Thesis for the Degree of Doctor of Philosophy

Coverage-driven Energy-efficient Deployment and Self-organization in Sensor Networks

Xiaoling Wu

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

August, 2008

Coverage-driven Energy-efficient Deployment and Self-organization in Sensor Networks

Xiaoling Wu

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

August, 2008

Coverage-driven Energy-efficient Deployment and Self-organization in Sensor Networks

by

Xiaoling Wu

Supervised by

Prof. Sungyoung Lee, Ph.D

Submitted to the Department of Computer Engineering

and the Faculty of Graduate school of

Kyung Hee University in partial fulfillment

of the requirement for Degree of

Doctor of Philosophy

Dissertation Committee:

Chairman, Choong Seon Hong, Ph.D ________________

Een-Kee Hong, Ph.D ________________

Jinsung Cho, Ph.D ________________

Sungwon Lee, Ph.D ________________

Sungyoung Lee, Ph.D ________________

Brian J. d’Auriol, Ph.D ________________

Abstract

Due to small size, low cost, and many other attractive features of sensor nodes, wireless sensor networks (WSNs) have been adapted to a vast array of applications in both military and civil sectors, such as military surveillance, smart homes, and remote environment monitoring.

Modern day requirements of various applications necessitate large scale sensor deployment in as efficient a way as possible. In this thesis we investigate the problem of increasing the sensor network's sense and detect sensitivity in both space and time. All of the existing literature surveyed as part of this research indicates that all of the solutions address singular aspects of this problem. My research is novel in that I address this problem holistically. The research goal is to maximize coverage and minimize energy consumption in sensor deployment and self-organization in WSNs.

The thesis first introduces a comprehensive taxonomy for WSNs deployment and self-organization. Three sensor relocation algorithms are proposed to match the mobility degree of sensor nodes, particle swarm optimization based algorithm (PSOA), relay shift based algorithm (RSBA) and energy efficient fuzzy optimization algorithm (EFOA). PSOA regards the sensors in the network as a swarm, and reorganizes the sensors by the particle swarm optimization (PSO) algorithm, in the full sensor mobility case. RSBA and EFOA assume relatively limited sensor mobility to further reduce energy consumption. We propose a novel method for the redeployment of mobile nodes in a hybrid sensor network consisting of a collection of both static nodes and mobile nodes. In such a sensor network, the locomotion ability of mobile nodes helps the autonomous deployment to enhance the network coverage. An optimal decision of a

sensor node moving direction is made based on Analytical Hierarchy Process (AHP). Four factors contributing to the optimal deployment are considered and they are coverage hole, obstacle avoidance, hot spot, and the boundary effect, respectively. I also propose a network maintenance strategy in the post-deployment phase based on the sensor node importance level ranking. Simulation results show that our approach not only achieves fast and stable deployment but also greatly improves the network coverage and prolongs the lifetime.

In order to enable efficient self-organiztion in static sensor networks, I propose a VQ-LBG based approach for cluster formation in WSNs. The most distinguishing feature of the proposed method is that both energy efficient cluster formation and fast data compression can be guaranteed. Experiment shows its great improvement over other related methods. The thesis then presents a sleep scheduling scheme for balancing energy consumption rates in a single hop cluster based network using AHP. Three factors are considered contributing to the optimal nodes scheduling decision and they are the distance to cluster head, residual energy, and sensing coverage overlapping, respectively. I also propose an integrated sleep scheduling and routing scheme for WSNs by AHP. The sleep scheduling is redesigned to adapt the multi-hop case. For the proposed routing protocol, the distance to the destination location, remaining battery capacity, and queue size of candidate sensor nodes in the local communication range are taken into consideration for next hop relay node selection.

In summary, this thesis provides an important link between the crucial problems of coverage, connectivity, energy management, and self-organization in wireless sensor networks. It is expected to lead to even more efficient protocols for node deployment, state management, recovery and routing protocols for energy-constrained sensor networks.

Table of Contents

List of Figures

List of Tables

Nomenclature

PSO	Particle swam optimization
GA	Genetic algorithm
AHP	Analytical Hierarchy Process
QoS	Quality of Service
A_i	the area covered by the i^{th} node
N	the total number of nodes
A	the area of the ROI
P	a grid point
s	a sensor node
$d_{ij}(x, y)$	the Euclidean distance between s at (x, y) and P at (i, j)
E_{elec}	energy dissipated per bit to run the transmitter or the receiver circuit
ε_{fs}	amplifier constant
ε_{mp}	amplifier constant
d	distance
E_{elec}	electronics energy
$S_i(P_j)$	the probability that node i can sense grid point j
$C(P_j)$	the probability grid point j is sensed by the whole network
C	coverage
CR	consistency ratio, which is defined as the ratio of CI to RI
RI	random index

λ_{max}	eigenvalue
CI	consistency index
W	eigenvector
a_{ij}	the ratio of the i^{th} factor weight to the j^{th} factor weight

Chapter 1 Introduction

1.1 Motivation

1.1.1 Wireless sensor networks

Due to advances in wireless communications and electronics over the last few years, the development of networks of low-cost, low-power, multifunctional sensors has received increasing attention [1, 2, 3, 4, 5, 6, 7, 8]. These sensors are small in size and able to sense, process data, and communicate with each other, typically over an RF (radio frequency) channel. A wireless sensor network (WSN) is designed to observe environment, collect and process data, and transmit sensed information to interested users. Basic features of sensor networks are:

· Self-organizing capabilities

· Short-range broadcast communication and multihop routing

· Dense deployment and cooperative effort of sensor nodes

· Frequently changing topology due to fading and node failures

· Limitations in energy, transmit power, memory, and computing power

These characteristics, particularly the last three, make WSNs different from other wireless ad hoc or mesh networks. WSNs are important for a number of applications such as coordinated target detection and localization, surveillance, ubiquitous health care and environmental monitoring. Break-throughs in miniaturization, hardware design techniques, and system software have led to cheaper sensors and fueled recent advances in WSNs [9].

A typical sensor network consists of a large number of nodes. A sensor node integrates hardware and software for sensing, data processing, and communication. Sensor nodes can be deployed readily in large numbers in various environments.

- Communication rage: the maximum distance that a node can communicate with another node is characterized by the communication unit on the sensor node, e.g., for the RF sensors used in the Berkeley mote [10], the maximum communication range is approximately 100 ft.
- Sensing coverage: the sensing area of a sensor node depends on the type of physical sensors used on that node, e.g., a range sensor such as the Polaroid 6500 ultrasonic ranging module, which is commonly used in robotics applications, is able to detect a target from 6 inches away up to a distance of 35 feet [11].

An important consideration in sensor networks is the amount of energy required for sensing, computation, and communication. Except for physical damage and system failures (e.g., software bugs), the lifetime of a sensor node depends exclusively on battery capacity; hence information exchange and data dissemination must be carried out using efficient communication protocols. A number of system architectures, communication protocols, and data aggregation algorithms have been proposed in the literature for retrieving and processing sensed data with low energy consumption [12, 13, 14, 15, 16, 17, 18].

WSNs are typically organized in an ad hoc manner, e.g., through random sensor deployment and ad hoc networking protocols. Nevertheless, a number of methods have recently been developed to organize the sensor network in hierarchical clusters [19, 20] to improve the sensing coverage [21, 41, 22, 23, 24, 25, 26, 27] and reduce the energy consumption in information processing [28, 29, 30].

1.1.2 Coverage-driven sensor deployment

In many applications such as remote surveillance, manual deployment of sensor nodes is infeasible. Nodes may be deployed from aircrafts, for example. As a result, the sensing and communication coverage cannot be controlled. In early works, researchers assume that sensor nodes are static and that a large number of redundant nodes are deployed in order to achieve a desired level of coverage. This however may introduce high cost and still cannot guarantee coverage, especially in the presence of obstacles. Recently, mobile sensors [31] have been developed and are expected to be applied in practice shortly. This motivates us to adopt mobile sensor nodes to address the coverage problem. The introduction of mobile sensor nodes, however, brings forth new issues such as how mobile nodes interact with each other and with static nodes, and brings forth new cost, performance, and communication tradeoffs. I have investigated these topics in a stepwise manner. First, I assume a system with all mobile sensor nodes, categorize their mobility into full and limited, and study how the nodes can effectively collaborate to improve coverage in a distributed fashion. Then, I consider a more realistic system including a mixed set of mobile and static sensor nodes, that is, hybrid sensor networks.

To deploy mobile sensors from where they are initially distributed, three sensor relocation algorithms are proposed to match the mobility degree of sensor nodes, particle swarm optimization based algorithm (PSOA), relay shift based algorithm (RSBA) and energy efficient fuzzy optimization algorithm (EFOA). PSOA regards the sensors in the network as a swarm, and reorganizes the sensors by the particle swarm optimization (PSO) algorithm, in the full sensor mobility case. RSBA and EFOA assume relatively limited sensor mobility, i.e., the movement distance is bounded by a threshold, to further reduce energy consumption. Based on these algorithms, individual nodes independently calculate their required movement to increase coverage. The different types of the calculation algorithms, tuned by multiple parameters, allow users to choose different levels of mobility, coverage, energy consumption and depending on system requirements.

With a hybrid sensor network that consists of both mobile and static sensors, it becomes a NP-hard problem to place mobile sensors to maximize sensing coverage. To tackle this NP-hard problem in a distributed fashion, I have proposed analytic hierarchy process (AHP) based deployment algorithm [32] to move sensors. It is different from the previous methods since it not only incorporates various environmental factors such as hot spot and obstacles in the design but also provides the optimal decision for mobile nodes movement. After an initial random deployment of static sensors, a certain amount of mobile nodes are deployed randomly into the monitored environment without changing the existing deployment of static sensor nodes. In order to increase the network coverage and uniformity, the mobile nodes are relocated according to the proposed scheme. The decision of moving direction of each mobile node is made according to AHP method, in which a set of criteria is evaluated and the optimal alternative is selected.

1.1.3 Self-organization

Wireless sensor networks can be deployed in inhospitable terrain or in hostile environments to provide continuous monitoring and information processing for a wide variety of applications [33, 34].

The network topology control for large number of randomly placed sensors was studied in the recent years emphasizing the limited battery power. Generally, there are three methods that can be considered as possible networking protocols: direct communication, multi-hop routing, and clustering. In order to send information from a very high number of sensor nodes to the base station, it is necessary and economical to group sensors into clusters to prevent redundant information transmission and prolong network lifetime. Given the parameters for variation of energy consumption in the nodes, there are some main problems: How many sensors should be connected to each cluster head (CH), how many clusters are needed, how to select CH, and where each CH should be positioned. Another typical challenge is that due to the resource limitation of sensor nodes (CPU, memory, bandwidth, and energy), the collected information from sensor nodes in the cluster has to be compressed quickly and precisely for transmission.

Many wireless sensor nodes have low cost and small form factors; therefore, they can be deployed in large numbers with high redundancy. In this way, the network can be made fault-tolerant. Since nodes are deployed in a redundant fashion, not every node in the sensor network needs to be active for sensing and communication all the time. By selecting only a subset of nodes to be active and keeping the remaining nodes in a sleep state, the energy consumption of the network is reduced, thereby extending the operational lifetime of the sensor network. Fewer active nodes also places less demand on the limited network bandwidth. Sleeping nodes serve as backup nodes to replace the failing active nodes without affecting the quality of service in the sensor network.

The selection of active nodes must be carried out with coverage and connectivity as important considerations. A procedure for selecting active nodes should provide the highest possible coverage of the sensor field, and it should also ensure network connectivity for routing and information dissemination.

I propose a sleep scheduling scheme for balancing energy consumption rates in a single hop cluster based network using AHP [35]. I consider three factors contributing to the optimal nodes scheduling decision and they are the distance to CH, residual energy, and sensing coverage overlapping, respectively. I also propose an integrated sleep scheduling and geographical multi-path routing scheme for WSNs by AHP. The sleep scheduling is redesigned to adapt the multi-hop case. For the proposed routing protocol [36], the distance to the destination location, remaining battery capacity, and queue size of candidate sensor nodes in the local communication range are taken into consideration for next hop relay node selection. It can reduce the packet loss rate and link failure rate since the buffer capacity is considered.

1.2 Problem statement

My research goal is to maximize coverage, i.e. sensing coverage, and to minimize energy consumption in sensor deployment and self-organization in WSNs.

Modern day requirements of various applications necessitate large scale sensor deployment in as efficient a way as possible. So in this thesis I investigate the problem of increasing the sensor network's sense and detect sensitivity in both space and time.

- **Space:** increase the coverage area so that unsensed regions can become sensed, subject to criteria of mobility and communication energy consumption.
- **Time:** prolong the sensor networks' sensing capability by conserving energy.

All of the existing literature surveyed as part of this research indicates that all of the solutions address singular aspects of this problem. My research is novel in that I address this problem holistically, that is, I address multiple facets of this problem as part of the whole context of WSNs.

1.3 Contributions

In this thesis, I focus on two aspects of wireless sensor networks: coverage-driven sensor deployment, and energy-efficient self-organization. The contributions are listed as follows:

- Three sensor relocation algorithms are proposed to match the mobility degree of sensor nodes, particle swarm optimization based algorithm (PSOA), relay shift based algorithm (RSBA) and energy efficient fuzzy optimization algorithm (EFOA). PSOA regards the sensors in the network as a swarm, and reorganizes the sensors by the particle swarm optimization (PSO) algorithm, in the full sensor mobility case. RSBA and EFOA assume relatively limited sensor mobility, i.e., the movement distance is bounded by a threshold, to further reduce energy consumption.
- Propose a network maintenance strategy in the post-deployment phase based on the sensor node importance level ranking to improves the network coverage and prolongs the lifetime.
- Propose a novel method for the redeployment of mobile nodes in a hybrid sensor network consisting of a collection of both static nodes and mobile nodes. In such a sensor network, the locomotion ability of mobile nodes helps the autonomous deployment to enhance the network coverage. An optimal decision of a sensor node moving direction is made based on AHP. Four factors contributing to the optimal deployment are considered and they are coverage hole, obstacle avoidance, hot spot, and the boundary effect, respectively.

- In order to enable energy-efficient self-organization, I propose a VQ-LBG based approach for cluster formation in WSN. The most distinguishing feature of the proposed method is that both energy efficient cluster formation and fast data compression can be guaranteed.
- Propose a sleep scheduling scheme for balancing energy consumption rates in a single hop cluster based network using AHP. Three factors are considered contributing to the optimal nodes scheduling decision and they are the distance to cluster head (CH), residual energy, and sensing coverage overlapping, respectively.

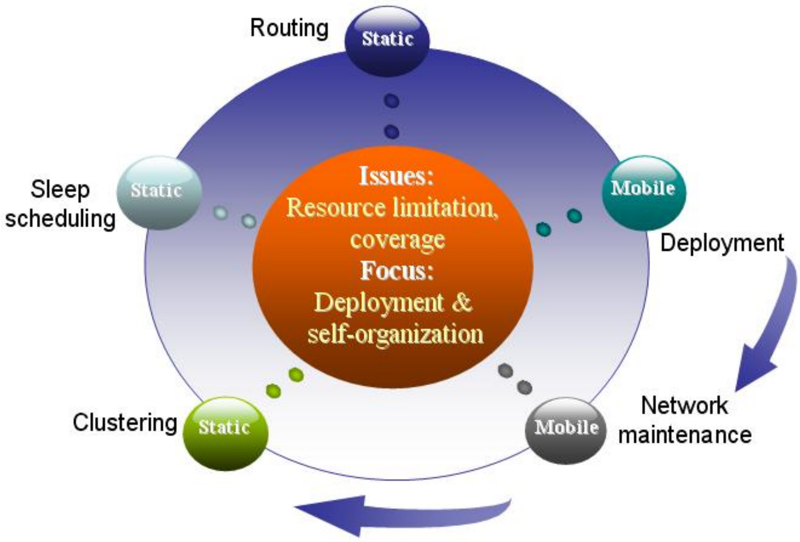


Fig. 1.1 Research diagram

- Propose an integrated sleep scheduling and geographical multi-path routing scheme for WSNs by AHP. The sleep scheduling is redesigned to adapt the multi-hop case. For the proposed routing protocol, the distance to the destination location, remaining battery capacity, and queue size of candidate sensor nodes in the local communication range are taken into consideration for next hop relay node selection.

In summary, this thesis provides an important link between the crucial problems of coverage, energy management, connectivity, and self-organization in wireless sensor networks, as shown in Fig. 1.1. It is expected to lead to even more efficient protocols for node deployment, state management, recovery and routing protocols for energy-constrained sensor networks.

1.4 Thesis organization

The rest of my thesis is organized as follows. I survey related work and introduce a comprehensive taxonomy for mobility-assisted sensor deployment and self-organization in Chapter 2. I present the mobility-assisted relocation algorithms to deploy both pure mobile sensor networks and hybrid sensor networks and also a maintenance strategy in post-deployment stage in Chapter 3. Two energy efficient self-organization schemes that are vector-quantization based clustering and integrated sleep scheduling and routing are presented in Chapter 4. In Chapter 5, I present the performance evaluation for the proposed algorithms, and finally conclude the thesis in Chapter 6.

Chapter 2 Taxonomy and related work

2.1 Taxnomy

Due to the unfamiliar nature of such environments, deployment of sensors has become a challenging problem and has received considerable attention recently. Sensor deployment cannot be performed manually when the environment is unknown or inhospitable such as remote inaccessible areas, disaster fields and toxic urban regions. To scatter sensors by aircraft is one possible solution. However, using this scheme, the actual landing position cannot be predicted due to the existence of wind and obstacles such as trees and buildings. Consequently, the coverage may not be able to satisfy the application requirements. Some researchers suggest simply deploying large amount of static sensors to increase coverage; however it often ends up harming the performance of the network [37]. Moreover, there are situations where sensor deployment is restricted by the environment, for example, during in-building toxic-leaks detection [38] chemical sensors must be placed inside a building from the entrance of the building. In such cases it is necessary to take advantage of mobile sensors which can move to the appropriate places to provide the required coverage. This approach is different from the some of the work [39, 40, 41] which assume that the environment is sufficiently known and under control.

In this chapter, I introduce a comprehensive taxonomy framework for wireless sensor networks (WSN) self-deployment in which three sensor relocation algorithms are proposed according to the mobility degree of sensor nodes. The first one, particle swarm optimization based algorithm (PSOA), regards the sensors in the network as a swarm, and reorganizes the sensors by the particle swarm optimization (PSO) algorithm, in the full sensor mobility case. The other two, relay shift based algorithm (RSBA) and energy efficient fuzzy optimization algorithm (EFOA), assume relatively limited sensor mobility, i.e., the movement distance is bounded by a threshold, to further reduce energy consumption. Simulation results show that our approaches greatly improve the network coverage as well as energy efficiency compared with related works. The contributions include the comprehensive collection of algorithms for mobile sensor network self-deployment within the context of a generally applicable taxonomy.

In this chapter, I present a brief overview of the previous work on the coverage driven deployment of mobile sensor networks and self-organiztion of static sensor networks that is most relevant to our study.

A taxonomy (Fig 2.1) for WSN self-deployment and self-organization is introduced. I take the initial deployment as the first level, in which most of the existing research work makes an assumption of random distribution. Three categories of the full, limited and zero mobility are then considered as the top 2nd level of Fig. 2.1. The three relocation and scheduling cases for sensor network self-deployment corresponding to the three categories of sensor nodes mobility degree are extensively studied and integrated. The two objectives are coverage (in space) and energy (in time) which are competing. The coverage objective desire “spread out” the nodes which consumes energy.

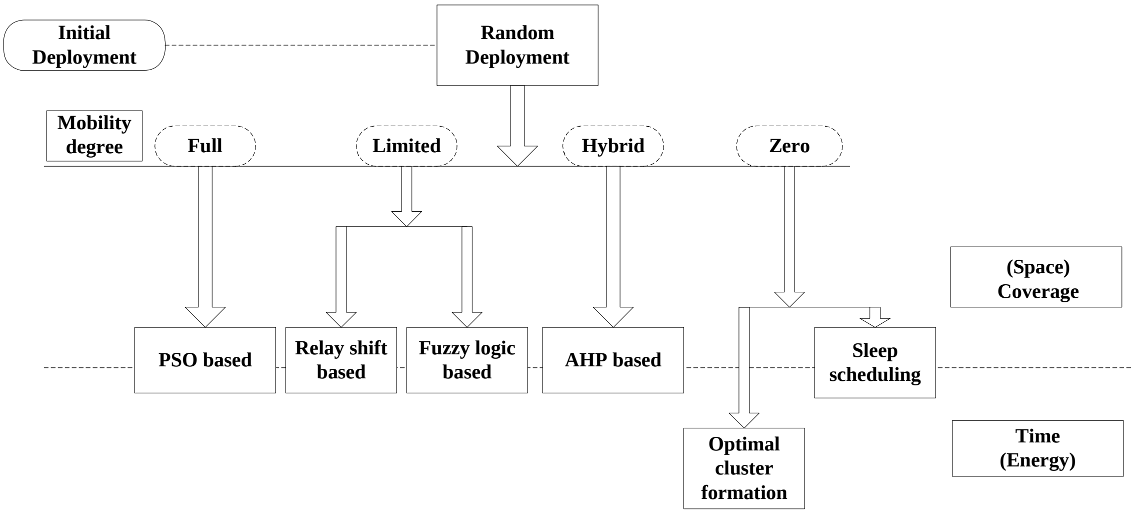


Fig. 2.1 Taxonomy integrating deployment and self-organization schemes

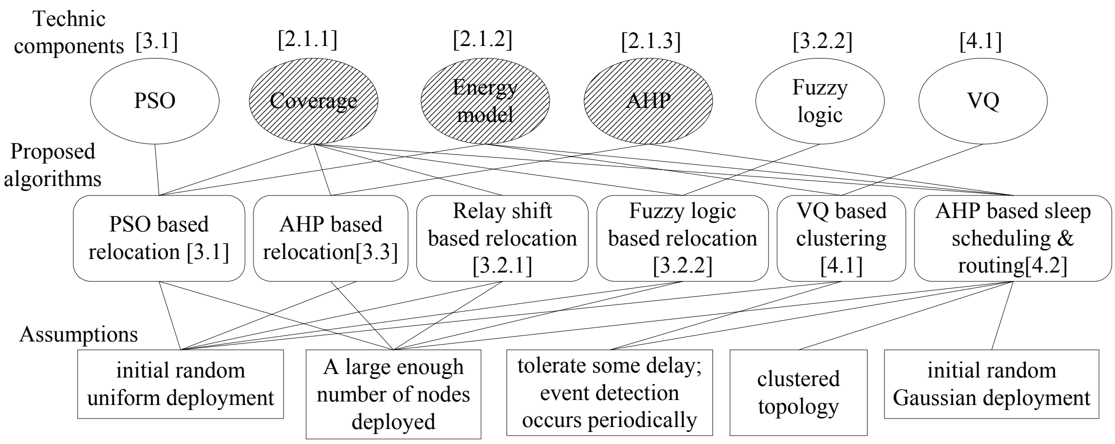
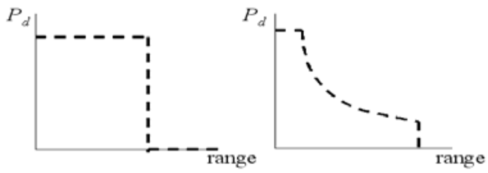


Fig. 2.2 Proposed algorithms together with the technic components and their assumptions

The proposed algorithms together with the technic components used and their assumptions are summarized in Fig 2.2. The numbers in square brackets indicate the sections in which the technic components or the proposed algorithms appear. The reused components (marked in shadow in Fig 2.2) in the whole thesis are coverage models and their calculations, energy model and Analytical Hierarchy Process (AHP) and they are introduced in the next subsections.

