A STUDY OF SCHEDULING AND DEPLOYMENT ALGORITHMS FOR COVERAGE PROBLEM IN WIRELESS SENSOR NETWORKS

A thesis submitted during 2012 to the University of Hyderabad in partial fulfillment of the award of a Ph.D. degree in Computer Science

by

S. MINI

DEPARTMENT OF COMPUTER AND INFORMATION SCIENCES
SCHOOL OF MATHEMATICS & COMPUTER/INFORMATION SCIENCES
UNIVERSITY OF HYDERABAD
(P.O.) CENTRAL UNIVERSITY
HYDERABAD - 500 046, INDIA

MARCH-2012

CERTIFICATE

This is to certify that the thesis entitled "**A STUDY OF SCHEDULING AND DEPLOYMENT ALGORITHMS FOR COVERAGE PROBLEM IN WIRELESS SENSOR NETWORKS**" submitted by **S. Mini** bearing Reg. No. 07MCPC13 in partial fulfillment of the requirements for the award of **Doctor of Philosophy in Computer Science** is a bonafide work carried out by her under my supervision and guidance.

The thesis has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

Dr. Siba K. Udgata
Supervisor
Department of Computer and
Information Sciences
University of Hyderabad

Prof. P.N. Girija
Head
Department of Computer and
Information Sciences
University of Hyderabad

Prof. T. Amaranath
Dean
School of Mathematics &
Computer/Information Sciences
University of Hyderabad

DECLARATION

I, **S. Mini**, hereby declare that this thesis entitled "**A STUDY OF SCHEDULING AND DEPLOYMENT ALGORITHMS FOR COVERAGE PROBLEM IN WIRELESS SENSOR NETWORKS**" submitted by me under the guidance and supervision of **Dr. Siba K. Udgata** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date:

Name: **S. Mini**

Signature of the Student:

Regd. No. **07MCPC13**

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Siba K. Udgata, whose enthusiasm, patience, kindness, as well as his academic experience, have been invaluable to me. He has been a great advisor and mentor. I feel fortunate to have got a chance to work under his dedicated guidance.

Thanks to Prof. T. Amaranath, Dean, MCIS and Prof. P.N. Girija, Head, DCIS for providing me the facilities to do research work. I would also like to thank former Heads of the Department Prof. Arun Agarwal and Prof. C.R. Rao.

I sincerely thank Prof. Arun K. Pujari for not only giving me the opportunity to work in his laboratory but also for being a great motivator. I am grateful to Dr. Samrat L. Sabat for his help and fruitful discussions at various stages of this research.

My thanks to Prof. Hrushikesha Mohanty, Dr. S. Durga Bhavani and Dr. Alok Singh for their time, insightful comments, and commitment as members of my doctoral committee. Thanks to Dr. Vineet Nair for his support. I am grateful to the AI lab staff and DCIS office staff for their constant help.

I am obliged to Dr. K. Bommanna Raja, my M.E. project supervisor, for his inspiring lectures and conversations, and for motivating me to pursue research in computer science.

No words can describe my gratitude to my parents and brother who have been constantly supportive and encouraging. I am forever indebted to them. Special thanks to Dr. Ajith Nair who has greatly influenced this thesis.

The informal support and encouragement of many friends has been indispensable, and I would like particularly to acknowledge the contribution of Regina, Saipriya and Yashaswini who have been with me from the very beginning of my research. They have made my stay much more enjoyable. Thanks to Monisha for her support. Special thanks to my friends Sushanth, Gayathri, Indu, Poorna, Vinija, Deepa, Usha, Susmitha, Sreejith and Santhosh.

My heartfelt gratitude to Mamata Mam, Sandeepan and Shreeya. Also I would like to thank Sumagna, Hla, Ja, Subrat, Sapna, Uma, Sandhya, Rusydi, Sreedhar, Amer, Supriya, and Shyam for their help. My thanks to all my labmates for their company and co-operation over the years. I would specially thank Deepak, Alefiah, Moumita, Komal, Kiran, Gouri, Ninjashree, Vasavi, Swamy, Sridhar and Kiranmayi for all the good times. A note of thanks to Sruti, Sudhansu and Swathi for their help and support.

I am thankful to the University of Hyderabad and then the Department of Science and Technology, Govt. of India for providing me with fellowship to pursue this research.

Most of all, I thank the Almighty for blessing me with the strength, light and wisdom to pursue my work.

March 2012 S. Mini

ABSTRACT

Network lifetime maximization and quality of coverage are of paramount interest in wireless sensor networks. This is because sensor nodes are battery powered and it is essential to use the available energy optimally. Minimizing energy consumption will lead to maximization of the network lifetime. Depending on the application, energy can be efficiently used through good energy utilization methods. Sensor node deployment and sensor scheduling are two main problems which concern optimal energy use. We address sensor scheduling and sensor node deployment for target coverage problem in wireless sensor networks. Each application will have a different coverage requirement. Keeping this coverage requirement as the main objective, we identify methods to reduce energy consumption for enhancing the network lifetime and improving the quality of coverage.

This thesis first looks at random deployment of sensor nodes. Random deployments are usually dense. This may be the only way of deployment for some harsh environments such as a battle field or a disaster region. A scheduling mechanism which activates only a subset of sensor nodes at a time that is just enough to satisfy the required coverage level and the remaining nodes are set to sleep for conserving energy is initially proposed. This, in turn, prolongs the lifetime of the sensor network. The proposed heuristic performs better than the existing heuristics Greedy-Maximum Set Covers (Greedy-MSC) and High Energy and Small Lifetime (HESL). Then a scheduling scheme for M-connected coverage problem is proposed which requires the subset of sensor nodes to be M-connected and to satisfy the required coverage. This leads to a higher network lifetime and reliable data delivery. The proposed heuristic performs better than Communication Weighted Greedy Cover (CWGC). These scheduling mechanisms will be useful only for random dense sensor node deployments where each sensor node has a fixed sensing range.

If the application permits deterministic deployment of nodes and if the sensor nodes

are limited, quality of sensing and energy conservation can be enhanced by restricting the sensing range requirement. Sensing models can be categorized as binary sensing model or probabilistic sensing model. In a binary sensing model, the target is either fully monitored or not monitored. A probabilistic coverage model considers the effect of distance and medium on the sensing ability of a node. A method for optimal deployment of sensor nodes such that the required sensing range is minimum for both binary and probabilistic sensing models is also presented in this thesis. Artificial Bee Colony (ABC) algorithm is used to compute the optimal deployment locations. A comparison with Particle Swarm Optimization (PSO) shows that ABC algorithm performs better than PSO for the given problem.

Finally the thesis gives a more general approach where a single round deterministic deployment of sensor nodes with fixed sensing range is permitted. A heuristic is initially proposed to solve this problem. The heuristic performs better than the random sensor node deployment. Computing optimal deployment locations using ABC algorithm outperforms the heuristic. The upper bound of network lifetime for a given region with sensor nodes monitoring targets can be theoretically computed. This helps in identifying the optimal positions wherein the network lifetime would be maximum. A scheduling mechanism is then used to schedule these nodes so that this theoretical upper bound of network lifetime can be achieved.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

WSN	Wireless Sensor Network
MANET	Mobile Ad hoc NETwork
GPS	Global Positioning System
MC-MIP	Maximum Covers using Mixed Integer Programming
LP-MSC	Linear Programming-Maximum Set Covers
Greedy-MSC	Greedy-Maximum Set Covers
HESL	High Energy and Small Lifetime
CTC	Connected Target Coverage
CWGC	Communication Weighted Greedy Cover
LEACH	Low-Energy Adaptive Clustering Hierarchy
MTE	Minimum-Transmission-Energy
SMART	Scan-based Movement-Assisted sensoR deploymenT method
ATRI	Adaptive TRIangular deployment
PCA	Probabilistic Coverage Algorithm
GA	Genetic Algorithm
PS-EA	Particle Swarm inspired Evolutionary Algorithm
ORRD	Obstacle-Resistant Robot Deployment
SI	Swarm Intelligence
ACO	Ant Colony Optimization
PSO	Particle Swarm Optimization
ABC	Artificial Bee Colony

NOTATION

S	Set of sensor nodes
T	Set of targets
s_r	Sensing Range
c_r	Communication Range
m	Number of sensor nodes
n	Number of targets
u	Upper bound of network lifetime
p	Probability
ST	Coverage matrix
CT	Connectivity matrix
b	Remaining battery power of sensor nodes
w	Weight of sensor node
E_{min}	Minimum usable energy
e	Energy consumption rate
B	Set of bees
nb	Number of bees
$nlife$	Network lifetime
α	Physical medium characterisitcs
d_{ij}	Distance between sensor node S_i and target T_j
C	Set of sensor covers

CHAPTER 1

INTRODUCTION

1.1 Wireless Sensor Networks

A wireless sensor network consists of a group of sensor nodes that can coordinate their actions through wireless communication and aim at performing tasks such as exploration, surveillance, or monitoring and tracking target points over a specific region (Cassandras and Li, 2005). Sensors can monitor physical or environmental conditions such as temperature, humidity, vehicular movement, lightning condition, pressure, soil makeup, noise levels, the presence or absence of certain kinds of objects, mechanical stress levels on attached objects, and the current characteristics such as speed, direction, and size of an object. Examples of sensors include seismic, infrared, thermal, visual, acoustic and radar. Transmission media, power consumption, scalability, production costs, fault tolerance etc. are some design constraints of a sensor network (Akyildiz *et al.*, 2002).

A similarity of WSNs (Wireless Sensor Networks) and MANETs (Mobile Ad hoc NETworks) is that both involve multihop communication. But they differ in the nature of applications and routing requirements. Since the data being collected by multiple sensors are based on common phenomena, there is likely to be some redundancy in the data being communicated by the various sources in sensor networks (Krishnamachari *et al.*, 2002). Contrary to the nodes in a MANET, nodes in a WSN are battery powered and cannot be recharged or replaced easily (Akyildiz *et al.*, 2002). This makes the power consumption important in WSN, whereas it is of no major concern in a MANET. Mobility rate is also lower in a WSN as compared to a MANET.

There are two types of WSNs: structured and unstructured (Yick *et al.*, 2008). An unstructured WSN is one that contains a dense collection of sensor nodes. Sensor nodes may be randomly deployed into the field. Once deployed, the network is left unattended to perform monitoring and reporting functions. Network maintenance in an unstructured WSN becomes tedious as managing connectivity and detecting failures is difficult due to the presence of large number of nodes. In a structured WSN, all or some of the sensor nodes are deployed in a pre-planned manner. The advantage of a structured network is that fewer nodes can be deployed with lower network maintenance and management cost. Fewer nodes can be deployed now since nodes are placed at specific locations to provide coverage while ad hoc deployment can have uncovered regions.

Since nodes are battery powered, protocols designed for a WSN must be energy efficient. This will maximize the network lifetime. The level of accuracy required for the sensed data varies with the application. A good sensor network protocol design includes low energy, low latency, high accuracy and fault tolerance (Tilak *et al.*, 2002). Energy constraints combined with large deployments pose many challenges to the design and management of WSNs (Al-Karaki and Kamal, 2004). Since some networks need unattended operation for months at a time, energy has to be managed carefully in a WSN (Krishnamachari *et al.*, 2002).

States of a Sensor Node: A sensor node will be in one of the following states (Cassandras and Li, 2005):

- Sensing: The node monitors the region/object using an integrated sensor, digitizes the information, processes it, and stores the data in its onboard buffer. The data will later be sent to the base station.
- Relaying: A relaying node receives data from other nodes and forwards it towards their destination.
- Sleeping: For a sleeping node, most device functions are either shut down or work in low-power mode. A sleeping node does not participate in either sensing or relaying. However, it wakes up from time to time and listens to the communication channel in order to answer requests from other nodes. Upon receiving a request, a state transition to sensing or relaying may occur.
- Dead: A dead node is no longer available to the sensor network. This state might be the result of no energy or some damage. A dead node cannot re-enter any other state.

1.1.1 Applications

WSNs are employed in a vast variety of fields, such as: environmental monitoring (e.g., temperature, humidity), military, monitoring disaster areas providing relief, file exchange, conferencing, home, health (monitoring patients and assisting disabled patients), commercial applications including managing inventory and monitoring product quality.

1.1.2 Research challenges

Energy efficiency, cost, and application requirements are some challenges that the design of a WSN must deal with (Yick *et al.*, 2008). An efficient WSN requires the optimization of both hardware and software. Hardware includes using low cost tiny sensor nodes while software addresses issues such as network lifetime, robustness, self-organization, security, fault tolerance, and middleware. Application requirements vary in terms of computation, storage, and user interface and consequently there is no single platform that can be applied to all applications.

Coverage and Connectivity Issues

The problem of coverage (Gu *et al.*, 2011) determination is to determine if a currently deployed sensor network provides the desired quality of monitoring with its set of currently active sensors. Coverage in a WSN needs to guarantee that the region is monitored with the required degree of reliability. A coverage model of sensor nodes would depend on the distance between the point of interest and the closest node. Therefore, locations of sensor nodes constitute the basic input for the algorithms that examine coverage of the network (Wang *et al.*, 2010). In a deterministic deployment, the required coverage degree affects the number of sensors to be deployed, the placement of these sensors, connectivity, and energy (Yick *et al.*, 2008). Coverage can either mean area coverage or target coverage. Area coverage can further be classified as full coverage or partial coverage. In full coverage, every point in it must be covered by at least one sensor without allowing any uncovered points. But in some applications, partial coverage is sufficient. For example, temperature or pressure sensing in environmental monitoring applications, where reading at one point is adequate for a region since it may have the same readings in its surrounding area. The overall coverage depends highly on the sensing ranges and deployment scheme of the nodes (Poe and Schmitt, 2009). Connectivity scales the adequacy with which the nodes are able to communicate. If a large number of nodes fail due to lack of energy, a part of the network may get completely disconnected from the rest (Jain and Liang, 2005).

Though sensing and communication ranges are generally assumed to be circular discs, it need not be the same for all cases (Chen and Koutsoukos, 2007). If the sensor network is dense, the possibility of the network being connected is more. The network size might shrink due to sensor node failures or because of energy exhaustion. Connectivity depends on the, possibly random, distribution of nodes (Al-Karaki and Kamal, 2004). It becomes an issue if the communication range is limited. The position of the sensor nodes have a direct influence on the coverage and connectivity (Younis and Akkaya, 2008). The k-connectivity problem targets at determining whether every pair of sensors in the targeted area is connected by at least k paths and if not, how additional sensors can be deployed to achieve the goal (Ma and Liu, 2007).

Sensor Node Deployment

Sensor nodes can be deployed randomly or can be placed in pre-computed locations. Some methods of deployment are: dropping from a plane, delivering in an artillery shell, rocket or missile, throwing by a catapult (from a ship board, etc.), placing in factory, and placing one by one either by a human or a robot (Akyildiz *et al.*, 2002). After deployment, topology changes are due to change in sensor nodes' position, reachability (due to jamming, noise, moving obstacles, etc.), available energy, malfunctioning, and task details.

The computation and communication costs associated with deployment make it a complex task (Heo and Varshney, 2005). The choice of the deployment scheme (random/deterministic) depends highly on the type of sensors, application and the environment that the sensors will operate in. In random deployment, the sensor nodes are scattered randomly (Al-Karaki and Kamal, 2004). This is the only way of deployment for harsh environments such as a battle field or a disaster region. In deterministic deployment, the sensors are placed at pre-computed locations. It is necessary when sensors are expensive or when their operation is significantly affected by their position. Deterministic deployment is one of the design optimization strategies. In such case, the coverage of the monitored region can be ensured through careful planning of node densities and fields of view and thus the network topology can be established at setup time. In addition to coverage, the nodes' positions affect numerous network performance metrics such as energy consumption, delay and throughput. For example, large distances between nodes weaken the communication links, lower the throughput and increase the energy consumption (Younis and Akkaya, 2008).

Fault-Tolerance

Sensor nodes are unreliable units that are subject to failures. Sensor nodes have limited power supply based on battery. Although there are different schemes to balance load and energy usage among sensor nodes, certain nodes may still drain out of power too soon and die out. The failed nodes may cause some area in the sensor network uncovered, some neighbor nodes unable to send data packets to the sink, or even cause network partitioned if enough nodes become unavailable (Du and Lin, 2005). Sensors may fail due to the surrounding physical conditions or because they ran short of energy. It may be difficult to replace existing sensors (Tilak *et al.*, 2002). Once a wireless sensor network has been deployed, sensors may fail over time for various reasons such as environmental factors (wind, rain, thunderstorms), battery debonding, reprogramming failures, etc.

Some sensor nodes may fail or be blocked due to lack of power, physical damage, or environmental interference (Al-Karaki and Kamal, 2004). The failure of sensor nodes should not affect the overall task of the sensor network. If many nodes fail, MAC and routing protocols must accommodate formation of new links and routes to the data collection base stations. This may require actively adjusting transmit powers and signaling rates on the existing links to reduce energy consumption, or rerouting packets through regions of the network where more energy is available. Therefore, multiple levels of redundancy may be needed in a fault-tolerant sensor network.

Power Management

To guarantee basic levels of system performance such as connectivity, throughput and delay, power management is essential (Pantazis and Vergados, 2007). Sensor node lifetime shows a strong dependence on battery lifetime. The protocols used in WSNs, including reliability protocols, must be concerned with energy efficiency (Hassanein and Luo, 2006). In a multihop ad hoc sensor network, each node plays the dual role of data originator and data router. The failure of few nodes can cause significant topological changes and might require re-routing of packets and re-organization of the network. Hence, power conservation and power management take on additional importance. In MANETs, power consumption has not been the primary consideration because of replaceable power sources. But in sensor networks, power efficiency directly influences the network lifetime. The main task of a sensor node in a sensor field is to detect events, perform quick local data processing, and then transmit the data. Power consumption can hence be divided into three domains: sensing, communication, and data processing (Akyildiz *et al.*, 2002). It can also be categorized as communication related and non-communication related which includes processing or sensing (Chang and Tassiulas, 2004).

The communication energy is defined as the sum of the energy required to transmit data, using a transceiver (radio), and the energy required for the data processing (to perform encoding and decoding). WSNs should operate with the least possible energy required in order to increase the lifetime of the sensor nodes, ensuring at the same time the network connectivity and availability. The cost of power management is a key issue to validate the effectiveness of a power control scheme. By analyzing the cost of a power control protocol in different aspects, qualitatively or quantitatively, its usefulness and drawbacks can be clearly specified. The operational part of the sensor is determined by the application it supports, e.g. the frequency of data collection, the packet size of the collected data, the duration of measurements, the kind of the sensor node employed and its characteristics (Pantazis and Vergados, 2007). Sensor-specific power management is necessary because the characteristics and energy consumption of sensors differ depending on their hardware architecture and design method (Kim *et al.*, 2008).

Existing power conservation mechanisms (Anastasi *et al.*, 2009) for WSNs may be classified into two main categories : Active and Passive (Pantazis and Vergados, 2007). Active refers to mechanisms that achieve energy conservation by smartly utilizing energy efficient network protocols (by not turning-off the radio interface), while Passive refers to mechanisms that save a node's power by turning-off the radio (transceiver) interface module. Energy-aware QoS in wireless sensor networks will certainly ensure guaranteed bandwidth, or delay, through the duration of connection as well as providing the use of most energy-efficient path. For applications where the system is expected to operate for long durations, energy becomes a severe bottleneck and much effort has been spent on the efficient use of battery energy (Kansal *et al.*, 2007). While shutdown techniques can yield substantial energy savings in idle system states, additional energy savings are possible by optimizing the sensor node performance in the active state (Sinha and Chandrakasan, 2001).

Localization

The basic assumption in many applications is that sensor nodes have to know their positions. For example, the sensed data must combine with location information, for a server instantly to know where an event has happened (Sheu *et al.*, 2010). With location-based routing protocols, both routing and data forwarding are determined from the geographic location. If the positions of sensor nodes can be located more accurately, the data transmission of the network will be more efficient (Liao *et al.*, 2008). The location of a node can be computed by a central unit (the sink node) or in a distributed manner, the latter of which is more common in WSNs (de Oliveira *et al.*, 2009). The problem of estimating spatial-coordinates of the node is referred to as localization (Sichitiu and Ramadurai, 2004; Dulman *et al.*, 2008; Yick *et al.*, 2008). It is important in WSNs since sensor nodes may be randomly deployed. There are several constraints for choosing a localization technique, such as the limited computational power and the restricting environment conditions. A simple solution consists of equipping all nodes with global positioning systems (Mourad *et al.*, 2009). Global Positioning System (GPS) cannot be used in WSNs as GPS can work only outdoors and cannot work in the presence of any obstruction. Moreover, GPS receivers are expensive and not suitable in the construction of small cheap sensor nodes (Al-Karaki and Kamal, 2004; Zhang *et al.*, 2010; Wang and Xu, 2010). Hence, there is a need to develop other means of establishing a coordinate system without relying on an existing infrastructure.

Localization systems for WSNs usually employ a small set of nodes which are aware of their own coordinates (called anchors) which will distribute this information to regular nodes in the network, helping them estimate their own positions (Chen *et al.*, 2008). Some nodes of sensor network should be equipped with a GPS device, while the others get their positions automatically by a localization scheme. In general, the location-aware nodes are called anchor nodes, and the remaining nodes are called normal nodes (Sheu *et al.*, 2010).

Many localization algorithms have been proposed over the past few years. The large body of solutions for the sensor node localization problem can be categorized based on whether the localization techniques are Range-based or Range-free, whether the localization algorithms are Centralized or Distributed, and whether localization results are Deterministic or Probabilistic (Teng *et al.*, 2009). Range based techniques (Chen *et al.*, 2008) require special hardware for estimating the distance between anchors and regular nodes, which may become prohibitively expensive. Range-free techniques, on the other hand, do not impose such demand as an anchor informs other nodes about its own position through message passing. After finishing the distance-from-anchor estimation process, a regular node can determine its own position through a variety of methods, such as multilateration, and triangulation. If necessary, an optional step is performed, in which regular nodes exchange messages among themselves to refine their locations.

Almost all existing localization algorithms consist of two stages (Liu *et al.*, 2010): 1) measuring geographic information from the ground truth of network deployment; 2) computing node locations according to the measured data. Geographic information includes a variety of geometric relationships from coarse-grained neighbor-awareness to fine-grained inter-node ranges (e.g., distance or angle). Based on physical measurements, localization algorithms solve the problem that how the location information from beacon nodes (anchor nodes) spreads network-wide. Generally, the design of localization algorithms largely depends on a wide range of factors, including resource availability, accuracy requirements, and deployment restrictions; and no particular algorithm is an absolute favorite across the spectrum.

Using mobile nodes to assist self-localization of a sparse sensor network is a new research direction (Wu *et al.*, 2007). Sparse sensor networks are deployed due to environment constraints, e.g., range measurements are missing due to obstructions in a room, or to reduce cost by minimizing the number of sensors. In an extreme case, the static sensors do not have range measurements between themselves since they are transmitters or receivers only. For sparse sensor networks, all the existing localization algorithms fail to work properly due to the lack of distance or connectivity data to uniquely calculate the geo-locations.

For static WSNs, once node positions have been determined, they are unlikely to change. On the other hand, mobile sensors must frequently estimate their position, which takes time and energy, and consumes other resources needed by the sensing application (Amundson and Koutsoukos, 2009). Furthermore, localization schemes that provide high accuracy positioning information in WSNs cannot be employed by mobile sensors, because they typically require centralized processing, take too long to run, or make assumptions about the environment or network topology that do not apply to dynamic networks. Localization in multihop environments (Wang *et al.*, 2007) is even more challenging, since nodes are often multiple hops away from anchor nodes, thereby increasing the uncertainty in location.

Routing

Compared to wired networks, wireless networks have limitations such as high cost of transmission, limited processing capabilities, and limited energy resources. Hence the routing approaches that worked well for wired networks might not perform the same way for WSNs (Raicu *et al.*, 2005). The sensor nodes when randomly deployed can be considered as ad hoc networks since there is no architecture or hierarchy in place; though a dynamic organization is possible later.

Routing in WSNs (Al-Karaki and Kamal, 2004) is very challenging due to the inherent characteristics that distinguish these networks from other wireless networks like mobile ad hoc networks or cellular networks. First, due to the relatively large number of sensor nodes, it is not possible to build a global addressing scheme for the deployment of a large number of sensor nodes as the overhead of ID maintenance is high. Thus, traditional IP-based protocols may not be applied to WSNs. Furthermore, sensor nodes that are deployed in an ad hoc manner need to be self-organizing as the ad hoc deployment of these nodes requires the system to form connections and cope with the resultant nodal distribution especially that the operation of the sensor networks is unattended. In WSNs, sometimes getting the data will be more important than knowing the IDs of which nodes sent the data. Second, in contrast to typical communication networks, almost all applications of sensor networks require the flow of sensed data from multiple sources to a particular base station. This, however, does not prevent the flow of data to be in other forms (e.g., multicast or peer to peer). Third, sensor nodes are tightly constrained in terms of energy, processing, and storage capacities. Thus, they require careful resource management.

The routing protocols for WSNs can be classified based on the network structure as flat network routing, hierarchical network routing, and location-based routing (Ming and Wong, 2007). In flat network routing, all nodes have equal functionality and they co-operate to perform the sensing tasks. The hierarchical network routing divides the network into clusters or grids in order to achieve scalability and energy efficiency. In hierarchical routing, nodes maintain a hierarchy of clusters that provides an addressing and routing scheme (Iwanicki and van Steen, 2009). The location-based routing relies on the node positions, which can be obtained from GPS device attached to the sensor to handle the data routing.

Multipath routing improves the intrusion tolerance (Deng *et al.*, 2006). Multipath routing (Ganesan *et al.*, 2001) allows the establishment of multiple paths between source and destination. The benefits of multipath routing are load balancing, whereby the traffic between a source-destination pair is split across multiple paths, and ensuring reliable data delivery.

Since the design requirements of a WSN is application driven, routing protocols which suffices the requirement need to be developed. For example, low latency precision tactical sensor has a different requirement from that of a periodic weather monitoring task. There might be redundant data in a weather monitoring system. Another interesting issue for routing protocols is the consideration of node mobility. Most of the current protocols assume that the sensor nodes and the base station are stationary. However, there might be situations such as battle environments where the base station and possibly the sensors need to be mobile. In such cases, the frequent update of the position of the command node and the sensor nodes and the propagation of that information through the network may excessively drain the energy of nodes (Al-Karaki and Kamal, 2004).

Secure routing (Chang and Tassiulas, 2004) is vital to the acceptance and use of sensor networks for many applications (Karlof and Wagner, 2003). Transmission power consumption is closely coupled with the route selection. Routing protocol design (Mao *et al.*, 2011) for wireless networks is often guided by two essential requirements: minimize energy cost or maximize network throughput. Traditional routing protocols do not take into account that a node contains only a limited energy supply (Schurgers and Srivastava, 2001). Assuming the network lifetime as the time for the first node in the WSN to fail, a perfect routing protocol would slowly and uniformly drain energy among nodes, leading to the death of all nodes nearly at the same time. Typically, an ideal routing protocol would avoid the fast drain of sensor nodes with high energy consumption (Bouabdallah *et al.*, 2009). Depending on the type of sensors used in the network, the detection probability of a sensor may decay with distance, environmental conditions, and hardware configuration (Carter and Ragade, 2009).

1.2 Motivation

Network lifetime and quality of coverage are major factors which measure the efficiency of a wireless sensor network. Energy efficient protocols should be designed so that network lifetime can be maximized. There are different ways by which energy consumption can be controlled. Depending on the application, energy saving mechanisms might vary. For example, energy preservation mechanism for a deterministic deployment of sensor nodes might not be applicable to a random deployment of nodes. It is hence important to identify the coverage requirement, accessibility of the region, method of deployment, availability and nature of sensor nodes, sensing model etc.

The coverage requirement of the application also has an impact on the network lifetime. Some applications will require only few sensor nodes to monitor the targets/location points, whereas some might require a high number of sensor nodes to monitor the targets/location points. Each target/location point in some applications will have a different coverage requirement. This complicates the problem further. Communication of the monitored data/information to the base station also depends on the application.

If the whole area has to be monitored and the number of nodes are limited, optimal deployment patterns will be useful. This may not always be helpful for target coverage problem. Other methods to preserve energy and enhancing quality of coverage, such as restricting sensing range should be taken into account in such cases. Apart from this, sensing models too play a major part in a sensor node deployment. In a binary sensing model, nodes can sense all the location points within its sensing range with full confidence. But in a probabilistic sensing model, environment and distance also affect sensing. If the sensor nodes have pre-specified sensing range, they can either be deployed in a single round or multiple rounds. Multi-round sensor node deployment may incur high cost. So a single round deployment and scheduling would maximize the network lifetime and minimize cost.

The challenge thus opens up with the design of energy preserving mechanisms to enhance network lifetime and to ensure the quality of coverage considering the nature of application, coverage requirement, availability and nature of sensor nodes.

1.3 Contributions of this Thesis

In this thesis, we propose solutions to the following problems in wireless sensor networks:

1. Optimal deployment patterns are useful only if it is possible to place the sensors

where desired. In some deployment scenarios, such as, when deploying in harsh terrains, placing individual sensors at desired locations may not be feasible. We consider random deployment of sensor nodes to achieve target coverage. Random deployments are dense and we propose a heuristic to solve direct base station connected simple, k and Q coverage problems. A sensor scheduling scheme to prolong network lifetime is proposed such that at any time, only minimal number of sensor nodes that are required to satisfy the required coverage are in active state. This heuristic could maximize the network lifetime and meet the theoretical upper bound for all experimented cases. It performs better than recent existing methods.

