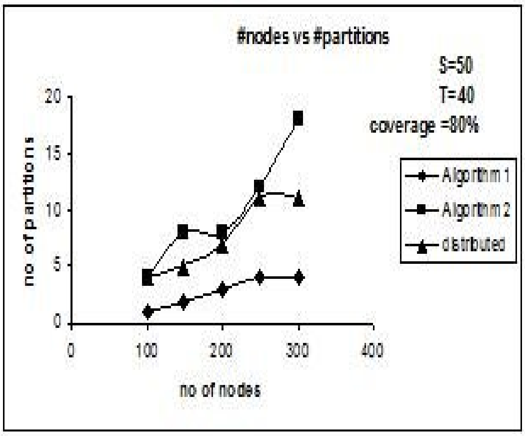


Fig. 11

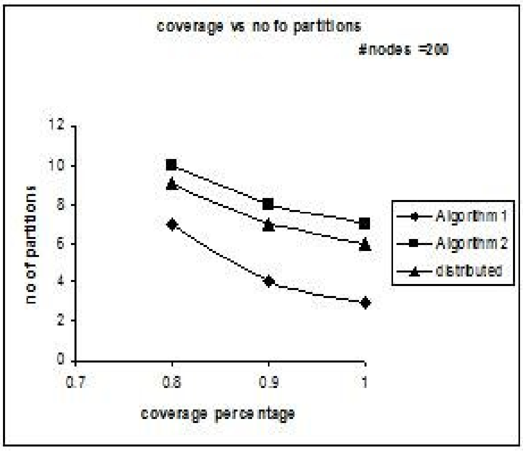


Fig. 12

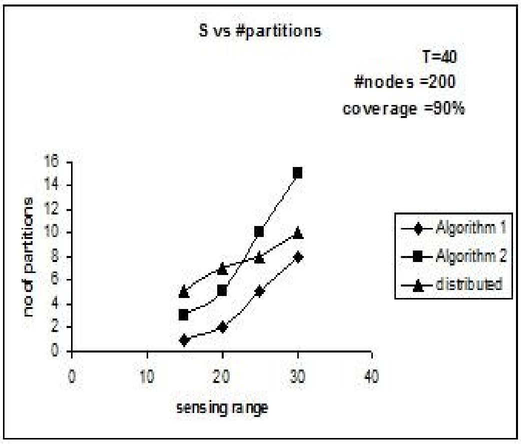


Fig. 13

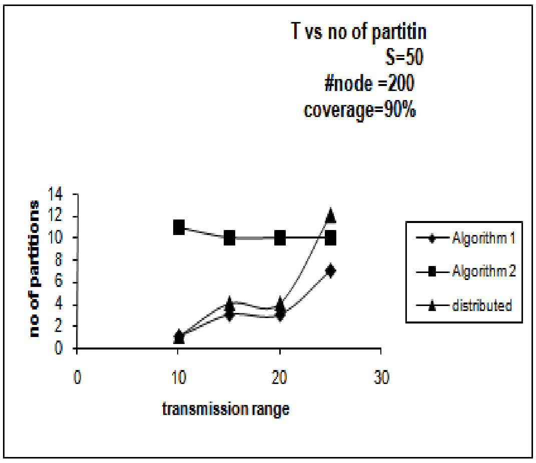


Fig. 14

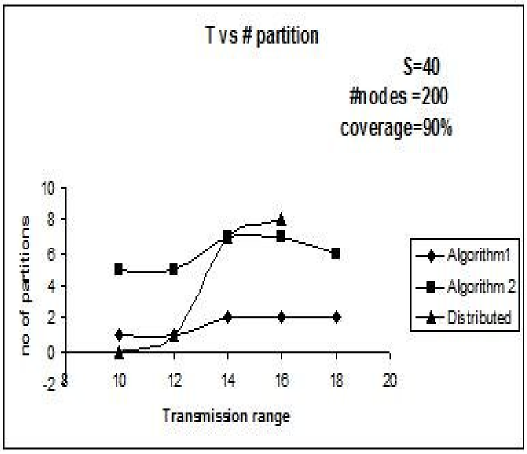


Fig. 15

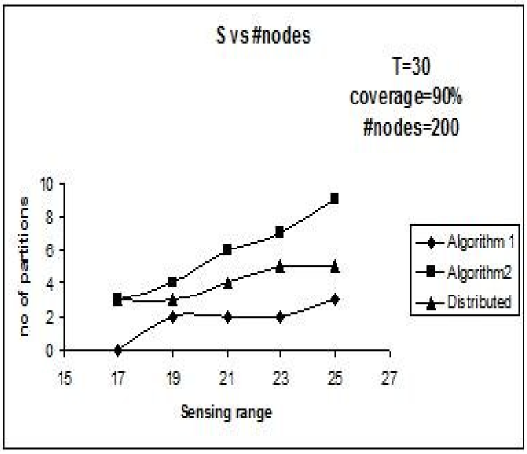


Fig. 16

If the leader selection threshold is less, in other words, leader selection probability is more, then the number of partitions given by distributed algorithm will be lesser. Since, we are considering the connectedness among the nodes in a partition, it will be highly probable that the leader nodes will not be able to satisfy the coverage requirement. Below, in Fig. 17, the variation of the number of partitions with number of nodes with leader selection threshold as 0.90 is shown.