2.1.1 Coverage

Coverage is one of the measurement criteria of Quality of Service (QoS) of a sensor network.



(a) Binary sensor model (b) probabilistic sensor model

Fig. 2.3 Sensor coverage models

The coverage of each sensor can be defined either by a binary sensor model or a probabilistic sensor model as shown in Fig. 2.3. In the binary sensor model, the detection probability of the event of interest is 1 within the sensing range; otherwise, the probability is 0. In this case coverage is defined as the ratio of the union of areas covered by each node and the area of the entire Region Of Interest (ROI), as shown in Eq. (2-1) [51]. Generally ROI indicates the area in which sensor nodes need to be deployed. Here, the covered area of each node is defined as the circular area within its sensing radius. Perfect detection of all interesting events in the covered area is assumed.

$$C = \frac{\bigcup_{i=1,\ldots,N} A_i}{A} \tag{2-1}$$

where

A_i is the area covered by the i^{th} node;

N is the total number of nodes;

A stands for the area of the ROI.

In order to prevent recalculating the overlapped area, the coverage here is calculated using Monte Carlo method by creating a uniform grid in the ROI. All the grid points located in the sensing area are labeled 1 otherwise 0, depending on whether the Euclidean distance between each grid point and the sensor node is longer or shorter than sensing radius, as shown in Fig 2.4 (dashed circle indicating the sensing area boundary). Then the coverage can be approximated by the ratio of the summation of ones to the total number of the grid points.

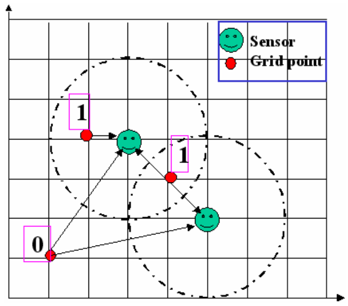


Fig. 2.4 Sensing coverage calculation

If a node is located well inside the ROI, its complete coverage area will lie within the ROI. In this case, the full area of that circle is included in the covered region. If a node is located near the boundary of the ROI, then only the part of the ROI covered by that node is included in the computation.

Although the binary sensor model is simpler, it is not realistic as it assumes that sensor readings have no associated uncertainty. In reality, sensor detections are imprecise, so that the coverage needs to be expressed in probabilistic terms. In many cases, cheap sensors such as omnidirectional acoustic sensors or ultrasonic sensors are used. Some practical examples include AWAIRS at UCLA/RSC, Smart Dust at UC Berkeley, the USC-ISI network, the DARPA SensIT systems/networks, the ARL Advanced Sensor Program systems/networks, and the DARPA Emergent Surveillance Plexus (ESP). For omnidirectional sensors, a longer distance between the sensor and the target generally implies a greater loss in the signal strength or a lower signal-to-noise ratio. This suggests that we can build an abstract sensor model to express the uncertainty in sensor responses. In other words, a sensor node that is closer to a target is expected to have a higher detection probability about the target existence than the sensor node that is further away from the target.

$$c_{ij}(x, y) = \begin{cases} 0 & if \quad r + r_e \le d_{ij}(x, y); \\ e^{-\lambda a^{\beta}}, & if \quad r - r_e < d_{ij}(x, y) < r + r_e; \\ 1 & if \quad r - r_e \ge d_{ij}(x, y). \end{cases} \tag{2-2}$$

The sensor field is represented by a grid. An individual sensor node s on the sensor field is located at grid point (x, y). Each sensor node has a detection range of r. For any grid point P at (i, j), I denote the Euclidean distance between s at (x, y) and P at (i, j) as $d_{ij}(x, y)$, i.e., $d_{ij}(x, y)=\sqrt{(x-i)^2+(y-j)^2}$. Eq. (2-2) expresses the coverage $c_{ij}(x, y)$ of a grid point at (i, j) by sensor s at (x, y), in which $r_e(r_e<r)$ is a measure of the uncertainty in sensor detection, $a = d_{ij}(x, y)-(r-r_e)$, and λ and β are parameters that measure detection probability when a target is at a distance greater than r_e but within a distance from the sensor. The distances are measured in units of grid points. In fact, the sensing behavior of almost all the omnidirectional range sensing devices including not only chemical sensors but also infrared, ultrasound, and acoustic sensors etc., can be modeled by probabilistic sensor detection model which is shown in Fig. 2.2(b). Fig. 2.2(b) also illustrates the translation of a distance response from a sensor to the confidence level as a probability value about this sensor response. The coverage for the entire grid sensor field is calculated as the fraction of grid points that exceeds the threshold c_{th}.

2.1.2 Energy model

According to the radio energy dissipation model, in order to achieve an acceptable Signal-to-Noise Ratio (SNR) in transmitting an *l* bit message over a distance *d,* the energy expended by the radio is given by [14]:

$$E_T(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & if\ d \le d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & if\ d > d_0 \end{cases} \tag{2-3}$$

where E_{elec} is the energy dissipated per bit to run the transmitter or the receiver circuit, ε_{fs} and ε_{mp} are amplifier constants, and *d* is the distance between the sender and the receiver. By equating the two expressions at $d=d_0$, we have $d_0 = \sqrt{\varepsilon_{fs} / \varepsilon_{mp}}$. Here electronics energy is set as E_{elec}=50*nJ/bit*, whereas the amplifier constant is taken as ε_{fs} =10*pJ/bit/m*^{2}, ε_{mp} = 0.0013*pJ/bit/m*^{2}.

In both cases, to receive *l* bit message, the radio expends:

$$E_R(l) = lE_{elec} \tag{2-4}$$

2.1.3 Analytical Hierarchy Process

The Analytical Hierarchy Process (AHP) [42] is a multiple criteria decision-making method which decomposes a complex problem into a hierarchy of simpler and more manageable sub problems. These sub-problems are usually called decision factors and weighted according to their relative dominances to the problem. AHP synthesizes their importance to the problem, and finds the best solution.

AHP performs following four main steps: decomposition, pair-wise comparison, local weight calculation, and weight synthesis.

A. Structuring Hierarchy

Structuring a problem as a hierarchy of multiple criteria is the first step of implementing AHP. The decision factors of the problem are identified and inserted into the hierarchy. The overall objective is placed at the topmost level of the hierarchy. The subsequent level presents the decision factors. The solution alternatives are located at the bottom level.

B. Calculating Local Weights

The second step is the evaluation stage where each factor is compared to all other factors within the same parent. Local weights consist of two parts: the weight of each decision factor to the goal and the weight of each nominee to each factor. Both of them are calculated with the same procedure. Taking the former as an example, I describe the procedure as the following three steps.

1) Making Pairwise Comparison

The evaluation matrices are built up through pairwise comparing each decision factor under the topmost goal. The comparison results are based upon user expertise experience by asking questions such as "Which is more important and by how much?" These initial values are captured in square matrix A as

$$A = \left(a_{ij}\right)_{n\times n} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \tag{2-5}$$

where a_{ij} denotes the ratio of the i^{th} factor weight to the j^{th} factor weight, and n is the number of factors.

The smaller one in a pair is chosen as a unit and the larger one is estimated as a multiple of that unit and assigned a number based on the perceived intensity of importance. Similarly, the reciprocals of these numbers are used to show the inverted comparison results. We thus obtain a reciprocal matrix where the entries are symmetric with respect to the diagonal. The fundamental 1 to 9 scale can be used to rank the judgments as shown in Table 2.1.

Table 2.1. A fundamental 1 to 9 scale

Number Rating	Verbal Judgment of Preferences
1	Equally
3	Moderately
5	Strongly
7	Very
9	Extremely

2, 4, 6, 8 indicate the medium value of above pairwise comparison.

2) Calculating Weight Vector

For the given matrix A in Eq. (2-5), its eigenvalue equation is written as $AW = \lambda_{max}W$, where W is a non-zero vector called eigenvector, and λ_{max} is a scalar called eigenvalue. W and λ_{max} appear as a pair and cannot be taken apart. After standardizing the eigenvector W, I regard the vector element of W as the local weight of each decision factor approximately, which can be denoted as:

$$\mathbf{w}_j^T = \{w_1, w_2, \cdots, w_n\} \tag{2-6}$$

As a result, the weights of the decision factors can be achieved by calculating the eigenvector of AHP matrix and the eigenvalue that approximately equals the number of assessed elements.

3) Checking for Consistency

If every element in Eq. (2-5) satisfies the equations $a_{ij} = 1/a_{ji}$ and $a_{ik} \cdot a_{kj} = a_{ij}$, the matrix A is the consistency matrix. However, the evaluation matrices are often not perfectly consistent due to people's random judgments. These judgment errors can be detected by a consistency ratio (CR), which is defined as the ratio of consistency index (CI) to random index (RI). CI can be achieved by

$$CI = (\lambda_{\max} - n)/(n-1), \tag{2-7}$$

where $\lambda_{\max}$ is the eigenvalue and

$$\lambda_{\max} = (1/n)\sum_{i=1}^{n}(AW)_i / W_i \,. \tag{2-8}$$

Table 2.2. Random index

n	1	2	3	4	5	6	7	8	9	10	11
RI	0	0	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49	1.51

The *RI* is given in Table 2.2 [43]. When $CR \leq 0.1$, the judgment errors are tolerable and the weight coefficients of the global weight matrix W_j are the weights of decision factor under the topmost goal. Otherwise, the pairwise comparisons should be adjusted until matrix A satisfies the consistency check, i.e. matrix A needs to be reinitialized.

C. Calculating Global Weights

From above steps, we can obtain not merely the weights of decision factors towards the topmost goal from W_j but also the weights of alternatives towards each factor. If there are k

candidates, all the k weight matrixes of alternatives under n factors construct a $k\times n$ matrix, denoted as $W_{n_i/j}$, i=1, 2, … k, j=1, 2, …n.

The global weight of each alternative can be achieved through multiplying the local weight by its corresponding parent. So the final weight matrix in the symbol of W_{n_i} is calculated as

$$W_{n_i} = W_{n_i/j} \cdot W_j \,, \tag{2-9}$$

where the final weight of each alternative is calculated as

$$W_{n_i} = \sum_{j=1}^{n} W_{n_i/j} \cdot W_j \,. \tag{2-10}$$

The larger the final weight of alternative, the higher the probability it is eligible to be selected.

2.2 Related work

2.2.1 Coverage

In WSNs with zero mobility, i.e., stationary sensor networks, there are many previous studies which have focused on characterizing coverage. The authors of [44] consider a grid-based sensor network and derive the conditions for the sensing range and failure rate of sensors to guarantee that an area is fully covered. In [45], the authors propose several algorithms to find paths that are most or least likely to be detected by sensors in a sensor network. Path exposure of moving objects in sensor networks is formally defined and studied in [39], where the authors propose an algorithm to find minimum exposure paths, along which the probability of a moving object being detected is minimized. The best and worst coverage problem is explored in [46]. They propose an optimal polynomial time worst and average case algorithm for coverage calculation for homogeneous isotropic sensors. They also present several experimental results and analyze potential applications, such as using best and worst-case coverage information as heuristics to deploy sensors to improve coverage. In [47], the authors define several important coverage measures for a large-scale stationary sensor network, namely, the area coverage, detection coverage, and node coverage. Under the assumption that sensor location follows a Poisson point process, the authors obtain analytical results for the coverage measures under a Boolean sensing model and a general sensing model. A more thorough survey of the sensor network coverage is provided by [48].

2.2.2 Mobility-assisted deployment

While the coverage of stationary sensor networks has been extensively studied and relatively well understood, a class of work has only recently appeared where full mobility of sensors is utilized to achieve desired deployment [49, 50, 51, 52, 53, 54, 55, 56]. Typically in such works, the sensors detect lack of desired deployment objectives, then estimate new locations, and move to the resulting locations. In [38, 22], the authors propose a virtual-force-based sensor movement strategy to enhance network coverage after an initial random placement of sensors. The virtual forces repel the nodes from each other and from obstacles to ensure that the initial configuration of nodes quickly spreads out to maximize coverage area. However they assume that global information regarding other nodes is available. Several distributed energy-efficient deployment algorithms are proposed in [51]. In order to achieve an energy-efficient node topology for a longer system lifetime, they employ a synergistic combination of cluster structuring and a peer-to-peer deployment scheme. Besides that, an energy-efficient deployment algorithm based on Voronoi diagrams is also proposed there. In [50], the authors propose several algorithms that identify existing coverage holes in the network and compute the desired target positions where sensors should move in order to increase the coverage. The main difference among all of their proposed algorithms is how the desired positions of sensors are computed. In [53], the authors propose a proxy-based sensor deployment protocol. Instead of moving iteratively, sensors calculate their desired positions based on a distributed algorithm, move logically, and exchange new logical locations with their new logical neighbors. Actual movement occurs at one time when sensors determine their final locations. The proposed protocol can greatly reduce the energy consumption while maintaining similar coverage. In [49], a scan-based movement-assisted sensor deployment method that uses scan and dimension exchange to achieve a balanced state is proposed. Using the concept of load balancing, it achieves good performance especially when applied to uneven distribution sensor networks. The authors of [47] study the dynamic aspects of the coverage of a mobile sensor network that depend on the sensor movement process. The results show that sensor mobility can be exploited to improve network coverage. For mobile targets, they take a game theoretic approach and derive optimal mobility strategies for sensors and targets from their own perspectives. In [55], the authors examine the optimization of wireless sensor network layouts using a multi-objective genetic algorithm (GA) in which two competing objectives are considered, total sensor coverage and the lifetime of the network. However the computation of this method is not inexpensive. In [56], fuzzy logic theory is applied to handle the uncertainty in full mobility sensor deployment problem. Their approach achieves fast and stable deployment and greatly increases the field coverage as well as communication quality. However, their fuzzy inference rules only consider two aspects, number of neighbors of each sensor and the average Euclidean distance between sensor node and its neighbors, without energy consumption included at all, which is one of the most critical issues in sensor networks.

In fact, the mobility of sensors is limited in most cases. To this extent, a class of Intelligent Mobile Land Mine Units (IMLM) [57] to be deployed in battlefields have been developed by Defense Advanced Research Projects Agency (DARPA). The IMLM units are employed to detect breaches, and move with limited mobility to repair them. This mobility system is based on a hopping mechanism and the hop distance is dependent on the amount of fuel and the propeller dynamics. Some other techniques can also provide such kind of mobility, for instance, sensors supplied by spring actuation etc. This type of model normally trades off mobility with energy consumption [58]. Moreover, in many applications, the latter goals outweigh the necessity for advanced mobility, making such mobility models quite practical in the future. In fact, [58] is one of the very few papers which deal with the mobility limited deployment optimization. The mobility in the sensors they consider is restricted to a flip. However coverage is the only considered objective in their paper and their approach is not feasible in network partition case.

With the same goal as the existing research work in mind, that is, to improve the sensing coverage in a predefined area with low energy consumption and with connectivity guaranteed, I propose three different relocation algorithms, PSOA, RSBA and EFOA, in the cases of full sensor mobility and limited sensor mobility. I also indicate in the diagram that, in the zero mobility case, static topology control and scheduling schemes such as optimal number of cluster heads selection and cluster formation may be used. In general, cluster formation allows individual sensors to be grouped together for either communication or power efficiency. Cluster head is a node which manages the processing and relaying the information from its cluster members. In the next chapter, I will describe our proposed sensor relocation approaches in detail.

The above is a class of work that recently appeared where all the sensor nodes with mobility are utilized to achieve desired deployment. Typically in such works, all sensor nodes are relocated to maximize the coverage of a given target area with constraints on deployment time, the distance the sensors have to travel and the complexity of the protocol. The sensor network deployment scenario when only some of the sensors are mobile while others are static, that is, hybrid sensor networks [59], has also been under active research, especially in the field of robotics for exploration purposes. The movement capable sensors can help in network maintenance and repair by moving to appropriate locations within the topology to achieve desired level of coverage and connectivity, and to connect a possibly disconnected network.

In [60], Batalin et al. suggest a combined solution for the exploration and coverage of a given target area. The coverage problem is solved with the help of a constantly moving robot in a given target area. The mobile robot first performs the network deployment in the target area as it explores the unknown environment. The deployed static nodes then guide the robot to poorly covered areas. However, the algorithm does not consider the communications between the deployed nodes. All decisions are made by the robot by directly communicating with a neighbor sensor node. In fact, the deployment strategy and network repair policy can also benefit from the multi hop information derived out of a communicating sensor network.

Wang et al. [61] address the single coverage problem by moving the available mobile sensors in a hybrid network to heal coverage holes. The static sensors detect their local coverage holes by using Voronoi diagrams as in [50]. The mobile sensors also calculate coverage holes formed locally if they decide to leave their current position. The static sensors bid for the mobile sensors based on the size of their detected coverage hole. A mobile sensor compares the bids and decides to move if the highest bid received has a coverage hole size greater than the new hole generated in its original location due to its movement. However, the local broadcast may prevent the bid messages reaching mobile sensors if they are located farther than two hops. Moreover, the environment influences are not included in the design.

In [62], a hybrid sensor network is considered and a Voronoi diagram based approach is provided to estimate the amount of coverage holes in a sensing field. They also propose a collaborative algorithm (Coven) to estimate the number of additional mobile nodes to be deployed and relocated to fix the coverage holes. However, their collaborative algorithm doesn’t consider any environmental factors such as obstacle and hot spot.

Luo, R.C. et al. [63] propose a mechanism which divides the map into many grids, and sets up weighting fields generated by various environmental effects such that the deployed goal can be determined. Without changing the previous distribution of static nodes, the coverage and uniformity are improved by incrementally placing additional mobile nodes one by one into the monitored environment. Although this “grid method” has inexpensive computation, it provides only approximate result rather than an optimal one.

2.2.3 Sleep scheduling

Sleep scheduling which aims to conserve the energy of the sensor nodes has been studied in the literature. In [99], nodes are allowed to sleep based on routing information, and nodes switch between sleep and active state based on the traffic of the network. As a modification to this basic algorithm, reactive features have been added to the node's schedule. The node would wake up more frequently based on the route discovery interval. Another widespread option is to turn off redundant nodes in the network [64]. In this scheme, the density of low power sensors is high enough to maintain the sensing coverage of the entire network even when some nodes are turned off. Each node studies the activity of their neighbors and decides to sleep if the coverage can be maintained by the active nodes. A back-off based approach has been used to prevent neighboring nodes to turn themselves off simultaneously.

In [65], a few nodes are selected as coordinators which would then decide the sleep/awake schedule of the other nodes in the network. While coordinators are awake at all times, the other nodes in the network sleep in order to conserve the overall network energy. In [66] nodes are randomly selected to go to the sleep mode and in [67] a Linear Distance-based Scheduling (LDS) technique has been used to define the sleep schedule of the nodes in a cluster based homogeneous network. In [68], the authors release the single hop communication assumption of [67] and introduce a Hop-based Sleeping Scheduling (HSS) algorithm in a circular sensor network which is divided by a number of levels. The overall result of these sleep schedules is a considerable reduction in the energy consumption of the sensor network.

In [69], the authors propose a cross-layer sleep-scheduling-based organizational approach, called SS-Trees, in order to increase monitoring coverage and operational lifetime of mesh-based USNs. An integer linear programming (ILP) formulation and an iterative algorithmic approach are suggested to determine the feasible SS-Tree structures for these purposes. The ILP approach requires the determination of objective functions and several constraints, which is often complicated. Hence the proposed AHP based approach is different and simpler in that we only need to give the estimated weight to several factors as an input for AHP to finish the whole process of optimal decision making without knowing the objective functions and constraints. From this point-of-view, AHP is easier to carry out with the achievement of the same performance goals.

2.2.4 Sensor clustering

The research community is actively looking into these challenges. [14] proposes the LEACH protocol, which is a hierarchical self-organized cluster-based approach for monitoring applications. The data collection area is randomly divided into several clusters. Based on time division multiple access (TDMA), the sensor nodes transmit data to the cluster heads, which aggregate and transmit the data to the base station. A new set of cluster heads are chosen after specific time intervals. A node can be re-elected only when all the remaining candidates have been elected. The work in [70] shows that a 2-tier architecture is more energy efficient when hierarchical clusters are deployed at specific locations. [71] describes a multi-level hierarchical clustering algorithm, where the parameters for minimum energy consumption are obtained using stochastic geometry. HEED [72] selects cluster heads through *O(1)* time iteration according to some metric and adopts the multi-hop communication to further reduce the energy consumption. PEGASIS [29] improves the performance of LEACH and prolongs the network lifetime greatly with a chain topology. But the delay is significant although the energy is saved. In [100], particle swarm optimization (PSO) is used to find the optimal cluster head positions for sensor network deployment. There are some other related works [73, 74] which efficiently use energy through clustering. However, none of the above clustering approaches can guarantee both energy efficient sensors clustering and fast data compression. For example, in their proposed cluster based topology, sensing values inside one cluster may not provide the closest correlations, thus the optimal data compression can not be guaranteed. To the best of our knowledge, [75] is the first paper that suggests a sensor cluster formation method with fast data compression while minimizing energy consumption in sensor networks.