2. We address M-connected coverage problem for random deployment of sensor nodes. M-connected coverage is required 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 information at other nodes including base station. We propose a heuristic to solve M-connected simple, k and **Q** coverage problems. This heuristic performs better than CWGC method.

3. We also consider a case where the number of sensor nodes is very less and a possibility of deterministic deployment exists. Such deployments are used when the nodes are expensive and/or are limited in number. The sensors must be placed in exact locations since there is only limited number of nodes extracting information. We use Artificial Bee Colony (ABC) algorithm and Particle Swarm Optimization (PSO) to solve this problem so as to minimize the sensing range requirement to achieve simple/k/**Q** coverage for a binary sensing model. This in turn saves energy and improves quality of coverage. ABC algorithm performs better than PSO for this problem.

4. We use artificial bee colony algorithm to solve sensor node deployment problem to achieve probabilistic simple/k/**Q** coverage. We compute the optimal deployment locations such that all the targets are covered with the required minimum probability and with the required level of coverage.

5. Finally we look into a more general model, where the user can deterministically deploy the given sensor nodes (with fixed sensing range) and decide a schedule as of which sensor nodes should be active at what time. Initially we propose a heuristic to

solve sensor node deployment problem. We also use ABC to compute optimal deployment locations of sensors. Computing deployment locations using ABC algorithm is powerful than the heuristic. After computing deployment locations, we use a heuristic to schedule the sensor nodes such that the theoretical upper bound of network lifetime is achieved.

1.4 Thesis Organization

Chapter 2 gives a detailed discussion of coverage problems in wireless sensor networks. Heuristics to solve direct base station connected coverage problem and M-connected coverage problem for all sorts of target coverage requirements are proposed in this chapter. This is applicable to sensor deployments which are dense and random. Scheduling mechanisms are proposed to improve the network lifetime. Chapter 3 discusses sensor deployment problem for binary sensing model as well as probabilistic sensing model. The assumption here is that the number of targets is higher than the number of sensors to be deployed; and deterministic deployment of sensor nodes is permitted. Energy consumption can be reduced by using the deterministic deployment scheme proposed. In Chapter 4, we assume that the sensor deployment is deterministic and the sensor nodes have predefined sensing range. We use an optimal deployment scheme for sensor deployment and then propose scheduling mechanisms so that the theoretical upper bound could be achieved. Conclusions and future directions in Chapter 5 end the thesis.

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

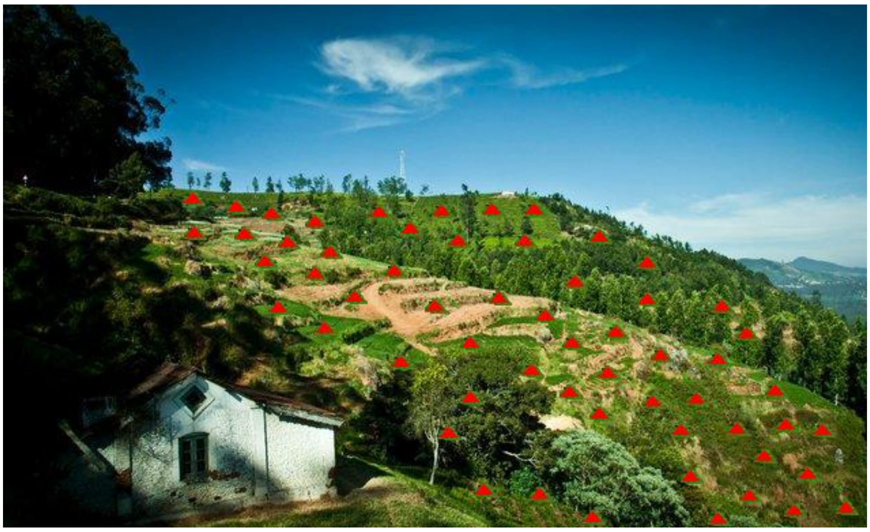


Figure 2.1: Random sensor node deployment

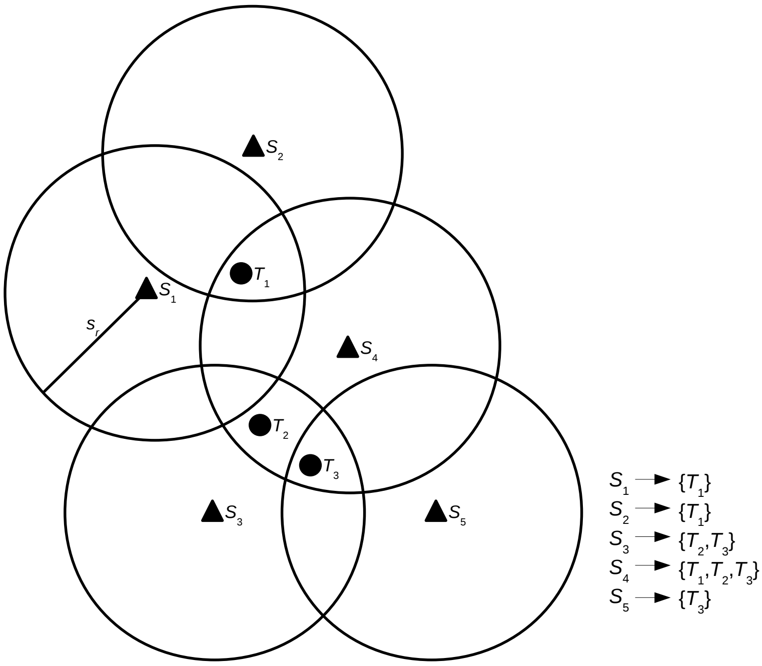


Figure 2.2: Direct Base Station Connected Coverage

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

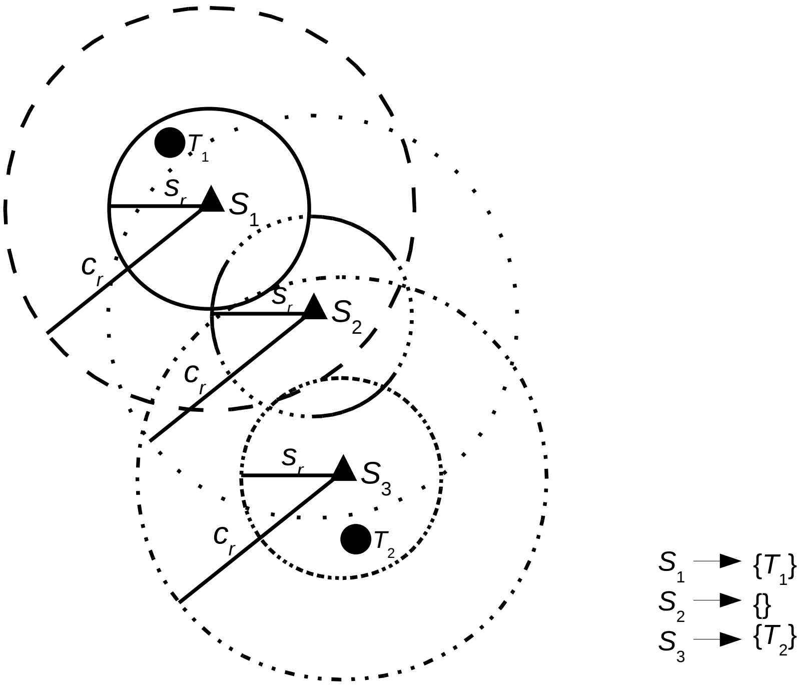


Figure 2.3: Connected Coverage

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 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 factor 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/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/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,

$$ST_{ij} = \begin{cases} 1 & \text{if } S_i \text{ monitors } T_j \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

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,

$$u = min_j \left\lfloor \frac{\frac{\sum_i ST_{ij} * b_i}{q_j}}{e_i} \right\rfloor \tag{2.2}$$

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,

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

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

The connectivity matrix is defined as,

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

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 (2010*a*) show how to achieve energy-efficient k-coverage of a region of interest. Ammari and Das (2010*b*) 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_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

Input: ST, B
Initialize k/**Q**, max_run
for r = 1 to max_run **do**
  for $iteration$ = 1 to $\sum_{i=1}^{m} b_i$ **do**
    if cover possibility exists **then**
      Calculate weight due to remaining energy (according to Equation 2.5)
      Determine cover based on priority (Algorithm 2)
      Optimize cover (Algorithm 3)
      Activate optimized cover and reduce battery power (Algorithm 4)
    else
      $break$
    end if
  end for
  Calculate network lifetime (*nlife*)
  if *nlife* $<$ *u* **then**
    Tune weight deciding component and/or parameter (according to Equation 2.9)
  else
    $break$
  end if
end for

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

$$w_1 = \mathbf{b_i} \tag{2.5}$$

2. *Weight due to covered targets*

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

$$wst_{ij} = \frac{ST_{ij}}{\sum_{i=1}^{m} ST_{ij}} \tag{2.6}$$

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

$$wt_i = \sum_{j=1}^{n} wst_{ij} \tag{2.7}$$

$$w_2 = wt_i \tag{2.8}$$

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

$$w = w_1^{\beta} + w_2^{\gamma} + w_3^{\delta} \tag{2.9}$$

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

Input: Sorted S in descending order of w
Output: Cov_S
Initialize $Cov\_S = \phi$
for i = 1 to m **do**
  if S_i contributes to coverage **then**
    $Cov\_S = Cov\_S \cup \{S_i\}$
  end if
  if coverage requirement met **then**
    $break$;
  end if
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

Input: Cov_S
Output: $Opt.Cov\_S$
Initialize $Opt.Cov\_S = \phi$
for i = $length(Cov\_S)$ down to 1 **do**
    if $Cov\_S - \{S_i.Cov\_S\}$ meets k/**Q** coverage requirement **then**
        $Ignore\ S_i.Cov\_S$
        $Cov\_S = Cov\_S - \{S_i.Cov\_S\}$
    else
        $Opt.Cov\_S = Opt.Cov\_S \cup \{S_i.Cov\_S\}$
    end if
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_{min}. When the battery power reaches E_{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

```
Input: Opt.Cov_S
for i = 1 to length(Opt.Cov_S) do
    S_i.state = true
    decrement b_i
    if b_i ≤ E_min then
        for j = 1 to n do
            M_ij = 0
        end for
    end if
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

Input: Sorted S in descending order of battery power
Output: Cov_S
Initialize $Cov\_S = \phi$
for i = 1 to m **do**
  if S_i contributes to coverage **then**
    $Cov\_S = Cov\_S \cup \{S_i\}$
  end if
  if coverage requirement met **then**
    $break$;
  end if
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 $S_i \in Cov\_S$, $1 \leq i \leq length(Cov\_S)$, 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

Input: Cov_S
Output: $Opt.Cov\_S$
Initialize $Opt.Cov\_S = \phi$
for i = $length(Cov\_S)$ down to 1 **do**
  if $Cov\_S - \{S_i.Cov\_S\}$ meets 1/k/**Q** coverage requirement **then**
    $Cov\_S = Cov\_S - \{S_i.Cov\_S\}$
  else
    $Opt.Cov\_S = Opt.Cov\_S \cup \{S_i.Cov\_S\}$
  end if
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

