IEICE TRANS. COMMUN., VOL.E90–B, NO.8 AUGUST 2007

PAPER

Mobility-Assisted Relocation for Self-Deployment in Wireless Sensor Networks*

Xiaoling WU[†a)], ***Nonmember***, **Jinsung CHO**[†], ***Member***, **Brian J. D'AURIOL**[†], ***Nonmember***, ***and*** **Sungyoung LEE**[†], ***Member***

SUMMARY Sensor network deployment is very challenging due to the hostile and unpredictable nature of environments. The field coverage of wireless sensor networks (WSNs) can be enhanced and consequently network lifetime can be prolonged by optimizing the sensor deployment with a finite number of mobile sensors. In this paper, we introduce a comprehensive taxonomy for WSN self-deployment in which 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. In the zero mobility case, static topology control or scheduling schemes can be used such as optimal cluster formation. Simulation results show that our approaches greatly improve the network coverage as well as energy efficiency compared with related works.

key words: *deployment, sensor networks, mobility, coverage, energy-efficiency*

1. Introduction

Sensor networks which are composed of tiny and resource constrained computing devices, have been widely employed for monitoring and controlling applications in physical environments [1]. 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 [5]. Moreover, there are situations where sensor deployment is restricted by the environment, for example, during in-building toxic-leaks detection [6] 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 [2]–[4] which assume that the environment is sufficiently known and under control.

In this paper, we 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 of this paper include the comprehensive collection of algorithms for mobile sensor network self-deployment within the context of a generally applicable taxonomy.

The rest of the paper is organized as follows. Section 2 introduces related work and a comprehensive taxonomy framework integrating the different deployment schemes. Section 3 thoroughly explains the proposed three sensor relocation algorithms. In Sect. 4, extensive experiments and performance evaluations of the proposed method are presented. We conclude with a summary and discuss future work in Sect. 5.

2. Related Work and Taxonomy

In this section, we present a brief overview of the previous work on the coverage driven deployment of both stationary and mobile sensor networks that is most relevant to our study. A more thorough survey of the sensor network coverage is provided by [7].

We introduce a taxonomy framework (Fig. 1) for WSN self-deployment in this section. We 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. 1. The three relocation

Manuscript received August 28, 2006.
Manuscript revised February 2, 2007.

[†]The authors are with the Department of Computer Engineering, Kyung Hee University, Korea.
[*]Prof. Jinsung Cho is the corresponding author.
a) E-mail: xiaoling@oslab.khu.ac.kr

DOI: 10.1093/ietcom/e90–b.8.2056

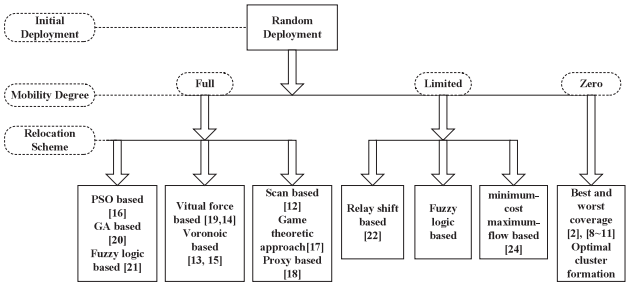


Fig. 1 Taxonomy framework integrating different deployment schemes.

and scheduling cases for sensor network self-deployment corresponding to the three categories of sensor nodes mobility degree are extensively studied and integrated.

In WSNs with zero mobility, i.e., stationary sensor networks, there are many previous studies which have focused on characterizing coverage. The authors of [8] 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 [9], 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 [2], 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 [10]. 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 [11], 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.

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 [12]–[21]. Typically in such works, the sensors detect lack of desired deployment objectives, then estimate new locations, and move to the resulting locations. In [14], [19], 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 [15]. 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 [13], 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 [18], 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 [12], 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 [17] 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 [20], 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 [21], 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, as we have earlier discussed in [22]. To this extent, a class of Intelligent Mobile Land Mine Units (IMLM) [23] 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 [24]. Moreover, in many applications, the latter goals outweigh the necessity for advanced mobility, making such mobility models quite practical in the future. In fact, [24] is one of the very few papers which deal with the mobility lim-

ited 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, we propose three different relocation algorithms, PSOA, RSBA and EFOA, in the cases of full sensor mobility and limited sensor mobility. We 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 section, we will describe our proposed sensor relocation approaches in detail.

3. Proposed Relocation Schemes

We 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.

3.1 Relocation in Full Mobility Environment: PSOA

In the full sensor mobility case, we propose particle swarm optimization (PSO) based algorithm for movement assisted relocation. PSO, originally proposed by Eberhart and Kennedy [25] 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 [26]–[29].

All of 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 $X_i = (x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ where i represents the particle number. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \ldots, p_{iD})$, 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 $V_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{iD})$. Let $d \in [1, 2, \ldots, D]$, the updating equation [30] is as follows,

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

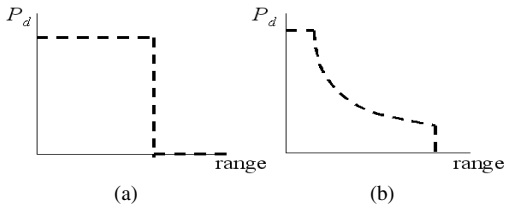


Fig. 2 Sensor coverage models (a) Binary sensor and (b) probabilistic sensor model.

$$x_{id} = x_{id} + v_{id} \quad (2)$$

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. Generally, 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.