2.2.5 Routing

Many routing protocols have also been developed for ad hoc networks, which can be summarized into two categories: table-driven (e.g., destination sequenced distance vector [76], cluster switch gateway routing [77]) and source-initiated on-demand (e.g., ad hoc on-demand distance vector routing [78], dynamic source routing (DSR) [79]). In [80], Lee and Gerla propose a Split Multi-path Routing protocol that builds maximal disjoint paths, where data traffic is distributed in two roots per session to avoid congestion and to use network resources efficiently. A Multi-path Source Routing (MSR) scheme is proposed in [81], which is an extension of DSR. Their work focuses on distributing load adaptively among several paths. Nasipuri and Das [82] present the On-Demand Multi-path Routing scheme which is also an extension of DSR. In their scheme, alternative routes are maintained, which can be utilized when the primary one fails.

In sensor networks, location is often more important than a specific node ID. For example, in sensor networks for target tracking, the target location is much more important than the ID of reporting node. Therefore, some location-aware routing schemes have been proposed for USNs. A greedy geographic forwarding with limited flooding to circumvent the voids inside the network is proposed in [83], and some properties of greedy geographic routing algorithms are studied in [84]. Jain et al [85] proposes a geographical routing using partial information for USNs.

Chapter 3 Mobility-assisted relocation schemes

3.1 Full mobility case: PSOA

I propose three different relocation methods for movement assisted self-deployment of sensors according to the mobility degree of sensor nodes. The common goal of the suggested schemes is to improve the sensing coverage in a predefined area with low energy consumption.

In the full sensor mobility case, I propose particle swarm optimization (PSO) based algorithm for movement assisted relocation. PSO, originally proposed by Eberhart and Kennedy [86] in 1995, and inspired by social behavior of bird flocking, has come to be widely used as a problem solving method in engineering and computer science [87, 88, 89, 90].

All of the particles have fitness values, evaluated by the fitness function to be optimized. PSO is initialized with a group of random solutions and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" factors. The first one, called *pbest*, is the best fitness it has achieved so far and it is also stored in memory. Another "best" value obtained so far by any particle in the population, is a global best and called *gbest*.

The PSO formulae define each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots\ldots, x_{iD})$ where i represents the particle number. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, p_{i3} \ldots\ldots p_{iD})$, and the index of the best particle among all the particles in the population is represented by the symbol g. A velocity along each dimension is denoted as $V_i = (v_{i1}, v_{i2}, v_{i3} \ldots\ldots v_{iD})$. Let $d \in [1, 2, \ldots, D]$, the updating equation [91] is as follows,

$$v_{id} = \varpi \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times rand() \times (p_{gd} - x_{id}) \tag{3-1}$$

$$x_{id} = x_{id} + v_{id} \tag{3-2}$$

where ω is the inertia weight, and c_1 and c_2 are acceleration coefficients.

The role of the inertia weight ω is considered to be crucial for the convergence of PSO. A suitable value for the inertia weight ω balances the global and local exploration ability and, consequently, reduces the number of iterations required to locate the optimum solution. So it is better to initially set the inertia to a large value, in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions. Thus, a time-decreasing inertia weight value is used.

PSO shares many similarities with genetic algorithm (GA). Both algorithms start with a group of a randomly generated population, have fitness values to evaluate the population with random techniques. Compared with GA, PSO is easier to implement, has fewer parameters to adjust, and requires only primitive mathematical operators. Because of its inexpensive computation and fast convergence rate, PSO is a potential algorithm to optimize deployment in a sensor network.

I assume that each node knows its position in the problem space, all sensor members in a cluster are homogeneous and cluster heads (CHs) are more powerful than sensor members. Sensing and communication coverage of each node are assumed to have a circular shape without any irregularity. The design variables are *2D* coordinates of the sensor nodes, $\{(x_1, y_1), (x_2, y_2), \ldots\}$. Sensor nodes are assumed to have certain mobility. PSOA includes two stages, the first is to optimize coverage by relocating sensors and the second is cluster formation when nodes have settled down during the first stage and don't move again.

I consider coverage as the first optimization objective. The coverage of each sensor is defined either by a binary sensor model or a probabilistic sensor model as shown in Fig. 2.2; both are used here. I take the Eqs. (2-1) and (2-2) as the coverage objective functions.

After optimization of coverage, all the deployed sensor nodes move to their own positions. Our goal then becomes to minimize energy usage in a cluster based sensor network topology by finding the optimal cluster head (CH) positions. So cluster formation used to optimize energy consumption here is actually in a static sensor network manner. We are now in the second stage of PSOA.

Assume that the sensor nodes inside a cluster have short distance *dis* to CH but each CH has long distance *Dis* to the base station. For each sensor node inside a cluster, to transmit an *l*-bit message a distance *dis* to CH, the radio expends

$$E_{TS}(l, dis) = lE_{elec} + l\varepsilon_{fs} dis^2 \tag{3-3}$$

For CH, however, to transmit an *l*-bit message a distance *Dis* to base station, the radio expends

$$E_{TH}(l, Dis) = lE_{elec} + l\varepsilon_{mp} Dis^4 \tag{3-4}$$

So the energy loss of a sensor member in a cluster is

$$E_s(l, dis) = l(100 + 0.01dis^2) \tag{3-5}$$

The energy loss of a CH is

$$E_{CH}(l, Dis) = l(100 + 1.3\times 10^{-6} \times Dis^4) \tag{3-6}$$

Since the energy consumption for computation is much less than that for communication, I neglect computation energy consumption here.

Assume m clusters with n_j sensor members in the j^{th} cluster C_j. The total energy loss E_{total} is the summation of the energy used by all sensor members and all the m CHs:

$$E_{total} = l\sum_{j=1}^{m}\sum_{i=1}^{n_j}(100 + 0.01dis_{ij}^2 + \frac{100}{n_j} + \frac{1.3\times10^{-6}Dis_j^4}{n_j}) \tag{3-7}$$

Because only two terms are related to distance, we can just set the fitness function as:

$$f = \sum_{j=1}^{m}\sum_{i=1}^{n_j}(0.01dis_{ij}^2 + \frac{1.3\times10^{-6}Dis_j^4}{n_j}) \tag{3-8}$$

From Eq. (3-8) we can minimize the energy dissipation in the sensor network by reducing the distance from each node to its CH and the CH to the remote base station. I use the PSO algorithm to find the optimal CH positions in the sensor field when the minimized energy consumption is achieved.

3.2 Limited mobility case

3.2.1 Relay Shift Based Algorithm (RSBA)

Let *G(V, E)* be the graph defined on *V* with edges *uv* ∈E iff $uv \leq R$. Here *uv* is the Euclidean distance between nodes *u* and *v*, *R* is the communication range. I assume that sensor nodes know their locations using one of the GPS-less localization techniques such as received signal strength so that CH can get the position information of its sensor members.

There are 4 steps for implementing RSBA:

Step 1: Randomly deploy nodes in the network.

Step 2: Detect coverage holes by voronoi diagram and detect redundant sensor nodes. I set a distance threshold value T_1. Calculate the longest linear distance between two points A and B along the uncovered area perimeter and create a virtual node point at the center of the straight line AB. If the distance between two neighbors is less than T_1, regard them as redundant nodes. Choose a redundant node nearest to the virtual node point in coverage hole.

Step 3: Use A* algorithm [92] to find a shortest path n_0-n_1-n_2-…-n_{n-1} from a redundant sensor n_0 to the destination n_{n-1} (added virtual node) in a coverage hole. The distance between n_{n-2} to n_{n-1} is bounded by R. A* algorithm is the most popular choice for pathfinding, because it is fairly flexible and can be used in a wide range of contexts. A* was developed to combine heuristic approaches like Best-First-Search (BFS) and formal approaches like Dijsktra's algorithm. It is like Dijkstra's algorithm in that it can guarantee a shortest path, while BFS cannot; and it is like BFS in that it works as fast as BFS which is faster than Dijsktra's algorithm. Take the advantage of A* algorithm, we can solve our problem more efficiently than our previous work [24] in which Dijsktra's algorithm was applied.

Step 4: Move sensor node n_{n-2} to the virtual node n_{n-1}, move n_{n-3} to n_{n-2} … finally move the redundant sensor n_0 to n_1, and leave the original location of sensor n_0 empty. The nodes coordinates can be updated by Eq (3-9):

$$NetLoc(n_i) = NetLoc(n_{i+1}), \quad i = 0, 1, \cdots, n-2 \tag{3-9}$$

$n_i \in$ nodes on shortest path from source to destination

n_0=source node

n_{n-1}=destination (virtual node)

The process is illustrated in Fig. 3.1 using an example of four sensors and one virtual node along the shortest path. Sensor node n_3 moves to the virtual node point n_4, n_2 moves to n_3 …

finally the redundant sensor n_0 moves to n_1, and leave the original location of n_0 empty. The network coverage is defined and calculated the same using Eq (2-1).

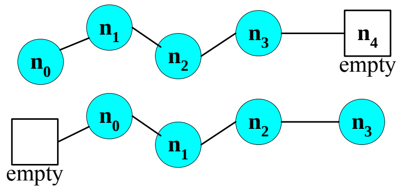


Fig. 3.1 Illustration of sensor nodes relay shift along the shortest path

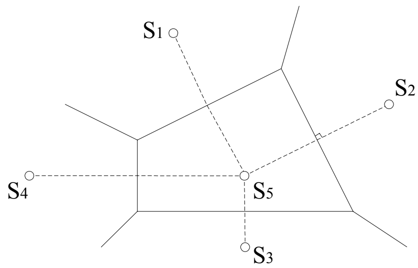


Fig. 3.2 Voronoi diagram

The coverage hole detection is based on Voronoi diagrams. As shown in Figure 3.2, each sensor node denoted by a S_i (i=1, 2, 3…), is enclosed by a Voronoi polygon. These polygons together cover the target field. The points inside one polygon are closer to the sensor node inside this polygon than the sensors positioned elsewhere. If this sensor cannot detect the events in its Voronoi polygon, no other sensor can detect it. Thus to detect coverage holes, each sensor only needs to check its own Voronoi polygon. If its sensing area cannot cover the polygon, there are coverage holes.

To construct the Voronoi polygon, sensors first calculate the bisectors of their neighbors and themselves. These bisectors (and possibly the boundary of the ROI) form several polygons. The smallest polygon encircling the sensor is the Voronoi polygon of this sensor.

The detailed procedure of the proposed RSBA method for sensor nodes reorganization is listed as follows:

1. Initialization

initial_node_locations (netXloc, netYloc);

sensing_range r;

communication_range R;

If distance (*i*, *j*) <= R

link *i* and *j*;

2. Detect coverage holes and create virtual nodes

Detect coverage holes based on voronoi diagram;

Calculate the longest length of two points A & B with coordinates (x_1, y_1) and (x_2, y_2) along the hole arc;

Calculate the center point C with coordinates (X_c, Y_c) of edge AB based on the following two equations, thus C becomes Virtual Nodes;

$Xc=(x_1+x_2)/2$; $Yc=(y_1+y_2)/2$;

3. Detect redundant nodes

threshold T_1;

If distance(i, j) <= T_1

If distance (i, C) < distance(j, C)

define i as source and C as destination;

4. Shortest Path Finding by A* algorithm

Function [path, totalCost] = A* (m, netCostMatrix, s, d) m: number of nodes in the network, s: source node index, d: destination node index,

path: node sequence of shortest path, totalCost: distance along shortest path

5. Sensor Nodes Movement

For k=1: length (path)-1

netloc (k) = netloc (k+1);

Update nodes link;

Calculate network coverage based on Eq. (2-1);

3.2.2 Energy-efficient Fuzzy Optimization Algorithm (EFOA)

A. Preliminaries of Fuzzy Logic System

The model of fuzzy logic system consists of a fuzzifier, fuzzy rules, fuzzy inference engine, and a defuzzifier. I have used the most commonly used fuzzy inference technique called Mamdani Method [93] due to its simplicity.

The process is performed in four steps:

1) Fuzzification of the input variables *energy, concentration and average distance to neighbors* - taking the crisp inputs from each of these and determining the degree to which these inputs belong to each of the appropriate fuzzy sets.
2) Rule evaluation - taking the fuzzified inputs, and applying them to the antecedents of the fuzzy rules. It is then applied to the consequent membership function.
3) Aggregation of the rule outputs - the process of unification of the outputs of all rules.
4) Defuzzification - the input for the defuzzification process is the aggregate output fuzzy set *moving distance* and the output is a single crisp number.

Information flows through the fuzzy inference diagram as shown in Fig. 3.3.

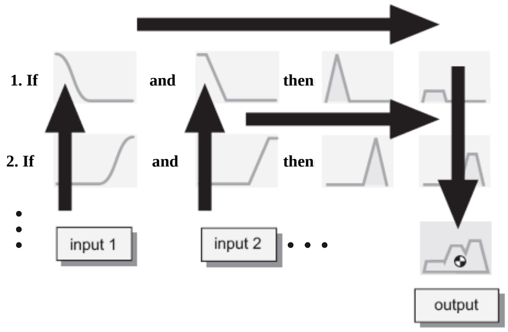


Fig. 3.3. Fuzzy inference diagram

B. Energy-efficient Fuzzy Optimization Algorithm

The same energy dissipation model presented in Sec. 2.1.2 is used here for calculation of energy consumption. Assume an area over which n nodes are uniformly distributed. For simplicity, assume the sink is located in the center of the field, and that the distance of any node to the sink or its CH is $\leq d_0$ as explained in Section 3.1.

Two main procedures are carried out in our algorithm: 1) Determine the next-step move distance for each sensor. 2) Determine the next-step move direction for each sensor. Expert knowledge for deployment problem is represented based on the following three descriptors:

- Node Energy - energy level available in each node, denoted by the fuzzy variable *energy*,
- Node Concentration - number of neighbors in the vicinity, denoted by the fuzzy variable *concentration*,
- Average distance to neighbors - average Euclidean distance between sensor node and its neighbors, denoted by the fuzzy variable d_n.

Table 3.1. Fuzzy rule base

No.	*En*	*Con*	d_n	d_m
1	low	low	close	close
2	low	low	moderate	vclose
3	low	low	far	vclose
4	low	med	close	moderate
5	low	med	moderate	close
6	low	med	far	vclose
7	low	high	close	moderate
8	low	high	moderate	close
9	low	high	far	close
10	med	low	close	moderate
11	med	low	moderate	close
12	med	low	far	close
13	med	med	close	far
14	med	med	moderate	moderate
15	med	med	far	close
16	med	high	close	far
17	med	high	moderate	moderate
18	med	high	far	moderate
19	high	low	close	far
20	high	low	moderate	moderate
21	high	low	far	moderate
22	high	med	close	vfar
23	high	med	moderate	far
24	high	med	far	moderate
25	high	high	close	vfar
26	high	high	moderate	far
27	high	high	far	far

Legend: vclose=very close, vfar=very far, med=medium, *En*=Energy, *Con*=Concentration

The linguistic variables used to represent the node energy and node concentration, are divided into three levels: *low, medium* and *high*, respectively, and there are three levels to represent the average distance to neighbors: *close, moderate* and *far*, respectively. The outcome to represent the moving distance d_m is divided into five levels: *very close, close, moderate, far* and *very far*. The fuzzy rule base includes rules like the following: IF the energy is *high* and the concentration is *high* and the distance to neighbor is *close* THEN the moving distance of sensor node *i* is *very far*.

Thus I use $3^3 = 27$ rules for the fuzzy rule base. I use triangle membership functions to represent the fuzzy sets *medium* and *moderate* and trapezoid membership functions to represent *low, high, close, vclose, far,* and *vfar* fuzzy sets. The membership functions developed and their corresponding linguistic states are represented in Table 3.1 and Figures 3.4 through 3.7.

For the defuzzification, the Centroid is calculated and estimated over a sample of points on the aggregate output membership function, using the following formula:

$$Cen = \left(\sum \mu_A(x) * x\right) / \sum \mu_A(x) \qquad (3\text{-}10)$$

where, $\mu_A(x)$ is the membership function of set *A*.

The control surface, or decision surface, is central in fuzzy logic systems and describes the dynamics of the controller and is generally a time-varying nonlinear surface. From Fig 3.8 and Fig 3.9 obtained by computation in Matlab Fuzzy Logic Toolbox [94], we can see that although the concentration for a certain sensor is high, the moving distance can be smaller than some sensor with higher energy or sensor with fewer neighbors but more crowded. With the assistance of control surface, the next-step moving distance can be determined.

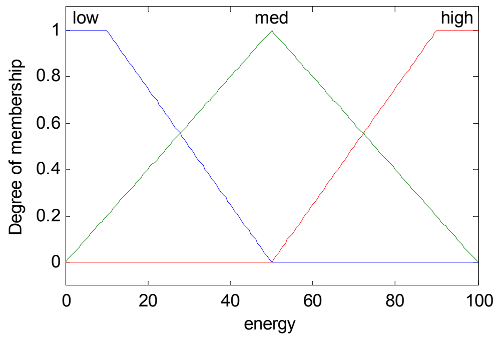


Fig. 3.4 Fuzzy set for fuzzy variable *energy*.

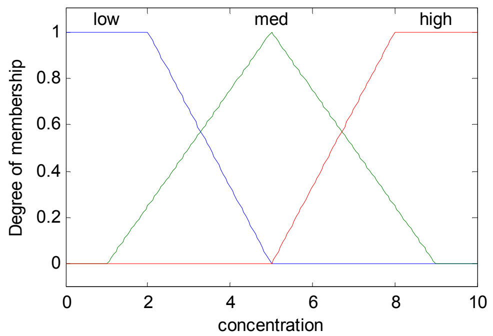


Fig. 3.5 Fuzzy set for fuzzy variable *concentration*.

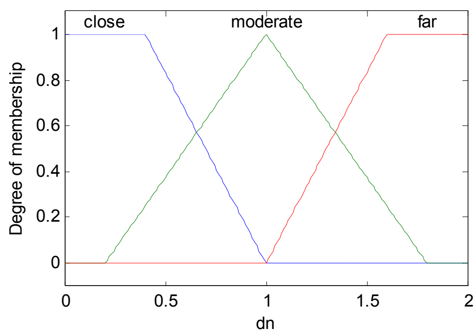


Fig. 3.6 Fuzzy set for fuzzy variable d_n.

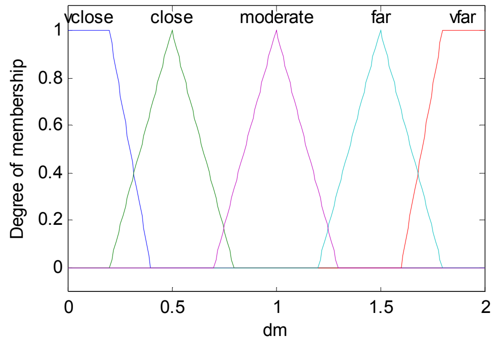


Fig. 3.7 Fuzzy set for fuzzy variable d_m.

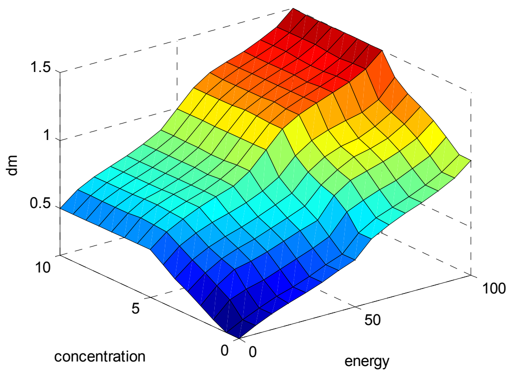


Fig. 3.8 Control surface (concentration, energy vs d_m).

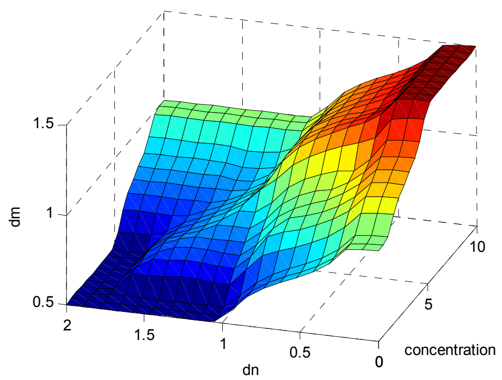


Fig. 3.9 Control surface (d_n, concentration vs d_m).

The next-step moving direction is decided by virtual force. Assume sensor i has k neighbors, $k=k_1+k_2$, in which k_1 neighbors are within threshold distance d_{th} to sensor i, while k_2 neighbors are farther than d_{th} distance to sensor i. The coordinate of sensor i is denoted as $C_i = (X_i, Y_i)$ and that of neighbor sensor j is $C_j = (X_j, Y_j)$. The next-step move direction of sensor i is represented as Eq. (3-11) and (3-12), thus sensor i clearly knows its next-step moving position by getting distance d_m and direction (angle α).

$$\vec{v} = \frac{1}{\left|\vec{C}_i - \vec{C}_j\right|^2}\left(\sum_{j=1}^{k_1}(\vec{C}_i - \vec{C}_j) + \sum_{j=1}^{k_2}(\vec{C}_j - \vec{C}_i)\right) \tag{3-11}$$

$$\tan(\alpha) = \frac{Y(\vec{v})}{X(\vec{v})} \tag{3-12}$$

The threshold distance d_{th} here is set to a proper value $\sqrt{3}r$ which is proved as follows. I attempt to make distance between two sensor nodes moderate, i.e., not very close and not very far. This kind of stable structure is illustrated in Fig. 3.10. Non-overlapped sensor coverage style is shown in Fig. 3.10(a), however, an obvious drawback here is that a coverage hole exists which is not covered by any sensor. Note that an alternative way is to allow overlap, as shown in Fig. 3.10 (b) and it ensures that all grid points are covered. Therefore, I adopt the second strategy.

In Fig. 3.10(b), it is obvious that $\triangle S_1S_2S_3$ is equilateral triangle. Because the sensing radius is r, through some steps of simple geometry calculations, we can easily derive the distance between two sensor nodes in the latter case $S_1S_2 = S_2S_3 = S_1S_3 = 2\times\frac{\sqrt{3}}{2}r = \sqrt{3}r$.

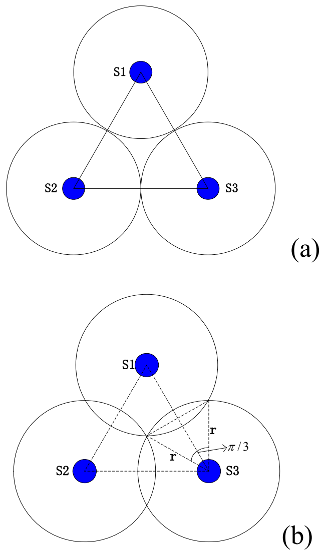


Fig. 3.10 Non-overlapped and overlapped sensor coverage cases

3.3 Mobile nodes relocation in hybrid sensor networks

In this section, I also solve the coverage problems in hybrid sensor networks by the proposed AHP based algorithm. It is different from the previous methods since it not only incorporates various environmental factors such as hot spot and obstacles in the design but also provides the optimal decision for mobile nodes movement. After a random deployment of static sensors, a certain amount of mobile nodes are deployed randomly into the monitored environment without changing the existing deployment of static sensor nodes. In order to increase the network coverage and uniformity, the mobile nodes are relocated according to our proposed scheme. The decision of moving direction of each mobile node is made according to AHP method, in which a set of criteria is evaluated and the optimal alternative is selected.