```
Input: S in descending order of battery power, Cov_S, Opt.Cov_S
Output: M_Connected_Cover
Initialize M_Connected_Cover = φ; flag = 0
Total_S = Opt.Cov_S ∪ {Cov_S − Opt.Cov_S} ∪ {S − Cov_S}
Rem_Nodes = Total_S − Opt.Cov_S
ToCheck = Opt_Cov
if ToCheck M-connected then
   M_Connected_Cover = ToCheck
else
   for i = 1 to length(Rem_Nodes) do
      ToCheck = ToCheck ∪ Rem_Nodes(i)
      if ToCheck M-connected then
         M_Connected_Cover = ToCheck
         break;
      else
         ToCheck1 = ToCheck
         j = 1
         while ToCheck1 ≠ φ do
            M_Connected_Subset = ToCheck1(j) ∪ Nodes in ToCheck satisfying
            M-connectivity starting with ToCheck1(j)
            if M_Connected_Subset meets coverage requirement then
               M_Connected_Cover = M_Connected_Subset
               flag = 1
               break;
            end if
            ToCheck1 = ToCheck1 − M_Connected_Subset
         end while
      end if
      if flag = 1 then
         break;
      end if
   end for
end if
```

M-Connected Cover Optimization

A node $S_i \in M\_Connected\_Cover$, $1 \leq i \leq length(M\_Connected\_Cover)$, represented as $S_i.M\_Connected\_Cover$ will not be added to the optimized cover set

$Opt.M\_Conn\_Cov$ if $M\_Connected\_Cover - \{S_i.M\_Connected\_Cover\}$ is M-connected and meets 1/k/**Q** coverage requirement. M-connected cover optimization algorithm is given in Algorithm 8.

Algorithm 8 M-Connected Cover Optimization

Input: $M\_Connected\_Cover$
Output: $Opt.M\_Conn\_Cov$
Initialize $Opt.M\_Conn\_Cov = \phi$
for i = $length(M\_Connected\_Cover)$ down to 1 **do**
    if $M\_Connected\_Cover - \{S_i.M\_Connected\_Cover\}$ meets k/**Q** coverage requirement and is M-connected **then**
        $Ignore\ S_i.M\_Connected\_Cover$
        $M\_Connected\_Cover = M\_Connected\_Cover - \{S_i.M\_Connected\_Cover\}$
    else
        $Opt.M\_Conn\_Cov = Opt.M\_Conn\_Cov \cup \{S_i.M\_Connected\_Cover\}$
    end if
end for

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_{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.

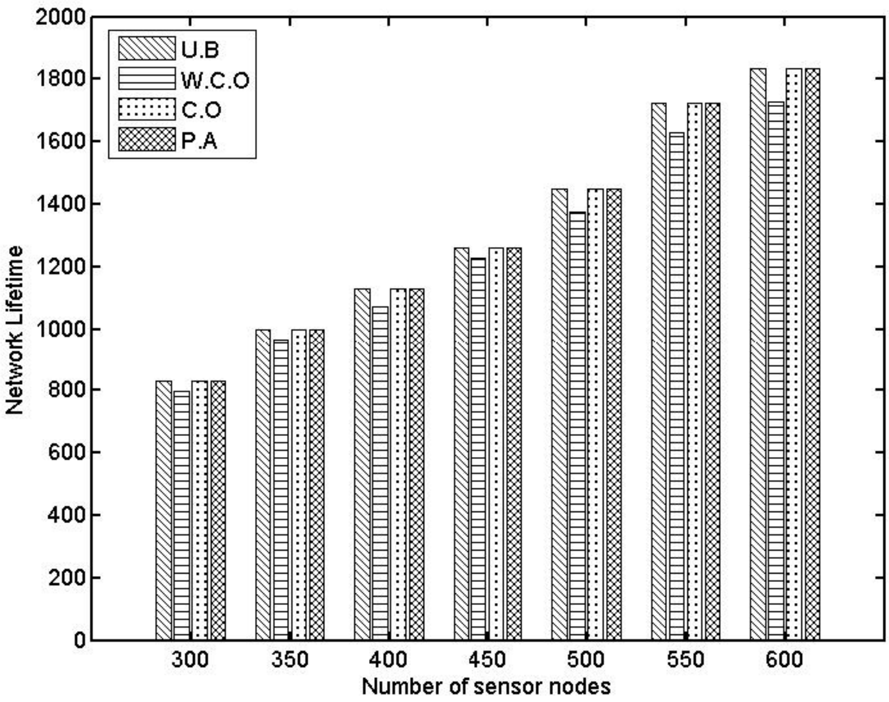


Figure 2.4: Comparison of upper bound (U.B), network lifetime without cover optimization (W.C.O), network lifetime with cover optimization (C.O) using only w_1 and network lifetime using proposed approach (P.A) for simple coverage problem

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 α 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

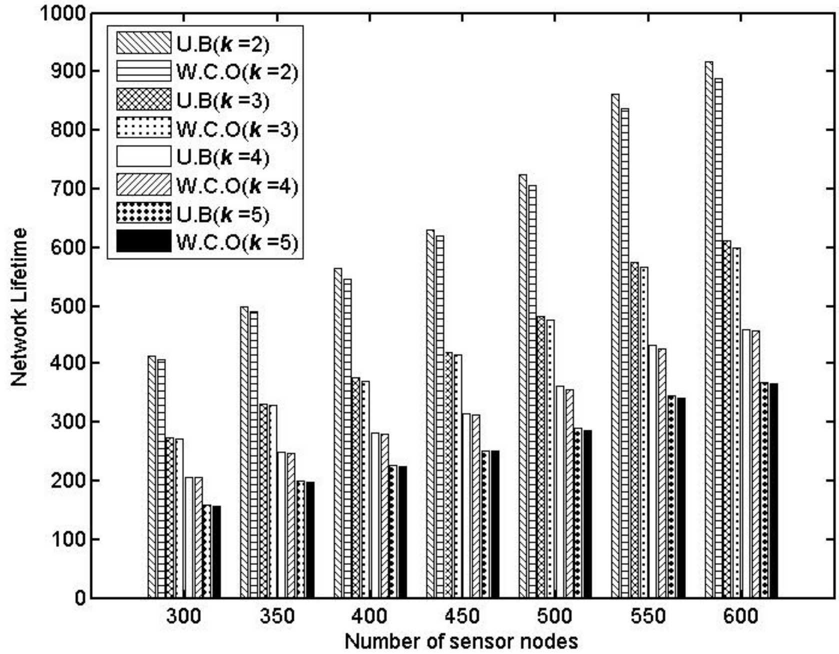


Figure 2.5: Comparison of upper bound and network lifetime without cover optimization for k-coverage problem

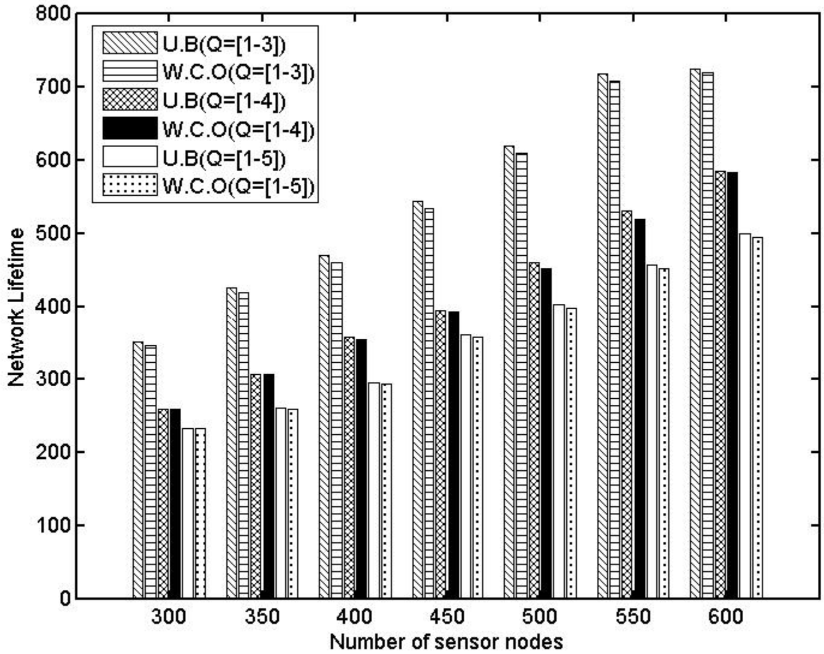


Figure 2.6: Comparison of upper bound and network lifetime without cover optimization for **Q**-coverage problem

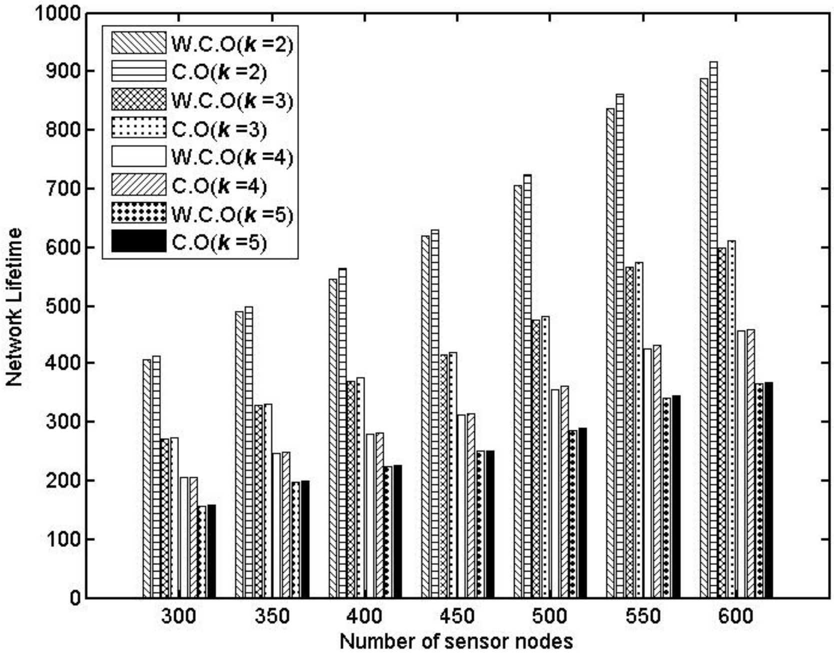


Figure 2.7: Comparison of network lifetime without cover optimization and with cover optimization for k-coverage problem

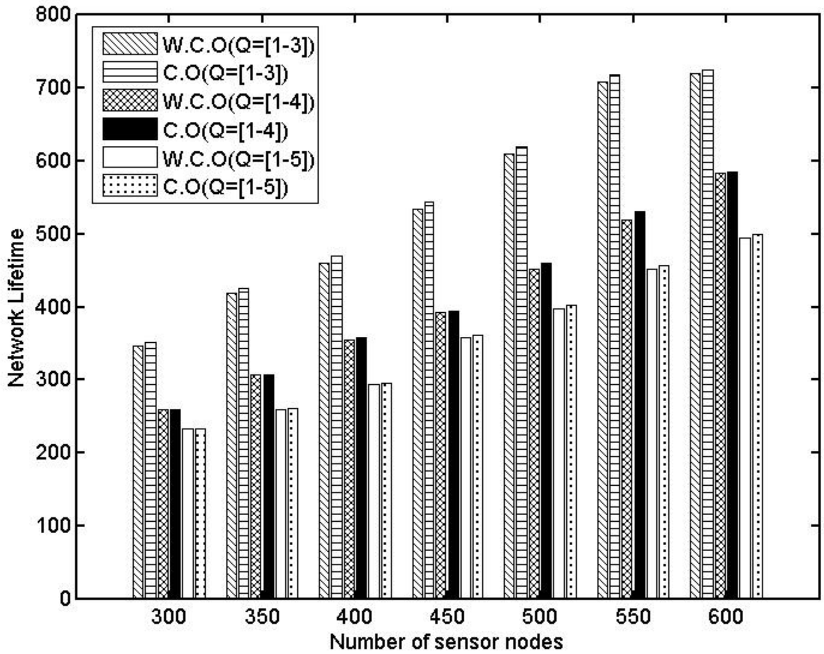


Figure 2.8: Comparison of network lifetime without cover optimization and with cover optimization for **Q**-coverage problem

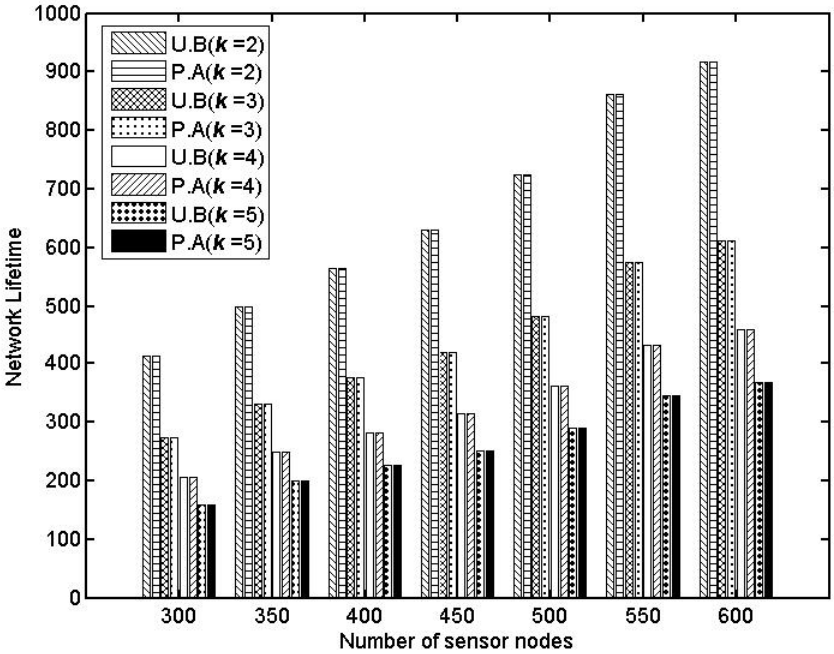


Figure 2.9: Comparison of upper bound and network lifetime using proposed method for k-coverage problem

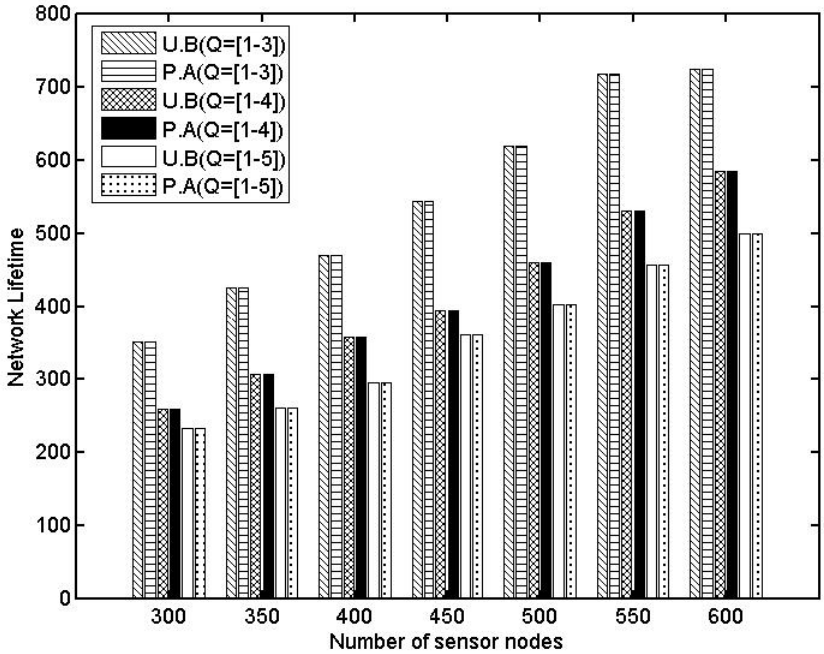


Figure 2.10: Comparison of upper bound and network lifetime using proposed method for **Q**-coverage problem

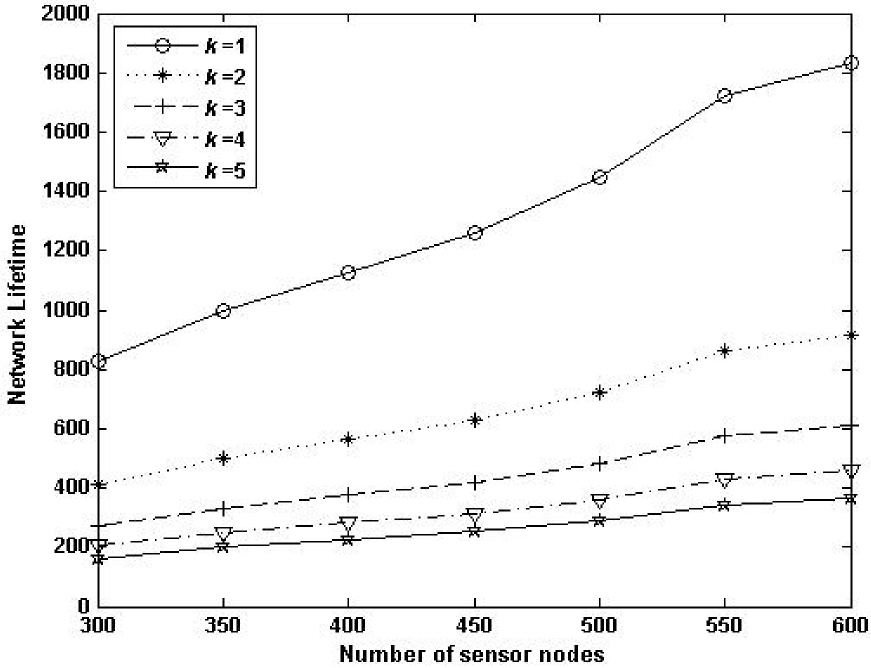


Figure 2.11: Impact of varying number of nodes & varying k for k-coverage problem

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.

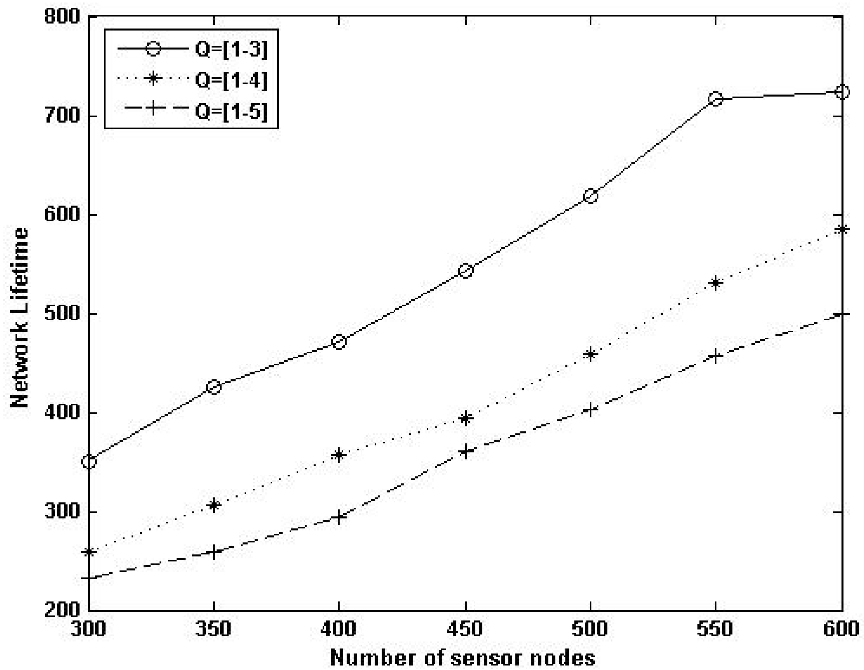


Figure 2.12: Impact of varying number of nodes & varying **Q** for **Q**-coverage problem

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

		k=1				k=2				k=3				k=4				k=5			
			R.B.P[4]				R.B.P[4]				R.B.P[4]				R.B.P[4]				R.B.P[4]		
N.S[1]	N.T[2]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]
300	25	**828**	795.56	827.96	**828**	**412**	407.08	412	**412**	**274.28**	271.52	274.28	**274.28**	**206**	205	206	**206**	**158.4**	157.76	158.4	**158.4**
350	25	**996**	963.48	995.96	**996**	**498**	490.48	497.96	**498**	**331.6**	328.92	331.6	**331.6**	**249**	247.56	249	**249**	**199.2**	198.2	199.2	**199.2**
400	25	**1128**	1071.3	1128	**1128**	**564**	545.72	564	**564**	**375.68**	370.64	375.68	**375.68**	**282**	278.88	282	**282**	**225.6**	223.84	225.6	**225.6**
450	25	**1260**	1227.6	1260	**1260**	**630**	620.44	630	**630**	**419.68**	415.56	419.68	**419.68**	**315**	312.92	315	**315**	**252**	251.08	252	**252**
500	25	**1448**	1375.6	1447.9	**1448**	**724**	705.08	723.92	**724**	**482.36**	474.92	482.32	**482.36**	**362**	355.88	361.92	**362**	**289.6**	285.44	289.56	**289.6**
550	25	**1724**	1627.2	1723.9	**1724**	**862**	837.64	861.92	**862**	**574.36**	566.32	574.36	**574.36**	**431**	425.76	430.96	**431**	**344.8**	341.96	344.8	**344.8**
600	25	**1836**	1727	1835.9	**1836**	**918**	887.52	917.96	**918**	**611.56**	598.84	611.52	**611.56**	**459**	455.4	458.96	**459**	**367.2**	365.28	367.2	**367.2**

[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

		Q=[1-3]				**Q**=[1-4]				**Q**=[1-5]			
			R.B.P[4]				R.B.P[4]				R.B.P[4]		
N.S[1]	N.T[2]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]	U.B[3]	W.C.O[5]	C.O[6]	P.A[7]
300	25	**350.4**	345	350.4	**350.4**	**258.92**	257.72	258.92	**258.92**	**232**	231.52	231.96	**232**
350	25	**425.04**	418.2	424.96	**425.04**	**306.32**	305.32	306.28	**306.32**	**259.32**	258	259.28	**259.32**
400	25	**470.44**	460	470.44	**470.44**	**356.88**	353.08	356.88	**356.88**	**294.72**	293.36	294.72	**294.72**
450	25	**543**	533.52	543	**543**	**393.96**	392.88	393.96	**393.96**	**359.76**	357.2	359.76	**359.76**
500	25	**618.44**	608.28	618.44	**618.44**	**459.56**	451.84	459.48	**459.56**	**402.68**	397.52	402.68	**402.68**
550	25	**716.44**	707	716.44	**716.44**	**529.88**	518.76	529.84	**529.88**	**456.2**	452.4	456.2	**456.2**
600	25	**724.4**	718.24	724.4	**724.4**	**584.56**	582.12	584.56	**584.56**	**498.8**	494.72	498.76	**498.8**

[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

	$k=1$			$k=2$			$k=3$			$k=4$			$k=5$		
N.S	U.B	G-MSC	P.A	U.B	G-MSC	P.A	U.B	G-MSC	P.A	U.B	G-MSC	P.A	U.B	G-MSC	P.A
300	828	817.8	828	412	400.24	412	274.28	256.44	274.28	206	170.88	206	158.4	120.76	158.4
350	996	992.44	996	498	488.68	498	331.6	311.96	331.6	249	226.6	249	199.2	173.84	199.2
400	1128	1113	1128	564	548.56	564	375.68	356.6	375.68	282	266.24	282	225.6	201.16	225.6
450	1260	1244.1	1260	630	618.48	630	419.68	400.52	419.68	315	295.84	315	252	229.36	252
500	1448	1432.5	1448	724	708.6	724	482.36	459.28	482.36	362	338.24	362	289.6	260.88	289.6
550	1724	1713.7	1724	862	850.32	862	574.36	560.96	574.36	431	416.04	431	344.8	327.48	344.8
600	1836	1815.3	1836	918	903.24	918	611.56	591.28	611.56	459	434.32	459	367.2	345.44	367.2

Table 2.4: Comparison of network lifetime using HESL and proposed approach for **Q**-coverage problem

N.T	**Q**=[1-3]			**Q**=[1-4]			**Q**=[1-5]		
	U.B	HESL	P.A	U.B	HESL	P.A	U.B	HESL	P.A
200	327.64	325.96	327.64	245.64	243.92	245.64	210.84	209.32	210.84
250	301	301	301	243.64	241.76	243.64	195.84	192.6	195.84
300	205.32	203.68	205.32	231.28	231	231.28	205.32	203.68	205.32
350	289.76	287.76	289.76	231.64	231	231.64	185.4	184.04	185.4
400	317	312.16	317	223.88	221.68	223.88	197.2	196.04	197.2

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 tx_1 units of time and the execution time of the algorithm which has to be evaluated be tx_2 units of time. 'tx_2/tx_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

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

	a[1]		b[2]		c[3]		d[4]		e[5]	
Alg[6]	G-MSC	P.A	G-MSC	P.A	G-MSC	P.A	HESL	P.A	HESL	P.A
T.C[7]	5.929	5.209	5.006	5.310	3.921	6.102	7.426	7.426	7.398	13.263
N.L[8]	1200	1200	1160	1200	960	1000	433	433	425	433

[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

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 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 $200\text{m} \times 200\text{m}$ 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.

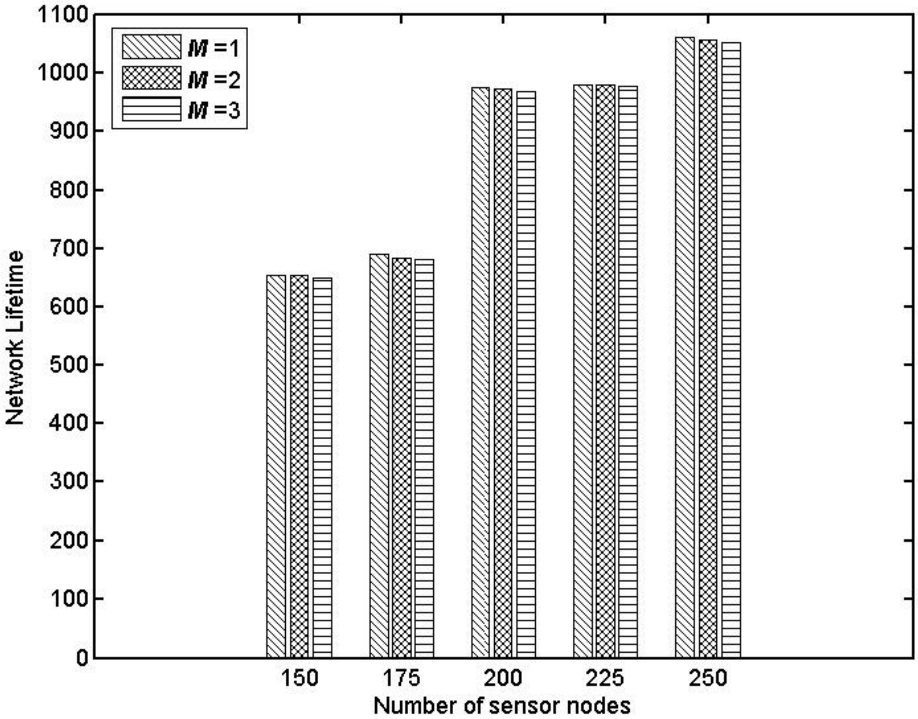


Figure 2.13: Network Lifetime for M-Connected simple coverage problem

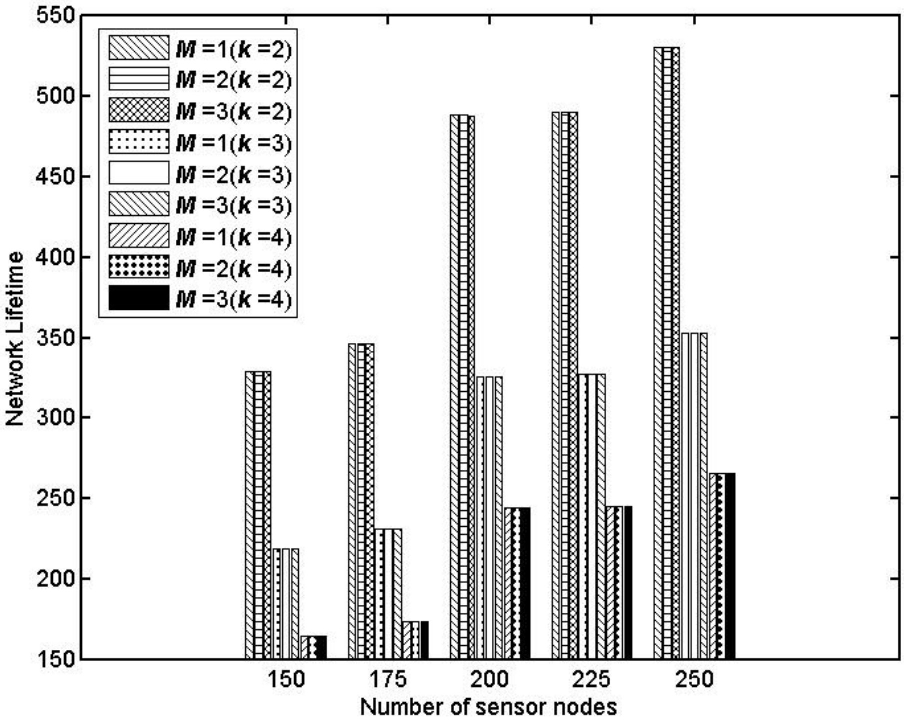


Figure 2.14: Network Lifetime for M-Connected k-coverage problem

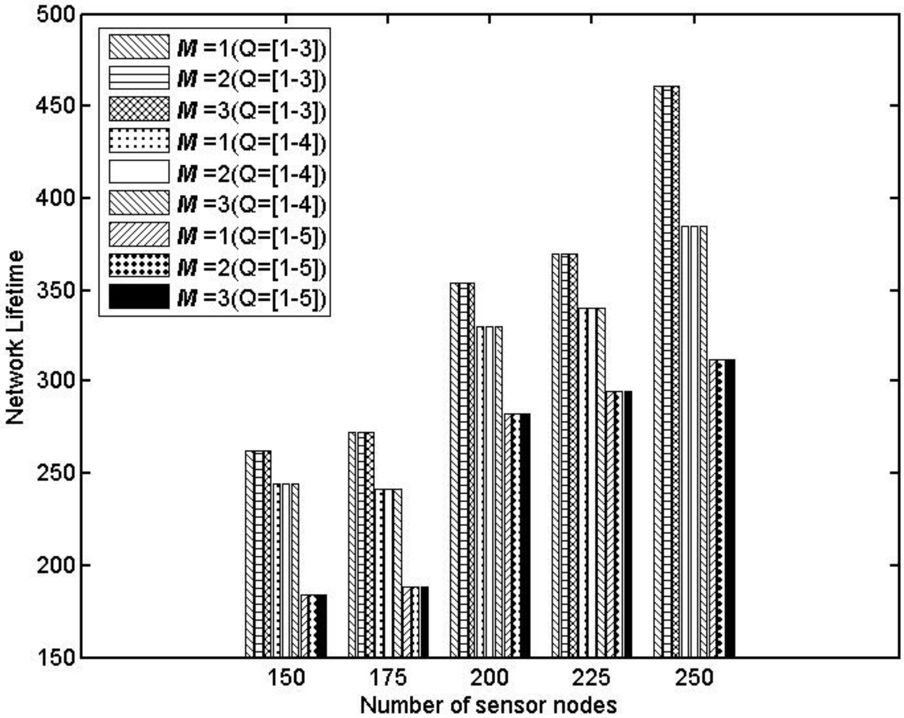


Figure 2.15: Network Lifetime for M-Connected **Q**-coverage problem

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

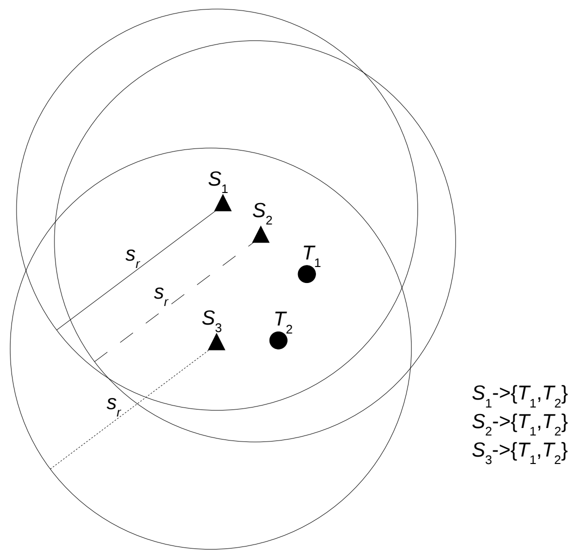


Figure 2.16: An example

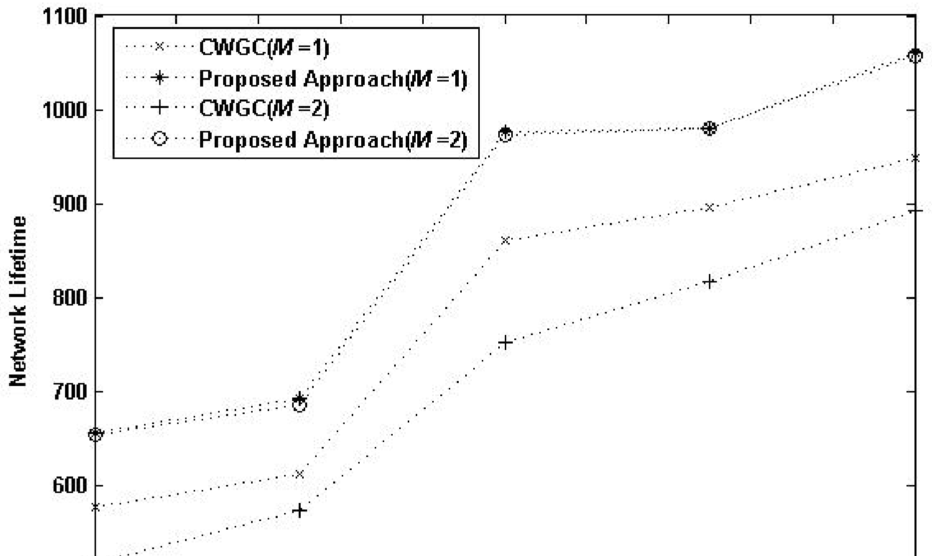


Figure 2.17: Comparison of CWGC and proposed approach for M-Connected simple coverage problem

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

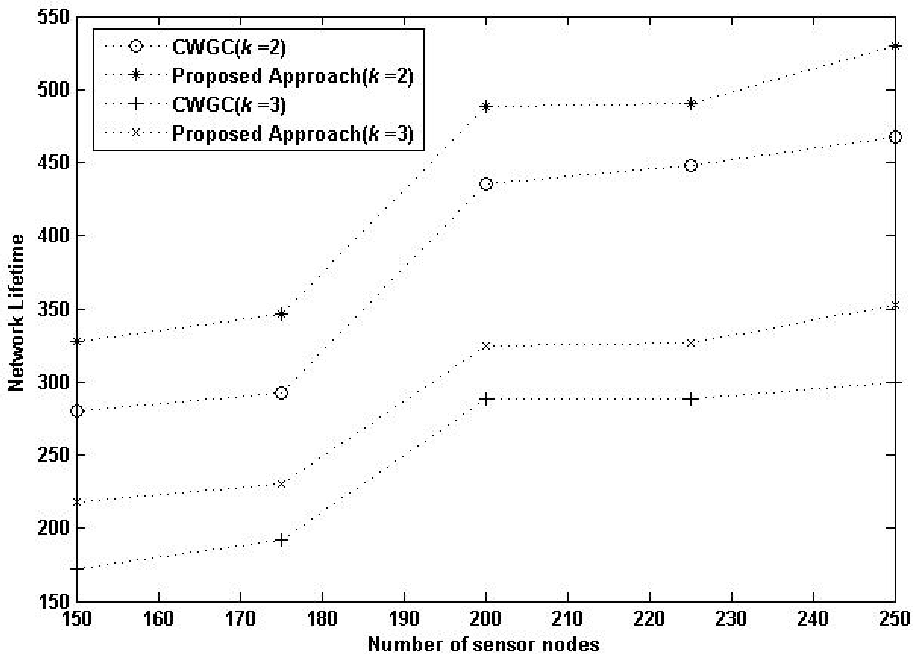


Figure 2.18: Comparison of CWGC and proposed approach for M-Connected k-coverage problem where $M = 2$

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.

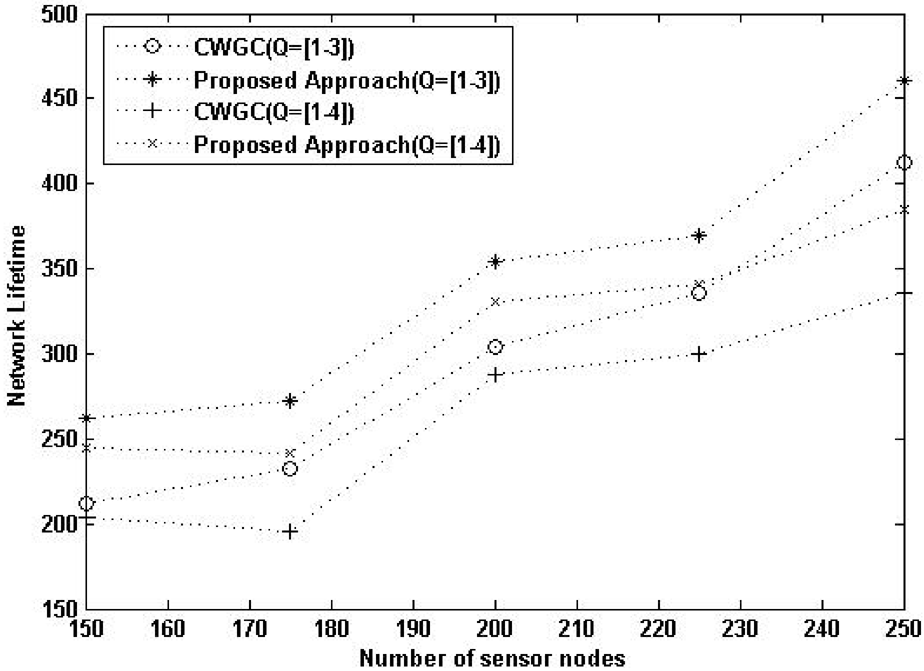


Figure 2.19: Comparison of CWGC and proposed approach for M-Connected **Q**-coverage problem where $M = 2$

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 information 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.

CHAPTER 3

SENSOR NODE DEPLOYMENT PROBLEM

3.1 Introduction

Since energy is the most critical issue in WSNs, it is necessary to optimize energy consumption in various ways. Using a proper node deployment scheme, energy consumption can be reduced and can thus extend the lifetime of WSNs (Poe and Schmitt, 2009). In addition to energy consumption, the number of nodes can be reduced and the quality of coverage can also be improved.

Maximization of network lifetime is one of the main challenges in wireless sensor networks. Energy can be efficiently used through proper energy utilization schemes, depending on the application. If the application permits deterministic deployment of nodes, energy wastage can be controlled by restricting the sensing range required and the quality of coverage can be improved in case of probabilistic coverage. Sensor nodes are deployed to achieve either area coverage or target coverage. Figure 3.1 shows a deployment to achieve area coverage. In such cases, if the sensors have fixed sensing range, optimal deployment patterns are preferred to minimize the number of sensor nodes required. In this chapter, we focus on target coverage problem. The number of sensor nodes to be deployed are limited and the sensing range has to be minimized to control energy usage and improve quality of coverage.

3.1.1 Types of Sensor Deployments

The deployment of sensor networks varies with the application considered. In some environments, it can be predetermined and be strategically hand placed. The deployment can also be undetermined: the sensors may be air-dropped or deployed by other means. The sensors can also be deterministically deployed by a robot, which can place them

Figure 3.1: Area Coverage

on the exact localization predetermined to optimize the use of network resources (Filipe *et al.*, 2004). While random node deployment is required in many applications, we need to explore possibility of other deployment methods since an inappropriate node deployment can increase the complexity of other problems like routing, clustering, data aggregation etc. in WSNs (Poe and Schmitt, 2009).

Depending on the density of nodes in a network, a sensor network deployment can be categorized as dense deployment or sparse deployment.

- Dense Deployment
  A dense deployment involves relatively large number of sensor nodes in the region of interest. It is used when higher level of coverage has to be satisfied.
- Sparse Deployment
  A sparse deployment includes only a few number of nodes in the region of interest. It is used when dense deployment is impossible because of cost of deployment or other factors (Mulligan and Ammari, 2010)

Based on the type of deployment, a sensor network deployment can be categorized as random deployment or deterministic deployment.

- Random Deployment
  Random deployment is suitable if no prior knowledge of the region is available. It is mainly used for military applications, inaccessible area, hostile region etc. However, random deployment does not always lead to effective coverage, especially if the sensors are overly clustered and there is a small concentration of sensors in certain parts of the sensor field (Zou and Chakrabarty, 2003).

- Deterministic Deployment
  Deterministic deployment is suitable for accessible regions. It is also prefered if powerful, sophisticated and expensive nodes are used which require careful planning and placement (Liu and Mohapatra, 2005). Non-sensitive applications usually use deterministic deployment.

3.1.2 Types of Sensing Models

Most research works assume that the sensing region of a sensor node is a sensing disc, that is, a sensor node has the uniform contribution in all directions of its sensing region. In the basic model, only if an event occurs within the sensing region, it is always assumed to be detected with probability 1 otherwise with probability 0. This idealized binary model has been extensively used to analyze the coverage problems of sensor networks. But in real deployment of sensor nodes, the sensing capabilities of sensor nodes have relations with the environment and then it is imperative to have practical considerations at the design stage. Such a sensing model is known as probabilistic sensing model. Thus, in general there are two sensing models: binary sensing model and probabilistic sensing model.

Binary Sensing Model

In a binary sensing model, the target is either monitored with full confidence or not monitored. Let $S = \{S_1, S_2, \ldots, S_m\}$ be the set of sensor nodes and $T = \{T_1, T_2, \ldots, T_n\}$ be the set of targets in a given region. A sensor node located at (x_1, y_1, z_1) can cover a target at (x_2, y_2, z_2) if the Euclidean distance between the sensor node and the target is less than or equal to the sensing range s_r.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \leq s_r \tag{3.1}$$

A binary sensing model is given by,

$$ST_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq s_r, \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

Figure 3.2: An example of binary sensing model

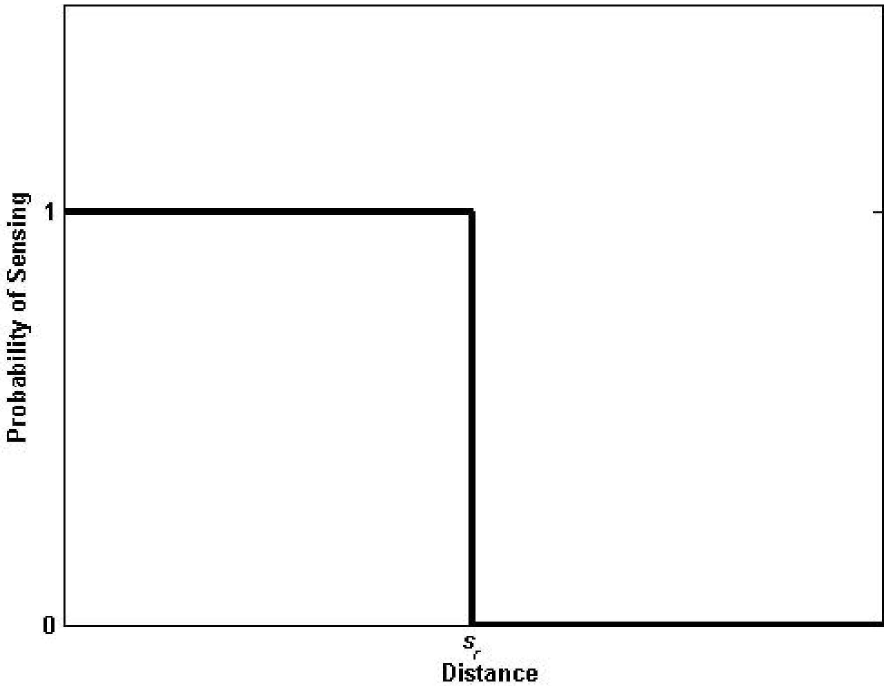


Figure 3.3: Binary Sensing

where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. d_{ij} corresponds to the Euclidean distance between S_i and T_j

Figure 3.2 shows a deployment which follows a binary sensing model. The sensor can monitor all the location points/objects within its sensing range, with full confidence.

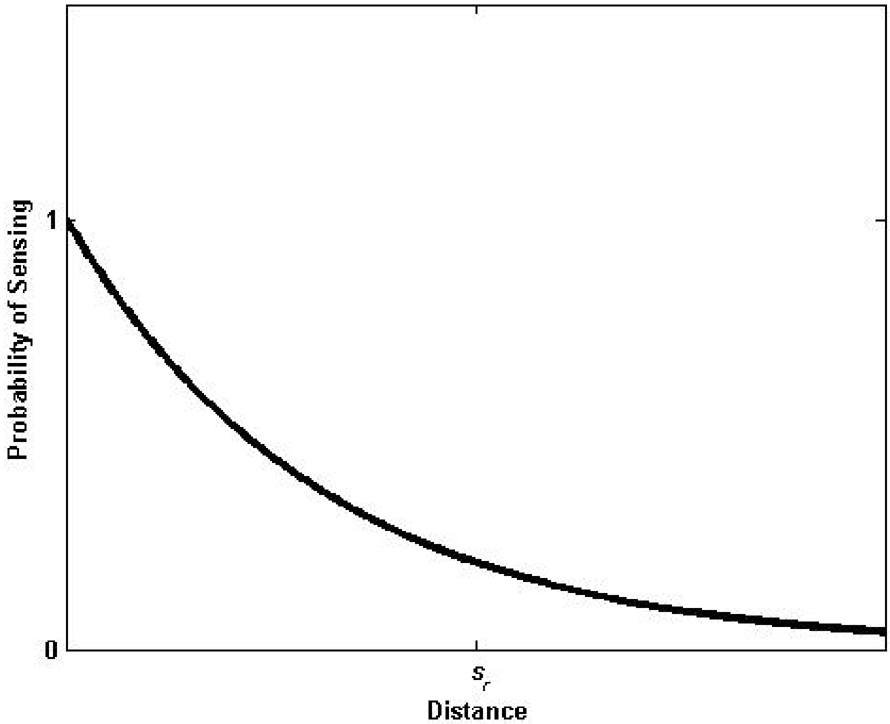


Figure 3.4: Probabilistic Sensing

Probabilistic Sensing Model

Most of the research done in coverage assumes that the sensing ability within a sensing area is deterministic, every point within the sensor's range will be seen by the sensor. This is not always the case with real sensors. A real sensor would be more likely to detect an event that is physically closer to the sensor than one that is near the edge of its sensing range due to attenuation of the signal. A probabilistic coverage model takes into consideration the effect of distance on the sensing ability of a node (Mulligan and Ammari, 2010).

Under probabilistic sensing models, the sensing range is no longer a disk. The overlap among sensing ranges of different sensors is not clearly defined. Therefore, the overlap minimization idea may not work with probabilistic coverage protocols that seek to optimize the number of activated sensors (Hefeeda and Ahmadi, 2010). With probabilistic model, the probability that the sensor detects a target depends on the relative position of the target within the sensors' sensing range. Basically the probability of detecting a target is assumed to diminish at an exponential rate with the increase in distance between a sensor and that target. A sensor can detect targets that lie in its line of sight.

Probabilistic coverage applies with some kinds of sensors e.g. acoustic, seismic etc., where the signal strength decays with distance from the source, and not with sensors that only measure local point values e.g. temperature, humidity, light etc. (Ahmed *et al.*, 2005). It can be used for applications which require a certain degree of confidence.

As in (Zou and Chakrabarty, 2005), we use the following exponential function to represent the confidence level in the received sensing signal:

$$ST_{ij} = \begin{cases} e^{-\alpha d_{ij}} & \text{if } d_{ij} \leq s_r, \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $0 \leq \alpha \leq 1$ is a parameter representing the physical characteristics of the sensing unit and environment.

The coverage of a target T_j which is monitored by multiple sensor nodes S_j is given by,

$$ST_j(S_j) = 1 - \prod_{S_i \in \mathbf{S}_j} (1 - ST_{ij}) \tag{3.4}$$

Figure 3.3 and Figure 3.4 represents binary sensing and probabilistic sensing respectively. In binary sensing (Figure 3.3), the probability of sensing immediately becomes zero when the distance of target from the sensor node is greater than or equal to the sensing range (s_r). But in probabilistic sensing (Figure 3.4), the probability of sensing slowly decreases with distance.

3.1.3 Sensor Deployment as an Optimization Problem

WSN issues such as node deployment, localization, energy-aware clustering and data-aggregation are often formulated as optimization problems. If the nodes can be deterministically deployed and if the targets are fairly large in number compared to the given

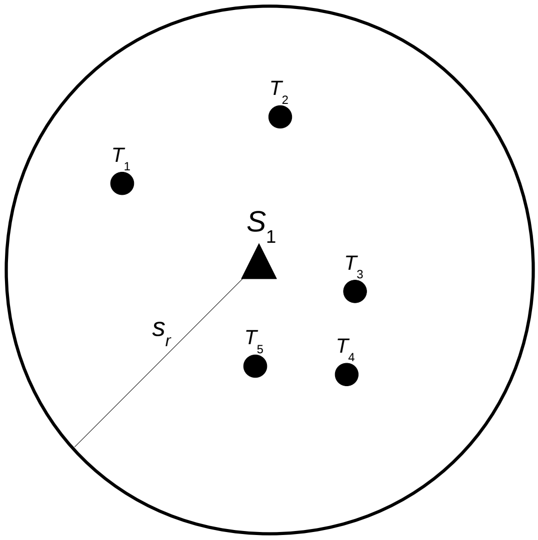


Figure 3.5: Random deployment with fixed sensing range

number of sensor nodes, energy usage can be restricted by limiting the sensing range requirement.

In this chapter, both the sensing models are considered. The binary sensing model is initially considered, where the aim is to reduce the sensing range required to achieve the required coverage. In probabilistic k-coverage, given a region with m sensor nodes and n targets, each target needs to be monitored by at least k sensors, where $1 \leq k \leq m$, as well as monitored with a required probability p for the network to function. Figure 3.5 shows a random deployment of sensor node S_1. Here S_1 has a pre-defined sensing range s_r. Figure 3.6 shows a case of deterministic deployment of S_1 where the sensing range is restricted/reduced so that it is optimal for all the targets to be monitored.

3.1.4 Swarm Intelligence to solve Optimization Problems

Swarm Intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates (Abraham *et al.*, 2006). It lends itself to forms of distributed control that may be much more efficient, scalable and effective for large, complex systems (Fleischer, 2003). Some examples of swarm

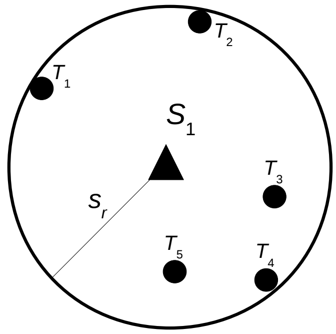


Figure 3.6: Deterministic deployment with optimal sensing range

systems are: ant colony, flock of birds, group of bees etc. The increased flexibility of collective structures in an insect colony triggered by simple modulations of the individual behavior opens interesting ways toward the design of self-adaptive artificial swarm intelligent systems (Garnier *et al.*, 2007).

The techniques evolved from the collective behaviors of swarm systems can be used to solve some complex problems. Inside a colony of insects, some features seem to be actively maintained constant and thus get out from the dynamic evolution of the colony. Robustness and flexibility are two important properties of these self-organized patterns. Robustness is the ability for a system to perform without failure under a wide range of conditions and flexibility is the ability for a system to readily adapt to new, different, or changing requirements. Robustness results from the multiplicity of interactions between individuals that belong to the colony. This ensures that, if one of the interactions fails or if one of the insects misses its task, their failure is quickly compensated by the other insects. This also promotes stability of produced patterns whereas individual behaviors are mostly probabilistic. Flexibility of self-organized systems is well illustrated by the ability of social insects to adapt their collective behaviors to changing environments and to various colony sizes (Deneubourg et al. 1986). The adaptations can occur without any change of the behavioral rules at the individual level. For instance, in the case of selection of shortest path in ants, a geometrical constraint applied on one of the two alternative paths increases the time needed by the ants to come back to their nest through this path and thus biases the choice toward the other path without any modification of the insects' behaviors (Garnier *et al.*, 2007). Some examples of swarm intelligence algorithms are Ant Colony Optimization (ACO) (Dorigo *et al.*, 1996; Dorigo and Blum, 2005), firefly algorithm (Yang, 2009; dos Santos Coelho *et al.*, 2011), harmony search (Geem *et al.*, 2001; dos Santos Coelho and Mariani, 2009).

Though there is a large body of literature in this area, we discuss two swarm intelligence algorithms we have used:

Particle Swarm Optimization

Particle Swarm Optimization (PSO) algorithm models the social behavior of bird flocking or fish schooling. PSO is a population-based stochastic optimization technique and well adapted to the optimization of nonlinear functions in multi-dimensional space. PSO consists of a swarm of particles moving in a search space of possible solutions for a problem. Every particle has a position vector representing a candidate solution to the problem and a velocity vector. Moreover, each particle contains a small memory that stores its own best position seen so far and a global best position obtained through communication with its neighbour particles (Kennedy and Eberhart, 1995).

Conceptually, it seems to lie somewhere between genetic algorithms and evolutionary programming. It is highly dependent on stochastic processes, like evolutionary programming. The adjustment toward pbest (particle best) and gbest (global best) by the particle swarm optimizer is conceptually similar to the crossover operation utilized by genetic algorithms. It uses the concept of fitness, as do all evolutionary computation paradigms (Eberhart and Kennedy, 1995).

It consists of a swarm of s candidate solutions called particles, which explore an nd-dimensional hyperspace in search of the global solution (n represents the number of optimal parameters to be determined). A particle i occupies position x_{id} and velocity v_{id} in the d^{th} dimension of the hyperspace, $1 \leq i \leq s$ and $1 \leq d \leq nd$. Each particle is evaluated through an objective function $f(x_1; x_2; \ldots; x_{nd})$, where $f : \mathbb{R}^{nd} \to \mathbb{R}$. The cost (fitness) of a particle close to the global solution is lower (higher) than that of a particle that is farther. PSO thrives to minimize (maximize) the cost (fitness) function. In the global-best version of PSO, the position where the particle i has its lowest cost is stored as ($pbest_{id}$). Besides, $gbest_d$, the position of the best particle. In each iteration

tr, velocity v and position x are updated using equation (3.5) and equation (3.6). The update process is iteratively repeated until either an acceptable $gbest$ is achieved or a fixed number of iterations tr_{max} is reached.

$$v_{id}(tr+1) = w.v_{id}(tr) + \varphi_1.r_1(tr).pbest_{id} - x_{id} + \varphi_2.r_2(tr).(gbest_d - x_{id}) \quad (3.5)$$

$$x_{id}(tr+1) = x_{id}(tr) + v_{id}(tr+1) \quad (3.6)$$

Here, φ_1 and φ_2 are constants, and $r_1(tr)$ and $r_2(tr)$ are random numbers uniformly distributed in [0,1].

The basic steps of PSO (Eberhart and Kennedy, 1995) are:

Initialize particles

Do

For each particle

Calculate the fitness value

If the fitness value is better than the best fitness value ($pbest$) in history

Set current value as the new $pbest$

End

Choose the particle with the best fitness value of all the particles as the $gbest$

For each particle

Calculate particle velocity according to velocity update equation (Equation(3.5))

Update particle position according to position update equation (Equation(3.6))

End

While maximum iterations or minimum error criteria is not attained

Artificial Bee Colony Algorithm