We 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.

3.1.1 Optimization of Coverage

We consider coverage as the first optimization objective. It is one of the measurement criteria of Quality of Service (QoS) of a sensor network.

The coverage of each sensor can be defined either by a binary sensor model or a probabilistic sensor model as shown in Fig. 2; both are used in this paper. 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. (3) [15]. Generally

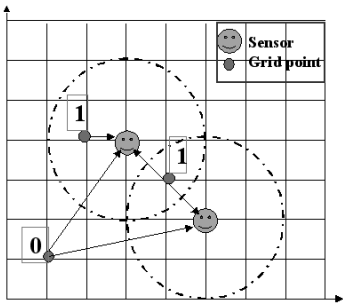


Fig. 3 Sensing coverage calculation (dashed circle indicating the sensing area boundary).

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.

$$C = \frac{\cup_{i=1,\ldots,N} A_i}{A} \tag{3}$$

where

A_i is the area covered by the ith 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 [16]. 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. 3. Then the coverage can be approximated by the ratio of the summation of ones to the total number of the grid points.

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 [14] 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.

$$c_{ij}(x,y) = \begin{cases} 0 & \text{if} \quad r + r_e \le d_{ij}(x,y); \\ e^{-\lambda a^\beta}, & \text{if} \quad r - r_e < d_{ij}(x,y) < r + r_e; \\ 1 & \text{if} \quad r - r_e \ge d_{ij}(x,y). \end{cases} \tag{4}$$

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), we denote the Euclidean distance between s at (x, y) and P at (i, j) as $d_{ij}(x, y)$, i.e., $d_{ij}(x, y) = \sqrt{(x-i)^2 + (y-j)^2}$. Equation (4) 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(b). Figure 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}.

3.1.2 Optimization of Energy Consumption

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.

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 [31]:

$$E_T(l,d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & \text{if } d \le d_0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & \text{if } d > d_0 \end{cases} \tag{5}$$

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 we set electronics energy 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:

$$E_R(l) = lE_{elec} \tag{6}$$

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

$$E_{TS}(l, dis) = lE_{elec} + l\varepsilon_{fs}dis^2 \quad (7)$$

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

$$E_{TH}(l, Dis) = lE_{elec} + l\varepsilon_{mp}Dis^4 \quad (8)$$

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

$$E_s(l, dis) = l(100 + 0.01dis^2) \quad (9)$$

The energy loss of a CH is

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

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

Assume m clusters with n_j sensor members in the jth 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:

$$E_{total} = l\sum_{j=1}^{m}\sum_{i=1}^{n_j}\left(100 + 0.01dis_{ij}^2 + \frac{100}{n_j} + \frac{1.3 \times 10^{-6}Dis_j^4}{n_j}\right) \quad (11)$$

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

$$f = \sum_{j=1}^{m}\sum_{i=1}^{n_j}\left(0.01dis_{ij}^2 + \frac{1.3 \times 10^{-6}Dis_j^4}{n_j}\right) \quad (12)$$

From Eq. (12) 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. We use the PSO algorithm to find the optimal CH positions in the sensor field when the minimized energy consumption is achieved.

3.2 Relocation in Limited Mobility Environment

3.2.1 Relay Shift Based Algorithm (RSBA)

Let $G(V, E)$ be the graph defined on V with edges $uv \in$E if $uv \leq R$. Here uv is the Euclidean distance between nodes u and v, R is the communication range. We assume that sensor nodes know their locations using one of the GPS-less localization techniques mentioned in [13] such as received signal strength so that CH can get the position information of its sensor members.

We have 4 steps for implementing RSBA:

Step 1: Randomly deploy nodes in the network.
Step 2: Detect coverage holes and redundant sensor nodes. We set two distance threshold value T_1 and T_2. If the longest linear distance between two nodes A and B along the uncovered area perimeter is larger than T_1, regard it as a coverage hole, and create a virtual node point at the center of the straight line AB. If the distance between two neighbors is less than T_2, regard them as redundant nodes. Choose a redundant node nearest to the virtual node point in coverage hole.

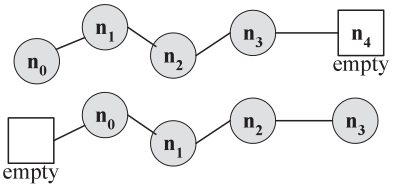


Fig. 4 Illustration of sensor nodes relay shift along the shortest path.

Step 3: Use A* algorithm [32] to find a shortest path $n_0-n_1-n_2-\ldots-n_{n-1}$ 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 [22] 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. (13):

$$NetLoc(n_i) = NetLoc(n_{i+1}), \quad i = 0, 1, \cdots, n-2 \quad (13)$$

$n_i \in$ nodes on shortest path from source to destination
n_0 = source node
n_{n-1} = destination (virtual node)

The process is illustrated in Fig. 4 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. (3).