3.3.1 Problem Statements

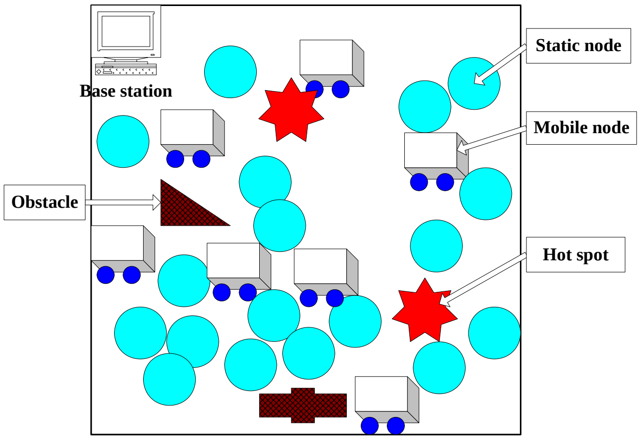


Fig. 3.11 Illustration of the design scenario

Assume that there have been some static nodes deployed in the monitored region. Then, some mobile nodes are randomly deployed into this system. The problem is how these mobile sensor nodes should be relocated for coverage enhancement under the constraint of environment factors. Fig. 3.11 is the illustration of our design scenario.

Some basic assumptions are made in our design. First, the networking system is a hybrid sensor network containing both mobile and static nodes. The static nodes must be pre-placed into the environment and the base station records all these locations. All nodes are equipped with the same sensing and communication devices. Second, the map should be well-known in detail including the distribution of obstacles. For example, to monitor a hazardous area suffered from terrible attacking, we need be familiar with the environment pattern and the distribution of static nodes. Third, the mobile nodes have only flip-based mobility as introduced in [95]. This type of model is adopted in most cases and normally trades-off mobility with energy consumption. The object of our proposed relocation scheme is to reduce the coverage holes and improve the network topology in specific environment after redeploying the additional mobile nodes.

To enable desired coverage while satisfying the environment requirements, we move the mobile nodes to proper locations according to their specific situation. In real environment, fours factors influence coverage directly, that is, the location and size of coverage hole in the network, the existence of hot spot and obstacle in the environment, and the boundary effect:

- Coverage hole: Areas not covered by any node. The direction to the nearest and largest coverage hole is preferred to be selected.
- Hot spot: The region in which events happen most frequently. The mobile node should ensure at least single coverage in hot spot. Thus the moving direction towards hot spot is also preferred.
- Obstacle: The mobile nodes need to avoid obstacles on their moving direction.
- Non-boundary: The mobile node is not preferred to move to the boundary since it will cause certain amount of sensing coverage loss.

The optimized next step moving direction determination is a multiple factors optimization problem and can be achieved using the AHP approach which is introduced in the next section.

3.3.2 Moving Direction Determination by AHP

The goal of the decision "choosing an appropriate moving direction" is at the top level of the hierarchy as shown in Fig. 3.12. The next level consists of the decision factors which are called criteria for this goal. At the bottom level there are 8 alternative directions to be evaluated.

From above steps, we can obtain not merely the weights of decision factors towards the topmost goal from W_j but also the weights of alternative directions towards each factor. It is assumed that there are eight directions. All the eight weight matrixes of alternatives under four factors construct a 8×4 matrix, denoted as $W_{n_i/j}$, i=1, 2, … 8, j=1, 2, 3, 4. The final weight of each alternative is calculated as

$$W_{n_i} = \sum_{j=1}^{4} W_{n_i/j} \cdot W_j \tag{3-13}$$

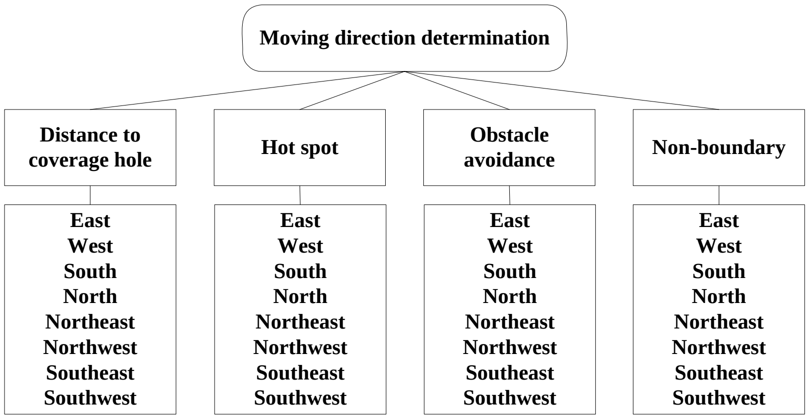


Fig. 3.12 AHP hierarchy for moving direction selection

The larger the final weight of the direction, the more important it towards enhancing the network topology quality. Thus, the direction with the largest weight is selected as next step moving direction of the mobile node.

3.4 Network Maintenance Strategy

After the first stage deployment, the network maintenance is also necessary to be considered due to the uncertain environment. Thus, it is actually the post-deployment stage after the fuzzy optimization deployment stage and a certain period of network operation. The characteristic of the network in this situation is heterogeneous. The proposed network maintenance strategy is based on the sensor node importance level ranking. First, I take the importance level calculation of the node n as an example. Assume the total number of nodes in the network is N. Let the probability that node i can sense grid point j be denoted by $S_i(P_j)$, and then the probability $C(P_j)$ that grid point j is sensed by the whole network is derived as:

$$C(P_j) = 1 - \prod_{i=1}^{N}(1 - S_i(P_j)) \\ = 1 - (1 - S_n(P_j)) \times \prod_{i \neq n}^{N}(1 - S_i(P_j)) \tag{3-14}$$

If delete node n, then the probability $C(P_j)$ becomes

$$C(P_j) = 1 - \prod_{i \neq n}^{N}(1 - S_i(P_j)) \tag{3-15}$$

For point j, the detection probability loss due to the deletion of node n becomes

$$\Delta C_n(P_j) = S_n(P_j) \times \prod_{i \neq n}^{N}(1 - S_i(P_j)) \tag{3-16}$$

Integrating the importance difference of each node in the network, the detection ability loss of the whole network after deleting node n is:

$$\Delta C_n = \sum_j \Delta C_n(P_j) \tag{3-17}$$

Considering the temporal gradient of sensing value in each grid point, the detection ability loss of the whole network after deleting node n is:

$$\Delta C_n = \sum_j \Delta C_n(P_j) \times \nabla(P_j) \tag{3-18}$$

in which $\nabla(P_j)$ is the temporal gradient of sensing value at grid point *j*. The higher the gradient value the more often the interesting events occurrence. I assume that sensor measurement physically has a range ($0\sim x_{max}$); if the sensing vale $v>x_{max}$, then let $v=x_{max}$.

According to importance level indicator ΔC_n, the importance level ranking of each node in the network can be sorted. Consequently we can either deploy several new sensor nodes close to the most important nodes or remove redundant nodes from "quiet" spot to the vicinity of those "busy" nodes as a backup.

Chapter 4 Self-organization of static sensor networks

The sensor deployment problem for cluster-based sensor networks, as described in previous chapter, focuses on mobile sensor networks and does not consider communication connectivity because multi-hop communication is not needed. In more general sensor network scenarios, we must also consider connectivity because it is needed for successful data delivery. However, most recent work in sensor network self-organization either focuses on one aspect such as energy-efficiency, maintaining the communication connectivity, the sensing coverage or output. The integration of some of these objectives is important in self-organization for sensor networks.

In this chapter, I present an energy-efficient clustering algorithm with fast data compression in sensor networks. It is a VQ-LBG based approach for cluster formation in WSN. The most distinguishing feature of the proposed method is that both energy efficient cluster formation and fast data compression can be guaranteed. I then propose a sleep scheduling scheme for balancing energy consumption rates in a single hop cluster based network using AHP. I consider three factors contributing to the optimal nodes scheduling decision and they are the distance to cluster head (CH), residual energy, and sensing coverage overlapping, respectively. I also propose an integrated sleep scheduling and geographical multi-path routing scheme for WSNs by AHP. The sleep scheduling is redesigned to adapt the multi-hop case. For the proposed routing protocol, the distance to the destination location, remaining battery capacity, and queue size of candidate sensor nodes in the local communication range are taken into consideration for next hop relay node selection. It can reduce the packet loss rate and link failure rate since the buffer capacity is considered.

4.1 Vector-quantization based clustering

WSN network topology control for large number of randomly placed sensors was studied in the recent years emphasizing the limited battery power. Generally, there are three methods that can be considered as possible networking protocols: direct communication, multi-hop routing, and clustering. In order to send information from a very high number of sensor nodes to the base station, it is necessary and economical to group sensors into clusters to prevent redundant information transmission and prolong network lifetime. Given the parameters for variation of energy consumption in the nodes, there are some main problems: How many sensors should be connected to each cluster head (CH), how many clusters are needed, how to select CH, and where each CH should be positioned. Another typical challenge is that due to the resource limitation of sensor nodes (CPU, memory, bandwidth, and energy), the collected information from sensor nodes in the cluster has to be compressed quickly and precisely for transmission.

The proposed approach is based on VQ-LBG design algorithm. It is a lossy data compression method based on the principle of block coding. It provides fast data compression process with minimum average distortion at CHs and uniform cluster formation. This would help in balancing the system load on each CH since all the clusters are balanced, and at the same time, the communication energy consumption will be significantly reduced due to the efficient data compression.

4.1.1 WSN model

In this section I describe our model of a wireless sensor network with nodes homogeneous in their initial amount of energy. I particularly present the energy model and how the optimal number of clusters can be computed. I assume that all nodes are distributed randomly over the sensor field.

Previous work have studied either by simulation or analytically [71], [96] the optimal probability of a node being elected as a cluster head as a function of spatial density when nodes are uniformly distributed over the sensor field. This clustering is optimal in the sense that energy consumption is well distributed over all sensors and the total energy consumption is minimum. Such optimal clustering highly depends on the energy model we use. For the purpose of this study I use similar energy model and analysis as proposed in [14].

According to the radio energy dissipation model, in order to achieve an acceptable Signal-to-Noise Ratio (SNR) in transmitting an *l* bit message over a distance *d,* the energy expended by the radio is given by Eq.(2-3), to receive *l* bit message, the radio energy expenses follow Eq. (2-4).

Assume an area $A = L \times L$ square meters over which *n* nodes are uniformly distributed. For simplicity, assume the sink is located in the center of the field, and that the distance of any node to the sink or its cluster head is $\leq d_0$.Thus, the energy dissipated in the cluster head node during a round is:

$$E_{CH}(l) = (\frac{n}{n_c} - 1)lE_{elec} + \frac{n}{n_c} lE_{DA} + lE_{elec} + l\varepsilon_{fs} d_{toBS}^2 \tag{4-1}$$

where n_c is the number of clusters, E_{DA} is the processing (data aggregation) cost of a bit per report to the sink, and d_{toBS} is the average distance between the cluster head and the sink. The energy used in a non-cluster head node is equal to:

$$E_{nonCH}(l) = lE_{elec} + l\varepsilon_{fs} d_{toCH}^2 \tag{4-2}$$

where d_{toCH} is the average distance between a cluster member and its cluster head. The expected squared distance from the nodes to the CH is given by:

$$E[d_{toCH}^2] = \frac{L^2}{2\pi n_c} \tag{4-3}$$

The energy dissipated in a cluster per round is given by:

$$E_{cluster} \approx E_{CH} + \frac{n}{n_c} E_{nonCH} \tag{4-4}$$

The total energy dissipated in the network is equal to:

$$E_{total} = l(2nE_{elec} + nE_{DA} + \varepsilon_{fs}(n_c d_{toBS}^2 + nd_{toCH}^2)) \tag{4-5}$$

By differentiating E_{total} with respect to n_c and equating to zero, the optimal number of constructed clusters can be found [97]:

$$n_{c-opt} \approx \sqrt{\frac{n}{2\pi}} \frac{L}{d_{toBS}} = \sqrt{\frac{n}{2\pi}} \frac{2}{0.765} \tag{4-6}$$

4.1.2. Cluster formation based on VQ-LBG

1) VQ design problem

I propose a data compression guaranteed clustering algorithm based on vector quantization (VQ). Our approach is designed according to the following observations:

(1) In sensor networks, the historical information exhibits similar patterns over time,

(2) different measurements are intrinsically correlated,

(3) coordinates of sensor nodes can be regarded as a feature pattern.

VQ is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm. LBG is a VQ design algorithm based on a training sequence. The use of a training sequence bypasses the need for multi-dimensional integration. A VQ that is designed using this algorithm are referred to as an LBG-VQ [98].

The VQ design problem can be stated as follows. Given a vector source with its statistical properties known, given a distortion measure, and given the number of codevectors, find the set of codebook and a cluster formation (partition) which results in the smallest average distortion.

I assume that there is a training sequence consisting of *M* source vectors:

$$\tau = \{X_1, X_2, ..., X_M\}.$$

This training sequence can be obtained from some large database. *M* is assumed to be sufficiently large so that all the statistical properties of the event source are captured by the training sequence. I assume that the source vectors are *k*-dimensional, e.g.,

$$X_m = \{x_{m,1}, x_{m,2}, ..., x_{M,k}\},\ m= 1, 2,\ldots, M$$

Let *N* be the number of codevectors and let $C = \{c_1, c_2, ..., c_N\}$, represents the codebook. Each codevector is *k*-dimensional, e.g., $c_i = \{c_{i,1}, c_{i,2}, ..., c_{i,k}\}$, *i*=1, 2,…, *N*. Let S_i be the encoding region associated with codevector c_i and let $P = \{S_1, S_2, ..., S_N\}$ denote the partition of the space. If the source vector X_{m} is in the encoding region S_i, then its approximation $Q(X_m) = c_i$, if $X_m \in S_i$.

Assuming a squared-error distortion measure, the average distortion is given by:

$$D_{ave} = \frac{1}{Mk}\sum_{m=1}^{M}\left\|X_m - Q(X_m)\right\|^2 \tag{4-7}$$

where $\|e\|^2 = e_1^2 + e_2^2 + ... + e_k^2$. The design problem can be briefly stated as follows: Given τ and N, find C and P such that D_{ave} is minimized.

2) Optimality criteria

If C and P are a solution to the above minimization problem, it must satisfy the following two criteria.

- Nearest Neighbor Condition:

$$S_i = \{X : \|X - c_i\|^2 \leq \|X - c_{n'}\|^2\} \quad \forall n' = 1,2,...,N\} \tag{4-8}$$

This condition means that the encoding region S_i should include all vectors that are closer to c_i than any of the other codevectors. For those vectors lying on the boundary, any tie-breaking procedure will do.

- Centroid Condition:

$$c_i = \frac{\sum_{X_m \in S_n} X_m}{\sum_{X_m \in S_n} 1}, \ i=1,2,...,N \tag{4-9}$$

It says that the codevector ci should be average of all those training vectors that are in encoding region S_i. We should ensure that at least one training vector belongs to each encoding region to guarantee that the denominator in the above equation is never 0.

3) LBG design algorithm

The LBG-VQ design algorithm is an iterative algorithm which alternatively solves the above two optimality criteria. The algorithm requires an initial codebook $C^{(0)}$. This initial codebook is obtained by the splitting method. In this method, an initial codevector is set as the average of the entire training sequence. This codevector is then split into two. The algorithm runs with these two vectors as the initial codebook. The two codevectors are splitted into four and the process is repeated until the desired number of codevectors is obtained. The algorithm is summarized below.

1) Given τ. Fixed $\varepsilon > 0$ to be a very small value.

2) Let $N = 1$ and $c_1^* = \frac{1}{M}\sum_{m=1}^{M} X_m$, and calculate $D_{ave}^* = \frac{1}{Mk}\sum_{m=1}^{M}\left\|X_m - c_1^*\right\|^2$.

3) Splitting: For *i=1,2,...,N*, set $c_i^{(0)} = (1+\varepsilon)c_i^*$, $c_{N+i}^{(0)} = (1-\varepsilon)c_i^*$ and set *N*=2*N*.

4) Iteration: Let $D_{ave}^{(0)} = D_{ave}^*$. Set the iteration index $i = 0$.

a) For *m*= 1, 2,…, *M*, find the minimum value of $\left\|X_m - c_n^{(i)}\right\|^2$ over all *n*=1, 2,…, *N*. Let n^* be the index which achieves the minimum. Set

$$Q(X_m) = c_{n^*}^{(i)} \tag{4-10}$$

b) For *n*=1, 2,…,*N*, update the codevector

$$c_n^{(i+1)} = \frac{\sum_{Q(X_m)=c_n^{(i)}} X_m}{\sum_{Q(X_m)=c_n^{(i)}} 1} \tag{4-11}$$

c) Set *i*=*i*+1.

d) Calculate $D_{ave}^{(i)} = \frac{1}{Mk}\sum_{m=1}^{M}\left\|X_m - Q(X_m)\right\|^2$

e) If $(D_{ave}^{(i-1)} - D_{ave}^{(i)}) / D_{ave}^{(i-1)} > \varepsilon$, go back to Step *a)*.

f) Set $D_{ave}^* = D_{ave}^{(i)}$. For *n*=1, 2,…,*N*, set $c_n^* = c_n^{(i)}$ as the final codevectors.

5) Repeat Steps 3) and 4) until the desired number of codevectors is obtained.

4) Proposed Algorithm

Considering both finding the initial cluster heads position for proper cluster formation and fast data compression, I propose our algorithm applied in sensor networks for both cluster formation and data compression as follows:

Step1: Create the codebook from training data set of the sensors.

Step2: Transmit the codebook to the base station.

Step3: Base station determines the CHs which has shortest distance to codebook.

Step4: Let the sensor collect data and fill the local buffer.

Step5: Compute the codebook update locally at CH and send to BS.

Step6: Sensor node with highest energy inside a fixed cluster becomes new CH.

Step7: Go to step 4 and repeat until the last node dies.

4.2 Integrated sleep scheduling and routing

Currently, energy supply is one of the fundamental bottlenecks in USNs. It is very costly and unpractical to replace sensor node batteries once they are deployed, both because of the large number of sensing nodes and the typically hazardous or unfriendly environment in which these nodes are deployed. Hence, prolonging network life becomes a primary concern in network design.

The sleeping technique has been used to conserve energy of battery powered sensors. Several researchers even suggest putting redundant sensor nodes into the network and allowing the extra sensors to sleep to extend network lifetime [99]. This approach is practical due to the low cost of individual sensors. When a sensor node is put into the sleep state, it is completely shut down, leaving only one extremely low power timer on to wake itself up at a later time. In a dense sensor network, rotating active and inactive sensors among the low power sensor members, some of which provide redundant data, is an intelligent way to manage sensors to extend its network lifetime. This leads to the following sleep scheduling problem: How does the CH or base station (BS) select which sensor nodes to be put into sleep, without compromising the sensing coverage capabilities of the whole network?

For a multi-hop USN rather than a single hop USN, an energy efficient routing protocol also needs to be considered. Hence, there exist more challenges than single hop networks, for example,

- The routing path (link) failure may happen during data transmission because of collision, node dying out (no battery), node busy, or other accidents. Some applications require real time information and data, which means retransmission is not possible. This motivates us to design a multi-path routing scheme for USNs.
- There exists energy constraint in USNs because most sensors are battery operated. This motivates us to consider energy aware routing.

In this chapter, I investigate the energy constraint problem in USNs and propose a sleep scheduling scheme in a single hop network based on Analytical Hierarchy Process (AHP). In addition, an integrated sleep scheduling and routing algorithm in a multi hop environment is proposed again based on AHP. In a single hop network, three factors contributing to the optimal nodes scheduling decision are considered and they are 1) distance to CH, 2) residual energy, and 3) sensing coverage overlapping, respectively. Our goal is to balance energy consumption in low power sensor nodes and extend the sensor network lifetime while maintaining adequate sensing coverage capabilities. In a multi hop network, I propose an integrated AHP based sleep scheduling and multipath routing scheme for USNs, each of which has three different criteria considered as well. I evaluate the efficiency of both proposed schemes in terms of energy consumption, lifetime and coverage, and compare with related work, that is, Linear Distance-based Scheduling (LDS) and random scheduling in single hop heterogeneous sensor networks, and Hop-based Sleeping Scheduling (HSS) algorithm and Geographical Multipath Routing (GMR) scheme in the multi hop case.

4.2.1 Single-hop case

I adopt the same radio model as stated in [14] with ε_{fs}=10$pJ/bit/m^2$ as amplifier constant, E_{elec}=50nJ/bit as the energy being dissipated to run the transmitter or receiver circuitry. It is assumed that the transmission between the common nodes or between the CH and its individual member node follows a second-order power loss model. The energy cost of transmission for sensor nodes at distance d from each other in transmitting an l-bit data is calculated as:

$$E_T(l,d) = lE_{elec} + l\varepsilon_{fs}d^2 \tag{4-12}$$

A typical sensor network could contain thousands of small sensors. In some specific applications, clustering has been employed to group a number of sensors, usually within a geographic neighborhood. In such a cluster based topology, sensors can be managed locally by a CH which is a node responsible for management in the cluster and for communication between the cluster and the base station.

I aim to enhance the efficiency of the given sensor network by enabling a balanced usage of energy across the nodes and an improved network lifetime without compromising network coverage. Figure 4.1 is the illustration of cluster based sensor network topology in which our proposed single hop sleep scheduling scheme is designed. I focus on energy consumption at the cluster level.

A. Assumptions

I consider the sleep node scheduling problem under several assumptions as follows:

- The target sensor network is heterogeneous with a large number of low power sensor nodes to serve as member nodes and a small number of more powerful nodes to serve as CHs. The motivation behind is to confine the complex hardware and additional battery to a few CH nodes. The low power nodes are simple in hardware and perform basic functions such as sensing and simple computations;
- A large number of sensor nodes are deployed over a sensing field, such that at least some sensor nodes can be put into the sleep state without degrading the sensing coverage of the network;
- The CHs can communicate directly with BS and vice- versa. Similarly, the CH can reach all the sensor members in the cluster in one hop and vice-versa. Thus, it is not needed for any routing strategy from the BS to any specific CH or from any CH to the individual sensor member.
- The application can tolerate some delay in reports from some sensors in each *round*.