Artificial Bee Colony (ABC) Algorithm (Karaboga and Basturk, 2007, 2008; Karaboga and Akay, 2009) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm.

In the ABC algorithm, the colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making decision to choose a food source, is called an onlooker and a bee going to the food source visited by itself previously is named an employed bee. A bee carrying out random search is called a scout. In the ABC algorithm, first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. For every food source, there is only one employed bee. In other words, the number of employed bees is equal to the number of food sources around the hive. The employed bee whose food source is exhausted by the employed and onlooker bees becomes a scout.

In the ABC algorithm, each cycle of the search consists of three steps: sending the employed bees onto the food sources and then measuring their nectar amounts; selecting the food sources by the onlookers after sharing the information of employed bees and determining the nectar amount of the food; determining the scout bees and then sending them onto possible food sources. At the initialization stage, a set of food source positions are randomly selected by the bees and their nectar amounts are determined. Then, these bees come into the hive and share the nectar information of the sources with the bees waiting on the dance area within the hive. The information about food sources is shared by dancing (waggle dance) in the designated dance area inside the hive. This mysterious dance is essential for colony communication, and contains information regarding a food patch: the direction in which it will be found, its distance from the hive and its quality rating (or fitness). The nature of dance is proportional to the nectar content of food source just exploited by the dancing bee. At the second stage, after sharing the information, every employed bee goes to the food source area visited by itself at the previous cycle since that food source exists in its memory, and then chooses a new food source by means of visual information in the neighborhood of the present one. At the third stage, an onlooker prefers a food source area depending on the nectar information distributed by the employed bees on the dance area. As the nectar amount of a food source increases, the probability with which that food source is chosen by an onlooker increases, too. Hence, the dance of employed bees carrying higher nectar recruits the onlookers for the food source areas with higher nectar amount. After arriving at the selected area, it chooses a new food source in the neighborhood of the one in the memory depending on visual information. When the nectar of a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one. Every bee colony has scouts that are the colony's explorers. The explorers do not have any guidance while looking for food. They are primarily concerned with finding any kind of food source. As a result of such behaviour, the scouts are characterized by low search costs and a low average in food source quality. Occasionally, the scouts can accidentally discover rich, entirely unknown food sources. In the case of artificial bees, the artificial scouts could have the fast discovery of the group of feasible solutions as a task. In ABC algorithm, one of the employed bees is selected and classified as the scout bee. The classification is controlled by a control parameter called "limit". If a solution representing a food source is not improved by a predetermined number of trials, then that food source is abandoned by its employed bee and the employed bee associated with that food source becomes a scout. The number of trials for releasing a food source is equal to the value of "limit", which is an important control parameter of ABC algorithm. There are three control parameters in the ABC: the number of food sources which is equal to the number of employed and onlooker bees, the value of limit and the maximum cycle number.

Artificial recruiting could similarly represent the measurement of the speed with which the feasible solutions or the good quality solutions of the difficult optimization problems can be discovered. The survival and progress of the bee colony are dependent upon the rapid discovery and efficient utilization of the best food resources. Similarly, the successful solution of difficult engineering problems is connected to the relatively fast discovery of good solutions especially for the problems that need to be solved in real time. In a robust search process, exploration and exploitation processes must be carried out together. In the ABC algorithm, while onlookers and employed bees carry out the exploitation process in the search space, the scouts control the exploration process.

This algorithm starts by associating all employed bees with randomly generated food sources (solution). In each iteration, every employed bee determines a food source in the neighborhood of its current food source and evaluates its nectar amount (fitness).

The i^{th} food source position is represented as $D_i = (X_i, Y_i)$. $F(D_i)$ refers to the nectar amount of the food source located at D_i. If an employed bee's new fitness value becomes better than the best fitness value achieved so far, then the employed bee moves to this new food source abandoning the old one, otherwise it remains in its old food source. When all employed bees have finished this process, they share the fitness information with the onlookers in side the hive, each of which selects a food source according to the probability. The probability depends on the quality of the food source. With this scheme, good food sources will get more onlookers than the bad ones. Each bee will search for a better food source around neighborhood patch for a certain number of cycle (limit), and if the fitness value will not improve within limit number of cycles, then that bee becomes scout.

The basic steps of ABC algorithm (Karaboga and Basturk, 2008) are:

Initialize

Repeat

Move the employed bees onto their food sources and determine their nectar amounts

Move the onlookers onto the food sources and determine their nectar amounts

Move the scouts for searching new food sources

Memorize the best food source found so far

Until (requirements are met)

We use Artificial Bee Colony algorithm to compute the optimal deployment locations so that the sensing range requirement is at minimum, satisfying k-coverage and probabilistic coverage.

3.2 Problem Definition

3.2.1 Deployment for Binary Sensing Model

Sensor Coverage

A sensor node located at (x_1, y_1, z_1) can cover a target at (x_2, y_2, z_2) if the Euclidean distance between the sensor node and the target is less than or equal to the sensing range s_r.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \leq s_r \tag{3.7}$$

Mean of location points

The mean value of the location points (x_L, y_L, z_L) for $L = 1, 2, \ldots, N$, is represented by (a_1, a_2, a_3), where

$$a_1 = \frac{\sum_{L=1}^{N}(x_L)}{N} \tag{3.8}$$

$$a_2 = \frac{\sum_{L=1}^{N}(y_L)}{N} \tag{3.9}$$

$$a_3 = \frac{\sum_{L=1}^{N}(z_L)}{N} \tag{3.10}$$

Simple Coverage

Given a set of targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V \times W$ region and a set of sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, the objective is to deploy the sensor nodes such that all the targets are continuously monitored and the network lifetime is extended by keeping the sensing range at minimum. In other words, the objective is to cover all the targets in a given region by at least one sensor node and to minimize the function

$$F = \forall_i((max(distance(S_i, H_g)))) \quad (3.11)$$

where H is the set of all targets monitored by S_i, $i = 1, 2, \ldots, m$, $g = 1, 2, \ldots, h$, where h is the total number of targets S_i monitors.

k-Coverage

Given a set of targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V \times W$ region and a set of sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, the objective is to deploy the sensor nodes such that all $T = \{T_1, T_2, \ldots, T_n\}$ is covered by at least k number of sensor nodes, $1 \leq k \leq m$ and the network lifetime is extended by keeping the sensing range at minimum. In other words, the objective is to cover all the targets in a given region by at least k sensor nodes and to minimize F (Equation(3.11)).

Q-Coverage

Given a set of targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V \times W$ region and a set of sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, the objective is to deploy the sensor nodes such that all $T = \{T_1, T_2, \ldots, T_n\}$ 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, $1 \leq q_j \leq m$, at any time and the network lifetime is extended by keeping the sensing range at minimum. In other words, the objective is to cover each target by at least q_j sensor nodes and to minimize F (Equation(3.11)).

Cluster Formation

Partitioning the targets into clusters will be a key to identify the position of sensor nodes. Each sensor node is associated to a cluster. Let the set of clusters to be formed be represented as $CL = \{CL_1, CL_2, \ldots, CL_m\}$. A target T_j belongs to CL_i if and only if $distance(T_j, S_i) \leq distance(T_j, S_l)\forall_l$ where $l = 1, 2, \ldots, m$; $l \neq i$ and $j =$

$1, 2, \ldots, n$. After computing clusters, if any $CL_i = \phi$, mark $CL = CL - \{CL_i\}$ it implies that S_i is not associated to any cluster. For simple coverage problem, each target is associated to exactly one cluster and for k and **Q**-coverage problems, each target is associated to minimum of k and q_j sensor nodes respectively.

3.2.2 Deployment for Probabilistic Sensing Model

Sensor Node Deployment to achieve probabilistic target k-coverage

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V \times W$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that each target is monitored by at least k-sensor nodes and with a total probability p and sensing range is minimum. The objective is to cover each target with at least k sensor nodes and probability p and to minimize the function

$$F = \forall_i((max(distance(S_i, H_g)))) \tag{3.12}$$

where H is the set of all targets monitored by S_i, $i = 1, 2, \ldots, m$, $g = 1, 2, \ldots, h$, where h is the total number of targets S_i monitors.

Sensor Node Deployment to achieve probabilistic target Q-coverage

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V \times W$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that $T = \{T_1, T_2, \ldots, T_n\}$ is monitored by **Q** = $\{q_1, q_2, \ldots, q_n\}$ number of sensor nodes such that target T_j needs to be monitored by at least q_j number of sensor nodes, where $1 \leq i \leq n$ and with a total probability p such that the required sensing range is minimum. The objective is to cover T_j with at least q_j sensor nodes and probability p and to minimize F ((Equation(3.12)))

3.3 Related Work

Due to the dynamic and distributed nature of deployment, it is a challenging task to obtain full coverage in the region of interest and to utilize energy of each sensor in a relatively fair fashion (Heo and Varshney, 2005). There are several approaches for sensor deployment with respect to area coverage problem. The sensors are assumed to have fixed sensing range in most cases. Optimal deployment patterns are proposed to achieve area coverage and to minimize the number of nodes to be deployed. Majority of the models use binary sensing model.

Chellappan *et al.* (2005) consider sensors' mobility to be restricted to a flip. The main aim is to maximize the coverage and to minimize the number of flips. A movement plan is determined such that this is achieved. The base station collects information regarding the number of nodes per region and a movement plan is decided as of where to flip. The solution is minimum-cost maximum flow based and is executed by the base station itself. The base station sends the movement plan to the corresponding sensors. Another solution is also proposed that does not need a centralized node. In this solution, the individual sensors collect region information and execute the solution separately to identify the movement plan.

Howard *et al.* (2002) address sensor deployment for mobile sensor networks, where each node has the capability to sense, communicate, compute and move. It can be used for environments which may be both hostile and dynamic. Mobile networks have self-deployment property which implies that they can start from some initial configuration and spread out such that maximum coverage of the field is achieved. A potential-field-based approach in which nodes are treated as virtual particles, subject to virtual forces, is proposed. These forces repel the nodes from each other and from obstacles. This ensures that the nodes spread out quickly to maximize the coverage area. Apart from replusive force, a viscous friction force which ensures that the network will finally reach a state of static equilibrium, is also present. The network will automatically be reconfigured if there is some change in the environment. Energy is saved by avoiding unnecessary moves.

Heo and Varshney (2005) propose a deployment method for a rectangular region with some number of sensor nodes. All sensor nodes are assumed to have identical capabilities for sensing, communication, computation and mobility. The sensing area and communication area are assumed to be circular discs. The aim is to find the positions and movements of nodes to achieve maximum coverage and to form a uniformly distributed wireless network in minimum time and with minimum energy consumption. To solve this, some heuristics are developed and the performance is evaluated based on coverage, uniformity, and the time and distance traveled until convergence

Wang *et al.* (2005) consider the sensing field as an arbitrary shaped region with obstacles. The proposed method can be applied on an indoor environment. The sensing field is initially divided into smaller sub-regions based on the shape of the field. The sensors are then deployed in these smaller sub-regions. The sensing field is modeled as an arbitrary polygon possibly with obstacles. Results show that only few sensors are required to ensure full coverage of the sensing field and connectivity of the network as compared to other methods.

Chang *et al.* (2009*b*) propose a robot-deployment algorithm that overcomes unpredicted obstacles and employs full-coverage deployment with a minimal number of sensor nodes. Node placement and spiral movement policies are proposed for the robot to deploy sensors and an obstacle surrounding movement policy is proposed to reduce the impacts of an obstacle upon deployment. When the robot encounters an obstacle, its algorithm switches to obstacle state wherein the robot adopts the obstacle surrounding movement policy to move and deploy sensors, reducing the impacts of obstacles on deployment. Simulation results show that the proposed algorithm significantly reduces the number of deployed sensors and improves the resistance to obstacles

Tan *et al.* (2009) design schemes to maximize sensing coverage and also guarantee connectivity for a network with arbitrary sensor communication/sensing ranges or node densities, at the cost of a small moving distance. The moving distance is minimized which will ultimately reduce energy consumption. This can be applicable to all fields; irrespective of the field layout or shape of the obstacle.

Dhillon and Chakrabarty (2003) propose two algorithms for sensor placement in a sensor field. The objective is to optimize the number of sensors and to determine the deployment locations. The proposed algorithms address coverage optimization under the constraints of imprecise detections and terrain properties. These algorithms are targeted at average coverage as well as at maximizing the coverage of the most vulnerable grid points. The issue of preferential coverage of grid points is also modeled.

Tang *et al.* (2006) study relay node placement problem. The cluster-based network model consists of relay and sensor nodes. Relay nodes serve as cluster heads and form a connected network topology for information dissemination. The relay nodes are capable of aggregating data packets from the sensor nodes in their clusters and transmitting them to the sink node via wireless multi-hop paths. The objective is to place the fewest number of relay nodes in the playing field of a sensor network such that each sensor node can communicate with at least one relay node and the network of relay nodes is connected

Du and Lin (2005) propose to improve sensor network performance by deploying a small number of mobile sensors in addition to a large number of static sensors. Mobile sensors are used to increase sensing coverage, provide better routing and connectivity for sensor networks. Under-covered areas are identified and mobile sensors are moved towards that area. Simulations show that coverage and network performance can be significantly improved by having a small number of mobile sensors.

Shen *et al.* (2006) propose Grid Scan which is applied to calculate the basic coverage rate with arbitrary sensing radius of each node. The basic goal of this approach is to provide a better coverage with less nodes. It can be used to ensure k-coverage of the area. A re-deployment approach is followed which is used to get equivalent coverage rate using less number of sensor nodes or to achieve higher coverage rate with the same number of sensor nodes.

Bartolini *et al.* (2008) propose SNAP & SPREAD, an algorithm for autonomous deployment of mobile sensors. It applies to area coverage problem. Decisions regarding the behavior of each sensor are based on locally available information and do not require any prior knowledge of the operating conditions.

Wu and Yang (2005) propose SMART (Scan-based Movement-Assisted sensoR deploymenT method) that deals with the deployment of mobile sensors. The movement-assisted Sensor deployment deals with moving sensors from an initial unbalanced state to a balanced state. SMART uses scan and dimension exchange to achieve a balanced state and also addresses communication holes in sensor networks.

Ma and Yang (2005) present ATRI (Adaptive TRIangular deployment) algorithm which maximizes coverage area and minimizes coverage gaps for large scale unattended mobile sensor networks. The positions are adjusted close to equilateral triangulations, which is proved to be the optimal layout to provide the maximum no-gap coverage. It requires only the location information of nearby nodes. This is cost effective since it avoids communication cost for exchanging global information.

Andersen and Tirthapura (2009) consider sensor deployment in a three dimensional space to achieve a desired degree of coverage. Another objective is to minimize the number of sensor placed. In this paper, sensor deployment is modeled as a discrete optimization problem.

Ahmed *et al.* (2005) propose a probabilistic coverage algorithm to evaluate area coverage in a randomly deployed wireless sensor network. The coverage issues in wireless sensor networks are investigated based on probabilistic coverage and propose a distributed Probabilistic Coverage Algorithm (PCA) to evaluate the degree of confidence in detection probability provided by a randomly deployed sensor network. Coverage has been considered in terms of maximal support and breach paths, exposure, quality of surveillance and area coverage etc. This work is more related to area coverage. Area coverage checks whether every point in the target area is covered at least by a sensor node such that there is no coverage hole in the target area.

Zou and Chakrabarty (2005) address the problem of selecting a subset of nodes that are active for both sensing and communication. The active node selection procedure is aimed at providing the highest possible coverage of the sensor field, i.e., the surveillance area. It also assures network connectivity for routing and information dissemination.

Carter and Ragade (2009) address area coverage problem for probabilistic deployment of sensor nodes. The problem is formulated as an optimization problem with the objective to minimize cost while covering all points above a given probability of detection coverage threshold. A probabilistic coverage matrix is defined to decouple the coverage determination method from the model. A Genetic Algorithm (GA) approach is presented to solve the optimization problem. Hefeeda and Ahmadi (2010) propose and evaluate a fully distributed, probabilistic coverage protocol. Experimental study shows that the protocol activates less sensors than the others while maintaining the same level of coverage.

ABC algorithm is used to solve optimization problems. Karaboga and Basturk (2007) compare the performance of the ABC with that of GA, Particle Swarm Optimization (PSO) and Particle Swarm inspired Evolutionary Algorithm (PS-EA) which are also swarm intelligence and population based algorithms as the ABC algorithm. In order to demonstrate the performance of the ABC algorithm, PSO, PS-EA, GA and ABC algorithms are tested on five high dimensional numerical benchmark functions that have multimodality. From the simulation results it is concluded that the proposed algorithm has the ability to get out of a local minimum and can be efficiently used for multivariable, multimodal function optimization. The results show that ABC outperforms the other algorithms.

Here we use ABC algorithm to find out the optimal deployment locations of sensor nodes such that the required sensing range is minimum for simple/k/**Q** coverage achieved. We consider binary sensing model as well as probabilistic sensing model.

3.4 Proposed Method

3.4.1 Deployment for Binary Sensing Model

The targets are stationary. A solution is a set of locations where the sensor nodes can be deployed to cover all the targets as required with minimum sensing range. Initial solutions are randomly generated. Let the solution population be B. The solution corresponding to a bee a is denoted as $B_a = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_m, y_m, z_m)\}$ where $a = 1, 2, \ldots, nb$, nb represents total number of bees and m represents total number of nodes to be deployed.

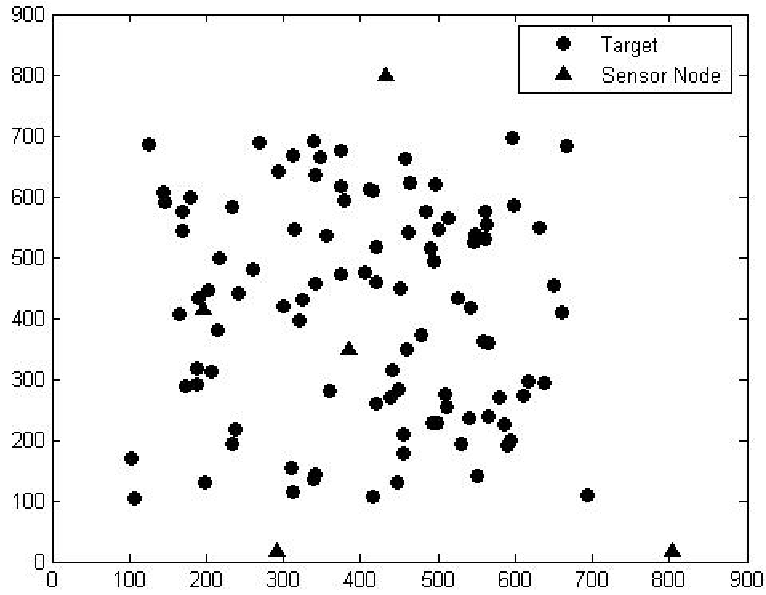


Figure 3.7: Random deployment of sensor nodes within a given region

The initial task is to form clusters according to their location. Each cluster has a sensor node associated as cluster centroid with it. The Euclidean distances of the targets and the sensor locations are calculated. Clusters are formed based on this distance measure. Clusters are generated in such a way that no sensor location in a solution is left idle without being part of a cluster.

Figure 3.7 shows a 900m$\times$900m region where 100 targets need to be monitored with 5 sensor nodes. It is noticeable that though the region is 900m$\times$900m, it is enough if the sensor nodes are randomly deployed within the region where the targets are clustered. The sub-region where the targets are clustered is marked in Figure 3.8. Figure 3.9 shows the sensor deployment after identifying this sub-region.

The number of targets in a cluster will be less if the sensor to which the cluster is associated is located at a remote place. The number of clusters formed is exactly equal to the number of sensor nodes to be deployed. The employed bees return with the solution having cluster centroids. All the deployment locations in a solution are replaced by the corresponding cluster centroid. The pseudocode for forming clusters is given in Algorithm 9.

The Euclidean distance between each target and the sensor location to which it is

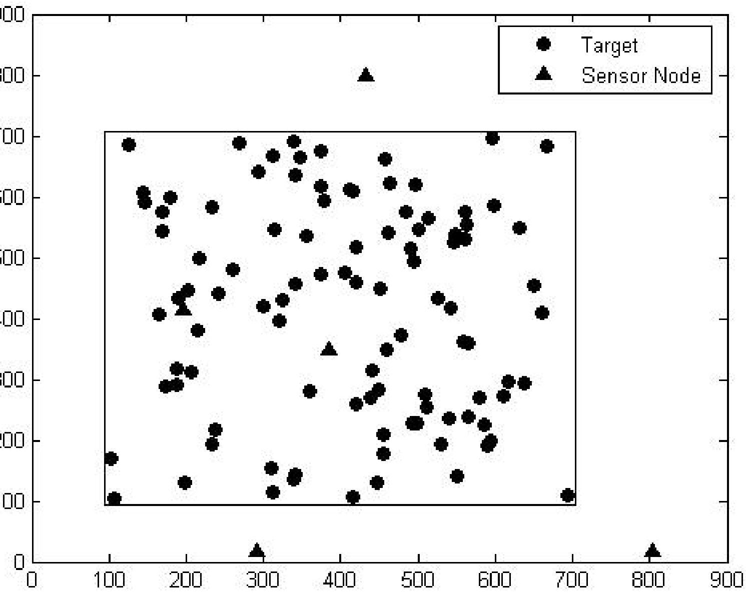


Figure 3.8: Sub-region where targets are clustered

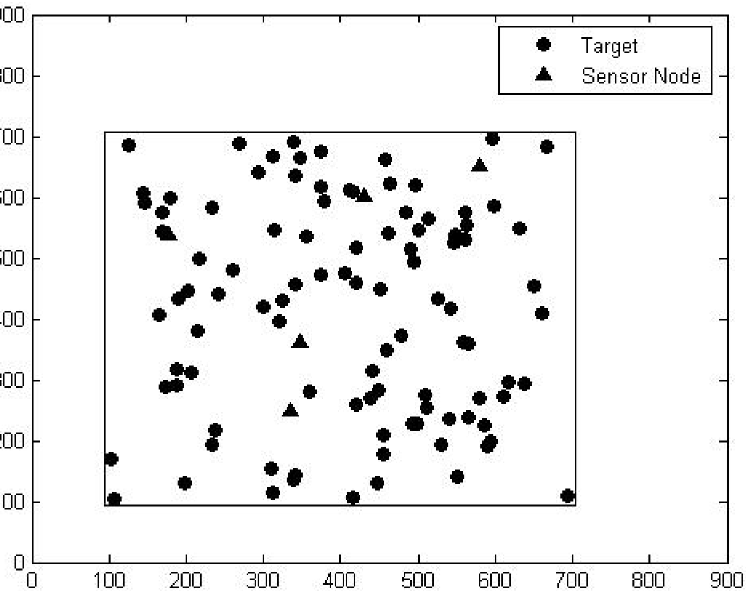


Figure 3.9: Sensors deployed within the sub-region where targets are clustered

associated is used as the fitness function to evaluate the solutions. Each sensor node is associated to a cluster, where a cluster corresponds to the set of targets monitored by the sensor node. Let $D_i = (D_{i1}, D_{i2}, D_{i3})$ be the initial position of i^{th} cluster. $F(D_i)$ refers to the nectar amount at food source located at D_i. After watching the waggle dance of

Algorithm 9 Cluster Formation