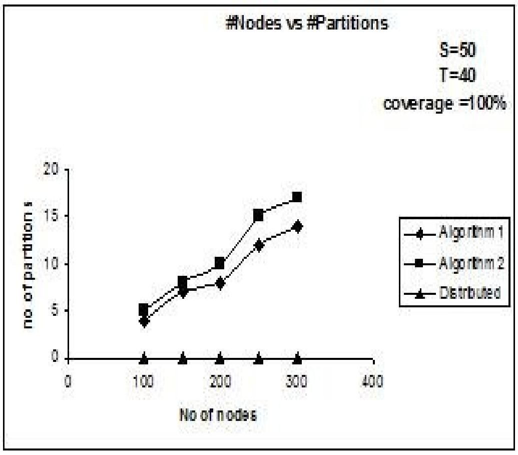


Fig. 17

Next, we have simulated to compare *Algorithm 3* with *Algorithm 2*. From Fig. 18, it can be easily seen that *Algorithm 3* practically can improve the number of partitions achieved by *Algorithm 2*. Here, we have shown the maximum partitions possible for 100% coverage, number of partitions returned by *Algorithm 2* and improvement by *Algorithm 3*.

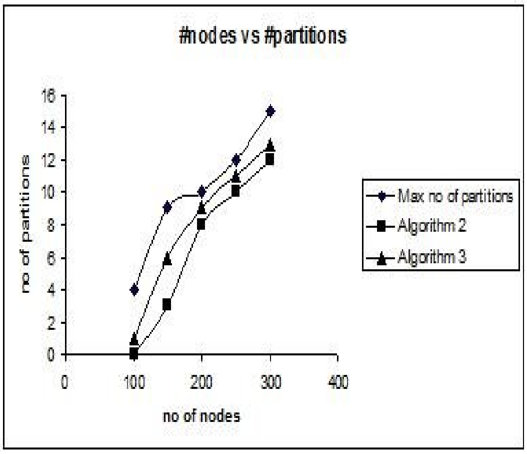


Fig. 18

4.3 Summary

In this chapter, we have compared the performance of all the algorithms proposed in chapter 3. In summary, the simulation study reveals the interesting fact that the performance of the distributed algorithm is comparable with the centralized ones, *Algorithm 2* shows the best overall performance. By *Algorithm 3* we can improve the number of partitions achieved by *Algorithm 2* by merging the remaining partitions. If it is not possible to improve the number of partitions then, at least the sensors used in those partitions will be allocated to some successful partition and hence, the coverage given by those partitions will be improved.

Chapter 5

Conclusion

Wireless sensor networks enable efficient monitoring of physical environments. The important operating constraint is available energy. In order to maximize efficient use of battery power in sensor nodes, we have addressed the connected coverage problem of selecting maximum number of mutually exclusive connected sets of sensor nodes with required coverage for a given query region. The idea is to keep one such set of sensors active at a time to provide the necessary coverage and connectivity and all such sets are activated in a round-robin fashion. So if we obtain k such covers, and one set is activated every k times, k-time enhancement can be achieved in the life time of the sensor network. We have designed two centralized greedy algorithms and also a distributed algorithm. It is interesting to see, that the performance of distributed algorithm is almost comparable with the centralized ones for the given networks under study, which is the most suitable one for the self organizing sensor networks.

Bibliography

[1] Zo Abrams, Ashish Goel, and Serge Plotkin. Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of IPSN,04*, pages 424–432, 2004.

[2] Wei An, Fang-Ming Shao, and Huajun Meng. The coverage-control optimization in sensor network subject to sensing area. *Comput. Math. Appl.*, 57(4):529–539, 2009.

[3] Chi fu Huang and Yu-Chee Tseng. The coverage problem in a wireless sensor network. pages 115–121. ACM Press, 2003.

[4] Chong Liu, Kui Wu, Yang Xiao, and Bo Sun. Random coverage with guaranteed connectivity: Joint scheduling for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(6):562–575, 2006.

[5] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM*, pages 1380–1387, 2001.

[6] Paolo Santi. *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, 2 edition, 2005.

[7] Sanjay Shakkottai, R. Srikant, and Ness Shroff. Unreliable sensor grids: Coverage, connectivity and diameter. In *Proceedings of IEEE INFOCOM*, pages 1073–1083, 2003.

[8] M.M. Singh, M.P. Gore. A solution to sensor network coverage problem. In *2005 IEEE International Conference on Personal Wireless Communications(ICPWC 2005)*, pages 77– 80, 2007.

[9] Sasa Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE International Conference on Communications*, pages 472–476, 2001.

[10] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *SenSys '03: Proceedings of the 1st international*

conference on Embedded networked sensor systems, pages 28–39, New York, NY, USA, 2003. ACM.

[11] Li-Hsing Yen, Chang Wu Yu, and Yang-Min Cheng. Expected -coverage in wireless sensor networks. *Ad Hoc Networks*, 4(5):636–650, 2006.

[12] H. Zongheng Zhou Das, S. Gupta. Connected k-coverage problem in sensor networks. In *13th International Conference on Computer Communications and Networks,2004(ICCCN 2004)*, pages 373–378, 2004.