B. Network Parameters

The user-defined parameters used in defining the network are listed below:

1) Fraction of nodes selected to sleep in a given round, 'r': This is the fraction of the total number of nodes in the network that are selected to sleep in each *round*.
2) Threshold limit, 'θ': This denotes the fraction of nodes in the network, which, when dead, determines the lifetime of the network.

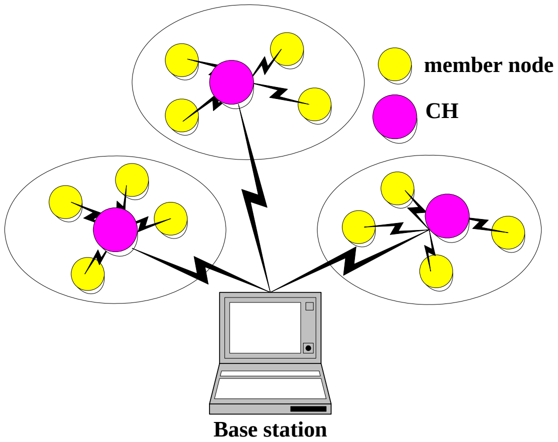


Fig. 4.1 Cluster based sensor network topology.

C. Sleep Scheduling Scheme by AHP

In our design, three factors influence the load balance and coverage directly, that is, 1) distance to CH, 2) residual energy, and 3) sensing coverage overlapping:

1) Distance to CH: Distance of a node to its CH. It can be approximated by the signal strength of radio transmission. The node with longest distance to the CH is preferred to be put into sleep.
2) Residual energy: Remaining battery of the sensor node. The initial energy is predefined. In addition, the energy consumption for transmission is calculated using Eq. (7) by CH.

3) Sensing coverage overlapping: Overlapped sensing range of a node by neighbor nodes. The node with larger overlapping degree, i.e., the node with higher redundancy, is desired to be selected as sleeping node.

This optimized sleep scheduling process is a multiple factors optimization problem and can be achieved by AHP, which is used to select the nodes eligible to sleep in one cluster. It is carried out in three steps:

Step 1: Collect information and formulate the sleeping nodes selection problem as a decision hierarchy of independent factors.

Step 2: Calculate the relative local weights of decision factors or alternatives of each level.

Step 3: Synthesize the above results to achieve the overall weight of each alternative nodes and choose the one with largest weight as the eligible sleeping node.

The goal of the decision "select a node eligible to sleep" is at the top level of the hierarchy as shown in Fig. 4.2. The next level consists of the three decision factors. At the bottom level there exist the m alternative sensor nodes to be evaluated.

In AHP modeling, the evaluation matrix A, here denoted as A1, is determined based on Eq. (3-21) as follows:

$$
A1 = \begin{array}{c|ccc}
 & \text{Distance to CH } (\alpha) & \text{Residual energy } (\beta) & \text{Sensing range overlapping } (\gamma) \\
\hline
\alpha & \alpha/\alpha & \alpha/\beta & \alpha/\gamma \\
\beta & \beta/\alpha & \beta/\beta & \beta/\gamma \\
\gamma & \gamma/\alpha & \gamma/\beta & \gamma/\gamma
\end{array}
$$

$$= \begin{bmatrix} 1 & 2/1 & 3/1 \\ 1/2 & 1 & 2/1 \\ 1/3 & 1/2 & 1 \end{bmatrix}$$

where the three criteria (distance to CH, residual energy and sensing range overlapping) are denoted by α, β and γ respectively. The selection of these initial values is motivated by our choice that "Distance to CH" is the most important, than "Residual energy" is the next important followed by "Sensing range overlapping" as the least important factor. This choice reflects a typical set of parameters for energy conservation.

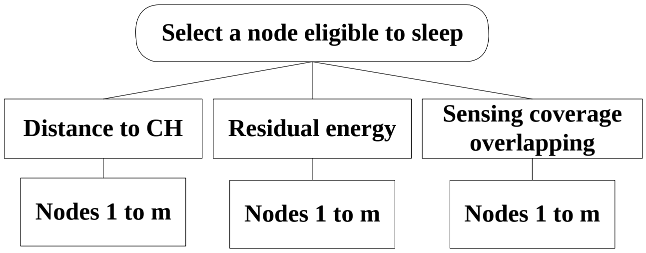


Fig. 4.2 AHP hierarchy for sleeping nodes selection in a single hop case.

The computed eigenvector W = [0.5396 0.2970 0.1634]. It indicates the local weight of the distance to CH, residual energy, and sensing coverage overlapping respectively so that we can see clearly that the distance to CH is the most important criterion, and sensing coverage overlapping is the least. According to Eq. (3-24), we can get the eigenvalue $\lambda_{max} = 3.0093$. Consequently, consistency ratio can be calculated as $CR = 0.0047 < 0.1$, thus matrix A satisfies the consistency check.

Each sensor node determines the weight matrices of alternatives under three factors and then gets global weight based on its specific situation. Afterwards, its eligibility as a sleeping node can be finally decided.

If there are eight candidate nodes in each cluster, all the eight weight matrixes of alternatives under three factors construct a 8×3 matrix, denoted as $W_{n_i/j}$, i=1, 2, … 8, j=1, 2, 3. The final weight of each alternative is calculated using Eq. (3-26) with n =3. The larger the final weight of node, the higher the probability of node which is eligible to be put into sleep. Thus, the r fraction of nodes with the largest weight are selected as the sleeping nodes in the current *round*.

4.2.2 Multi-hop case

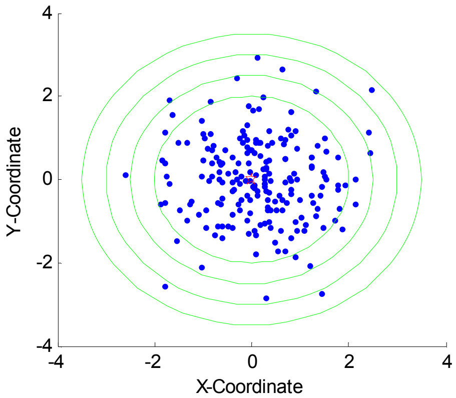


Fig. 4.3 Ring based multihop network topology.

In multihop environment, I investigate the sleep scheduling problem as well as multipath routing problem and propose an integrated AHP based sleep scheduling and routing (A-SR) scheme in a ring based multi-hop network topology with the destination (or BS) at the center, as shown in Fig. 4.3. In the existing multi hop sleep scheduling scheme (e.g., [68]), although routing is integrated, residual energy is the only factor considered. In the existing geographical routing approach (e.g., [85]), the path selection doesn't consider the remaining battery capacity of each node, which is a very important factor for energy constraint sensor networks. In our A-

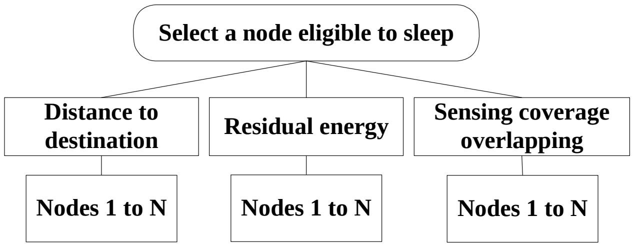
SR sleep scheduling part, *distance to destination, residual energy,* and *sensing coverage overlapping* are included, with the latter two factors the same as the single hop case. In contrast with the single hop case, the node with shorter distance to destination is preferred to be put into sleep since they are more energy hungry as a relay. In routing part, *distance to the destination, residual energy*, and *queue size* of each sensor node are included, with the former two factors the same as the proposed multi hop sleep scheduling. Our scheme is a fully distributed approach where each sensor only needs the above parameters, and I use AHP to handle these parameters in the A-SR.

In our A-SR scheme, I only keep the second assumption from single hop case. And I additionally assume that the event detection by the nodes in the network occurs periodically and all nodes are synchronized. Thus our A-SR can be executed round by round based on the period. For A-SR routing part, the detailed explanations of the three criteria for next hop relay node selection are given as follows:

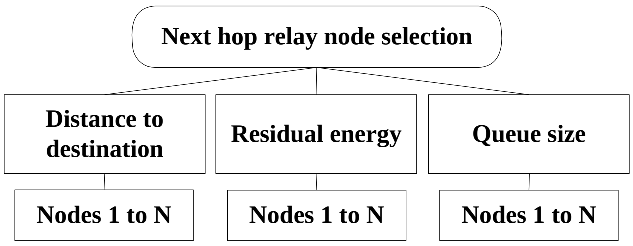
1) Distance to destination: Distance of a node to BS which is the destination. The geographical location of destination is known to the source node (as in [85]), and the physical location of each sensor node can be estimated easily if the locations of three sensor nodes (within a communication range) are known in a USN. The node with shorter distance to the destination is preferred to be selected.
2) Residual energy: Remaining battery of the sensor node. The energy consumption for transmission and reception can be calculated using Eqs. (4-12) and (3-6).
3) Queue size: It indicates the buffer capacity at the node. This parameter helps avoid packet drops due to congestion at the receiver.

The optimized node selection in multipath routing is also a multiple factors optimization problem and can be achieved using AHP.

In our A-SR for M-path routing, the source node select M nodes in its communication range for the first hop relay. Assume there are N ($N > M$) nodes in its communication range, nodes that are farther to the destination node than the source node are not considered. Choosing M nodes from remaining eligible nodes is based on AHP (as will be described in detail). Starting the second hop, each node in the M-path selects its next hop node also using AHP.



(a) AHP hierarchy for sleep nodes selection



(b) AHP hierarchy for next hop relay nodes selection

Fig. 4.4 AHP hierarchy for decision making in a multi-hop network.

In the AHP hierarchy model, the goal of the decision "next hop relay node selection" is at the top level of the hierarchy as shown in Fig. 4.4 (b). The next level consists of the three decision factors and at the bottom level there exist the N alternative sensor nodes to be evaluated.

We assume that each sensor node keeps a table which has some information about its neighbor nodes: locations, battery level, and queue size. The table is updated periodically by the locally-broadcasted information (beacon) from each neighbor node. We define a time interval T, during which the three parameters (locations, battery level, and queue size) do not change very much. This time interval T is the shortest time duration that a sensor node will send another beacon. Each sensor examines itself the status of the three parameters in every period T, and if a certain parameter has changed above a threshold, it will locally broadcast a beacon.

In the route discovery phase, the source node uses AHP model to evaluate all eligible nodes (closer to destination) in its communication range based on the parameters of each node: distance to destination, residual energy, and queue size. The source node chooses the top M nodes based on the local weight that this node will be selected. And the source node sends a Route Acknowledgement (RA) packet to each desired node, and each desired node will reply using a REPLY packet if it is available. The structure of RA and REPLY is summarized in Table 4.1. If after a certain period of time, the source node did not receive REPLY from some desired node, it will pick the node with highest weight among the remaining N-M nodes. In the second hop, the selected node in each path will choose its next hop node using the same process. As illustrated in Fig. 4.5, node B needs to choose one node from four eligible nodes C, D, E, and F based on their three parameters, and sends RA packet to the selected node and waits for REPLY. If the top one node is unavailable (for exmaple, selected by another path), then the top second node will be selected. Consequently, M paths can be set up.

For A-SR sleep scheduling part, we only present its AHP hierarchy model which is shown in Fig. 4.4 (a), due to the similarity to the single hop case.

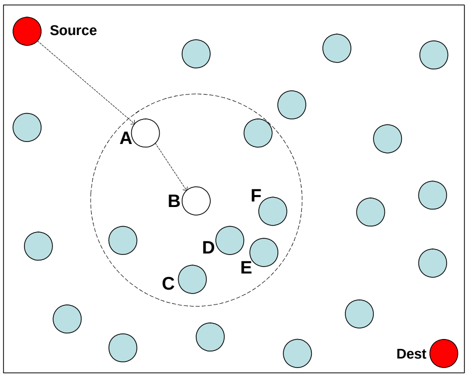


Fig. 4.5 Illustration of next hop node selection.

For energy analysis, I adopt the previously presented transmission energy model, i.e., Eq. (4-12). In multihop networks, the energy for reception and data aggregation also need to be taken into account. Thus, to receive an l-bit data, the radio energy expenses follow Eq. (3-6), and the energy for data aggregation is set as $E_{DA} = 5\ nJ/bit$, the same as [14].

In AHP modeling, the evaluation matrices for A-SR, here denoted as A2, is determined based on Eq. (3-21) as follows:

$$A2 = \begin{array}{c} \\ \alpha \\ \beta \\ \gamma \end{array} \begin{array}{c} \begin{array}{ccc} \alpha & \beta & \gamma \end{array} \\ \begin{bmatrix} 1 & 2/1 & 3/1 \\ 1/2 & 1 & 2/1 \\ 1/3 & 1/2 & 1 \end{bmatrix} \end{array}$$

where the three criteria, shown in Fig. 4.4 (a) and (b) from left to right, are denoted by α, β and γ respectively.

The computed eigenvector W has the same value as the single hop case since I assumed the same evaluation matrix. So we can observe that the distance to destination is the most important criterion, and sensing coverage overlapping and queue size are the least. We can again get the eigenvalue $\lambda_{max} = 3.0093$, and consequently matrix A2 satisfies the consistency check.

Each sensor node determines the weight matrixes of alternatives under three factors and then gets global weight based on its specific situation. Its eligibility as next hop relay node and sleep node can be finally decided by the AHP hierarchy model.

Table 4.1 RA and REPLY message structure

Type	Desired Node ID	Self Node ID	Dest_X	Dest_Y	Src_ID

Chapter 5 Performance Evaluation

5.1 Experimental methodology

This chapter is organized following the presentation of the earlier algorithms, to wit, firstly, experiments pertaining to the mobility-assisted relocation algorithms, secondly, to the self-organization methods.

The purpose of the experiments relating to deployment is to confirm via simulation important aspects relating to the performance of the dual coverage and energy consumption, by categorizing the algorithms to match the mobility metrics and by comparing the set of algorithms in each category, giving some insight on choosing protocols in different situations, lastly, to confirm via simulation practical and realistic implementation issues. These performance-related experiments compare the proposed algorithms in this dissertation to random deployment-based algorithms because the nature of the initial deployment assumptions (Chapter 2) leads to random distribution of sensor nodes. In addition, a second perfomance comparison with the performance of a unifromly distributed sensor network or related work is conducted to verify that all the proposed algorithms have similar performance or outperform a standard coverage-driven deployment algorithm.

The purpose of those experiemnts that relate with network maintenance is to confirm predictive expectations of the model presented in Section 3.4. The purpose of the experiments relating to sef-organization is mainly to test the effectiveness of our algorithms in decreasing energy consumption without compromising the coverage, secondly to confirm via simulation practical implementation issues.

We analyze the performance of our protocols from two aspects: deployment quality and energy consumption. Deployment quality is measured by the sensing coverage and the time (number of rounds) to reach this coverage. As for movement, we use moving distance and the number of movement as the metrics.

5.2 Mobility-assisted relocation

5.2.1 PSOA

5.2.1.1 Optimization of Coverage

A. Binary Model Case

The PSO starts with a “swarm” of sensors randomly generated. As shown in Fig. 5.1 is a randomly deployed sensor network with coverage value 0.4484 calculated using Eq. (3-3). A linear time decreasing inertia weight value from 0.95 to 0.4 is used in order to make better global exploration of the search space first and to get more refined solutions gradually. The range is decided according to [91]. Acceleration coefficients c_1 and c_2 both are set to 2 as proposed in [91]. For this performance study, I select a large scale deployment of 50x50 square sensor network because the prevalent modern-day view prefers it. For optimizing coverage, we have used 20 particles, which are denoted by all sensor nodes coordinates and the maximum number of generations we are running is 500. The maximum velocity of the particle is set to be 50. The sensing range of each sensor is set to be 5 units. An upper bound on the coverage is given by the ratio of the sum of the circle areas (corresponding to sensors) to the total area of the sensor field. In this simulation, the upper bound evaluates to be 0.628, which is calculated from the perfect uniform distribution case without any overlapped area. The coverage is calculated as a fitness value in each generation.

Fig. 5.2 is the coverage optimization results. The coverage improvement verses number of iterations in one run is shown in Fig. 5.2 (a) and the final achieved coverage values for six runs are shown in Fig. 5.2 (b). Compared with the upper bound 0.628 (which is achieved in a uniformly distributed ideal coverage driven situation), the difference between the average value 0.58 for six runs and upper bound is small.

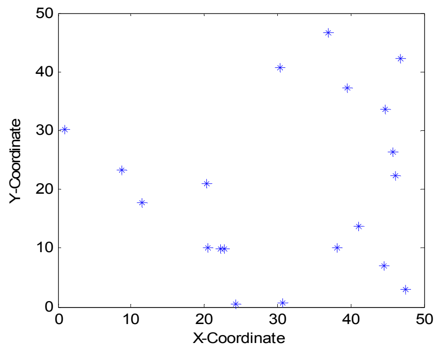
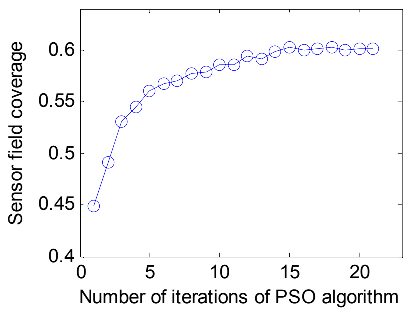
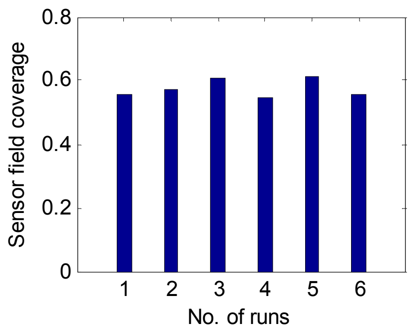


Fig. 5.1 Randomly deployed sensor network (Coverage value=0.4484).



(a) Coverage improvement



(b) Coverage for six runs

Fig. 5.2 Optimal coverage results (binary sensing model)

B. Probabilistic Model Case

In probabilistic model case, I use a randomly deployed sensor network as shown in Fig. 5.3, with coverage value 0.31 calculated by Eq. (3-4) and approximate method mentioned in section 3.1.1. PSO algorithm parameters are set the same as binary model case, however, the other parameters of sensor models are set to be r=5, r_e=3, λ=0.5, β=0.5, c_{th}=0.7.

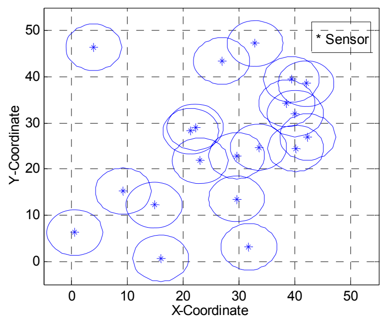
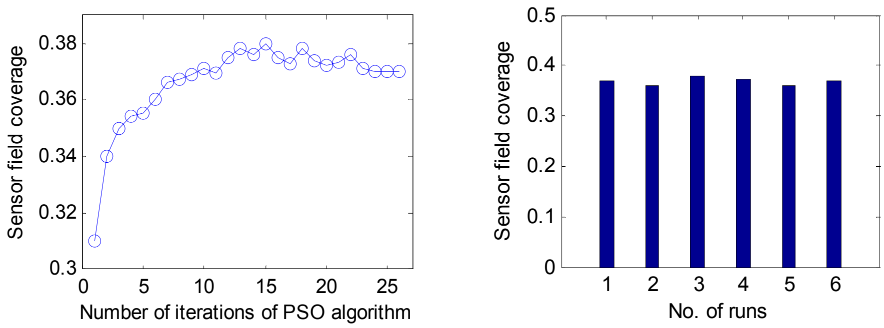


Fig. 5.3. Randomly deployed sensor network with r=5 (Coverage value=0.31)



(a) Coverage improvement (b) Coverage for six runs

Fig. 5.4 Optimal coverage achieved using PSO algorithm (probabilistic sensing model)

Fig. 5.4 (a) shows the improvement of coverage during the execution of the PSO algorithm. Note that the upper bound for the coverage for the probabilistic sensor detection model (roughly 0.38) is lower than the upper bound for the case of binary sensor detection model (roughly 0.628). This is due to the fact that the coverage for the binary sensor detection model is the fraction of the sensor field covered by the circles. For the probabilistic sensor detection model, even though there are a large number of grid points that are covered, the overall number of grid points with coverage probability greater than the required level is fewer. I also show the achieved coverage for six runs in Fig. 5.4 (b), and the average is nearly 0.37 which has little difference from the upper bound.

5.2.1.2 Optimization of Energy Consumption

I have compared the average distance traveled (an indicator of moving energy consumption), PSO has a result of 49 and the ideal uniformly distribution strategy take 112, which shows a large moving energy saving of PSO.

After the optimization of coverage, all sensors move to their final locations in setup phase. Now the coordinates of potential CHs are set as particles in the sensor network. The communication range of each sensor node is 15 units with a fixed remote base station at (25, 80). I start with a minimum number of clusters acceptable in the problem space to be 4. The node, which will become a CH, will have no restriction on the transmission range. The nodes are organized into clusters by the base station. Each particle will have a fitness value, which will be evaluated by the fitness function (12) in each generation. Our purpose is to find the optimal location of CHs. Once the position of the CH is identified, if there is no node in that position then a potential CH nearest to the CH location will become a CH.

I also optimized the placement of CH in the *2-D* space using GA. I used a simple GA algorithm with single-point crossover and selection based on a roulette-wheel process. The coordinates of the CH are the chromosomes in the population. For our experiment I use 10 chromosomes in the population. The maximum number of generations allowed is 500. In each evolution I update the number of nodes included in the clusters. The criterion to find the best solution is that the total fitness value should be minimal.

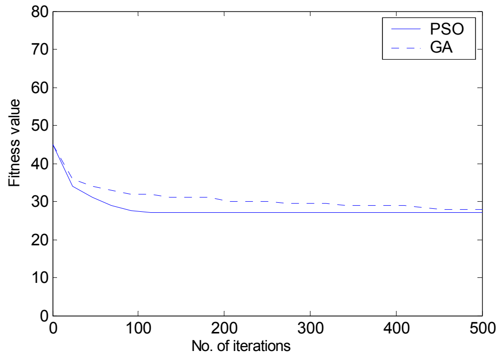


Fig. 5.5 Comparison of convergence rate between PSO and GA based on Eq. (3-12).