```
for each $B_a$ do
    $var = 0$
    repeat
        if $var = 0$ then
            Calculate distance between each target and all the sensor locations
            Form clusters by assigning targets to 1/$k$/**Q** sensor nodes which are at minimum distance
            if all sensor nodes form cluster then
                Move the sensor location to centroid of all target location points that are associated with it
                $var = 1$
            else
                Move sensors without assigned targets to random target locations
            end if
        end if
    until $var = 1$
end for
```

employed bees, an onlooker goes to the region of D_i with probability G_i defined as,

$$G_i = \frac{F(D_i)}{\sum_{l=1}^{m} F(D_l)} \tag{3.13}$$

where m is the total number of food sources. The onlooker finds a neighborhood food source in the vicinity of D_i by using,

$$D_i(t+1) = D_i(t) + \delta_{ij} \times v \tag{3.14}$$

where δ_{ij} is the neighborhood patch size for j^{th} dimension of i^{th} food source, v is random uniform variate $\in$ [-1, 1] and calculates the fitness value. It should be noted that the solutions are not allowed to move beyond the edge of the region. The new solutions are also evaluated and compared using the fitness function. If any new solution is better than the existing one, the new one is retained and old one is discarded. Scout bees search for a random feasible solution. The solution with the least sensing range is finally chosen as the best solution. The pseudocode of proposed method is given in Algorithm 10.

Algorithm 10 Proposed Method of sensor deployment for binary sensing model

Initialize the solution population B
Evaluate fitness (Equation(3.11))
Produce new solutions based on cluster centroids
Choose the fittest bee
$cycle = 1$
repeat
  Search for new solutions in the neighborhood
  if new solution better than old solution **then**
    Memorize new solution and discard old solution
  end if
  Replace the discarded solution with a newly randomly generated solution through a scout bee
  Memorize the best solution
  $cycle = cycle + 1$
until cycle = maximumcycles

3.4.2 Deployment for Probabilistic Sensing Model

Let the solution population be B. The region is assumed to have only stationary targets. Each solution $B_a = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_m, y_m, z_m)\}$ where $a = 1, 2, \ldots, nb$, nb the total number of bees and m the total number of nodes to be deployed, corresponds to a bee. Unlike the method described in Section3.4.1, in this case the initial solution is generated in such a way that all the targets can be probabilistically covered, and no sensor node is left idle without contributing to probabilistic 1/k/**Q**-coverage. The sensor nodes which can make each target T_j meet the required probability is then identified. Let this subset be R_j. If R_j satisfies 1/k/**Q**-coverage requirement of T_j, T_j is assigned to each sensor node in R_j. If it does not satisfy 1/k/**Q**-coverage, then identify the nearest nodes which do not belong to R_j that can make T_j 1/k**Q**-covered, along with R_j. T_j is assigned to these new sensor nodes in addition to R_j. Each sensor node is then moved to the center of all the targets which are assigned to it. This move is not allowed if some target will not be probabilistically covered due to this shift of location.

The Euclidean distance between each target and the sensor location to which it is associated is used as the fitness function to evaluate the solutions. Each sensor node is associated to a cluster, where a cluster corresponds to the set of targets monitored by the sensor node. Let $D_i = (D_{i1}, D_{i2}, D_{i3})$ be the initial position of i^{th} cluster. $F(D_i)$ refers to the nectar amount at food source located at D_i. After watching the waggle dance of employed bees, an onlooker goes to the region of D_i with probability G_i defined as in Equation 3.13. The onlooker finds a neighborhood food source in the vicinity of D_i by using Equation 3.14. It should be noted that the solutions are not allowed to move beyond the edge of the region. The new solutions are also evaluated by the fitness function. If any new solution is better than the existing one, choose that solution and discard the old one. Scout bees search for a random feasible solution. The solution with the least sensing range is finally chosen as the best solution. Algorithm 11. explains the proposed scheme.

Algorithm 11 Proposed Method of sensor deployment for probabilistic sensing model

Initialize the solution population B
Evaluate fitness ((Equation(3.12)))
Produce new solutions based on probabilistic 1/k/**Q**-coverage
Choose the fittest bee
$cycle = 1$
repeat
  Search for new solutions in the neighborhood
  if new solution better than old solution **then**
    Memorize new solution and discard old solution
  end if
  Replace the discarded solution with a newly randomly generated solution
  Memorize the best solution
  $cycle = cycle + 1$
until cycle = maximumcycles

3.5 Results and Discussion

3.5.1 Deployment for Binary Sensing Model

Experimental set up

We consider a 200m×200m×20m region for experiments. For ABC algorithm, the number of bees (colony size) is taken as 10 with equal number of employed bees and onlookers. The limit value is 50, number of cycles is 500 and the number of runs is 3. For PSO algorithm, the number of particles is 10. The number of targets to be monitored is varied from 100 to 250. Three instances are considered for each case. The number of sensors to be deployed is varied from 10 to 30. k takes values 1, 3 and 5. **Q** ranges from 1-5 and 3-5. The best sensing range obtained over all runs, the mean of sensing range of all runs and the standard deviation are reported. Experiments are conducted using MATLAB 7.

Analysis of Results

Impact of varying k and Q

The value of k is initially set as 1, which implies simple coverage problem. The optimal deployment locations and the required sensing range are computed using the proposed method. The same is done for $k = 3$ and $k = 5$. An increase in sensing range is observed but it is evident that the sensing range requirement does not increase in proportion to the increase in k. The same is observed for **Q**-coverage requirement also. **Q** which has values ranging from 1 to 5 and 3 to 7 are used as coverage requirement criteria. Table 3.1 and Table 3.2 show the sensing range requirement for k and **Q** coverage problems respectively. Figure 3.10 shows an instance where 10 sensor nodes are to be deployed in a region and k takes values 1, 3 and 5. Figure 3.11 shows an instance where **Q**-coverage requirement has to be satisfied with the least required sensing range. Both the figures clearly show that sensing range requirement does not increase in proportion with the coverage requirement.

Impact of varying number of sensor nodes

The number of sensor nodes to be deployed in the region is varied from 10 to 30. The number of clusters increase as the number of sensor nodes increase. The sensing range requirement decreases when more number of nodes are to be deployed. Figure 3.12 and Figure 3.13 show this decrease in sensing range requirement when the number of sensor nodes are increased, for k and **Q** coverage problems respectively.

Table 3.1: Sensing Range Requirement for k-coverage problem

N.T [1]	Instance	N.S [2]	k=1				k=3				k=5			
			Best	Mean	S.D [3]	S.A [4]	Best	Mean	S.D[3]	S.A[4]	Best	Mean	S.D.[3]	S.A [4]
100	1	10	38.88	38.88	0	38.91	92.23	93.47	1.07	92.4	109.89	109.99	0.16	109.96
		20	24.34	25.37	0.92	24.41	53.61	53.94	0.54	53.68	64.96	64.96	0	65.08
		30	19.45	20	0.96	19.59	41.76	41.82	0.05	41.85	56.08	56.17	0.15	56.16
	2	10	39.38	40.07	0.83	39.42	90.91	91.25	0.29	91	104.15	104.7	0.47	104.21
		20	24.61	25.34	1.05	24.72	54.43	54.86	0.61	54.51	61.62	61.62	0	61.7
		30	19.46	19.72	0.22	19.53	46.91	47.6	0.59	46.99	57.17	58.87	1.54	57.2
	3	10	35.66	36.55	0.98	35.78	87.77	88.38	0.61	87.84	108.71	108.9	0.32	108.86
		20	26.2	26.46	0.46	26.23	55.24	55.52	0.24	55.32	65.37	65.79	0.74	65.45
		30	18.79	19.18	0.6	18.91	39.51	39.51	0	39.59	59.43	59.44	0.01	59.53
150	1	10	38.21	38.21	0	38.35	86.31	86.7	0.34	86.44	106.06	106.55	0.74	106.16
		20	26.53	27.22	0.6	26.93	53.63	54.38	0.65	53.73	67.34	67.81	0.78	67.5
		30	21.48	21.8	0.45	21.56	41.99	42.24	0.42	42.32	56.41	56.68	0.45	56.51
	2	10	39.53	39.8	0.48	39.61	99.44	99.89	0.59	99.91	108.68	108.68	0	108.74
		20	26.64	26.89	0.22	26.7	54.62	55.54	1.33	54.83	70.06	70.22	0.14	70.12
		30	21.02	21.39	0.33	21.12	43.23	43.67	0.43	43.51	62.05	62.05	0	62.14
	3	10	40.93	40.93	0	41	91.05	91.09	0.08	91.13	109.16	109.78	1.08	109.28
		20	25.7	26.63	0.8	25.85	53.86	54.34	0.42	53.96	66.21	66.21	0	66.3
		30	21.39	22.28	0.91	21.5	42.81	43.37	0.65	42.92	58.6	58.83	0.37	58.73
200	1	10	42.24	42.3	0.08	42.33	95.85	96.58	0.68	95.96	111.42	111.85	0.69	111.54
		20	30.14	30.29	0.27	30.22	58.95	59.71	0.71	59.05	70.6	70.6	0	70.72
		30	23.39	23.71	0.56	23.52	47.74	48.37	0.85	47.9	66	66.96	0.92	66.11
	2	10	41.22	42.03	1.13	41.32	99.37	99.77	0.35	99.49	111.54	111.74	0.17	111.63
		20	28.95	29.09	0.17	29.08	56.74	56.94	0.34	56.89	70.22	70.22	0	70.34
		30	23.18	23.43	0.24	23.36	47.13	47.13	0	47.56	59.36	59.8	0.38	59.45
	3	10	42.51	43.08	0.64	42.68	98.06	98.12	0.06	98.19	114.09	114.27	0.27	114.18
		20	29.57	29.73	0.28	29.66	59.2	59.72	0.53	59.34	71.54	71.88	0.6	71.6
		30	24.01	24.54	0.77	24.13	47.49	47.73	0.25	47.61	62.02	63.17	1.53	62.09
250	1	10	41.27	41.75	0.67	41.33	99.53	99.85	0.3	99.6	110.04	110.29	0.22	110.1
		20	29.68	30.14	0.46	22.73	56.7	57.08	0.33	56.8	67.94	67.97	0.03	68
		30	23.83	24.78	0.83	23.89	44.92	45.89	1.11	44.98	64.59	65.22	0.72	64.63
	2	10	41.76	41.93	0.29	41.83	96.86	96.99	0.12	96.94	108.48	108.48	0	108.53
		20	28.91	29.59	0.76	28.96	55.24	55.14	0.91	55.27	72.95	73.39	0.38	72.98
		30	23.02	23.87	0.99	23.07	45.3	45.74	0.59	45.38	60.79	60.96	0.16	60.84
	3	10	42.6	42.72	0.12	42.64	98.26	99.85	1.39	98.31	108.23	108.23	0	108.3
		20	28.37	29.39	0.88	28.44	59.58	59.92	0.5	59.64	71.82	72.28	0.45	71.89
		30	23.64	25.19	1.36	23.7	43.78	43.95	0.31	43.84	63.21	63.53	0.44	63.27

[1] Number of targets
[2] Number of sensor nodes
[3] Standard Deviation
[4] Sensitivity Analysis

Table 3.2: Sensing Range Requirement for **Q**-coverage problem

N.T[1]	Instance	N.S[2]	Q=1-5 Best	Q=1-5 Mean	Q=1-5 S.D[3]	Q=1-5 S.A[4]	Q=3-5 Best	Q=3-5 Mean	Q=3-5 S.D[3]	Q=3-5 S.A[4]
100	1	10	111.64	112.29	0.65	111.75	125.27	126.16	1.15	125.34
		20	65.03	65.96	0.81	66.03	89.5	89.65	0.17	89.65
		30	48.25	48.49	0.29	48.39	61.2	62.24	1.12	61.29
	2	10	112.96	115.28	2.01	113.1	146.83	147.63	1.01	146.96
		20	65.81	66.71	1.33	65.93	96.11	97.48	1.27	96.24
		30	48.48	49.14	0.58	48.62	65.11	65.79	0.62	65.2
	3	10	98.5	99.14	0.58	98.64	134.47	135.21	0.93	134.52
		20	61.17	61.62	0.4	61.25	93.5	93.93	0.62	93.58
		30	47.65	48.33	0.8	47.74	65.16	65.77	0.53	65.29
150	1	10	111.57	112.89	1.24	111.64	132.4	132.97	0.54	132.53
		20	70.76	71.16	0.41	70.89	93.12	93.72	0.65	93.22
		30	56.06	56.6	0.49	56.1	70.19	70.48	0.41	70.28
	2	10	109.87	110.16	0.29	109.98	133.55	133.93	0.43	133.67
		20	68.19	68.51	0.53	68.3	99.69	99.97	0.33	99.88
		30	55.83	56.04	0.23	55.89	72.07	72.12	0.06	72.19
	3	10	119.52	120.06	0.58	119.66	143.1	144.05	0.95	143.14
		20	72.36	72.78	0.49	72.5	99.59	101.81	1.96	99.74
		30	59.92	60.32	0.32	59.99	68.53	69.1	0.61	68.6
200	1	10	122.21	122.74	0.51	122.27	138.19	139.06	0.78	138.24
		20	78.75	79.52	0.73	78.81	101.28	101.88	0.55	101.31
		30	63.42	63.5	0.07	63.5	77.6	77.98	0.41	77.69
	2	10	123.74	124.6	0.74	123.8	142.2	143.88	1.61	142.25
		20	71.36	72.17	0.8	71.43	104.95	106.23	1.11	105
		30	59.83	60.67	0.74	59.88	68.49	68.89	0.35	68.54
	3	10	117.21	117.24	0.06	117.28	140.37	142.09	1.6	140.44
		20	70.14	70.22	0.11	70.21	101.79	101.95	0.25	101.83
		30	54.04	55	0.84	54.09	75.82	76.04	0.78	75.89
250	1	10	116.67	117.16	0.49	116.74	130.84	131.56	0.66	130.88
		20	74.47	74.86	0.56	74.54	97.89	98.14	0.41	97.91
		30	58.73	59.85	1	58.8	72.77	73.11	0.51	72.83
	2	10	125.74	127.41	1.46	125.79	139.19	139.33	0.13	139.24
		20	77.38	78.85	1.37	77.42	103.91	105.83	1.82	103.96
		30	57.3	57.94	0.56	57.36	72.83	73.21	0.33	72.9
	3	10	119.69	122.53	2.73	119.74	151.06	152.95	1.68	151.09
		20	77.12	78	0.94	77.17	117.98	119.39	1.44	118.01
		30	57.67	58.1	0.51	57.7	79.47	81.41	1.77	79.53

[1] Number of targets
[2] Number of sensor nodes
[3] Standard Deviation
[4] Sensitivity Analysis

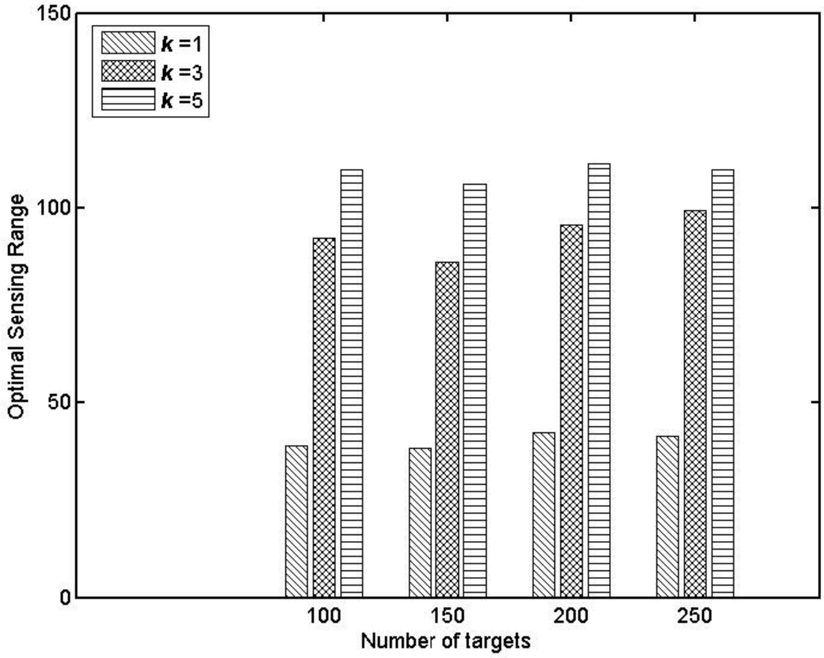


Figure 3.10: Sensing range requirement for k-coverage problem where number of sensor nodes to be deployed is 10

Impact of varying number of targets

The number of targets to be covered is varied from 100 to 250. Results show that the sensing range requirement need not essentially be high for higher number of targets. Sensing range requirement is highly dependent on the location of the targets to be covered. The results can also be used to find the minimum number of sensor nodes required to cover specific number of targets with a given sensing range in the 3-D region.

Sensitivity Analysis

Since it may be hard to deploy the sensors exactly at positions where sensing range is optimal, we conduct sensitivity analysis. We change the optimum deployment positions by ± 0.05 and calculate the new required sensing range. The variation in required sensing range is found to be of less significance. The analysis reveals that the deployment solutions obtained through the proposed ABC based method is a robust one and does not change significantly with a slight variation in the optimal deployment positions.

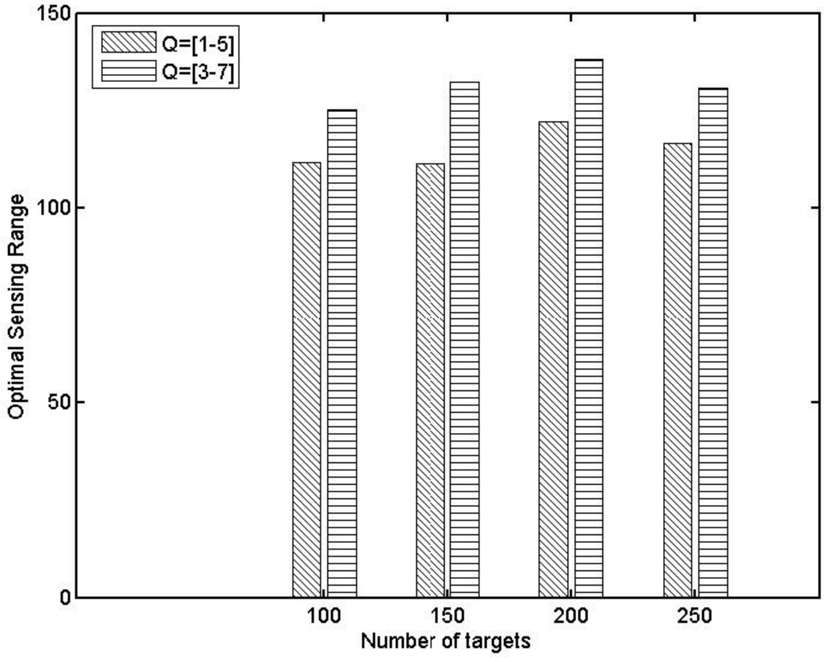


Figure 3.11: Sensing range requirement for **Q**-coverage problem where number of sensor nodes to be deployed is 10

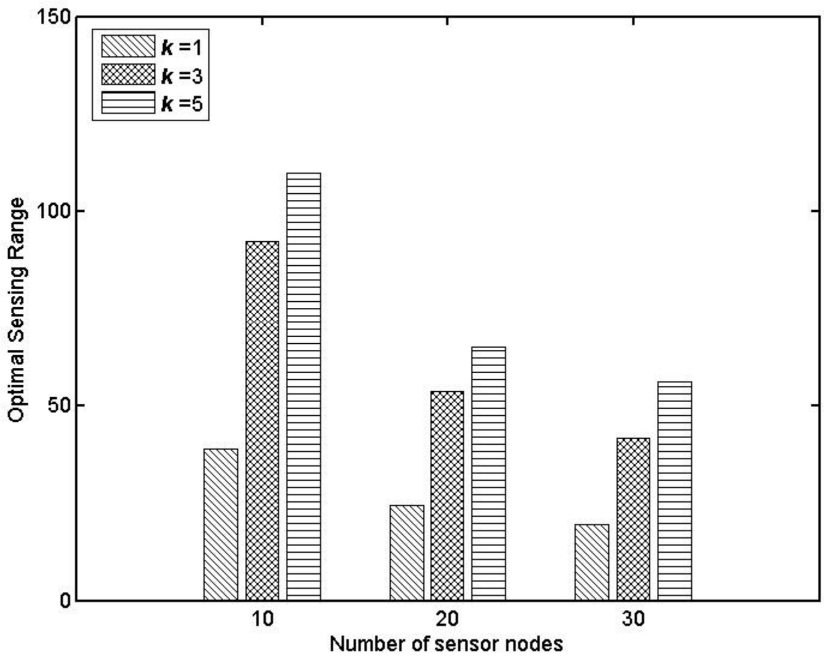


Figure 3.12: Sensing range requirement for k-coverage problem where number of targets is 100

Figure 3.14(a) shows a region where 5 sensor nodes are randomly deployed to cover 100 targets. It is evident that random positioning of sensor nodes will lead to higher

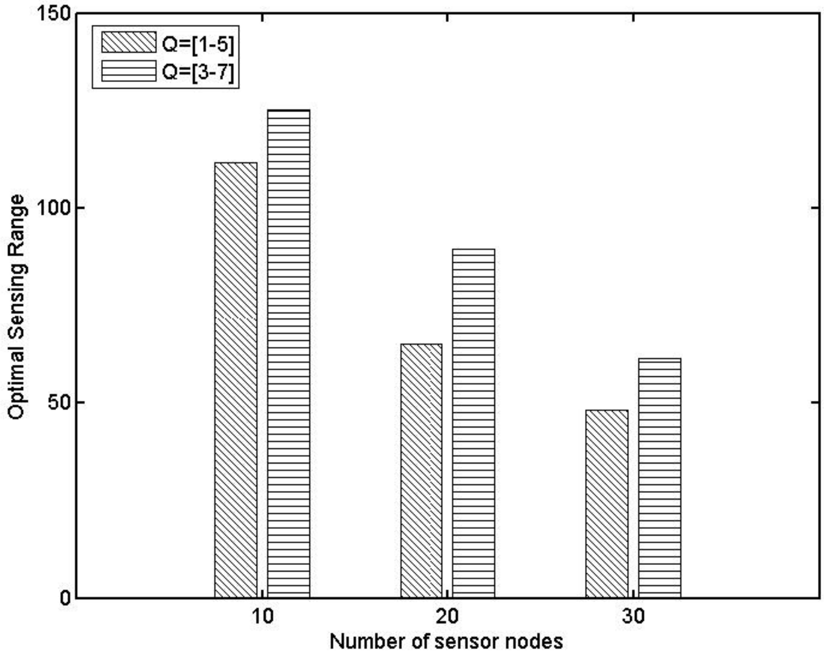
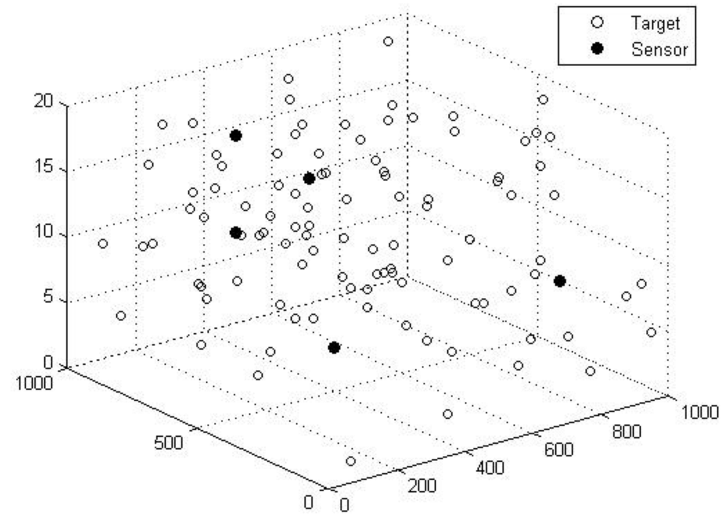


Figure 3.13: Sensing range requirement for **Q**-coverage problem where number of targets is 100

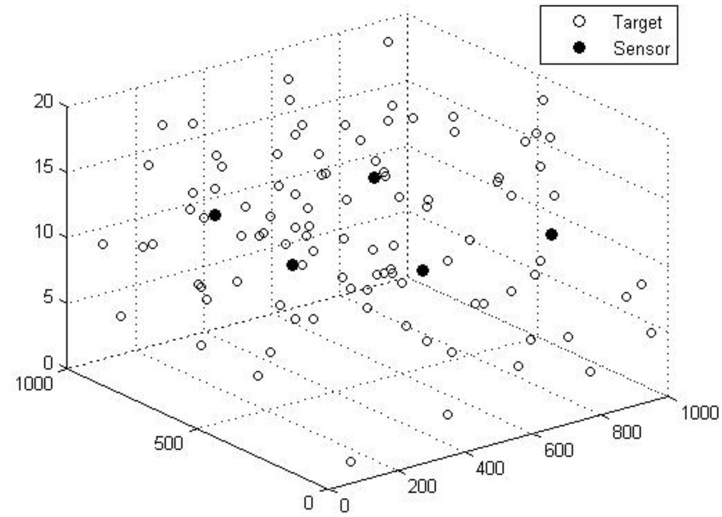
sensing range requirement. Figure 3.14(b) shows the sensor positioning using proposed method.

3.5.2 Performance comparison of ABC and PSO algorithm for sensor node deployment

Table 3.3 shows that ABC performs better than PSO for most cases of k-coverage problem. With 100 targets to be covered with 10 sensor nodes and $k = 1$, PSO and ABC performs equally well. The same is observed for $k = 5$. But for $k = 3$, ABC performs better than PSO. When 150, 200 and 250 targets are to be covered, ABC consistently performs better than PSO in minimizing the required sensing range. ABC achieves significant better results when 200 targets are to be covered with $k = 3$ and $k = 5$. Table 3.4 shows a comparison of ABC and PSO for **Q** coverage problem. For all the cases, ABC outperform PSO.



(a) Random placement of 5 sensor nodes



(b) Positioning using proposed scheme

Figure 3.14: Random Deployment vs Deterministic Deployment

Table 3.3: Comparison-ABC and PSO for k-coverage problem

		k=1		k=3		k=5	
N.T	N.S	PSO	ABC	PSO	ABC	PSO	ABC
100	10	38.8848	38.8848	94.3126	92.2272	109.8947	109.8947
150	10	38.3071	38.2093	87.1409	86.3083	107.4020	106.0641
200	10	42.7753	42.2403	97.3731	95.8521	116.4750	111.4267
250	10	42.5557	41.27	100.5250	99.5288	110.39394	110.0405

Table 3.4: Comparison-ABC and PSO for **Q**-coverage problem

		Q=[1-5]		Q=[3-7]	
N.T	N.S	PSO	ABC	PSO	ABC
100	10	111.7030	111.6439	125.5504	125.2670
150	10	111.6057	111.5673	132.4023	132.4023
200	10	122.2658	122.2077	138.8689	138.1892
250	10	117.2917	116.6738	131.9910	130.8441

Table 3.5: Sensing Range for Probabilistic Coverage

		Sensing Range		
α	Probability	Best	Mean	Standard Deviation
	0.6	2.0616	2.0616	0
0.05	0.7	2.0616	2.0616	0
	0.8	2.0616	2.0616	0
	0.9	2.0616	2.0616	0
	0.6	2.0616	2.0616	0
0.1	0.7	2.0616	2.0616	0
	0.8	2.0616	2.0616	0
	0.9	3.8748	3.9071	0.0465
	0.6	2.0616	2.0616	0
0.15	0.7	2.0616	2.0616	0
	0.8	3.8586	4.0558	0.3406
0.2	0.6	2.0616	2.0616	0
	0.7	3.5618	3.6170	0.0927

3.5.3 Deployment for Probabilistic Sensing Model

Initially, we consider a $10 \times 10 \times 10$ grid for experiments. The number of targets is 10 and the number of sensor nodes is 5. The number of bees (colony size) is taken as 10 with 5 employed bees and 5 onlookers. The number of cycles is 500, limit for neighborhood search is 20 and the number of runs is 3. MATLAB 2007a is used for implementation.

Probabilistic Coverage

Initially, we compute the sensing range required for probabilistic coverage without considering k. The required probability for coverage is varied from to 0.6 to 0.9. α is varied

Table 3.6: Sensing Range for Probabilistic Target k-Coverage

			Sensing Range		
α	Probability	k	Best	Mean	Standard Deviation
0.05	0.6	2	3.9249	3.9844	0.0531
		3	6.1847	6.2085	0.0315
	0.7	2	3.7814	3.9384	0.1550
		3	6.1974	6.2051	0.0089
	0.8	2	3.9514	3.9802	0.0251
		3	6.1901	6.2034	0.0142
	0.9	2	3.8407	3.9497	0.1330
		3	6.1847	6.2054	0.0187
0.1	0.6	2	3.9175	3.9870	0.1050
		3	6.1847	6.2235	0.0366
	0.7	2	3.9098	3.9505	0.0423
		3	6.2043	6.2235	0.0262
	0.8	2	3.9105	3.9433	0.0289
		3	6.1870	6.2013	0.0125
	0.9	2	3.9206	4.2680	0.3030
		3	6.1968	6.2175	0.0319
0.15	0.6	2	3.9538	3.9658	0.0107
		3	6.1847	6.2187	0.0296
	0.7	2	3.9236	3.9490	0.0304
		3	6.1847	6.2129	0.0289
	0.8	2	3.9286	4.0131	0.0757
		3	6.2005	6.2144	0.0128
0.2	0.6	2	3.9319	3.9531	0.0198
		3	6.1851	6.1876	0.0042
	0.7	2	3.9119	3.9789	0.0749
		3	6.1847	6.1963	0.0113

from 0.05 to 0.2. Table 3.5 shows the sensing range requirement for probabilistic coverage. When $\alpha = 0.05$, the required sensing range does not change for any of the required probability. But when α increases to 0.1, the sensing range required increases for a detection probability of 0.9. Similarly, a variation in sensing range is observed at 0.8 for $\alpha = 0.15$ and at 0.7 for $\alpha = 0.2$. This implies that for higher α, the sensing range requirement varies at a smaller detection probability. It is also observed from Table 3.5 that for higher value of α, no solution exists for higher probability value (for $alpha = 0.2$, probability greater than 0.7, there is no solution).

Table 3.7: Sensing Range for Probabilistic Target **Q**-Coverage

			Sensing Range		
α	Probability	Q	Best	Mean	Standard Deviation
0.05	0.6	1-2	2.0616	2.0616	0
		1-3	3.9751	4.0235	0.0439
		1-4	3.9427	3.9547	0.0208
	0.7	1-2	2.0616	2.0616	0
		1-3	3.7723	3.8672	0.0836
		1-4	3.9147	3.9761	0.0719
	0.8	1-2	2.0616	2.0616	0
		1-3	3.8876	3.9565	0.0815
		1-4	3.8286	3.9514	0.1233
	0.9	1-2	2.0616	2.0616	0
		1-3	3.943	3.9721	0.0297
		1-4	3.9254	3.9858	0.055
0.1	0.6	1-2	2.0616	2.0616	0
		1-3	3.8579	3.9443	0.1061
		1-4	3.9178	3.9591	0.0611
	0.7	1-2	2.0616	2.0616	0
		1-3	3.8544	3.9946	0.1216
		1-4	3.8623	3.9178	0.1078
	0.8	1-2	2.0616	2.0616	0
		1-3	3.9269	3.9494	0.0264
		1-4	3.9652	4.0359	0.0627
	0.9	1-2	3.6976	3.9003	0.1771
		1-3	4.0306	4.1559	0.1095
		1-4	4.1482	4.3142	0.1648
0.15	0.6	1-2	2.0616	2.0616	0
		1-3	4.0044	4.0378	0.0289
		1-4	3.9331	3.9378	0.004
	0.7	1-2	2.0616	2.0616	0
		1-3	3.9766	3.9827	0.0061
		1-4	3.9273	3.9803	0.0912
	0.8	1-2	3.4953	3.6227	0.1207
		1-3	3.8904	4.0533	0.168
		1-4	3.9161	4.0423	0.1204
0.2	0.6	1-2	2.0616	2.0616	0
		1-3	3.9354	3.9735	0.0449
		1-4	3.8028	3.9391	0.1228
	0.7	1-2	3.4768	3.5369	0.0941
		1-3	3.9114	4.0044	0.1294
		1-4	3.8978	3.9552	0.0516

Probabilistic k-Coverage

To observe the difference in sensing range required for probabilistic k-coverage, k takes values 2 and 3. The required probability for coverage is varied from 0.6 to 0.9 and α is varied from 0.05 to 0.2.

Table 3.7 shows the sensing range requirement for probabilistic k-coverage. For a constant detection probability, the sensing range requirement may or may not increase with k. This is because for some cases, more than k sensor nodes may have to monitor a target for satisfying probabilistic coverage. Due to the same reason, there are instances where probabilistic coverage and probabilistic k-coverage requires the same sensing range. For example, $\alpha = 0.05$ and probability = 0.6 for probabilistic coverage requires sensing range of 2.0616 units, and $\alpha = 0.05$, probability = 0.6, $k = 1$ for probabilistic k-coverage requires the same sensing range. But in this case, when $k = 2$, the sensing range required increases.

Probabilistic Q-Coverage

α is assumed to take values 0.05, 0.1, 0.15 and 0.2. We vary Q-values 1-2, 1-3 and 1-4. The required probability is set to 0.6, 0.7, 0.8 and 0.9. The sensing range required depends highly on **Q**. With α at 0.15, the targets cannot be covered with a probability 0.9 and with $\alpha = 0.2$, the targets cannot be covered with a probability 0.8 or higher.

We also consider a $100 \times 100 \times 20$ grid for experimentation. Three instances of 100 targets being monitored by 10 sensor nodes are considered. k is varied from 1 to 5. The required probability is set to 0.8 and α is assumed to be 0.01. Table 3.8 shows the sensing range requirement for this set-up. There is no significant variation in standard deviation even with an increase in k. This shows that the method is a reliable one even for higher k. Table 3.9 shows the sensing range required for **Q**-coverage problem. With **Q**-coverage requirement, for probability 0.8 and 0.9, the sensing range required for **Q** values 1-2, 1-3 and 1-4 are the same.

Table 3.8: Sensing Range for Probabilistic Target k-coverage (100$\times$100$\times$20 grid)

		Sensing Range		
k	Instance	Best	Mean	Standard Deviation
1	1	19.2000	19.3030	0.2485
	2	19.2428	19.3317	0.1487
	3	18.6335	19.0473	0.3584
2	1	28.3267	28.5713	0.2630
	2	29.5044	29.7045	0.3058
	3	28.3521	28.5780	0.1957
3	1	41.8409	42.5757	0.6741
	2	43.9295	44.0164	0.1469
	3	43.7273	44.0340	0.5125
4	1	49.8290	49.8973	0.0611
	2	50.3353	50.5659	0.2514
	3	50.7494	51.2090	0.7510
5	1	51.6985	52.2129	0.4852
	2	52.192	52.498	0.3183
	3	54.0340	54.4637	0.5567

Table 3.9: Sensing Range for Probabilistic Target **Q**-coverage (100$\times$100$\times$20 grid)

		Sensing Range		
Probability	Q	Best	Mean	Standard Deviation
0.6	1-2	28.0531	28.0764	0.0378
	1-3	28.0543	28.0998	0.0517
	1-4	28.0545	28.0560	0.0027
0.7	1-2	28.0538	28.0955	0.0422
	1-3	28.0545	28.1372	0.0737
	1-4	28.0545	28.1345	0.1386
0.8	1-2	28.0545	28.4431	0.3564
	1-3	28.0545	28.4431	0.3564
	1-4	28.0545	28.4431	0.3564
0.9	1-2	28.9500	29.2468	0.2888
	1-3	28.9500	29.2468	0.2888
	1-4	28.9500	29.2468	0.2888

3.6 Conclusion

This chapter concentrates on WSN deployments where the deployment of sensor nodes is carefully planned and controlled instead of random deployment. Though there are deployments where sensor nodes have to be randomly deployed, there are applications where the deployment can be made much more efficient by optimal deployment. Both the sensing models; binary and probabilistic are taken into consideration.

For a binary sensing model, we propose an ABC based method to find optimum sensor deployment positions in a 3-D terrain in order to satisfy different target coverage criteria, namely, simple, k-coverage and $\mathbf{Q}$-coverage. Extensive simulations are carried out with varying number of sensor nodes, number of targets, k-values and values of $\mathbf{Q}$ to find the minimum sensing range requirement. We notice that sensing range requirement does not increase in the same proportion with increase in k or $\mathbf{Q}$ requirements. An increase in number of sensor nodes to be deployed, decreases the sensing range requirement. But for a given number of sensor nodes, an increase in the number of targets to be covered need not always make the sensing range requirement high. This method is also suitable to find the optimal number of sensor nodes required to satisfy a coverage criteria.

We also explore the use of both probabilistic coverage, probabilistic k-coverage and probabilistic $\mathbf{Q}$-coverage models for target coverage problem and propose a method to compute the optimal deployment locations so that the sensing range requirement is minimum. ABC algorithm is used to solve probabilistic coverage, probabilistic k-coverage and probabilistic $\mathbf{Q}$-coverage. The variation in sensing range is studied for a range of detection probabilities (p), coverage requirement (k) and physical medium characteristics (α). ABC algorithm proves to be reliable in getting the optimal deployment locations. The standard deviation of obtained sensing range among various runs does not change significantly for a larger region or for higher values of k/$\mathbf{Q}$ which justifies the robustness of the proposed algorithm.

CHAPTER 4

SENSOR NODE DEPLOYMENT AND SCHEDULING PROBLEM

4.1 Introduction

Energy efficient protocols should be designed to extend the lifetime of a wireless sensor network (Cheng *et al.*, 2008). This is since sensor nodes operate on batteries and replacing batteries every week in building networks is a laborious task and replacing them in a less friendly environment may not be possible (Hohlt *et al.*, 2004). Some sensitive applications require higher-order coverage, meaning that the phenomenon of interest should be simultaneously monitored by multiple sensors (Bellazreg *et al.*, 2010). k-coverage is often discussed in the context of reliability or fault tolerance (Ma and Liu, 2007). Energy usage should be optimized for all sensor operations, which include sensing, computation, and communication (Yao and Giannakis, 2005). It is also important to maintain a balance of energy consumption in the network so that certain nodes do not die much earlier than others. In order to maintain some energy balance in the network, it is useful to adopt deterministic deployment and use scheduling mechanisms further. Sensor scheduling (Ma and Liu, 2007; Chachra and Marefat, 2006) preserves energy since some sensors that share common sensing region and tasks switched into sleep mode.

Random deployment of sensor nodes may result in an inefficient WSN, where some areas are densely deployed while the other areas have a low density deployment (Chang *et al.*, 2009*a*). Dense deployment in some areas increases the hardware cost, whereas a sparse deployment in the other areas results in coverage holes or network partitions. As a result, there is no guarantee for full coverage, and considerable hardware costs are needed.

Optimal deployment patterns is useful if it is possible to place the sensors where desired. In some deployment scenarios, such as when deploying in harsh terrains, placing individual sensors at desired locations may not be feasible. In these circumstances, sensor node deployments may follow some probability distribution. In deterministic deployment, the details of the region will be known and since a provision of deploying nodes at specific locations prevail, there exist two ways by which network lifetime can be maximized. One is at deployment phase and the other is at scheduling phase. Given a region with targets being monitored by sensor nodes, the upper bound of network lifetime can be mathematically computed. This information can be used for computing locations which would be appropriate for coverage to be satisfied as well as network lifetime to be maximum. Once the deployment locations are computed, sensor nodes can be scheduled to achieve the optimum lifetime. Sensor node deployment and scheduling in this way contributes equally to extend the network lifetime.

Both sensor node deployment and scheduling are important to ensure prolonged network lifetime. Traditionally, the problems of sensor placement and scheduling have been considered separately from each other. A balanced performance is crucial for most applications. One approach of sensor node deployment is to deploy few nodes with large batteries. This might not be effective since it is sensitive to node failures (Krause *et al.*, 2009). Additionally, packaging constraints can limit the size of the battery deployed with the nodes. Hence it can be more effective to deploy a larger number of nodes with smaller batteries, that are activated only a fraction of the time. Later, scheduling becomes a crucial task to obtain longer lifetime.

Some planned deployments (Ma and Liu, 2007) aim at minimizing the number of sensors to be deployed. Splitting sensor node deployment into multiple rounds may incur higher deployment cost, e.g., an aircraft needs to fly along the deployment line multiple times to accomplish the task (Yang and Qiao, 2010). Deploying a large number of scheduled sensors has the benefit that it allows trading off power and accuracy. The deployed network might have several modes of operation: a scheduled mode of operation, where only a small fraction of sensors is active, and a "high density" mode where all (or a larger fraction of) sensors are activated. For example, in traffic monitoring, once a traffic congestion is detected (during scheduled mode), the high density mode could be used to accurately identify the boundary of the congestion (Krause *et al.*, 2009).

The network performance with respect to target detection, is directly related to the placement of the sensors within the field of interest (Lazos *et al.*, 2007). Different sensor node deployment strategies can cause very different network topology, and thus different degrees of sensor redundancy (Wu *et al.*, 2005). A good sensor node deployment with sufficient number of sensors which ensures a certain degree of redundancy in coverage so that sensors can rotate between active and sleep modes is required to balance the workload of sensors (Wu and Yang, 2007).

4.2 Problem Definition

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that all targets are covered as per the coverage requirement and schedule the nodes such that the network lifetime is maximum. The objective is
1. To deploy the sensor nodes such that the network lifetime upper bound is maximum and
2. To schedule the sensor nodes so as to achieve the optimal network lifetime

4.2.1 Upper Bound of Network Lifetime

Assume m sensor nodes $\{S_1, S_2, \ldots, S_m\}$ randomly deployed to cover the region R with n targets $\{T_1, T_2, \ldots, T_n\}$. 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 coverage matrix is defined as,

$$ST_{ij} = \begin{cases} 1 & \text{if } S_i \text{ monitors } T_j \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

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

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,

$$u = min_j \left\lfloor \frac{\frac{\sum_i ST_{ij} * b_i}{q_j}}{e_i} \right\rfloor \tag{4.2}$$

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

4.2.2 Sensor Node Deployment

Sensor Node Deployment to achieve 1-coverage

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that each target is monitored by at least one sensor node and the network lifetime is maximum. The objective is to maximize u such that each target is monitored by at least one sensor node.

Sensor Node Deployment to achieve k-coverage

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that each target is monitored by atleast k-sensor nodes and to maximize u.

Sensor Node Deployment to achieve Q-coverage

Given a set of n targets $T = \{T_1, T_2, \ldots, T_n\}$ located in $U \times V$ region and m sensor nodes $S = \{S_1, S_2, \ldots, S_m\}$, place the nodes such that each target T_j, $1 \leq j \leq n$, is covered by at least q_j sensor nodes and to maximize u.

4.2.3 Sensor Node Scheduling

1-Coverage Scheduling

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. Network lifetime $\sum_{P=1}^{Y} t_P$ is maximized

k-Coverage Scheduling

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. Network lifetime $\sum_{P=1}^{Y} t_P$ is maximized

Q-Coverage Scheduling

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. Network lifetime $\sum_{P=1}^{Y} t_P$ is maximized

4.3 Related Work

Most of the existing works on sensor node deployment problem focus on area coverage. Bartolini *et al.* (2010) propose GREASE which is a distributed deployment algorithm. Environments with obstacles are considered for sensor node deployment. This reduces the sensing range of sensor nodes that are deployed. Full coverage can be achieved if the number of available sensors exceeds a given threshold.

The performance of multiround sensor node deployment is studied by Yang and Qiao (2010). Optimal strategies that use fewest sensors to cover a barrier are also derived. The results show that a simple two-round sensor node deployment leads to efficient barrier coverage. A solution to the movement-assisted sensor deployment problem is proposed by Wu and Yang (2007) using global network information. The monitoring area is a 2-D grid-based mesh and the focus is on load balance solutions in WSN that minimize total moving distance of sensors.

Chang *et al.* (2009*a*) present Obstacle-Resistant Robot Deployment (ORRD) algorithm to near-deploy optimal number of sensors over the monitoring region, which achieves full coverage. A node placement policy, a serpentine movement policy, and obstacle-handling rules are presented. An environment that contains several irregular obstacles is considered to identify the number of required sensors to achieve full coverage.

Ibrahim *et al.* (2007) adds an available set of relays to enhance the connectivity of a network. The network is not allowed to be disconnected. But sensors at some point of time might not be able to send its own data as well as relay the other sensors' data due to loss of energy, leading to a coverage gap in the network area. This necessitates adding relays to ensure connectivity.

From the perspective of coverage, Onur *et al.* (2005) address quality of the deployment and propose quality measures, which indicate if the deployment provides sufficient coverage, or whether redeployment is required or not. Lazos *et al.* (2007) address wireless sensor deployment for detecting mobile targets. The target detection problem is mapped to a line-set intersection problem and analytic expressions for the probability of detecting mobile targets are derived.

Bredin *et al.* (2005) consider the problem of deploying or repairing a sensor network to ensure a specified level of multi-path connectivity (k-connectivity) between all nodes. This is required to guarantee fault tolerance against node failures and high capacity through multi-path routing. Chin (2009) propose a sensor deployment approach which uses optimal sensor distribution as a reference. Initially, the optimal sensor distribution for the given terrain is calculated. A clustering based approach is then developed to guide sensors to appropriate locations. Regions with and without obstacles are considered for deployment.

Wu *et al.* (2005) estimate redundant sensing areas among neighboring wireless sensors. Methods to estimate the degree of redundancy without the knowledge of location or directional information are presented. Meng *et al.* (2010) address the area coverage redundancy problem for randomly deployed WSNs. Liu *et al.* (2007) propose a mathematical method for calculating the coverage fraction in WSNs. It is assumed that the deployed sensor nodes can cover the whole area. If a node's sensing area is monitored by its active neighbor nodes, it can be considered as a redundant node. A lightweight node scheduling (LNS) algorithm is proposed based on this idea. It prolongs the network lifetime of the sensor network by turning off redundant nodes without using location information.

Liu *et al.* (2005) aim at designing and analyzing sensor node scheduling algorithms. There is only one subset of sensor nodes active at any given time. Since the network is assumed to be dense, each subset might be sufficient to cover most of the area. Blind points are areas that cannot be monitored by any sensor node for a given time period. The algorithm cannot guarantee the elimination of blind points. Blind points at this time can be covered at some other time, as long as it is within the sensing range of certain sensor nodes.

Yen and Cheng (2009) assume random deployment of sensor nodes and consider powering off redundant sensors temporarily. This helps to conserve energy while retaining sufficient network coverage. A range-based sleep scheduling (RBSS) protocol which needs sensor-to-sensor distance but no location information, is proposed. RBSS attempts to approach an optimal sensor deployment pattern that demands the minimal number of working (awake) sensors while preserving 100% area coverage. It requires the ability to estimate transmission distances between neighboring sensors.

Yu *et al.* (2007) present techniques to implement a query mechanism to collect data periodically over sensor networks. A scheduling algorithm to schedule sensor node operations to achieve contention-free communication in aggregate monitoring applications is proposed. A deterministic transmission schedule can be computed for each sensor node which in turn guarantees the successful completion of data aggregation within a known time. The algorithm helps achieve significant amount of savings on power consumption over CSMA based alternative approaches.

Keshavarzian *et al.* (2006) consider the design of efficient wakeup scheduling schemes for energy constrained sensor nodes that adhere to the bidirectional end-to-end delay constraints posed by applications that require immediate notification of rare but urgent events and also fast delivery of time sensitive actuation commands. Makhoul and Pham (2009) address the area coverage problem of scheduling the activity of randomly deployed nodes to extend the network lifetime in the context of surveillance applications. Xiao *et al.* (2006) consider the optimal power scheduling problem for the decentralized estimation of a noise-corrupted deterministic signal in an inhomogeneous sensor network.

Choi and Das (2006) propose a coverage-adaptive random sensor scheduling for application-aware data-gathering in wireless sensor networks. The goal is to maximize the network lifetime. In each round, k data reporters (sensors) which can meet the desired sensing coverage specified by the users/applications is identified. Geometric probability theory and a randomization technique is used to select these k data reporters. A data gathering tree is formed by the selected k data reporters for a round. This avoids wait-and-forward delay which may result from the random sensor scheduling. The entire monitored area is covered within a fixed delay since all sensors have an equal opportunity to report sensed data periodically.

Kompella and Snoeren (2003) present a practical realization of lazy packet scheduling that attempts to minimize the total transmission energy in a broadcast network.

This is achieved by dynamically adjusting each node's transmission power and rate on a per-packet basis. Yao and Giannakis (2005) focus on a single-hop sensor data collection although the scheduling protocol can also be used in multihop networks with multilevel hierarchy to support communication between nodes and cluster heads, and between cluster heads and higher level gateways. Arai *et al.* (2008) address a sensor scheduling problem for a class of systems whose measurements are influenced by state dependent noise. The sensor scheduling problem is formulated as a model predictive control problem with single sensor measurement per time.

Bellazreg *et al.* (2010) investigate the effect of radio temporal irregularities on sensor deployment and sleep scheduling and claim that the sensor coverage range varies according to time because the available energy decreases. Chang and Chang (2008) propose node placement, topology control, and scheduling protocol to prolong the sensor network lifetime, balance the power consumption of sensor nodes, and avoid collision. This applies to area coverage problem. The whole monitoring area is partitioned into a number of equal-sized zones. Then two node-placement techniques, namely Distance-based and Density-based deployment schemes, are proposed to balance the power consumption of sensor nodes. Distance-based scheme controls the node placement distance and use power control mechanism to balance the power consumption. Nodes that are closer to the sink node will be deployed with a smaller distance to their neighbors. The Density-based scheme deploys more sensor nodes in the zone closer to the sink node since the sensor nodes in the zone closer to the sink node consume more energy for packet forwarding. One sensor node in a zone stays active, while the other nodes sleep. Thus, sensor nodes in the higher density zone will have more chance to sleep for energy balance. A collision-free MAC scheduling protocol is also proposed to prevent the packet transmissions from collision.

Ma and Liu (2007) investigate sensor networks with directional sensing and communication capability and propose a method for deploying sensor nodes with directional sensing range, and subsequent connectivity checking and repairing. A deployment strategy, with sensors having adjustable sensing ranges to cover an area, is also proposed.

Sensor node deployment and sensor node scheduling have been mostly considered as separate problems. In this chapter, we address network lifetime maximization with respect to both deployment and scheduling.

4.4 Proposed Method

The proposed method has two phases: sensor node deployment and sensor scheduling. The nodes are initially deployed randomly. Based on the theoretical upper bound of network lifetime, we compute the optimal deployment locations using ABC algorithm. A heuristic is then used to schedule the sensor nodes such that the network lifetime is maximum. Algorithm 12 describes the proposed method.

Algorithm 12 Proposed Approach

```
Input: S, T
Output: Optimal location of S and sensor schedule
Deploy S randomly
Compute network lifetime upper bound using Equation 4.2
Recompute sensor node positions using ABC algorithm such that the upper bound
   of network lifetime is maximum
Design sensor schedule using the proposed heuristic for sensor scheduling such that
   the network lifetime upper bound is achieved
```

4.4.1 Sensor Node Deployment

Since the upper bound of network lifetime can be computed, we have to find the deployment locations such that the network lifetime is maximum. First we propose a heuristic to compute the deployment locations and then later we use ABC algorithm to compute the locations and make a comparative study on the performance of the heuristic and ABC based method to solve this problem.

A Heuristic for Sensor Node Deployment

Here we propose a heuristic for sensor node deployment (Algorithm 13). Initially, place the sensor nodes randomly. If any sensor node is idle (without monitoring any target), the node is moved to the least monitored targets' location. This is to ensure that all sensor nodes play their part in monitoring targets. Sort the sensor nodes based on the number of targets it cover. Place the sensor node at the middle of all the targets it covers. Find the next nearest target and place the sensor node in the middle of all these targets. If it can cover this new target along with targets it was already monitoring, allow this move, else discard the move. This is done till the sensor node cannot cover any new target. At the end, upper bound is computed. The drawback of this approach is that it depends on the initial position of the sensor nodes. Though it may perform well for dense deployments, consistency cannot always be guaranteed.

Algorithm 13 A heuristic for sensor node deployment

Place sensor nodes randomly
for $i = 1$ to m **do**
  if S_i does not monitor any target **then**
    Move S_i to the least monitored target
    Recompute sensor-target coverage matrix
  end if
end for
S = Sensor nodes sorted in ascending order of number of targets it cover
for $i = 1$ to m **do**
  repeat
    Place S_i at the center of all targets it cover
    Move S_i to the center of all targets it cover and its next nearest target
    if S_i can cover a new target **then**
      Recompute sensor-target matrix
    else
      Discard move
    end if
  until S_i cannot cover another target
end for
Compute upper bound of network lifetime using sensor-target coverage matrix

ABC based sensor node deployment

Algorithm 14 describes ABC algorithm. Let the solution population be B. The region has only stationary targets. Each solution $B_a = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ where $a = 1, 2, \ldots, nb$, where nb and m represents the total number of bees and the total number of nodes respectively, corresponds to a bee. The initial solution is generated in such a way that all the targets are covered, and each sensor node covers at least one target. The network lifetime is computed for each solution using Equation 4.2.

This network lifetime is used as the fitness function for evaluating the solutions.

Each sensor node is associated with a cluster, where a cluster corresponds to the set of targets monitored by the sensor node. Let $D_i = (X_i, Y_i)$ be the initial position of i^{th} cluster. $F(D_i)$ refers to the nectar amount at food source located at D_i. After watching the waggle dance of employed bees, an onlooker goes to the region of D_i with probability G_i defined as,

$$G_i = \frac{F(D_i)}{\sum_{l=1}^{m} F(D_l)} \tag{4.3}$$

where m is the total number of food sources. The onlooker finds a neighborhood food source in the vicinity of D_i as,

$$D_i(t+1) = D_i(t) + \delta_{ij} \times f \tag{4.4}$$

where δ_{ij} is the neighborhood patch size for j^{th} dimension of i^{th} food source, and f is a random uniform variate $\in$ [-1, 1]. It should be noted that the solutions are not allowed to move beyond the edge of the search region. The new solutions are evaluated using the fitness function Equation 4.2. If any new solution is better than the existing one, the old solution is replaced with a new solution. Scout bees search for a random feasible solution. The solution with the least sensing range is finally chosen as the best solution.

Algorithm 14 ABC Algorithm