Fig. 5.5 shows the convergence rate of PSO and GA. I ran the algorithm for both approaches six times and in every run PSO converges faster than GA which was used in [55] for coverage and lifetime optimization. The main reason for the fast convergence of PSO is due to the velocity factor of the particle.

Fig. 5.6 and Fig. 5.7 show the final cluster topology in the sensor network space after coverage and energy consumption optimization when the number of clusters in the sensor space is 4. We can see from the figure that nodes are uniformly distributed among the clusters compared with the random deployment as shown in Fig 5.1 and 5.3. The four red stars denote CHs, the blue tiny circles and diamonds are sensor members, and the dashed circles are communication range of sensor nodes. The energy saved is the difference between the initial fitness value and the final minimized fitness value. In this experiment, it is approximately 16.

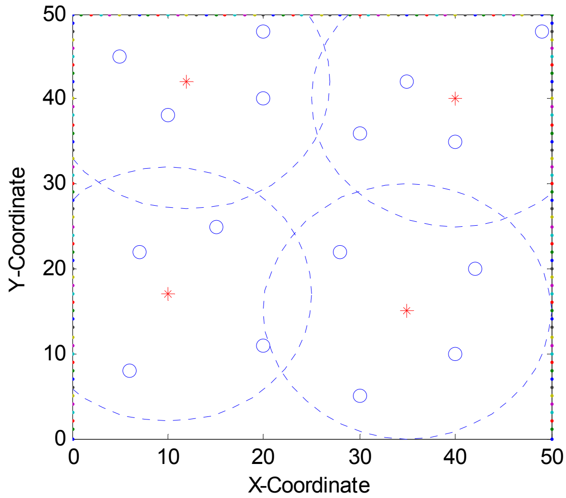


Fig. 5.6 Final cluster formation by PSO (Binary model case)

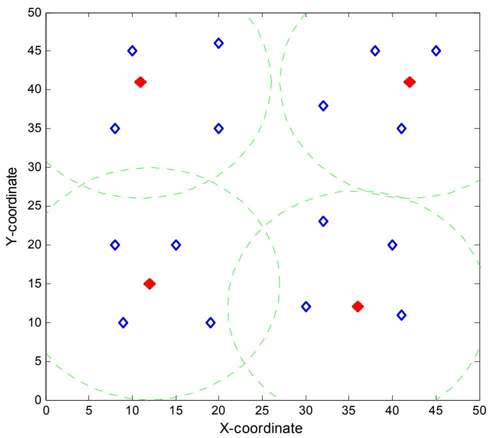


Fig. 5.7 Final cluster formation by PSO (Probabilistic model case).

5.2.2 RSBA

The performance of the proposed movement assisted algorithm RSBA is evaluated by simulation. For the convenience of comparison with related work, I set the initial parameters the same as in [51]: 30 randomly placed nodes in a region of size 10×10 are used for initial deployment; the r and R used in the experiment are 2 and 4 m, respectively. In Fig. 5.8, the coverage and connectivity of the initial random deployment before running the algorithms are shown. The green circles are used to show the sensing range *r* of the nodes. Communications are possible within the *R* between nodes that are connected by a dashed line.

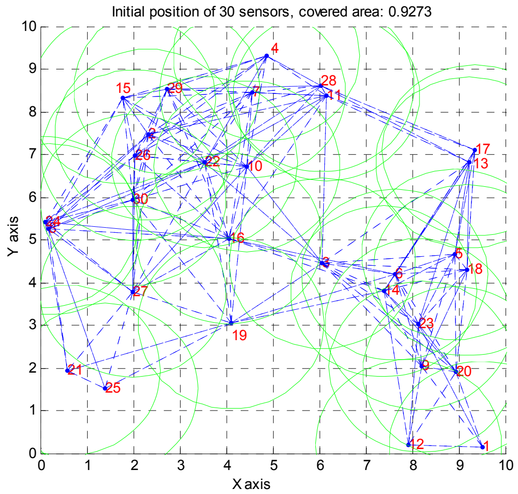


Fig. 5.8 Initial random deployment with sensing range 2m and communication range 4m.

Fig. 5.9 shows the detected virtual node points (labeled as 31 and 32) in coverage hole and the redundant nodes nearest to 31 and 32 are 14 and 17 respectively. Both the coverage holes and the redundant nodes are judged by CHs. This information is then broadcasted by CHs to the whole network. The parameter values needed are: threshold value T_1= r/4.

Fig. 5.10 shows the two shortest paths found (14→19→31 and 17→32) by A* algorithm from redundant nodes to virtual node points. This is also the actual path of individual nodes as they move by relay shift, in which sensor node move only one hop at a time which can guarantee the connectivity. For the initial distribution of Fig. 5.8, each node moves a distance of 2.6157 on average and the standard deviation of distance traveled is 0.5714. When the average distance traveled is small, the corresponding energy for locomotion is small. Also, when the standard deviation of distance traveled is small, the variation in energy remaining at each node is not significant and a longer system lifetime with desired coverage can be achieved. Fig. 5.11 shows the final node positions with desired coverage=0.9923 after executing RSBA. Note that the original 30 sensor nodes are finally reorganized and relabeled.

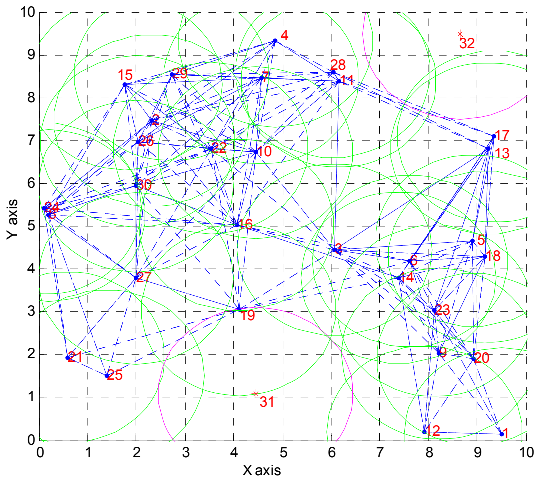


Fig. 5.9 Determine virtual node point in uncovered area and redundant nodes.

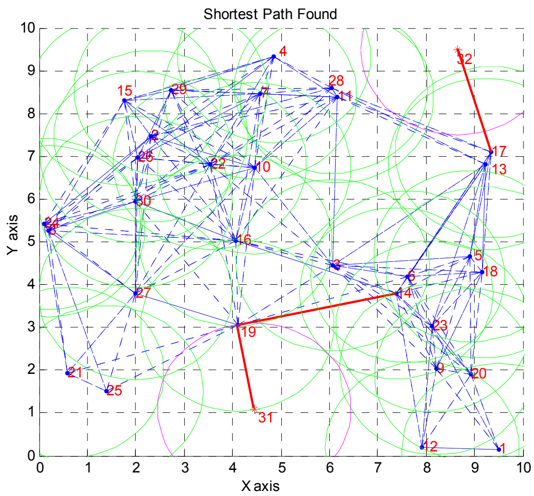


Fig. 5.10 Find shortest path by A* algorithm from redundant node to virtual node point.

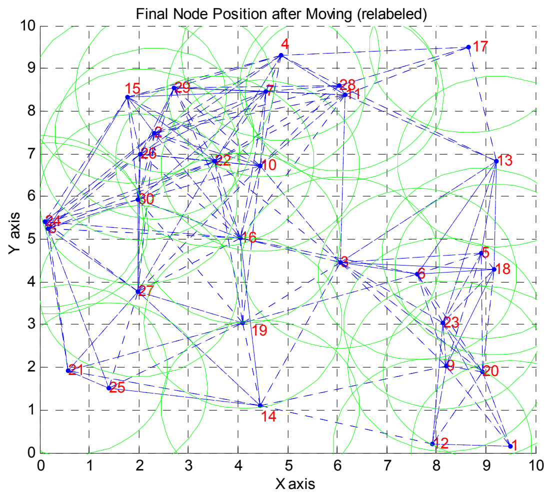


Fig. 5.11 Final node positions after executing proposed movement-assisted deployment algorithm

Next, the performances of RSBA are compared with DSSA, IDCA, and VDDA [51] in terms of coverage, movement distance until convergence, and time. Results are presented in Figures. 5.12–5.14. These results are obtained for different number of nodes dispersed over a fixed ROI of size 10×10, i.e., for different node densities to examine the relation between node densities and the performance metrics. The number of nodes varies from 20 to 40 and results are averaged over 10 runs (initial random distributions) for each node density.

Fig. 5.12 shows the improvement in coverage area from the initial random deployment for RSBA, DSSA, IDCA, and VDDA. All four algorithms exhibit a similar performance. Although the coverage of RSBA (99%~1) is not always the highest among the four algorithms, this number is often satisfactory for many application requirements.

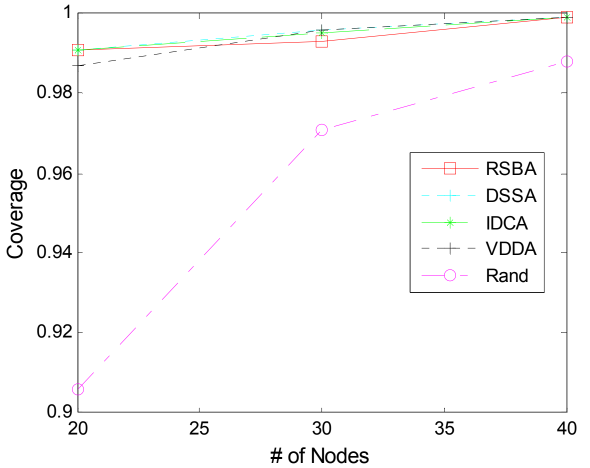


Fig. 5.12 Coverage comparison.

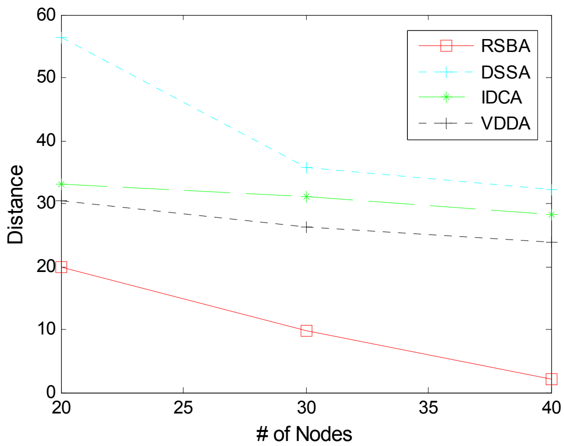


Fig. 5.13 Total distance traveled comparison.

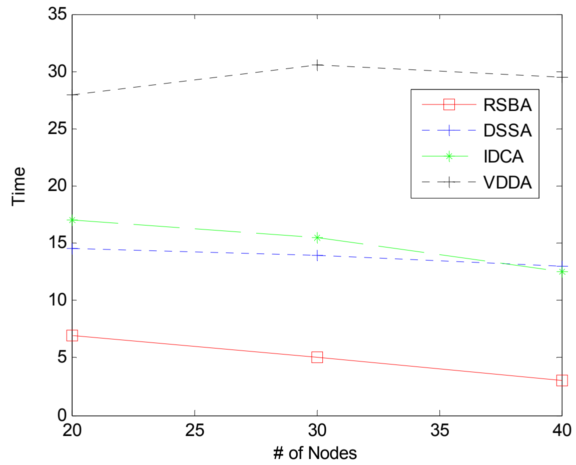


Fig. 5.14 Termination time comparison.

Fig. 5.13 shows the significant reduction of total distance traveled by RSBA compared with other three algorithms. In fact, distance moved here is used as the indicator of energy consumption. In RSBA, only very few numbers of nodes need to move and each sensor movement is bounded by only one hop. However, almost every node needs to move in the other three algorithms. So it is obvious that our proposed algorithm can save much more energy compared with related methods. Fig. 5.14 shows that RSBA leads to faster deployment than DSSA, IDCA, and VDDA. Termination time is measured in the number of iterations until the algorithms stop.

5.2.3 EFOA

For the convenience of comparison between EFOA and related work FOA, I set the initial parameters the same as in [56]: various number of sensors deployed in a field of 10×10 square area are investigated; the r and R used in the experiment are 1m and 2m (2m and 4m) respectively. So d_n should be ranged as 0~2 (0~4), not 0~10 as set by [56]. I assume each sensor is equipped with an omni antenna to carry out the task of detection and communication. Evaluation of our EFOA algorithm follows three criteria: field coverage, energy consumption and convergence. Results are averaged over 100 Monte Carlo simulations.

Fig. 5.15 shows that the coverage of the initial random deployment, FOA and EFOA when r=1m and R=2m. The FOA and EFOA algorithms have similar results that both of them can improve the network coverage by 20% ~ 30% in average.

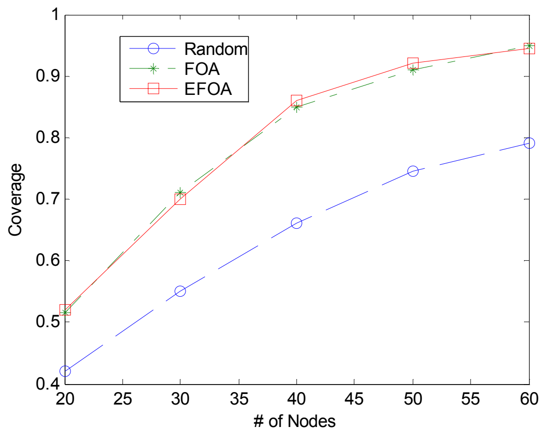


Fig. 5.15 Coverage vs. Number of Nodes (R=2, r=1).

Fig. 5.16 gives the results when r=2m and R=4m, the coverage comparison of 1) random deployment, FOA, EFOA and RSBA with binary sensing model and 2) random deployment, EFOA and RSBA with probabilistic sensing model (denoted as Random-Prob, EFOA-Prob, and RSBA-Prob). In the case of binary sensing model, when 20 sensors are deployed, initially the coverage after random deployment is around 86%. After FOA and EFOA algorithm are executed, the coverage reaches 97%. RABA even has higher coverage ratio up to 99%. The coverage is dramatically improved in the low density network. The coverage ratio in case of probabilistic sensing model has similar improvement pattern by EFOA and RSBA compared with random deployment. The above two figures indicate that instead of deploying large amount of sensors, the desired field coverage could also be achieved with fewer sensors.

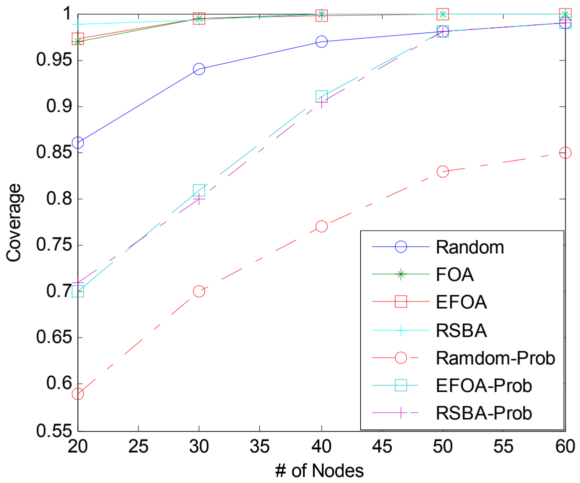


Fig. 5.16 Coverage vs. Number of Nodes (R=4, r=2).

Fig 5.17 shows the total number of nodes that remain alive over time where each node begins with 2J of energy and when R=4 and r=2. The number of nodes in EFOA remains same for a long time and they die out quickly almost at the same time, while the first node dies the earliest in FOA and RSBA in between. The reason is that after some operation time, the network display heterogeneous characteristics, however, FOA doesn't consider the residual energy of nodes, so the energy difference among sensors becomes significant as time goes on. Network lifetime is the time span from the deployment to the instant when the network is considered nonfunctional. When a network should be considered nonfunctional can be generally considered as the instant when the first sensor dies or a percentage of sensors die and the loss of coverage occurs. In RSBA, the uniformity is worse than EFOA but better than FOA. Thus the lifetime is prolonged in EFOA compared with FOA.

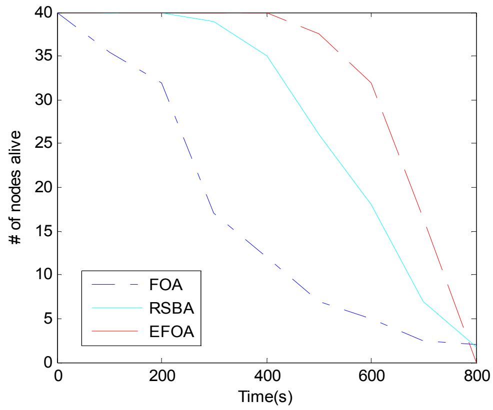


Fig. 5.17 # of nodes alive over time where each node begins with 2 J of energy. (R=4, r=2).

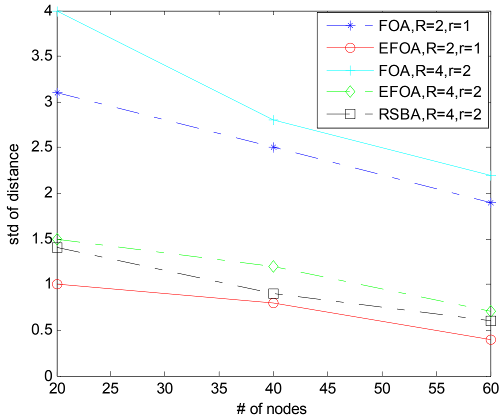


Fig. 5.18 Standard deviation of distance traveled verses number of nodes.

Fig. 5.18 shows EFOA has much lower standard deviation of distance compared with FOA in both cases when R=4, r=2 and R=2, r=1 while slightly higher than RSBA in the former case with various number of nodes. When the standard deviation of distance traveled is small, the variation in the energy remaining at each node is not significant and thus a longer system lifetime with desired coverage can be achieved. However, in case many sensors don’t need to move in RSBA, although the standard deviation is low, it causes lower uniformity and thus slightly shorter lifetime compared with EFOA.

5.2.4 Mobile nodes relocation in hybrid sensor network

In order to evaluate the relocation scheme by AHP, I compare it with random deployment case. In our simulation, the 50*m* by 50*m* square monitored area is divided into 100 uniform square grids. Each grid has the same length of 5*m*, and all nodes equip with identical sensors with sensing radius equal to 5*m*. The communication range is set as 10*m* to ensure the network connectivity. The moving style of a mobile node is flip by flip until the stable status is achieved. One flip distance is assumed to be 2*m*.

In AHP modeling, the matrix A is determined as follows according to Section 3.3:

$$A = \begin{array}{c} \\ \text{Coverage hole} \\ \text{Hot spot} \\ \text{Obstacle} \\ \text{Non-boundary} \end{array} \begin{array}{c} \begin{array}{cccc} \text{Coverage hole} & \text{Hot spot} & \text{Obstacle} & \text{Non-boundary} \end{array} \\ \begin{bmatrix} 1 & 3/1 & 2/1 & 5/1 \\ 1/3 & 1 & 1/2 & 3/1 \\ 1/2 & 2/1 & 1 & 4/1 \\ 1/5 & 1/3 & 1/4 & 1 \end{bmatrix} \end{array}$$

The computed eigenvector W = [0.4729 0.1699 0.2844 0.0729]. It indicates the local weight of coverage hole, hot spot, obstacle and non-boundary, respectively, so that we can see clearly that coverage hole is the most important criterion, and non-boundary is the least. According to Eq. (3-20), we can get the eigenvalue $\lambda_{max} = 4.0505$. Consequently, consistency radio can be calculated as $CR = 0.02 < 0.1$, thus matrix A satisfies the consistency check.

Each mobile sensor node determines the weight matrixes of alternatives under four factors[1] and then gets global weight based on its specific location and environment characteristics. Its moving direction can be finally selected by the AHP model.

In contrast to random deployment which achieves desired coverage with 70 static sensors deployed, the proposed AHP based scheme can achieve the same amount of coverage (k coverage can be guaranteed in hot spot with $k \geq 1$) using only a combination of approximate 20 static and 20 mobile sensors.

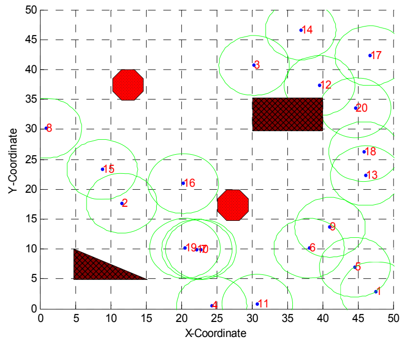


Fig. 5.19 Initial static sensor nodes placement

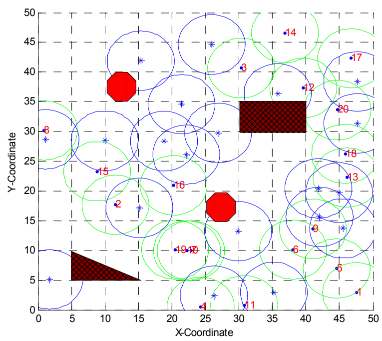


Fig. 5.20 Random mobile nodes deployment

[1] Coverage holes positions and area can be calculated by using Voronoi diagrams as in [50].

In Fig. 5.19, the static node locations and coverage of the initial random deployment before running the algorithms are shown. Tiny points with red numerical label beside represent the positions of 20 static nodes. The two red small disks denote the hot spots, the rectangle and triangle denote the obstacles in the environment. It's obvious that many uncovered areas exist and the hot spots are not well covered. Fig. 5.20 shows the random 20 mobile nodes deployment without change of original static sensors placement. The tiny stars denote the mobile node positions, and the green and blue circles represent the sensing range of the static sensor and mobile sensor respectively. The final mobile node positions with desired coverage after executing AHP based algorithm are shown in Fig. 5.21.

Fig. 5.22 provides the coverage ratio comparison between proposed AHP based redeployment and random deployment. The proposed scheme is only compared with the random deployment case because of our different assumptions from other existing mobile node relocation schemes. In AHP based redeployment, the coverage is achieved by deploying a hybrid sensor network in which mobile nodes occupy a half and the environment has a 3% obstacle area. The coverage here is defined as the ratio of the union of all sensor nodes' sensing areas to the whole monitored environment excluding obstacles. For the detailed explanation of coverage ratio calculation method, please refer to Chapter 3, Section 3.1.1. Note that, as the number of mobile nodes increases (the total number of nodes also increase), the coverage increases sharply because the sensing field becomes more flexible by movement of sensors.