```
Initialize the solution population B
Evaluate fitness
cycle = 1
repeat
    Search for new solutions in the neighborhood
    if new solution is better than old solution then
        Memorize new solution and discard old solution
    end if
    Replace the discarded solution with a newly randomly generated solution
    Memorize the best solution
    cycle = cycle + 1
until cycle = maximumcycles
```

4.4.2 A Heuristic for Sensor Scheduling

As mentioned earlier, another objective of this paper is to schedule the sensor nodes such that the theoretical upper bound of network lifetime can be achieved.

To achieve this, we propose a weight-based 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 15 Heuristic for sensor node scheduling

Input ST, b
Initialize k/**Q**, max_run, priority calculated using battery power
for r = 1 to max_run **do**
  for $iteration$ = 1 to $\sum_{i=1}^{m} b_i$ **do**
    if cover possibility exists **then**
      Determine cover based on priority
      Optimize cover
      Activate optimized cover and reduce battery power
    else
      $break$
    end if
  end for
  Calculate network lifetime (*nlife*)
  if *nlife* $< u$ **then**
    Consider weight due to covered targets to compute priority
  else
    $break$
  end if
end for

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 it has. Cover sets are decided based on this priority.

The base station calculates weight for each sensor node by considering two factors:

a) Weight due to covered targets and b) Weight due to the remaining energy

$a)\ Weight\ Assigned\ by\ the\ node\ itself$

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

$b)\ Weight\ due\ to\ covered\ targets$

All sensor nodes are assigned weights based on the targets it covers as:

$$w_i = \sum_{j=1}^{n} \frac{ST_{ij}}{\sum_{i=1}^{m} ST_{ij}} \tag{4.5}$$

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, subsequently 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: If a sensor node that monitors a target turns off, it 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.

The proposed heuristic (Algorithm 15) initially finds the network lifetime using weight assigned by the nodes itself (battery power). If the obtained network lifetime does not match the theoretical upper bound of network lifetime, the weight due to covered targets is considered to compute the lifetime.

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 (Algorithm 16). 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 16 Cover Formation