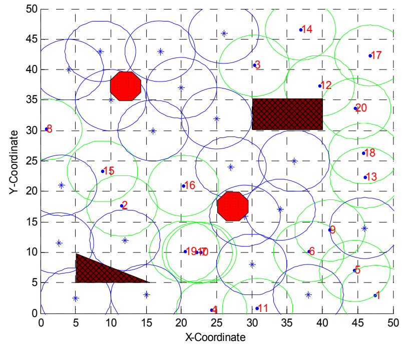


Fig. 5.21 Mobile nodes relocated

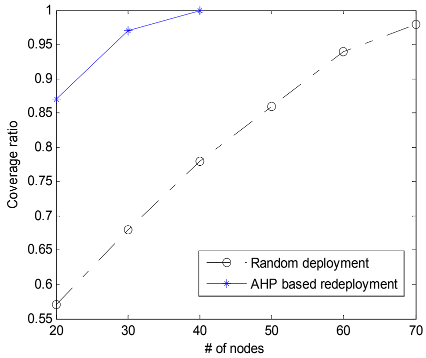
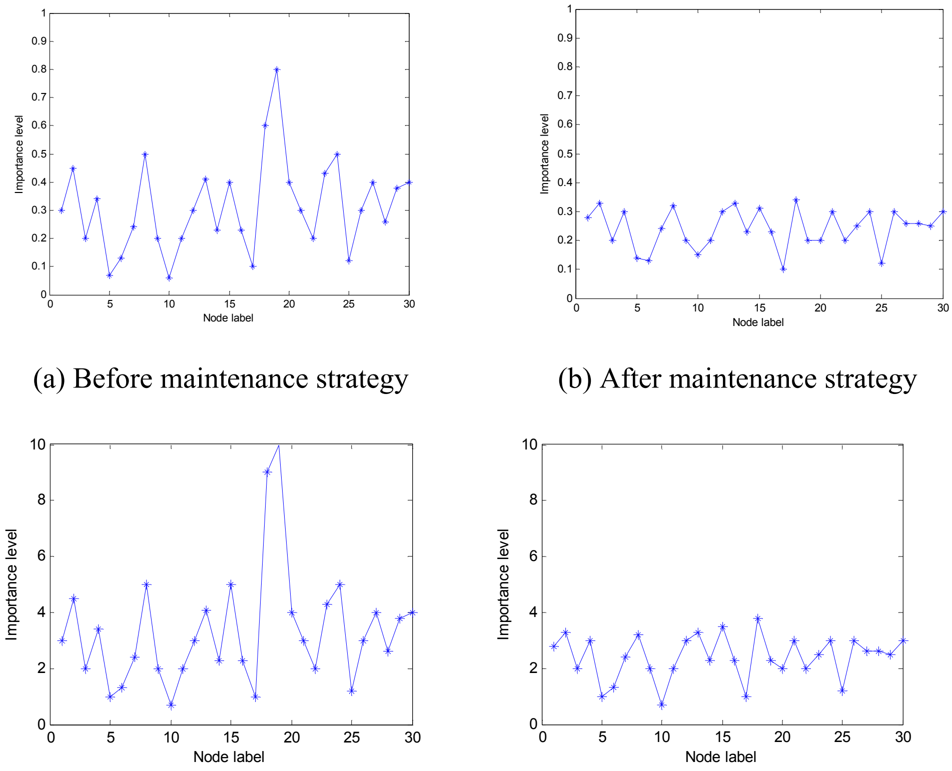


Fig. 5.22 Coverage vs. total number of nodes

5.2.5 Network maintenance strategy

The network maintenance strategy is simulated thereafter as Figure 5.23 shows. The parameter $x_{\max}$ is set to be 50, sampling period is 5s.Total number of nodes in the network is 30, and two of the most importance nodes are the nodes labeled as 18 and 19 which have the highest importance level. After adding four new nodes close to node 18 and 19, the importance level distribution become nearly uniform compared with the case before executing network maintenance strategy. Thus the working load of the "busy" nodes can be shared by the backup nodes and the lifetime can be further prolonged. The importance level distribution before and after maintenance strategy based on Eq. (3-17) are shown in Fig. 5.23 (a) and (b), and the results based on Eq. (3-18) are shown in Fig. 5.23 (a) and (b).



(a) Before maintenance strategy (b) After maintenance strategy

(c) Before maintenance strategy (d) After maintenance strategy

Fig. 5.23 Importance level verses node serial number

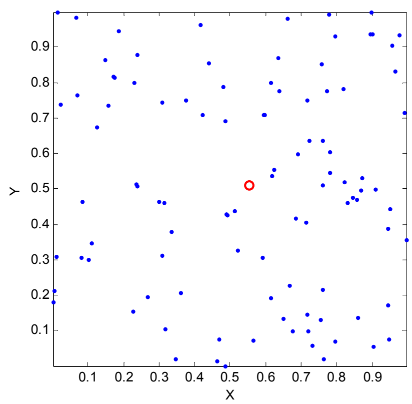
5.3 Self-organization of static sensor networks

5.3.1 Vector-quantization based clustering

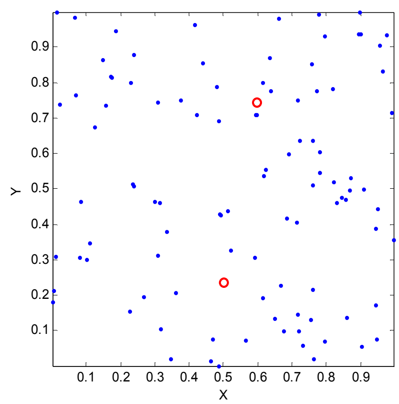
I simulate a wireless sensor network in a $100m \times 100m$ field. I set E_{elec} as E_{elec}=50nJ/bit, E_{DA} as E_{DA}=50$nJ/bi/report$ and the amplifier constant is taken as ε_{fs} =10$pJ/bit/m^2$, ε_{mp} = 0.0013$pJ/bit/m^2$, thus d_0 can be computed as about 87m. I set total sensor nodes number n=100, and the optimal number of constructed clusters can be approximately 8 according to Eq. (4-6).

So the number of codevectors N is 8. I take $\tau = \{X_1, X_2\}$ with X_1 as the location of sensor nodes, and X_2 as temperature and humidity sensing values. I regard the source vectors as 2-dimensional. So the sample data set is a 100×4 matrix. Without losing generality, I regard the sensing values to be the coordinates divided by 100. Thus when I simulate, I standardize the coordinates range as [0, 1] instead of [0, 100m].

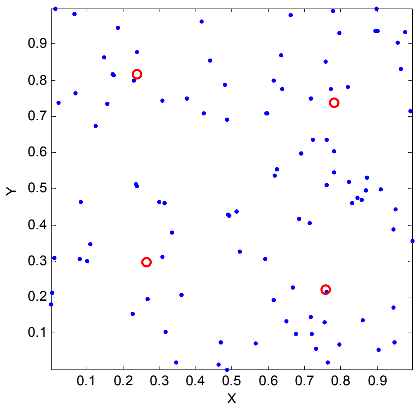
Figure 5.24 (a)-(d) show four snapshots of the VQ-LBG algorithm based codebook creation. The red circles represent codebook. Figure 5.25 is the final cluster formation in which the nodes with same color are bunched into one cluster. The sensor node in one cluster which has the shortest Euclidean distance to its codevector is selected as CH. Figure 5.26 is an example of randomly generated 8 cluster heads marked as red stars which form uneven cluster distribution used in LEACH. We can see that compared with VQ-LBG based cluster formation result, randomly generated cluster heads cannot always guarantee uniform cluster formation.



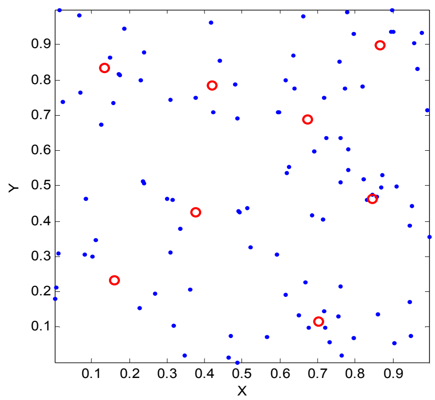
(a) 1 codevector



(b) 2 codevectors



(c) 4 codevectors



(d) 8 codevectors

Fig. 5.24 Snapshots of the VQ-LBG algorithm based codebook creation

Table 5.1 Loop count and distortion vs No. of centers during the execution

No. of centers	Loop count	distortion
2	9	10.834
4	6	4.20282
8	8	1.83539

Table 5.1 shows the loop count and distortion verses number of centers during the algorithm execution process. It is obvious that the distortion is decreased significantly after the final codebook with 8 centers is determined.

Figure 5.27 exhibits the distribution of the number of clusters in randomly selected 50 rounds in both proposed algorithm and LEACH. The number of clusters varies widely in each run in LEACH; on the other hand, the cluster number varies narrowly at the optimal range in proposed algorithm. Although the clusters are fixed in our algorithm and only the cluster head nodes are rotated, once the clusters are formed, there is no set-up overhead at the beginning of each round. Depending on the cost of forming adaptive clusters, our approach where the clusters are formed once and fixed and the cluster head position rotates among the nodes in the cluster is more energy efficient than LEACH. Besides that, our method at the same time guarantees fast data compression which is also an important issue in WSNs due to the scare resources of sensor node.

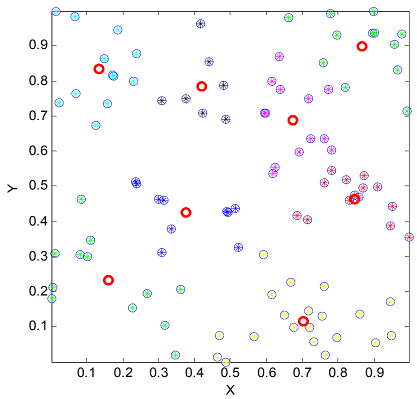


Fig. 5.25 Final cluster formation in which nodes with same color are bunched into one cluster

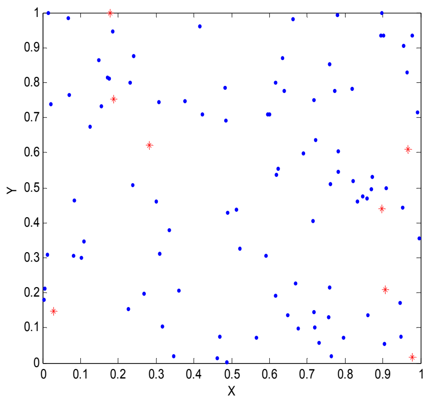


Fig. 5.26 An example of randomly generated 8 CHs marked as red stars which form uneven cluster distribution

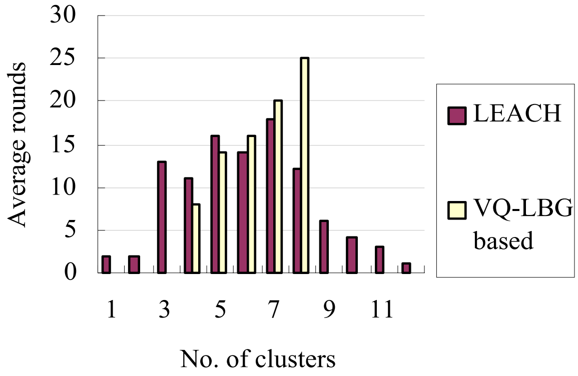


Fig. 5.27 The number of clusters in each round in both proposed algorithm and LEACH

5.3.2 Integrated sleep scheduling and routing

In order to evaluate the single hop sleep nodes scheduling scheme by AHP, we compare it with the upper (lower) bound (which optimizes merely the current factor), LDS [67] and random scheduling scheme. In our simulation, the 50*m* by 50*m* square monitored area is assumed. All nodes except CHs equip with identical sensors and the sensing and communication range are equal to 8*m* and 16*m* respectively. Initial energy in each node is 2*J*. I set the total number of nodes to be N_t=50 and number of static clusters to be 2. Thus the number of nodes in each cluster is 25 by assuming a uniform distribution of nodes.

Assume the CH plans to allow 25*r* nodes in its cluster to sleep in each cycle. In the random scheduling scheme, the CH randomly selects *r* fraction sensor nodes to sleep. At first, we compare the average energy consumption in a cluster by AHP based scheme and random scheduling scheme to show the energy that can be conserved by our scheme. Figure 5.28 provides the energy consumption verses the fraction of sleeping nodes of the three schemes. Furthermore, I also consider the ideal case where Eq. (3-6) is used to determine the minimum energy consumption which provides a lower bound on average energy consumption. It shows that the energy consumption in case of the proposed AHP based scheme is less than that of the random scheme, however slightly more than that of the LDS. The energy savings can be enhanced with an increasing value of *r*. For an *r* value of "0.7", the energy consumed by the AHP based scheme is 49.3% less than by random scheme and 9% more than LDS.

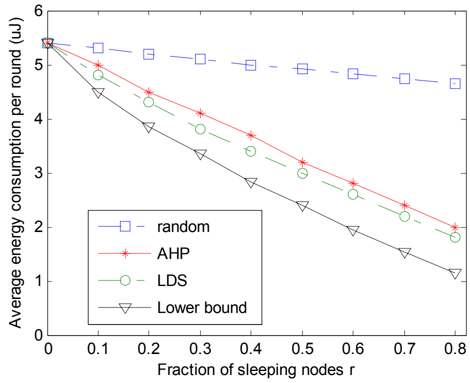


Fig. 5.28 Energy Consumption in the cluster per round.

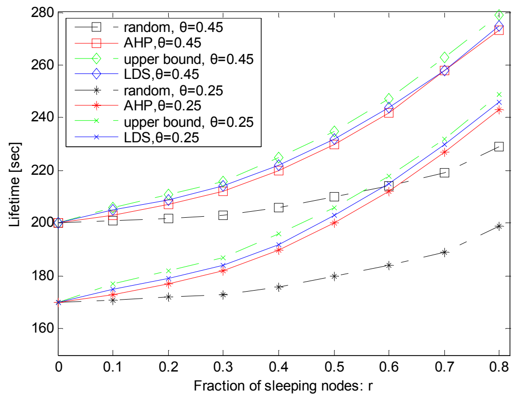


Fig. 5.29 Lifetime comparision.

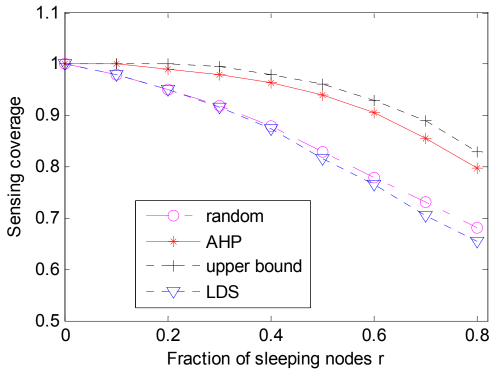


Fig. 5.30 Coverage verses the fraction of sleeping nodes.

Network lifetime can be defined as the time when a fraction of nodes, θ, run out of energy. In Fig. 5.29, I evaluate the lifetime of the three schemes and the upper bound for various values of *r* and θ. The length of each round is 5 seconds. We can see that the lifetime of both schemes is prolonged with the increasing of *r* and the proposed AHP based scheme greatly outperforms the random scheme and is close to LDS. This is in line with the analysis that the proposed scheme can balance the energy consumption among all the member nodes. We also can see that the lifetime of all the schemes increases with an increase of θ. This is because the network can be alive up to the time when θ fraction of nodes are drained of their energy.

Figure 5.30 provides the comparison of coverage ratio verses the fraction of sleeping nodes *r*. The coverage here is defined as the ratio of the union of all sensor nodes' sensing areas to the whole monitored environment. For the detailed explanation of coverage ratio calculation method, please refer to [100, 101]. Fig. 5.30 shows that for the three schemes (AHP, LDS [67] and random) the coverage ratio decreases with the increasing of the fraction of sleeping nodes, *r*. LDS shows similar sensing coverage ratio with random scheme (though different pattern) since the sensing coverage of the LDS scheme in the border area is lower than that in the central area, as sensor nodes close to the border are put into sleep with higher probability. However, in case of the proposed AHP based sleeping scheme, the coverage ratio still can maintain above the desired value of 0.98 when up to 30% nodes are put into sleep. It indicates that the tradeoff in terms of coverage is not very critical using the AHP based scheme. AHP based scheme outperforms the LDS and is close to the upper bound in that the AHP takes overlapping coverage as one of the impact factors while the LDS does not but only energy saving.

To evaluate the integrated sleep scheduling and routing in multi hop networks by AHP, I have used J-Sim [102] as the simulation environment. 60 sensors are randomly deployed in an area of 100m x 100m. The source and destination sensors are set as 2J initially, and 5 couples of source and destination nodes are communicating at the same time in this network. All the other sensors have initial energy of 0-2J. The buffer capacity of each sensor node has been taken as 5 packets with packet length 512 bit and bit rate 9.6kb/sec. The time interval T is set as 10s in our simulation. The source node select $M=3$ nodes in its communication range for the first hop relay. From the second hop, each node along the three paths selects only one node toward its next hop.

We compare our A-SR with Hop-based Sleeping Scheduling (HSS) algorithm [68], upper bound and the geographical multipath routing (GMR) [85] scheme where only distance to the destination is considered. In Fig. 5.31, I plot the simulation time versus the number of nodes dead. It shows that when 50% nodes (30 nodes) die out, the network lifetime for A-SR has been extended more than 40%. A-SR significantly outperforms GMR and has similar performance to HSS. In Fig. 5.32, we compare the packet loss rate of these three schemes. Packets are dropped either due to insufficient buffer capacity at the receiver or because of the lack of energy needed to transmit the packet. Observe that our A-SR outperforms the GMR and HSS with about 20% and 10% less packet loss respectively resulting in greater reliability. The average latency during transmission (end-to-end) is 424.23ms for our A-SR, 407.5ms for GMR and 422.8ms for HSS, and link failure rate for A-SR is 6.51%, but for GMR is 10.42% and for HSS is 10%. Due to the integrated sleep scheduling, in our proposed scheme, the network coverage ratio does not drop below the satisfactory value 0.97 when up to around 30% nodes are put into sleep.

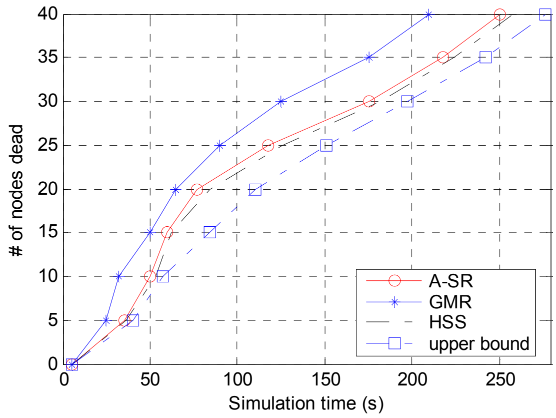


Fig. 5.31 Lifetime comparison.

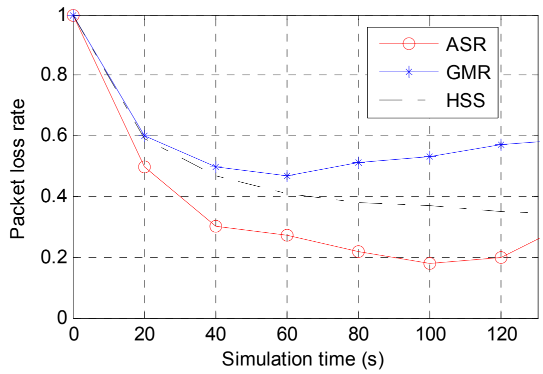


Fig. 5.32 Simulation time vs. packet loss rate.

Chapter 6 Conclusions and future work

In this thesis, the proposed techniques have addressed the problems of effective sensing coverage-driven energy-aware deployment and self-organization in WSNs. The results of this thesis provide an efficient infrastructure for sensor network management.

I first introduced a comprehensive taxonomy for coverage-driven energy-aware WSNs deployment and self-organization. Three sensor relocation algorithms were proposed according to the mobility degree of sensor nodes. The first one, PSOA, regards the sensors in the network as a swarm and reorganizes the sensors by PSO, in the full sensor mobility case. The other two, RSBA and EFOA, assume relatively limited sensor mobility further reduce energy consumption. Simulation results have shown that our approaches greatly improve the network coverage as well as energy efficiency compared with related works. In a hybrid sensor network which consists of a collection of both static nodes and mobile nodes, a novel method for the redeployment of mobile nodes was proposed. An optimal decision of the mobile sensor node moving direction is made based on AHP. Four factors contributing to the optimal deployment were considered which are coverage hole, obstacle avoidance, hot spot, and the boundary effect, respectively. The moving style is flip by flip until the stable status is achieved. Simulation results showed that the proposed approach could provide high coverage with limited movement distance without compromising connectivity. I also proposed a network maintenance strategy in the post-deployment phase based on the sensor node importance level ranking. Simulation results showed that our approach not only achieved fast and stable deployment but also greatly improved the network coverage and as well as extended the lifetime.

In the zero mobility case, i.e. a static sensor network, self-organization technique such as clustering and integrated sleep scheduling and routing were introduced. I have proposed a VQ-LBG algorithm based clustering approach for sensor networks which provides fast data compression process with minimum average distortion at CHs. The resulted uniform cluster distribution also balanced the system load on each CH since all the clusters were balanced, and at the same time, the communication energy consumption was significantly reduced due to the efficient data compression. In order to achieve energy-efficient self-organization, I proposed a sleeping scheduling scheme in a single hop network and an integrated sleep scheduling and routing protocol in a multi hop network based on AHP. In the single hop network, three factors contributing to the optimal nodes scheduling decision were considered and they are 1) distance to CH, 2) residual energy, and 3) sensing coverage overlapping, respectively. In the multi hop network, our proposed A-SR scheme includes distance to destination, residual energy, and sensing coverage overlapping for sleep scheduling, and distance to the destination, residual energy, and queue size of each sensor node for routing. To evaluate the performance, in the single hop network case, we evaluated the efficiency of our proposed scheme in terms of energy consumption, lifetime and coverage ratio, and compared with the upper (lower) bound, LDS and traditional random sleep scheduling scheme in heterogeneous clustered sensor networks. The proposed scheme was observed to improve network lifetime and conserve energy. We also evaluated the efficiency of the proposed scheme in the multi-hop environment and the results showed that it could extend the network lifetime much longer than the original geographical routing scheme which only considered distance to the destination location, and it had similar lifetime performance with HSS. Moreover, the proposed scheme could reduce the packet loss rate and link failure rate since the buffer capacity was considered. In both single hop and multi hop network environment, the sensing coverage capabilities were not compromised.

As a future work, the abstraction of mobility degree will be further studied in which both the physical factors and environmental factors are included, so that the continuous mobility metric can be generated and a generalized deployment scheme can be designed. Moreover, the realistic sensing range model will be derived and used in the network maintenance strategy.

Acknowledgement

I would like to express my gratitude to all those people who have helped make this thesis possible.

I am sincerely grateful to my advisor, Dr. Sungyoung Lee, for his guidance, inspiration, encouragement and support throughout this work. He guided me not only in research, but also in career; he is not only helping, but also generous. He supported me to attend international conferences again and again. Without his continuous and extensive support, this work will never have gone this far. His insistence on perfection and high-quality work have brought out the best in me. He helped me and guided me not only as an advisor, but also like a father.

I would like to express my special thanks to my technical advisors Prof. Jinsung Cho and Prof. Brian J. d'Auriol. Prof. Jinsung Cho not only helped me broaden my vision on the key issues in wireless sensor networks based on his own insights, but also enlightened me with the proper way of transforming preliminary ideas into effective solutions through independent thinking. He guided me to achieve one progress after another with his constructive and insightful suggestions. I can feel my improvement after countless discussion and communication with him. He taught me how to better organize a good paper. I still can remember he helped me on my paper when he was extremely busy with his administrative work in our department. He was also very generous in supporting my paper registration fee. Prof. Brian J. d'Auriol taught me how to view problems in a big picture. With his stimulating and intelligent supervision, I worked out the integration of different mobility assited deployment algorithms into a taxonomy framework with the mobility metric. He helped me how to deal with the reponse to the reviewers' comments. His suggestions were always greatly helpful. He could quickly point out my unclear expressions and explanations in my papers and guide me to work it out in a more explicit and intelligent way. In the thesis organization and revision process, he also took great effort and time helping me improve my thesis and presentation file.

I am grateful to the members of my thesis committee, Prof. Choongseon Hong, Prof. Enki Hong, Prof. Jinsung Cho, Prof. Sungwon Lee, Prof. Brian J d'Auriol and Prof. Sungyoung Lee for their encouragements and for sharing his valuable thoughts with me.

I would especially like to thank my husband Heo Hoon. I could never express enough my gratitude to him for his support, help, encouragement and his guidance to write a good journal paper, without which, I could not work out this thesis. I'm also very grateful to Hui Gao, Keith Blankenship and Yufei Blankenship for their serious revision and proofreading of my JIST journal paper.

I want to express my deepest gratitude and appreciation to my parents, who have always being there with me to share my pains and joys. I gratefully dedicate this thesis to them for their love, care, and support.

Finally, I would like to thank my lab collegues (Lei Shu, Jie Yang, Hui Xu, Min Meng, Yu Niu, Jin Wang, etc.) and friends (M. Julius Hossain, M. Mamun-Or-Rashid, etc.) for their cooperations and help.

Financial support received from the IITA (Institute of Information Technology Advancement), Korea, is gratefully acknowledged.

2008. 6. Xiaoling Wu

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A survey on sensor networks," IEEE Communications Magazine, pp. 102–114, 2002.

[2] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, “Next century challenges: scalable coordination in sensor networks,” Proc. IEEE/ACM MobiCom Conference, pp. 263–270, 1999.

[3] D. Estrin, L. Girod, G. Pottie and M. Srivastava, “Instrumenting the world with wireless sensor networks,” Proc. International Conference on Acoustics, Speech, and Signal Processing , vol. 4, pp. 2033–2036, 2001.

[4] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Piste, “System architecture directions for network sensors,” Proc. International Conference on Architectural Support for Programming Languages and Operating Systems, 2000.

[5] G. J. Pottie and W. J. Kaiser, “Wireless sensor networks”, Communications of the ACM, vol. 43, pp. 51–58, May 2000.

[6] H. Qi, S. S. Iyengar, K. Chakrabarty, “Distributed sensor networks- a review of recent research,” Journal of the Franklin Institute, vol. 338, pp. 655–668, 2001.

[7] L. Schwiebert, S.D.S. Gupta and J. Weinmann,“ Research challenges in wireless networks of biomedical sensors,” Proc. IEEE/ACM MobiCom, pp. 151–165, 2001.

[8] S. Tilak, N. B. Abu-Ghazaleh and W. B. Heinzelman, “A taxonomy of wireless micro-Sensor network models,” ACM Mobile Computing and Communications Review, vol. 6, no. 2, 2002.

[9] J. R. Agre and L. P. Clare, “An integrated architecture for cooperative sensing networks,” IEEE Computer Magazine, vol. 33, no. 5, pp. 106–108, 2000.

[10] Tinyos, University of California at Berkeley, http://www.tinyos.net, page accessed on Mar. 13, 2008.

[11] Polaroid Sonar, “Technical specifications for 6500 series sonar ranging module,” http://www-inst.eecs.berkeley.edu/_ee40/calbot/sensor datasheets/6500series.pdf, page accessed on Mar. 13, 2008.

[12] Emergent Surveillance Plexus (ESP): A Multidisciplinary University Research Initiative (MURI), http://strange.arl.psu.edu/ESP, page accessed on Nov. 13, 2008.

[13] W. B. Heinzelman, J. Kulik and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” Proc. IEEE/ACM MobiCom Conference, pp. 174–185, 1999.

[14] W. B. Heinzelman, A. Chandrakasan and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, vol. 1, pp. 660–670, 2002

[15] N. Heo and P. K. Varshney, “A Distributed self spreading algorithm for mobile wireless sensor networks,” Proc. IEEE Wireless Communications and Networking Conference, paper ID: TS48–4, 2003.

[16] Y. Zou and K. Chakrabarty, “Energy-aware target localization in wireless sensor networks,” Proc. IEEE Interantional Conference on Pervasive Computing and Communications, pp. 60–67, 2003.

[17] Y. Zou and K. Chakrabarty, “Target localization based on energy considerations in distributed sensor networks,” Ad Hoc Networks, vol. 1, pp. 261–272, 2003.

[18] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, Sungyoung Lee, and Yong-koo Lee, "An integrated sleep scheduling and routing in USN,"IEICE Trans. on Communications, 2007.

[19] R. Min, M. Bhardwaj, S. H. Cho, A. Sinha, E. Shih, A. Wang and A. Chandrakasan, “Low-power wireless sensor networks,” VLSI Design, 2001.

[20] C. Srisathapornphat, C. Jaikaeo, and C.-C. Shen, “Sensor information networking architecture,” International Workshops on Parallel Processing, pp. 23–30, 2000.

[21] K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho, “Grid coverage for surveillance and target Location in distributed sensor networks,” IEEE Transactions on Computers, vol. 51, pp. 1448–1453, 2002.

[22] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization in distributed sensor networks,” ACM Transactions on Embedded Computing Systems, vol. 3, pp. 61–91, 2004.

[23] Y. Zou and K. Chakrabarty, “Uncertainty-aware and coverage-oriented deployment for sensor networks,” Journal of Parallel and Distributed Computing, vol. 64–7, pp. 788-798, 2004.

[24] Xiaoling Wu, Niu Yu, Shu Lei, Jinsung Cho, Young-Koo Lee, Sungyoung Lee, "Relay Shift Based Self-Deployment for Mobility Limited Sensor Networks", The 3rd International Conference on Ubiquitous Intelligence and Computing (LNCS), Wuhan and Three Gorges, China, 2006

[25] Xiaoling Wu, Shu Lei, Wang Jin, Jinsung Cho, Sungyoung Lee, "Energy-efficient Deployment of Mobile Sensor Networks by PSO", IWSN 2006, LNCS, Harbin, China

[26] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, and Sungyoung Lee, "Optimal Deployment of Mobile Sensor Networks and Its Maintenance Strategy," Proc. of International Conference on Grid and Pervasive Computing (GPC 2007, LNCS), Paris, France, May, 2007.

[27] Xiaoling Wu, Lei Shu, Min Meng, Jinsung Cho, and Sungyoung Lee, "Coverage-driven Self-deployment for Cluster Based Mobile Sensor Networks", CIT-06, IEEE Computer Society, Seoul, Korea

[28] W. B. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless micro sensor networks," Proc. 33rd Annual Hawaii International Conference on System Sciences, pp. 3005–3014, 2000.

[29] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," Proc. IEEE Aerospace Conference, vol. 3, pp. 1125–1130, 2002.

[30] J. Liu, P. Cheung, L. Guibas and F. Zhao, "A dual-space approach to tracking and sensor management in wireless sensor networks," Proc. 1st ACM International Workshop on Wireless Sensor Networks and Applications, pp. 131–139, September 2002.

[31] G. T. Sibley, M. H. Rahimi and G. S. Sukhatme. Robomote: A Tiny Mobile Robot Platform for Large-Scale Sensor Networks. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2002.

[32] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, Sungyoung Lee and Hee Yong Youn , "Self-Deployment of Mobile Nodes in Hybrid Sensor Networks by AHP," Proc. of International

Conference on Ubiquitous Intelligence and Computing (UIC 2007, LNCS), Hong Kong, China, July 11-13, 2007.

[33] R. R. Brooks, P. Ramanathan and A. A. Sayeed, "Distributed target classification and tracking in sensor networks," Proceedings of the IEEE, vol. 91, pp. 1163–1171, 2003.

[34] H. Qi, Y. Xu and X. Wang, "Mobile-agent-based collaborative signal and information processing in sensor networks", Proceedings of the IEEE, vol. 91, pp. 1172–1183, August 2003.

[35] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, and Sungyoung Lee, "Sleep Nodes Scheduling in Cluster-based Heterogeneous Sensor Networks Using AHP," Proc. of International Conference on Embedded Software and System (ICESS 2007, LNCS), Daegu, Korea, May, 2007.

[36] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, and Sungyoung Lee, "Energy-aware Routing for Wireless Sensor Networks by AHP," Proc. of IFIP Workshop on Software Technologies for Future Embedded; Ubiquitous Systems (SEUS 2007, LNCS), Santorini Island, Greece, May, 2007.

[37] Sameer Tilak, Nael B. AbuGhazaleh, and Wendi Heinzelman, "Infrastructure tradeoffs for sensor networks", Proc. 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, Georgia, USA, pp. 49 – 58, 2002.

[38] A. Howard, M. J. Mataric and G. S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks", Autonomous Robots, Special Issue on Intelligent Embedded Systems, Vol. 13, Issue 2, pp. 113 – 126, Sept. 2002.

[39] S. Meguerdichian , F. Koushanfar, G. Qu and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks", Proc. of International Conference on Mobile Computing and Networking (Mobicom), Rome, Italy, pp. 139 – 150, 2001.

[40] S. Dhillon, K. Chakrabarty and S. Iyengar, "Sensor placement for grid coverage under imprecise detections", Proc. International Conference on Information Fusion, Annapolis, USA, Vol. 2, pp. 1581- 1587, Jul. 2002

[41] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and Kewal K. Saluja, "Sensor deployment strategy for detection of targets traversing a region", ACM Mobile Networks and Applications, Vol. 8, Issue 4, pp. 453 – 461, Aug. 2003.

[42] T. L. Saaty, Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process, RWS Publications, U.S.A., 2000

[43] Q. Y. Song and A. Jamalipour: A Network Selection Mechanism for Next Generation Networks. IEEE Int. Conf. Communication (ICC), Vol. 2 (2005) 1418-1422

[44] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: coverage, connectivity and diameter", Proc. IEEE Conference on Computer Communications (INFOCOM), San Francisco, Vol. 2, pp. 1073 – 1083, Mar./Apr. 2003.

[45] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks", Proc. IEEE Conference on Computer Communications (INFOCOM), Anchorage, Alaska, USA, pp. 1380 – 1387, Apr. 2001.

[46] Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava, "Worst and best-case coverage in sensor networks", IEEE transactions on mobile computing, Vol. 4, No. 1, pp. 84- 92, Jan. /Feb. 2005.

[47] B. Liu and D. Towsley, "A study on the coverage of large-scale sensor networks", Proc. 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04), Fort Lauderdale, Florida, USA, pp. 475 – 483, Oct. 2004.

[48] M. Cardei and J. Wu, "Coverage in wireless sensor networks", in Handbook of Sensor Networks, M.Ilyas and I. Mahgoub (eds.), CRC Press, ISBN: 0-8493-1968-4, 2004.

[49] J. Wu and S. Wang, "Smart: A scan-based movement-assisted deployment method in wireless sensor networks", Proc. IEEE Conference on Computer Communications (INFOCOM), Miami, Vol. 4, pp. 2313 – 2324, Mar. 2005.

[50] G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment", IEEE Transactions on Mobile Computing, Vol. 5, Issue 6, pp. 640 – 652, June 2006.

[51] Nojeong Heo and Pramod K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks", IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans, Vol. 35, No. 1, pp. 78 – 92, 2005.

[52] B. Liu, P. Brass, and O. Dousse, "Mobility improves coverage of sensor networks", Proc. 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Urbana-Champaign, Illinois, USA, pp. 300 – 308, May 2005.

[53] Guiling Wang, Guohong Cao, Tom La Porta, “Proxy-based sensor deployment for mobile sensor networks”, Proc. 1st IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS’04), Fort Lauderdale, Florida, USA, pp. 493 – 502, Oct. 2004.

[54] A. Howard, M. Mataric, and G. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem”, Proc. of DARS'02, Fukuoka, Japan, pp. 299 – 308, June 2002.

[55] Damien B. Jourdan, Olivier L. de Weck, “Layout optimization for a wireless sensor network using a multi-objective genetic algorithm”, Proc. IEEE 59th Vehicular Technology Conference (VTC 2004-Spring), Milan, Italy, Vol.5, pp. 2466 – 2470, May 2004.

[56] Haining Shu, Qilian Liang, “Fuzzy optimization for distributed sensor deployment”, Proc. of IEEE Wireless Communications and Networking Conference, New Orleans, USA, pp. 1903 – 1907, 2005.

[57] Defense Advanced Research Projects Agency (DARPA), “http://www.darpa.mil/ato/programs/shm/index.html”.

[58] Sriram Chellappan, Xiaole Bai, Bin Ma, and Dong Xuan, “Sensor networks deployment using flip-based sensors”, Proc. of IEEE International Conference MASS, Washington, DC, pp. 291 – 298, Nov. 2005.

[59] Ahmed N., Kanhere S. and Jha S.: The Holes Problem in Wireless Sensor Networks: A survey. Mobile Computing and Communications Review, 9(2) (2005) 4-18

[60] Maxim A. Batalin and Gaurav S. Sukhtame: Coverage, Exploration and Deployment by a Mobile Robot and Communication Network. Telecommunication Systems Journal, Special Issue on Wireless Sensor Networks, 26 (2) (2004)181–196

[61] Guiling Wang, Guohong Cao, Tom La Porta, and Piotr Berman: Bidding Protocols for Sensor Deployment. IEEE Trans. on Mobile Computing, Vol. 2, No. 5 (2007) 563-576

[62] Amitabha Ghosh: Estimating Coverage Holes and Enhancing Coverage in Mixed Sensor Networks. 29th Annual IEEE International Conference on Local Computer Networks (LCN'04), Tampa, Florida, U.S.A (2004) 68-76

[63] Luo, R.C., Tu, L.C., Chen, O.: Auto-deployment of Mobile Nodes in Wireless Sensor Networks Using Grid Method. IEEE International Conference on Industrial Technology (ICIT'05), Hong Kong, China (2005) 359- 364

[64] D. Tian and N. D. Georganas: A coverage-preserving node scheduling scheme for large wireless sensor networks. Proc of the 1st ACM international workshop on Wireless sensor networks and applications (2002) 32 – 41.

[65] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris: Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. ACM Wireless Networks, Vol. 8, No. 5 (2002).

[66] W. Ye, J. Heidemann, and D. Estrin: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. Proc of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), New York, NY (2002).

[67] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney: Scheduling Sleeping Nodes in High Density Cluster - based Sensor Networks. ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Energy Constraints and Lifetime Performance in Wireless Sensor Networks (2005).

[68] Yun Wang, Demin Wang, Weihuang Fu, and Dharma P. Agrawal: Hops-based Sleep Scheduling Algorithm for Enhancing Lifetime of Wireless Sensor Networks. Proc of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS), Canada (2006) 709-714

[69] Rick W. Ha, Pin-Han Ho and X. Sherman Shen: SS-trees: a cross-layer organizational approach for mesh-based wide-area wireless sensor networks. Proc. of the 2nd International Conference on Broadband Networks, Vol. 2, Boston, MA, USA (2005) 823- 832

[70] J. Pan, L. Cai, Y. T. Hou, Y. Shi, and S. X. Shen, "Optimal base-station locations in two-tiered wireless sensor networks", IEEE Transactions on Mobile Computing (TMC), 4(5), 2005, pp. 458–473.

[71] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks", In Proceedings of the IEEE Conference on Computer Communications. INFOCOM, 2003.

[72] O. Younis, et. al., "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks", IEEE Transactions on Mobile Computing, 3(4), 2004, pp. 660-669.

[73] H. Chan, et. al., “ACE: An Emergent Algorithm for Highly Uniform Cluster Formation”, the First European Workshop on Sensor Networks (EWSN), 2004.

[74] J. Kamimura, et. al., “Energy-Efficient Clustering Method for Data Gathering in Sensor Networks”, 1st Annual International Conference on Broadband Networks, 2004.

[75] Xiaoling Wu, Mohammed A. U. Khan, Jinsung Cho, Sungyoung Lee and Young-Koo Lee, "Energy-efficient Clustering with Fast Data Compression in Sensor Networks", International Conference on Hybrid Information Technology (ICHIT'06), Cheju Island, Korea, Nov 9-11, 2006, ISBN 0-7695-2674-8, IEEE Computer Society, pp. 403-408

[76] C. P. Bhagwat: Highly dynamic destination-sequenced distance vector routing. Proc. of ACM SIGCOMM (1994) 234-244

[77] Ching-Chuan Chiang, Hsiao-Kuang Wu, Winston Liu, Mario Gerla: Routing in clustered multihop mobile wireless networks with fading channel. Proc. IEEE Singapore Intl Conference on Networks (1997)

[78] C. E. Perkins and E. Royer: Ad hoc on demand distance vector routing. Proc. 2nd IEEE Workshop o Mobile Computing Systems and Applications (1999)

[79] D. Johnson and D. Maltz: Mobile Computing. Kluwer Academic Publishers (1996)

[80] S.J. Lee, and M. Gerla: Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. ICC (2001)

[81] L. Wang, Y. T. Shu, M. Dong, L.F. Zhang, and W.W. Yang: Multipath Source Routing in wireless Ad Hoc Networks. Canadian Conference on Electrical and Computer Engineering, vol. 1 (2000) 479-483

[82] A. Nasipuri, and S.R. Das: On-Demand Multipath Routing for Mobile Ad Hoc Networks. IEEE ICCCN (1999) 64-70

[83] G. G. Finn: Routing and addressing problems in large metropolitanscale internetworks. USC ISI Report ISI/RR-87-180 (1987)

[84] G. Xing, C. Lu, R. Pless, and Q. Huang: On Greedy Geographic Routing Algorithms in Sensing-Covered Networks. ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Tokyo, Japan (2004)

[85] R. Jain, A. Puri, and R. Sengupta: Geographical routing using partial information for wireless sensor networks. IEEE Personal Communications (2001) 48-57

[86] Kennedy J. and R. C. Eberhart, "Particle swarm optimization", Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942 – 1948, 1995.

[87] Ioannis N. Kassabalidis, Mohamed A. El-Sharkawi, Robert J. Marks, II, Luciano S. Moulin, and Alexandre P. Alves da Silva, "Dynamic security border identification using enhanced particle swarm optimization", IEEE Transactions on Power Systems, Vol. 17, No. 3, pp. 723 – 729, Aug. 2002.

[88] Ganesh K. Venayagamoorthy and Sheetal Doctor, "Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller", Proc. 39th IEEE IAS Annual Meeting on Industry Applications, Seattle, WA, USA, pp. 1200 – 1206, Oct. 2004.

[89] B. A. Kadrovach, G. B. Lamont, "A particle swarm model for swarm-based networked sensor systems", Proc. ACM symposium on Applied computing, Madrid, Spain, pp. 918 – 924, 2002.

[90] T. Hardin, X. Cui, R. K. Ragade, J. H. Graham, and A. S. Elmaghraby, "A modified particle swarm algorithm for robotic mapping of hazardous environments", Proc. the World Automation Congress, Spain, pp. 31 – 36, 2004.

[91] Yuhui Shi, Russell C. Eberhart, "Empirical study of particle swarm optimization", Proc. the 1999 Congress on Evolutionary Computation, Washington DC, Vol. 3, pp. 1948 – 1950, 1999.

[92] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to algorithms", 2nd Edition. MIT Press and McGraw-Hill, 2001.

[93] Indranil Gupta, Denis Riordan and Srinivas Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks", Proc. 3rd Annual Communication Networks and Services Research Conf., Halifax, Canada, pp. 255 – 260, May 2005.

[94] The MathWorks, Inc.，Fuzzy Logic Toolbox 2.2.3, http://www.mathworks.com/products/fuzzylogic/", 2008.

[95] Sriram Chellappan, Xiaole Bai, Bin Ma, and Dong Xuan: Sensor Networks Deployment Using Flip-based Sensors. Proc. of IEEE International Conference MASS'05 (2005)

[96] S.Bandyopadhyay, E.J.Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors", Computer Networks, vol. 44, No. 1, 2004, pp. 1–16.

[97] Georgios Smaragdkis, Ibrahim Matta and Azer Bestavros, "SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks", Proc. of the Int'l Workshop on SANPA, 2004.

[98] http://www.data-compression.com/vq.shtml, accessed in Apr. 2008.

[99] Y. Xu, J.Heidemann and D. Estrin, Adaptive energy-conserving routing for multihop ad hoc networks, Research report 527, USC/Information Sciences Institute (2000).

[100] Xiaoling Wu, Shu Lei, Yang Jie, Xu Hui, Jinsung Cho and Sungyoung Lee: Swarm Based Sensor Deployment Optimization in Ad hoc Sensor Networks. Proc. of ICESS’ 05/ LNCS, Xi’an, China, (2005) 533-541

[101] Xiaoling Wu, Jinsung Cho, Brian J. d'Auriol, and Sungyoung Lee, "Mobility-Assisted Relocation for Self-deployment in Wireless Sensor Networks," IEICE Trans. On ommunications, 2007.

[102] Ohio State University, and University of Illinois at Urbana-Champaign, J-Sim. http://www.j-sim.org/. Last updated 01/28/2005