```
Input: Sorted S in descending order of assigned weight
Output: Cov_S
Initialize Cov_S = φ
for i = 1 to m do
    if S_i contributes to coverage then
        Cov_S = Cov_S ∪ {S_i}
    end if
    if coverage requirement met then
        break;
    end if
end for
```

Cover Optimization

Once the coverage requirement is met, the obtained cover set is optimized (Algorithm 17). 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 $S_i \in Cov\_S$, $1 \le i \le length(Cov\_S)$, 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.

Algorithm 17 Cover Optimization

Input: Cov_S
Output: $Opt.Cov\_S$
Initialize $Opt.Cov\_S = \phi$
for i = $length(Cov\_S)$ down to 1 **do**
  if $Cov\_S - \{S_i.Cov\_S\}$ meets k/**Q** coverage requirement **then**
    $Ignore\ S_i.Cov\_S$
    $Cov\_S = Cov\_S - \{S_i.Cov\_S\}$
  else
    $Opt.Cov\_S = Opt.Cov\_S \cup \{S_i.Cov\_S\}$
  end if
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_{min}. When the battery power reaches E_{min}, the node becomes inactive and will not be able to monitor any more targets further. 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 (Algorithm 18).

As we assume that the number of sensors deployed in the region 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.

Algorithm 18 Cover Activation and Energy Reduction

Input: $Opt.Cov\_S$
for i = 1 to $length(Opt.Cov\_S)$ **do**
    $S_i.state = true$
    $decrement\ b_i$
    if $b_i \leq E_{min}$ **then**
        for j = 1 to n **do**
            $M_{ij} = 0$
        end for
    end if
end for

4.5 Results and Discussion

We consider a 500m×500m region for experiments. The number of targets is 25. The number of sensor nodes is varied from 100 to 250. Sensing range of each sensor node is fixed as 75m. Initially, each sensor node has 100 units of battery power. Energy consumption rate is 1 unit. Simulations are carried out using MatLab 2007a.

4.5.1 Sensor Node Deployment

Random Deployment

In random deployment, there is more chance of targets being not detected or targets not being covered with the required level of coverage. However, this may not hold true with dense deployment of nodes. There is another possibility of some targets being monitored by many sensor nodes, and some by very few sensor nodes. This difference in the number of sensor nodes monitoring each target will affect the network lifetime. The sensor nodes may be positioned in a better way so as to avoid this variation which will yield better lifetime. Though random deployment has these drawbacks, there are applications where random deployment is the only feasible strategy.

Figure 4.1, Figure 4.2 and Figure 4.3 show the network lifetime for various deployment methods. Random deployment does not yield as much network lifetime as other methods do. Figure 4.1 shows that for simple coverage problem, random deployment survives when 100, 150, 200 and 250 nodes are deployed. But Figure 4.2 shows that when 100 sensor nodes are to be randomly deployed, $k = 2$ will be satisfied, but $k = 3$ and $k = 4$ will not be satisfied, ultimately giving zero network lifetime. Figure 4.3 shows that for 100 sensor nodes and $\boldsymbol{Q}$ ranging from 1 to 5, random deployment does not work. This clearly shows that when the number of nodes to be deployed is high, random deployment can be done, but when the number of sensor nodes is not sufficient enough, random deployment fails.

Heuristic

The heuristic could consistently achieve better results compared to random deployment. Figure 4.1, Figure 4.2 and Figure 4.3 clearly show it. Figure 4.2 shows that when 100 sensor nodes are randomly deployed, $k = 3$ condition can not be satisfied. But with the heuristic, the network can be active for some time. The same is observed for $k = 4$. Random deployment yields zero network lifetime, but the heuristic can make the network survive for more time.

When the number of nodes is increased to 150, $k = 3$ and $k = 4$ requirement is possible even with random deployment. In these cases, heuristic can achieve much higher network lifetime. Figure 4.3 shows that the heuristic is able to satisfy $\boldsymbol{Q}$ ranging from 1 to 5 with 100 sensor nodes deployed, whereas random deployment fails. The results confirm that the heuristic performs better compared to random deployment.

ABC based Deployment

Though deterministic deployment may be time consuming, it helps in improving network lifetime. It might require only a few nodes to be active at a time to satisfy coverage requirement, as compared to random deployment. In the experiments, we have taken the number of bees as 10, number of cycles is 5000 and the number of runs is 5. The limit for neighborhood search is set as 30. Deployment using ABC algorithm could achieve much higher network lifetime compared to random deployment and the proposed heuristic. Irrespective of number of sensor nodes, coverage requirement etc., this method consistently achieves better network lifetime. Figure 4.1, Figure 4.2 and Figure 4.3 depict it.

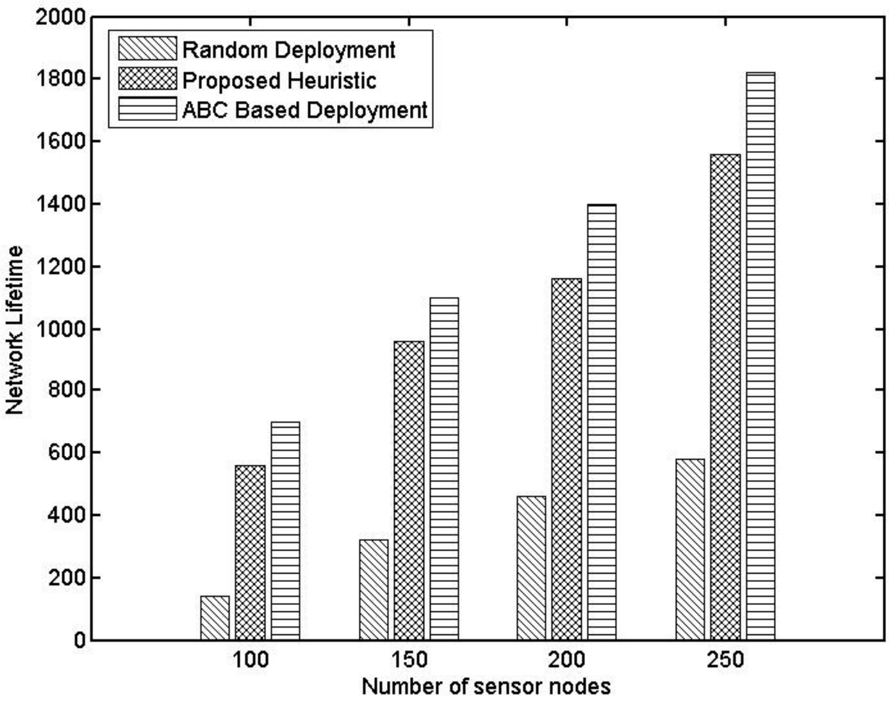


Figure 4.1: Network Lifetime for simple coverage problem using random deployment, proposed heuristic and ABC algorithm

Other Observations

$a) Varying\ number\ of\ Nodes$: The number of sensor nodes to be deployed in the region is varied from 100 to 250. The network lifetime increases when higher number of nodes are to be deployed. $b) Varying\ Coverage\ Requirement$: An increase in network lifetime is observed but it is evident that the network lifetime does not increase in proportion to the increase in k. The same is observed for **Q**-coverage requirement also.

4.5.2 Sensor Scheduling

Since the optimal deployment locations are known, now the sensor nodes have to be scheduled such that each sensor node need not be awake all the time. Theoretical upper bound helps to analyze how far the proposed heuristic is successful. With battery power as the priority deciding factor, though most instances could obtain network lifetime equal to upper bound, there are a few instances where the network lifetime does not

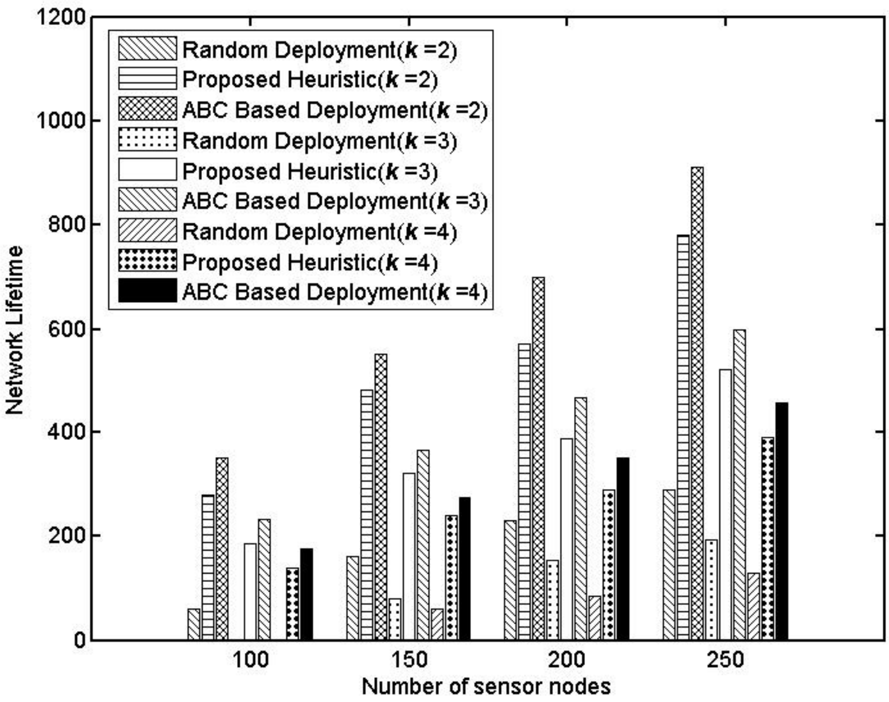


Figure 4.2: Network Lifetime for k coverage problem using random deployment, proposed heuristic and ABC algorithm

match the computed upper bound. For these instances, the second weight deciding factor is considered. An inverse of this helps to achieve theoretical upper bound. Thus we observe that there are a few cases where if the sensors monitoring more number of targets are kept in reserve for later use, theoretical and experimental network lifetime matches.

Figure 4.4 shows a comparison of upper bound of network lifetime and the network lifetime obtained using proposed approach for simple coverage problem. The proposed approach could achieve the theoretical upper bound in all the instances. Figure 4.5 and Figure 4.6 show the comparison of upper bound and the network lifetime obtained using proposed approach for k-coverage and **Q**-coverage problems respectively. The network lifetime obtained using proposed approach matches the theoretical upper bound for all these requirements as well. Figure 4.7 shows a comparison of the proposed method and Greedy-MSC (Cardei *et al.*, 2005). Greedy-MSC was proposed to solve simple coverage problem. In Greedy-MSC, a critical target (the target most sparsely covered) is selected initially. Once the critical target has been selected, the heuristic selects the

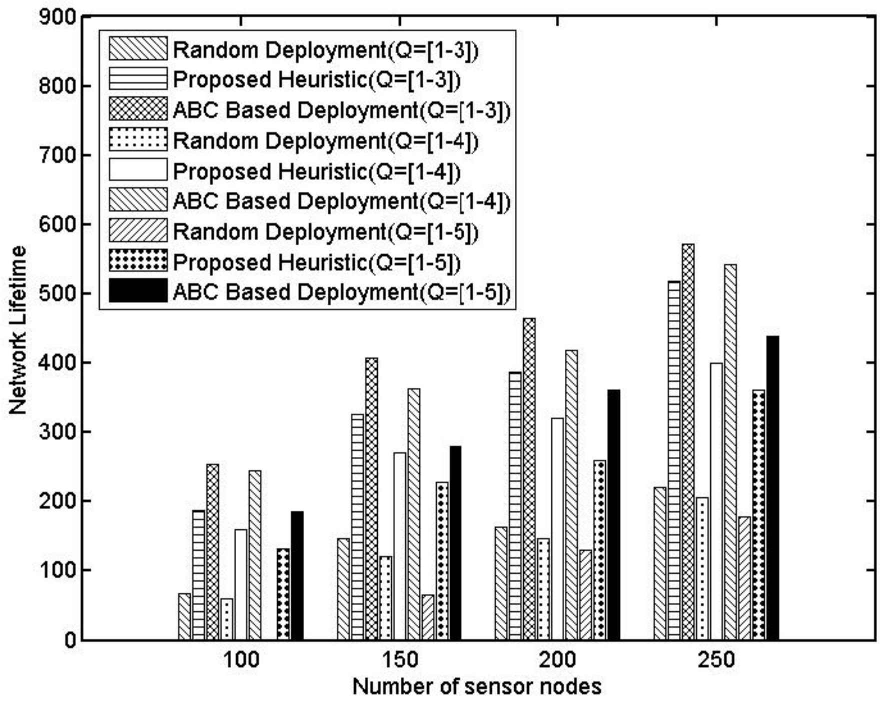


Figure 4.3: Network Lifetime for **Q**-coverage problem using random deployment, proposed heuristic and ABC algorithm

sensor with the greatest contribution that covers the critical target. Once a sensor has been selected, it is added to the current set cover, and all additionally covered targets are removed from the set of targets to be covered. When all targets are covered, the new set cover is formed. Our proposed method could achieve better network lifetime for simple coverage ($k = 1$) and higher values of k as shown in Figure 4.7.

Computing positions using ABC algorithm outperforms random deployment and heuristic. The heuristic proposed to schedule the sensor nodes meets the theoretical upper bound in all experimented cases.

4.6 Conclusion

We propose a heuristic to solve sensor node deployment problem when deterministic deployment of nodes with fixed sensing range is permitted. We also compute deployment locations for sensor nodes using artificial bee colony algorithm such that the network lifetime is maximum. Experimental results show that ABC based sensor node deploy-

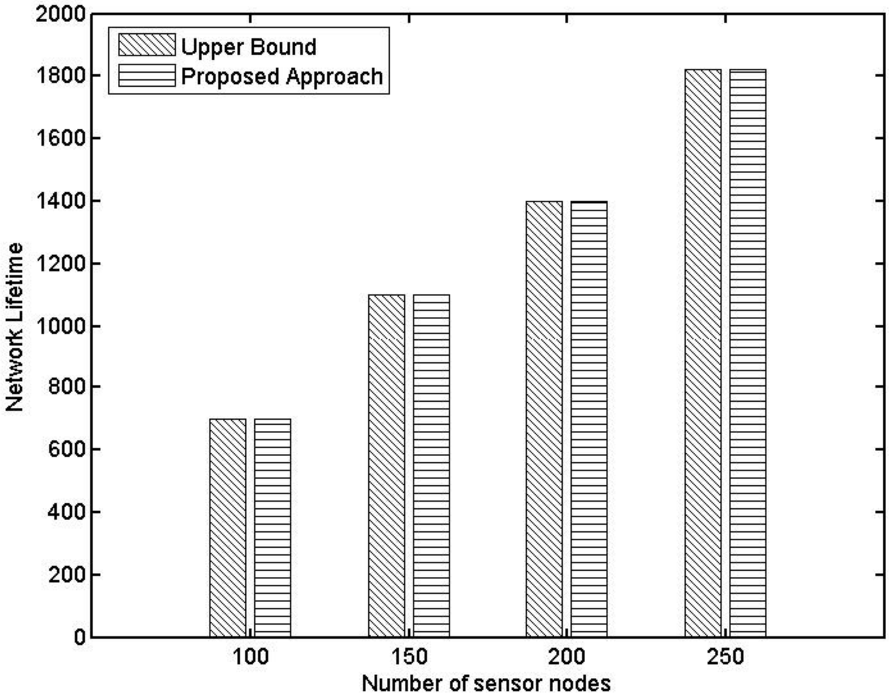


Figure 4.4: Network Lifetime for simple coverage problem after scheduling

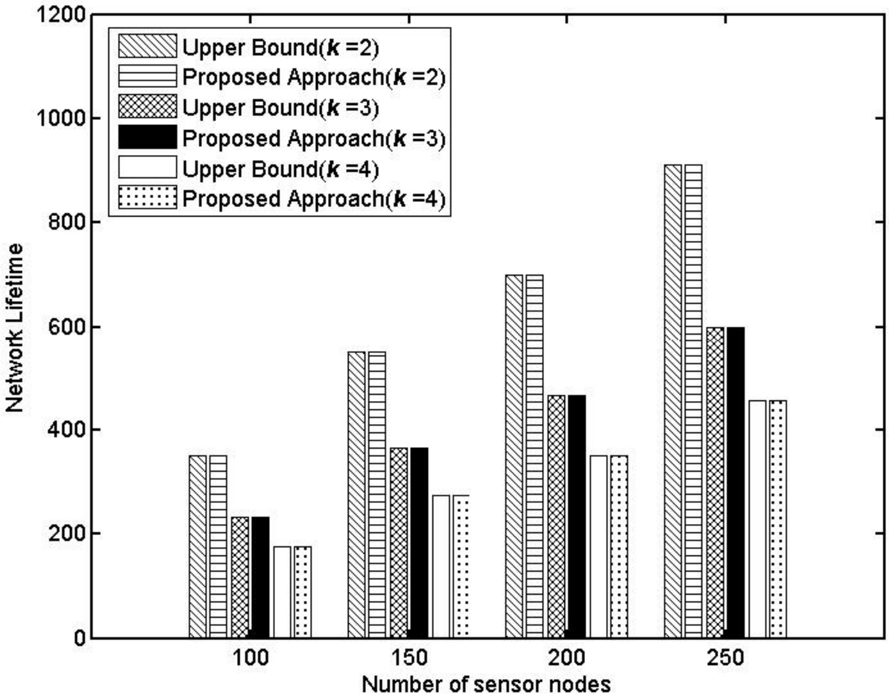


Figure 4.5: Network Lifetime for k-coverage problem after scheduling

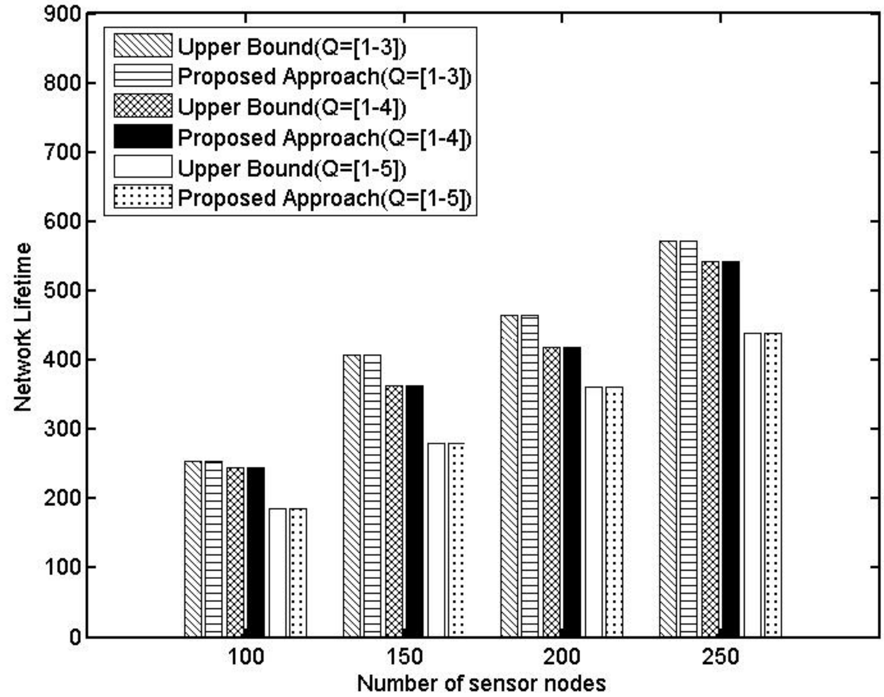


Figure 4.6: Network Lifetime for **Q**-coverage problem after scheduling

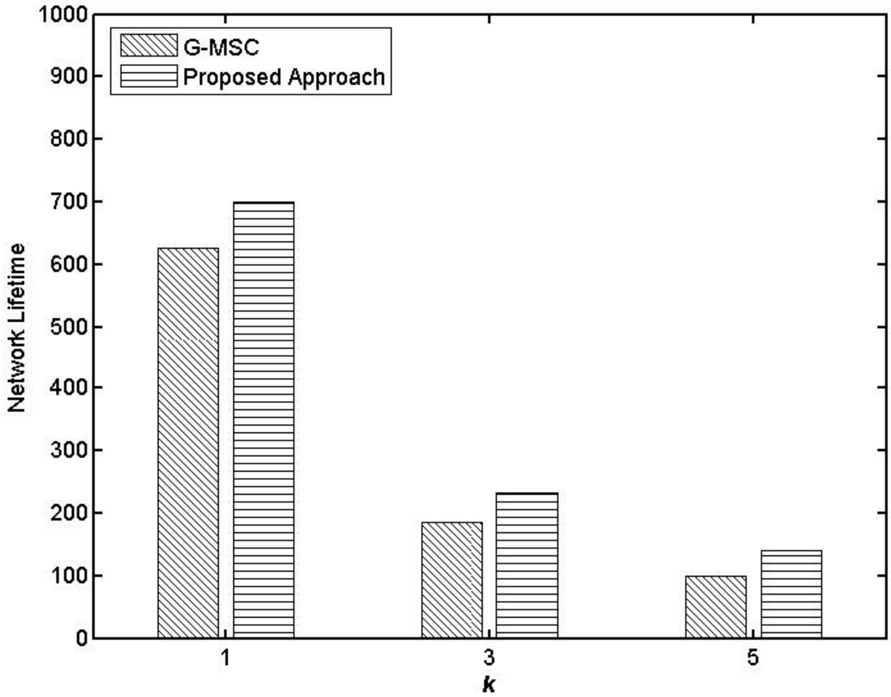


Figure 4.7: Comparison of proposed approach and G-MSC

ment is powerful than the heuristic. In order to avoid the battery drain of all nodes at a time, sensor node scheduling is done so that only minimum number of sensor nodes required for satisfying coverage requirement need to be active. The other nodes can be reserved for later use. This method helps to prolong the network lifetime. We use a heuristic which is powerful enough to schedule the sensor nodes in such a way that the network lifetime matches the theoretical upper bound of network lifetime. Network lifetime can be extended by using this method of deploying at optimal locations such that it achieves maximum theoretical upper bound and then scheduling them so as to achieve the theoretical upper bound.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The wide applicability of wireless sensor networks has opened up a lot of research challenges. Since sensor nodes are battery powered, designing energy efficient protocols become a major challenge. Efficient use of the available energy will lead to network lifetime maximization. We address deployment and scheduling problems to achieve different types of target coverage in wireless sensor networks.

The work described in this thesis looks at random deployment and deterministic deployment of sensors. Random deployment considers dense deployment; deterministic deployment considers sparse as well as dense deployments.

5.1 Summary of Contributions

We consider both deployment methods: random and deterministic. Chapter 2 addresses sensor scheduling for random deployment of sensor nodes for direct base station connected coverage as well as M-connected coverage. Chapter 3 looks at deterministic deployment of sensor nodes for both binary sensing model and probabilistic sensing model where the deployment locations are computed such that the required sensing range is minimum. Here, the number of sensor nodes is assumed to be limited. Chapter 4 addresses deterministic deployment where the sensor nodes have fixed sensing range. Sensor nodes are deployed at pre-computed locations and further scheduled. The thesis can be summarized as below:

1. *Given a region with dense randomly deployed sensor nodes, how to schedule the sensor nodes such that the network lifetime will be maximum with simple/k/***Q** *coverage (as required by the application) achieved?* A sensor scheduling scheme to prolong network lifetime is proposed such that at any time, only minimal number of sensor nodes that are required to satisfy the required coverage is active. The proposed heuristic solves

simple, k and **Q** coverage problems. The heuristic outperforms some existing methods and achieves the theoretical upper bound of network lifetime for all the experimented cases.

2. *Given a region with dense randomly deployed sensor networks, how to schedule the sensor nodes such that the network lifetime will be maximum with M-connected simple/k/***Q** *coverage (as required by the application) achieved?* We propose a heuristic which requires only a subset of sensor nodes which are M-connected and meets the required coverage to be active at a time. This heuristic performs better than CWGC.

3. *Given limited number of sensor nodes, where to deploy the nodes such that the required sensing range will be at minimum with simple/k/***Q** *coverage (as required by the application) achieved for a binary sensing model?* We use artificial bee colony algorithm to compute the optimal deployment locations followed by a sensitivity analysis test to check the robustness of the obtained solution. Restricting the sensing range in turn saves energy and enhances quality of sensing. Experimental results show that artificial bee colony algorithm performs better than PSO for this problem.

4. *Given limited number of sensor nodes, where to deploy the nodes such that the required sensing range will be at minimum with probabilistic simple/k/***Q** *coverage (as required by the application) achieved for a probabilistic sensing model?* We use artificial bee colony algorithm to compute the optimal deployment locations so that the required sensing range to achieve probabilistic simple/k/**Q** coverage is minimum.

5. *Given some sensor nodes (with fixed sensing range) that can be deterministically deployed, where to deploy them and how to schedule them so as to achieve the required target coverage level and maximize the network lifetime?* Initially we propose a heuristic to solve this problem. We also use artificial bee colony algorithm to compute the optimal deployment locations. Experimental results show that ABC algorithm performs better than the heuristic. After computing the optimal deployment locations, we use a heuristic to schedule the sensor nodes so that the theoretical upper bound of network lifetime could be achieved.

5.2 Future Work

There can be much more work done to improve the energy efficiency and to maximize the network lifetime of a wireless sensor network taking into consideration the different sensing models, communication models, coverage required as per application etc. Following are some extensions of this thesis we plan to make in the future:

1. *Study scheduling and deployment algorithms for deployments with multiple sinks.* In this thesis, we have assumed only one base station. In future, we propose to study these problems for deployments with multiple sinks.

2. *Modify routing algorithms to enhance network lifetime.* Existing routing algorithms can be modified for these problems to further increase the network lifetime.

3. *Study scheduling and deployment algorithms for regions with mobile targets.* We have assumed the targets to be static. We plan to modify these algorithms for mobile targets.

4. *Study connected coverage for deterministic sparse sensor node deployment.* In this thesis, we propose heuristics for M-connected coverage problem for random dense sensor node deployment. In future we plan to investigate on connected coverage problem for deterministic sparse sensor node deployment where the sensing range and communication range should be at the minimum with the targets being covered as required.

5. *Study probabilistic connected coverage for random sensor node deployment.* Connected coverage for dense random deployment of sensor nodes, where the sensing range of a sensor node is fixed, is addressed in this thesis. A method to deterministically deploy the sensor nodes to achieve probabilistic target coverage such that the required sensing range is minimum, is also proposed. Further extension would be to consider probabilistic connected coverage for random sensor node deployment.

6. *Study probabilistic sensor scheduling for deterministic deployment.* This thesis includes sensor scheduling mechanism of sensor nodes. It assumes a binary sensing model. Scheduling schemes to achieve probabilistic coverage for deterministic deployment need to be studied.

REFERENCES

1. **Abraham, A.**, **H. Guo**, and **H. Liu**, Swarm Intelligence: Foundations, Perspectives and Applications. *In Swarm Intelligent Systems, Studies in Computational Intelligence*. Springer, 2006.

2. **Ahmed, N.**, **S. S. Kanhere**, and **S. Jha**, Probabilistic Coverage in Wireless Sensor Networks. *In IEEE Conference on Local Computer Networks*, LCN '05. IEEE Computer Society, Washington, DC, USA, 2005. ISBN 0-7695-2421-4. URL `http://dx.doi.org/10.1109/LCN.2005.109`.

3. **Akyildiz, I. F.**, **W. Su**, **Y. Sankarasubramaniam**, and **E. Cayirci** (2002). Wireless sensor networks: a survey. *Computer Networks*, **38**, 393–422.

4. **Al-Karaki, J.** and **A. Kamal** (2004). Routing Techniques in Wireless Sensor Networks: A Survey. *IEEE Wireless Communications*, **11**(6), 6–28. ISSN 1536-1284.

5. **Ammari, H. M.** and **S. K. Das**, Clustering-Based Minimum Energy Wireless m-Connected k-Covered Sensor Networks. *In 5th European Conference on Wireless Sensor Networks*. 2008.

6. **Ammari, H. M.** and **S. K. Das**, Mission-Oriented k-Coverage in Mobile Wireless Sensor Networks. *In* **K. Kant**, **S. Pemmaraju**, **K. Sivalingam**, and **J. Wu** (eds.), *Distributed Computing and Networking*, volume 5935 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010*a*, 92–103.

7. **Ammari, H. M.** and **S. K. Das** (2010*b*). A Study of k-Coverage and Measures of Connectivity in 3D Wireless Sensor Networks. *IEEE Transactions on Computers*, **59**, 243–257. ISSN 0018-9340.

8. **Amundson, I.** and **X. D. Koutsoukos**, A Survey on Localization for Mobile Wireless Sensor Networks. *In 2nd International Conference on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT'09. Springer-Verlag, Berlin, Heidelberg, 2009. ISBN 3-642-04378-X, 978-3-642-04378-9. URL `http://dl.acm.org/citation.cfm?id=1813141.1813162`.

9. **Anastasi, G.**, **M. Conti**, **M. D. Francesco**, and **A. Passarella** (2009). Energy Conservation in Wireless Sensor Networks: A Survey. *Ad Hoc Networks*, **7**(3), 537–568. ISSN 1570-8705. URL `http://www.sciencedirect.com/science/article/pii/S1570870508000954`.

10. **Andersen, T.** and **S. Tirthapura**, Wireless Sensor Deployment for 3D Coverage with Constraints. *In Sixth International Conference on Networked Sensing Systems (INSS)*. 2009.

11. **Arai, S.**, **Y. Iwatani**, and **K. Hashimoto**, Fast and Optimal Sensor Scheduling for Networked Sensor Systems. *In IEEE Conference on Decision and Control*. 2008. ISSN 0191-2216.

12. **Bai, X.**, **Z. Yun**, **D. Xuan**, **T. H. Lai**, and **W. Jia**, Deploying Four-Connectivity and Full-Coverage Wireless Sensor Networks. *In INFOCOM*. 2008.

13. **Bartolini, N.**, **T. Calamoneri**, **E. Fusco**, **A. Massini**, and **S. Silvestri**, Snap and Spread: A Self-deployment Algorithm for Mobile Sensor Networks. *In* **S. Nikoletseas**, **B. Chlebus**, **D. Johnson**, and **B. Krishnamachari** (eds.), *Distributed Computing in Sensor Systems*, volume 5067 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-69169-3, 451–456.

14. **Bartolini, N.**, **T. Calamoneri**, **T. La Porta**, and **S. Silvestri**, Mobile Sensor Deployment in Unknown Fields. *In IEEE INFOCOM*. 2010. ISSN 0743-166X.

15. **Bellazreg, R.**, **M. Hamdi**, and **N. Boudriga**, On the Impact of Irregular Radio Propagation on Coverage Control and Sleep Scheduling in Wireless Sensor Networks. *In IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. 2010.

16. **Bouabdallah, F.**, **N. Bouabdallah**, and **R. Boutaba** (2009). On Balancing Energy Consumption in Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, **58**(6), 2909–2924. ISSN 0018-9545.

17. **Bredin, J. L.**, **E. D. Demaine**, **M. Hajiaghayi**, and **D. Rus**, Deploying Sensor Networks with Guaranteed Capacity and Fault Tolerance. *In 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05. ACM, New York, NY, USA, 2005. ISBN 1-59593-004-3. URL `http://doi.acm.org/10.1145/1062689.1062729`.

18. **Cardei, M.** and **D.-Z. Du** (2005). Improving Wireless Sensor Network Lifetime through Power Aware Organization. *ACM Wireless Networks*, **11**, 333–340.

19. **Cardei, M.**, **M. T. Thai**, **Y. Li**, and **W. Wu**, Energy-Efficient Target Coverage in Wireless Sensor Networks. *In IEEE Infocom*. 2005.

20. **Carter, B.** and **R. Ragade**, A Probabilistic Model for the Deployment of Sensors. *In IEEE Sensors Applications Symposium*. 2009.

21. **Cassandras, C. G.** and **W. Li** (2005). Sensor Networks and Cooperative Control. *European Journal of Control*, **11**(4-5), 436–463.

22. **Chachra, S.** and **M. Marefat**, Distributed Algorithms for Sleep Scheduling in Wireless Sensor Networks. *In IEEE International Conference on Robotics and Automation*. 2006. ISSN 1050-4729.

23. **Chang, C.-Y.**, **C.-T. Chang**, **Y.-C. Chen**, and **H.-R. Chang** (2009*a*). Obstacle-Resistant Deployment Algorithms for Wireless Sensor Networks. *IEEE Transactions on Vehicular Technology*, **58**(6), 2925–2941. ISSN 0018-9545.

24. **Chang, C.-Y.** and **H.-R. Chang** (2008). Energy-aware node placement, topology control and MAC scheduling for wireless sensor networks. *Computer Networks*, **52**, 2189–2204. ISSN 1389-1286. URL `http://dl.acm.org/citation.cfm?id=1383671.1384033`.

25. **Chang, C.-Y.**, **J.-P. Sheu**, **Y.-C. Chen**, and **S.-W. Chang** (2009*b*). An Oobstacle-Free and Power-Efficient Deployment Algorithm for Wireless Sensor Networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **39**, 795–806. ISSN 1083-4427. URL `http://dl.acm.org/citation.cfm?id=1656581.1656590`.

26. **Chang, J.-H.** and **L. Tassiulas** (2004). Maximum Lifetime Routing In Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, **12**(4), 609–619. ISSN 1063-6692.

27. **Chaudhary, M.** and **A. K. Pujari**, Q-Coverage Problem in Wireless Sensor Networks. *In ICDCN*. 2009.

28. **Chellappan, S.**, **X. Bai**, **B. Ma**, and **D. Xuan**, Sensor Networks Deployment Using Flip-Based Sensors. *In IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*. 2005.

29. **Chen, H.**, **M. Martins**, **P. Huang**, **H. C. So**, and **K. Sezaki**, Cooperative Node Localization for Mobile Sensor Networks. *In IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, volume 1. 2008.

30. **Chen, J.** and **X. Koutsoukos**, Survey on Coverage Problems in Wireless Ad Hoc Sensor Networks. *In IEEE SouthEastCon*. 2007.

31. **Cheng, Z.**, **M. Perillo**, and **W. Heinzelman** (2008). General Network Lifetime and Cost Models for Evaluating Sensor Network Deployment Strategies. *IEEE Transactions on Mobile Computing*, **7**(4), 484–497. ISSN 1536-1233.

32. **Chin, T.-L.**, Sensor Deployment for Collaborative Target Detection in the Presence of Obstacles. *In IEEE Global Telecommunications Conference*. 2009. ISSN 1930-529X.

33. **Choi, W.** and **S. K. Das** (2006). Coverage-adaptive random sensor scheduling for application-aware data gathering in wireless sensor networks. *Computer Communications*, **29**, 3467–3482. ISSN 0140-3664. URL `http://dl.acm.org/citation.cfm?id=1646670.1647174`.

34. **Costa, D. G.** and **L. A. Guedes** (2010). The Coverage Problem in Video-Based Wireless Sensor Networks: A Survey. *Sensors*, **10**, 8215–8247.

35. **de Oliveira, H.**, **A. Boukerche**, **E. Nakamura**, and **A. Loureiro** (2009). An Efficient Directed Localization Recursion Protocol for Wireless Sensor Networks. *IEEE Transactions on Computers*, **58**(5), 677–691. ISSN 0018-9340.

36. **Deng, J.**, **R. Han**, and **S. Mishra** (2006). Insens: Intrusion-Tolerant Routing for Wireless Sensor Networks. *Computer Communications*, **29**(2), 216–230.

37. **Dhillon, S.** and **K. Chakrabarty**, Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks. *In IEEE Wireless Communications and Networking*, volume 3. 2003. ISSN 1525-3511.

38. **Dorigo, M.** and **C. Blum** (2005). Ant colony optimization theory: A survey. *Theoretical Computer Science*, **344**, 243–278.

39. **Dorigo, M.**, **V. Maniezzo**, and **A. Colorni** (1996). Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **26**(1), 29–41. ISSN 1083-4419.

40. **dos Santos Coelho, L.**, **D. de Andrade Bernert**, and **V. Mariani**, A chaotic firefly algorithm applied to reliability-redundancy optimization. *In IEEE Congress on Evolutionary Computation (CEC)*. 2011. ISSN Pending.

41. **dos Santos Coelho, L.** and **V. C. Mariani** (2009). An improved harmony search algorithm for power economic load dispatch. *Energy Conversion and Management*, **50**(10), 2522–2526. ISSN 0196-8904. URL `http://www.sciencedirect.com/science/article/pii/S0196890409002192`.

42. **Du, X.** and **F. Lin**, Improving Sensor Network Performance by Deploying Mobile Sensors. *In 24th IEEE International Performance, Computing, and Communications Conference*. 2005. ISSN 1097-2641.

43. **Dulman, S. O.**, **A. Baggio**, **P. J. Havinga**, and **K. G. Langendoen**, A Geometrical Perspective on Localization. *In First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, MELT '08. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-189-7. URL `http://doi.acm.org/10.1145/1410012.1410032`.

44. **Eberhart, R.** and **J. Kennedy**, A New Optimizer Using Particle Swarm Theory. *In Sixth International Symposium on Micro Machine and Human Science*. 1995.

45. **Filipe, L.**, **M. Vieira**, **M. Augusto**, **M. Vieira**, **L. B. Ruiz**, **A. A. F. Loureiro**, **D. C. Silva**, and **A. O. Fern**, Efficient Incremental Sensor Network Deployment Algorithm. *In SBRC*. 2004.

46. **Fleischer, M.**, Foundations of Swarm Intelligence: From Principles to Practice. *In Conference on Swarming: Network Enabled C4ISR*. 2003.

47. **Ganesan, D.**, **R. Govindan**, **S. Shenker**, and **D. Estrin** (2001). Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, **5**, 11–25. ISSN 1559-1662. URL `http://doi.acm.org/10.1145/509506.509514`.

48. **Garnier, S.**, **J. Gautrais**, and **G. Theraulaz** (2007). The biological principles of swarm intelligence. *Swarm Intelligence*, 3–31.

49. **Geem, Z. W.**, **J. hoon Kim**, and **G. V. Loganathan** (2001). A New Heuristic Optimization Algorithm: Harmony Search. *Transactions of The Society for Modeling and Simulation International*, **76**, 60–68.

50. **Gu, Y.**, **H. Liu**, and **B. Zhao**, Target Coverage with QoS Requirements in Wireless Sensor Networks. *In IPC*. 2007.

51. **Gu, Y.**, **B.-H. Zhao**, **Y.-S. Ji**, and **J. Li** (2011). Theoretical Treatment of Target Coverage in Wireless Sensor Networks. *Journal of Computer Science and Technology*, **26**(1), 117–129. ISSN 1000-9000. URL `http://dx.doi.org/10.1007/s11390-011-1115-x`.

52. **Gupta, H.**, **S. R. Das**, and **Q. Gu**, Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. *In MobiHoc 2003*. 2003.

53. **Hassanein, H.** and **J. Luo**, Reliable Energy Aware Routing in Wireless Sensor Networks. *In Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems*. 2006.

54. **Hefeeda, M.** and **H. Ahmadi** (2010). Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, **21**(5), 579–593. ISSN 1045-9219.

55. **Hefeeda, M.** and **M. Bagheri**, Randomized k-Coverage Algorithms for Dense Sensor Networks. *In INFOCOM 2007*. 2007.

56. **Heinzelman, W. R.**, **A. Chandrakasan**, and **H. Balakrishnan**, Energy-Efficient Communication Protocol for Wireless Microsensor Networks. *In Hawaii International Conference on System Sciences*. 2000.

57. **Heo, N.** and **P. Varshney** (2005). Energy-Efficient Deployment of Intelligent Mobile Sensor Networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, **35**(1), 78–92. ISSN 1083-4427.

58. **Hohlt, B.**, **L. Doherty**, and **E. Brewer**, Flexible Power Scheduling for Sensor Networks. *In Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN '04. ACM, New York, NY, USA, 2004. ISBN 1-58113-846-6. URL `http://doi.acm.org/10.1145/984622.984653`.

59. **Howard, A.**, **M. J. Mataric**, and **G. S. Sukhatme**, Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem. *In 6th International Symposium on Distributed Autonomous Robotics Systems*. 2002.

60. **Huang, C.-F.** and **Y.-C. Tseng**, The Coverage Problem in a Wireless Sensor Network. *In 2nd ACM International Conference on Wireless Sensor Networks and Applications*. 2003.

61. **Ibrahim, A.**, **K. Seddik**, and **K. Liu**, Improving Connectivity via Relays Deployment in Wireless Sensor Networks. *In IEEE Global Telecommunications Conference*. 2007.

62. **Iwanicki, K.** and **M. van Steen**, On Hierarchical Routing in Wireless Sensor Networks. *In Information Processing in Sensor Networks, 2009. IPSN 2009. International Conference on*. 2009.

63. **Jain, E.** and **Q. Liang**, Sensor Placement and Lifetime of Wireless Sensor Networks: Theory and Performance Analysis. *In IEEE Global Telecommunications Conference*, volume 1. 2005.

64. **Kansal, A.**, **J. Hsu**, **S. Zahedi**, and **M. B. Srivastava** (2007). Power Management in Energy Harvesting Sensor Networks. *ACM Transactions on Embedded Computing Systems*, **6**. ISSN 1539-9087. URL `http://doi.acm.org/10.1145/1274858.1274870`.

65. **Karaboga, D.** and **B. Akay** (2009). A survey: algorithms simulating bee swarm intelligence. *Artificial Intelligence Review*, 61–85.

66. **Karaboga, D.** and **B. Basturk** (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, **39**, 459–471. ISSN 0925-5001. URL `http://dl.acm.org/citation.cfm?id=1295970.1296021`.

67. **Karaboga, D.** and **B. Basturk** (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, **8**, 687–697. ISSN 1568-4946. URL `http://dx.doi.org/10.1016/j.asoc.2007.05.007`.

68. **Karlof, C.** and **D. Wagner**, Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *In IEEE International Workshop on Sensor Network Protocols and Applications*. 2003.

69. **Kennedy, J.** and **R. Eberhart**, Particle Swarm Optimization. *In IEEE International Conference on Neural Networks*, volume 4. 1995.

70. **Keshavarzian, A.**, **H. Lee**, and **L. Venkatraman**, Wakeup Scheduling in Wireless Sensor Networks. *In 7th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '06. ACM, New York, NY, USA, 2006. ISBN 1-59593-368-9. URL `http://doi.acm.org/10.1145/1132905.1132941`.

71. **Kim, N.**, **S. Choi**, and **H. Cha**, Automated Sensor-specific Power Management for Wireless Sensor Networks. *In 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*. 2008.

72. **Kompella, R. R.** and **A. C. Snoeren**, Practical Lazy Scheduling in Sensor Networks. *In 1st international conference on Embedded networked sensor systems*, SenSys '03. ACM, New York, NY, USA, 2003. ISBN 1-58113-707-9. URL `http://doi.acm.org/10.1145/958491.958524`.

73. **Krause, A.**, **R. Rajagopal**, **A. Gupta**, and **C. Guestrin**, Simultaneous Placement and Scheduling of Sensors. *In International Conference on Information Processing in Sensor Networks*, IPSN '09. IEEE Computer Society, Washington, DC, USA, 2009. ISBN 978-1-4244-5108-1. URL `http://dl.acm.org/citation.cfm?id=1602165.1602183`.

74. **Krishnamachari, B.**, **D. Estrin**, and **S. Wicker**, Modelling Data-Centric Routing in Wireless Sensor Networks. *In IEEE INFOCOM*. 2002.

75. **Lazos, L.**, **R. Poovendran**, and **J. Ritcey**, On the Deployment of Heterogeneous Sensor Networks for Detection of Mobile Targets. *In 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops*. 2007.

76. **Li, Y.** and **S. Gao** (2008). Designing k-Coverage Schedules in Wireless Sensor Networks. *Journal of Combinatorial Optimization*, **15**, 127–146.

77. **Liao, W.-H.**, **K.-P. Shih**, and **Y.-C. Lee** (2008). A localization protocol with adaptive power control in wireless sensor networks. *Computer Communications*, **31**(10), 2496–2504. ISSN 0140-3664. URL `http://www.sciencedirect.com/science/article/pii/S014036640800193X`.

78. **Liu, C.**, **K. Wu**, and **V. King**, Randomized Coverage-Preserving Scheduling Schemes for Wireless Sensor Networks. *In NETWORKING'05*. 2005.

79. **Liu, M.**, **Y. Zheng**, **J. Cao**, **W. Lou**, **G. Chen**, and **H. Gong**, A Lightweight Scheme for Node Scheduling in Wireless Sensor Networks. *In* **J. Indulska**, **J. Ma**, **L. Yang**, **T. Ungerer**, and **J. Cao** (eds.), *Ubiquitous Intelligence and Computing*, volume 4611 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-73548-9, 579–588.

80. **Liu, X.** and **P. Mohapatra**, On the Deployment of Wireless Sensor Nodes. *In Third International Workshop on Measurement, Modelling, and Performance Analysis of Wireless Sensor Networks, in conjunction with MobiQuitous 2005*. 2005.

81. **Liu, Y.**, **Z. Yang**, **X. Wang**, and **L. Jian** (2010). Location, Localization, and Localizability. *Journal of Computer Science and Technology*, **25**, 274–297. ISSN 1000-9000. URL `http://dx.doi.org/10.1007/s11390-010-9324-2`. 10.1007/s11390-010-9324-2.

82. **Liu, Z.**, Maximizing Network Lifetime for Target Coverage Problem in Heterogeneous Wireless Sensor Networks. *In MSN*. 2007.

83. **Lu, M.**, **J. Wu**, **M. Cardei**, and **M. Li**, Energy-Efficient Connected Coverage of Discrete Targets in Wireless Sensor Networks. *In ICCNMC*. 2005.

84. **Ma, H.** and **Y. Liu** (2007). Some problems of directional sensor networks. *International Journal of Sensor Networks*, **2**, 44–52. ISSN 1748-1279. URL `http://dl.acm.org/citation.cfm?id=1359159.1359165`.

85. **Ma, M.** and **Y. Yang**, Adaptive Triangular Deployment Algorithm for Unattended Mobile Sensor Networks. *In* **V. Prasanna**, **S. Iyengar**, **P. Spirakis**, and **M. Welsh** (eds.), *Distributed Computing in Sensor Systems*, volume 3560 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-26422-4, 465–465.

86. **Makhoul, A.** and **C. Pham**, Dynamic Scheduling of Cover-Sets in Randomly Deployed Wireless Video Sensor Networks for Surveillance Applications. *In 2nd IFIP Conference on Wireless Days*, WD'09. IEEE Press, Piscataway, NJ, USA, 2009. ISBN 978-1-4244-5660-4. URL `http://dl.acm.org/citation.cfm?id=1834116.1834130`.

87. **Mao, X.**, **S. Tang**, **X. Xu**, **X.-Y. Li**, and **H. Ma** (2011). Energy-Efficient Opportunistic Routing in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, **22**(11), 1934–1942. ISSN 1045-9219.

88. **Meguerdichian, S.**, **F. Koushanfar**, **M. Potkonjak**, and **M. B. Srivastava**, Coverage Problems in Wireless Ad-hoc Sensor Networks. *In IEEE INFOCOM*. 2001.

89. **Meng, F.**, **H. Wang**, **G. Wei**, and **Z. Fan**, Energy-Efficient and Coverage-Specific Node Scheduling for Wireless Sensor Networks. *In ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile systems*, MSWIM '10. ACM, New York, NY, USA, 2010. ISBN 978-1-4503-0274-6. URL `http://doi.acm.org/10.1145/1868521.1868582`.

90. **Ming, Y.** and **V. W. S. Wong** (2007). An Energy-Efficient Multipath Routing Protocol for Wireless Sensor Networks. *International Journal of Communication Systems*, **20**(7), 747–766. URL `http://doi.wiley.com/10.1002/dac.843`.

91. **Mourad, F.**, **H. Snoussi**, **F. Abdallah**, and **C. Richard** (2009). Anchor-Based Localization via Interval Analysis for Mobile Ad-Hoc Sensor Networks. *IEEE Transactions on Signal Processing*, **57**(8), 3226–3239. ISSN 1053-587X.

92. **Mulligan, R.** and **H. M. Ammari** (2010). Coverage in Wireless Sensor Networks: A Survey. *Network Protocols and Algorithms*, **2**(2), 27–53.

93. **Onur, E.**, **C. Ersoy**, and **H. DeliÃğ** (2005). Quality of Deployment in Surveillance Wireless Sensor Networks. *International Journal of Wireless Information Networks, special issue on ad hoc sensor networks*, **12**, 61–67.

94. **Pantazis, N.** and **D. Vergados** (2007). A Survey on Power Control Issues in Wireless Sensor Networks. *IEEE Communications Surveys Tutorials*, **9**(4), 86–107. ISSN 1553-877X.

95. **Poe, W. Y.** and **J. B. Schmitt**, Node Deployment in Large Wireless Sensor Networks: Coverage, Energy Consumption, and Worst-Case Delay. *In Asian Internet Engineering Conference*, AINTEC '09. ACM, New York, NY, USA, 2009. ISBN 978-1-60558-614-4. URL `http://doi.acm.org/10.1145/1711113.1711127`.

96. **Raicu, I.**, **L. Schwiebert**, **S. Fowler**, and **S. K. S. Gupta** (2005). Local Load Balancing for Globally Efficient Routing in Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, **1**, 163–185.

97. **Raman, B.** and **K. Chebrolu** (2008). Censor Networks: A Critique of "Sensor Networks" from a Systems Perspective. *ACM SIGCOMM Computer Communication Review*, **38**(3), 75–78.

98. **Schurgers, C.** and **M. Srivastava**, Energy Efficient Routing in Wireless Sensor Networks. *In IEEE Military Communications Conference*, volume 1. 2001.

99. **Shen, X.**, **J. Chen**, and **Y. Sun**, Grid Scan: A Simple and Effective Approach for Coverage Issue in Wireless Sensor Networks. *In IEEE International Conference on Communications*, volume 8. 2006. ISSN 8164-9547.

100. **Sheu, J.-P.**, **W.-K. Hu**, and **J.-C. Lin** (2010). Distributed Localization Scheme for Mobile Sensor Networks. *IEEE Transactions on Mobile Computing*, **9**(4), 516–526. ISSN 1536-1233.

101. **Sichitiu, M.** and **V. Ramadurai**, Localization of Wireless Sensor Networks with a Mobile Beacon. *In IEEE International Conference on Mobile Ad-hoc and Sensor Systems*. 2004.

102. **Sinha, A.** and **A. Chandrakasan** (2001). Dynamic Power Management in Wireless Sensor Networks. *IEEE Design Test of Computers*, **18**(2), 62–74. ISSN 0740-7475.

103. **Slijepcevic, S.** and **M. Potkonjak**, Power Efficient Organization of Wireless Sensor Networks. *In IEEE International Conference on Communications*. 2001.

104. **Tan, G.**, **S. Jarvis**, and **A.-M. Kermarrec** (2009). Connectivity-Guaranteed and Obstacle-Adaptive Deployment Schemes for Mobile Sensor Networks. *IEEE Transactions on Mobile Computing*, **8**(6), 836–848.

105. **Tang, J.**, **B. Hao**, and **A. Sen** (2006). Relay Node Placement in Large Scale Wireless Sensor Networks. *Computer Communications*, **29**, 490–501. ISSN 0140-3664. URL `http://dl.acm.org/citation.cfm?id=1646656.1646891`.

106. **Teng, G.**, **K. Zheng**, and **W. Dong** (2009). Adapting Mobile Beacon-Assisted Localization in Wireless Sensor Networks. *Sensors*, **9**(4), 2760–2779. ISSN 1424-8220. URL `http://www.mdpi.com/1424-8220/9/4/2760/`.

107. **Tilak, S.**, **N. B. Abu-Ghazaleh**, and **W. Heinzelman** (2002). A Taxonomy of Wireless Micro-Sensor Network Models. *ACM Mobile Computing and Communications Review*, **6**, 28–36.

108. **Wang, F.**, **L. Qiu**, and **S. S. Lam** (2007). Probabilistic Region-based Localization for Wireless Networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, **11**, 3–14. ISSN 1559-1662. URL `http://doi.acm.org/10.1145/1234822.1234826`.

109. **Wang, J.**, **R. Ghosh**, and **S. Das** (2010). A survey on sensor localization. *Journal of Control Theory and Applications*, **8**, 2–11. ISSN 1672-6340. URL `http://dx.doi.org/10.1007/s11768-010-9187-7`. 10.1007/s11768-010-9187-7.

110. **Wang, L.** and **Q. Xu** (2010). GPS-Free Localization Algorithm for Wireless Sensor Networks. *Sensors*, **10**(6), 5899–5926. ISSN 1424-8220. URL `http://www.mdpi.com/1424-8220/10/6/5899/`.

111. **Wang, Y.-C.**, **C.-C. Hu**, and **Y.-C. Tseng**, Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks. *In First International Conference on Wireless Internet*. 2005.

112. **Wu, C.-H.**, **W. Sheng**, and **Y. Zhang**, Mobile Sensor Networks Self Localization based on Multi-dimensional Scaling. *In IEEE International Conference on Robotics and Automation*. 2007. ISSN 1050-4729.

113. **Wu, J.** and **S. Yang**, Smart: A Scan-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks. *In IEEE INFOCOM 2005*, volume 4. 2005. ISSN 0743-166X.

114. **Wu, J.** and **S. Yang** (2007). Optimal Movement-Assisted Sensor Deployment and Its Extensions in Wireless Sensor Networks. *Simulation Modelling Practice and Theory*, **15**(4), 383–399.

115. **Wu, K.**, **Y. Gao**, **F. Li**, and **Y. Xiao** (2005). Lightweight Deployment-Aware Scheduling for Wireless Sensor Networks. *Mobile Networks and Applications*, **10**, 837–852. ISSN 1383-469X. URL `http://dx.doi.org/10.1007/s11036-005-4442-8`.

116. **Xiao, J.-J.**, **S. Cui**, **Z.-Q. Luo**, and **A. Goldsmith** (2006). Power Scheduling of Universal Decentralized Estimation in Sensor Networks. *IEEE Transactions on Signal Processing*, **54**(2), 413–422. ISSN 1053-587X.

117. **Yang, G.** and **D. Qiao**, Multi-Round Sensor Deployment for Guaranteed Barrier Coverage. *In IEEE INFOCOM*. 2010. ISSN 0743-166X.

118. **Yang, X.-S.**, Firefly algorithms for multimodal optimization. *In Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications*, SAGA'09. Springer-Verlag, Berlin, Heidelberg, 2009. ISBN 3-642-04943-5, 978-3-642-04943-9. URL `http://dl.acm.org/citation.cfm?id=1814087.1814105`.

119. **Yao, Y.** and **G. Giannakis** (2005). Energy-Efficient Scheduling for Wireless Sensor Networks. *IEEE Transactions on Communications*, **53**(8), 1333–1342. ISSN 0090-6778.

120. **Yen, L.-H.** and **Y.-M. Cheng** (2009). Range-Based Sleep Scheduling (RBSS) for Wireless Sensor Networks. *Wireless Personal Communications*, **48**, 411–423. ISSN 0929-6212. URL `http://dl.acm.org/citation.cfm?id=1485093.1485095`.

121. **Yen, L.-H.**, **C. W. Yu**, and **Y.-M. Cheng** (2006). Expected k-Coverage in Wireless Sensor Networks. *Ad Hoc Networks*, **4**, 636–650.

122. **Yick, J.**, **B. Mukherjee**, and **D. Ghosal** (2008). Wireless sensor network survey. *Computer Networks*, **52**, 2292–2330. ISSN 1389-1286. URL `http://dl.acm.org/citation.cfm?id=1389582.1389832`.

123. **Younis, M.** and **K. Akkaya** (2008). Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey. *Ad Hoc Networks*, **6**, 621–655. ISSN 1570-8705. URL `http://dl.acm.org/citation.cfm?id=1346362.1346629`.

124. **Yu, X.**, **S. Mehrotra**, and **N. Venkatasubramanian**, Sensor Scheduling for Aggregate Monitoring in Wireless Sensor Networks. *In 19th International Conference on Scientific and Statistical Database Management*, SSDBM '07. IEEE Computer Society, Washington, DC, USA, 2007. ISBN 0-7695-2868-6. URL `http://dx.doi.org/10.1109/SSDBM.2007.42`.

125. **Zhang, D.**, **M. Xu**, **S. Wang**, and **B. Zhang**, Minimization of the Redundant Coverage for Dense Wireless Sensor Networks. *In 3rd International Conference on Embedded Software and Systems*. 2007.

126. **Zhang, G.**, **K. Sivanand**, **F. Chin**, and **K. C. Chung**, Quality of Service and Coverage Optimization in Wireless Sensor Networks. *In IEEE Vehicular Technology Conference*. 2008. ISSN 1550-2252.

127. **Zhang, S.**, **J. Cao**, **C. Li-Jun**, and **D. Chen** (2010). Accurate and Energy-Efficient Range-Free Localization for Mobile Sensor Networks. *IEEE Transactions on Mobile Computing*, **9**(6), 897–910. ISSN 1536-1233.

128. **Zhao, Q.** and **M. Gurusamy** (2008). Lifetime Maximization for Connected Target Coverage in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, **16**(6), 1378–1391.

129. **Zhou, Z.**, **S. Das**, and **H. Gupta**, Connected K-Coverage Problem in Sensor Networks. *In International Conference on Computer Communications and Networks (IC3N*. 2004.

130. **Zou, Y.** and **K. Chakrabarty**, Sensor Deployment and Target Localization Based on Virtual Forces. *In Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. INFOCOM 2003.*, volume 2. 2003. ISSN 0743-166X.

131. **Zou, Y.** and **K. Chakrabarty** (2005). A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks. *IEEE Transactions on Computers*, **54**(8), 978–991. ISSN 0018-9340.

LIST OF PAPERS BASED ON THESIS

1. Mini, S., Udgata, S.K., Sabat, S.L. Sensor Deployment and Scheduling for Target Coverage Problem in Wireless Sensor Networks *(communicated)* .

2. Mini, S., Udgata, S.K., Sabat, S.L. M-Connected Simple/k/Q-Coverage Problem in Wireless Sensor Networks *ISRN Sensor Networks (in press)*.

3. Mini, S., Udgata, S.K., Sabat, S.L. A Heuristic to Maximize Network Lifetime for Target Coverage Problem in Wireless Sensor Networks *Ad hoc & Sensor Wireless Networks*, 13(3-4), 251-269, (2011).

4. Mini, S., Udgata, S.K., Sabat, S.L. Sensor Deployment for Probabilistic Target k-Coverage using Artificial Bee Colony Algorithm *Swarm, Evolutionary, and Memetic Computing (SEMCCO 2011). LNCS*, 7076, 654-661, (2011).

5. Mini, S., Udgata, S.K., Sabat, S.L. Artificial Bee Colony based Sensor Deployment Algorithm for Target Coverage Problem in 3-D Terrain *Distributed Computing and Internet Technology (ICDCIT 2011). LNCS*, 6536, 313-324, (2011).

6. Mini, S., Udgata, S.K., Sabat, S.L. Sensor Deployment in 3-D Terrain using Artificial Bee Colony Algorithm *Swarm, Evolutionary, and Memetic Computing (SEMCCO 2010). LNCS*, 6466, 424-431, (2010).

7. Udgata, S.K., Sabat, S.L., Mini, S. Sensor Deployment in Irregular Terrain using ABC Algorithm *Proc. of IEEE BICA'09*, 296-300, (2009).