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. We have used the most commonly used fuzzy inference technique called Mamdani Method [33] 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.

B. Energy-efficient Fuzzy Optimization Algorithm

The same energy dissipation model as Eq. (5) and Eq. (6) 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 Sect. 3.1.2.

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.

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 we use $3^3 = 27$ rules for the fuzzy rule base. We 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 1 and Figs. 5 through 8.

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

$$Cen = \left(\sum \mu_A(x) * x\right) \Big/ \sum \mu_A(x) \tag{14}$$

Table 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

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. 9 and Fig. 10 obtained by computation in Matlab Fuzzy Logic Toolbox [34], 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.

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.

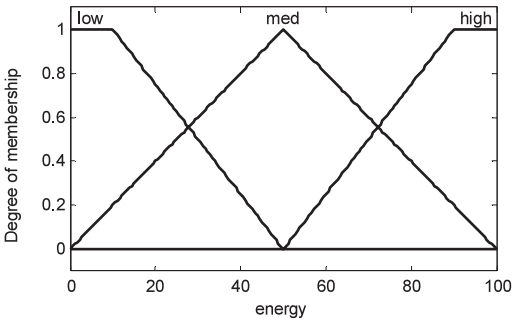


Fig. 5 Fuzzy set for fuzzy variable *energy*.

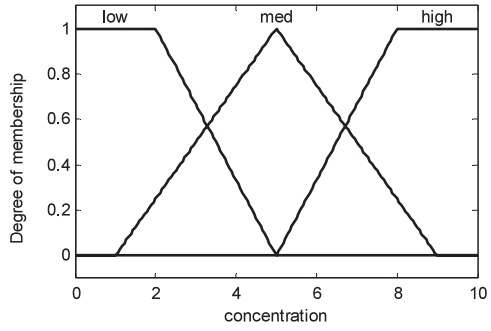


Fig. 6 Fuzzy set for fuzzy variable *concentration*.

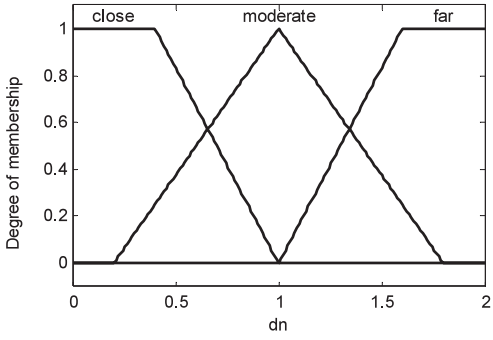


Fig. 7 Fuzzy set for fuzzy variable d_n.

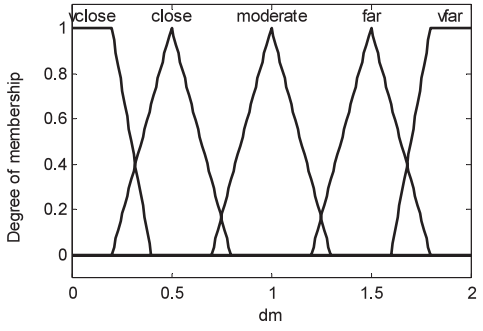


Fig. 8 Fuzzy set for fuzzy variable d_m.

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 Eqs. (15) and (16), thus sensor i clearly knows its next-step moving position by

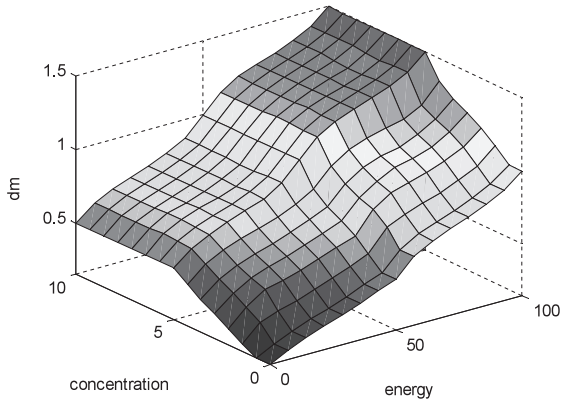


Fig. 9 Control surface (concentration, energy vs d_m).

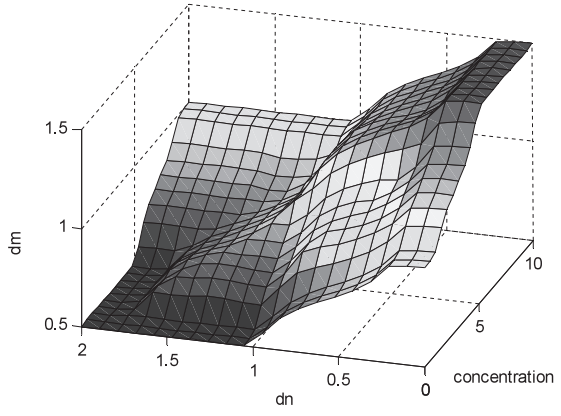


Fig. 10 Control surface (d_n, concentration vs d_m).

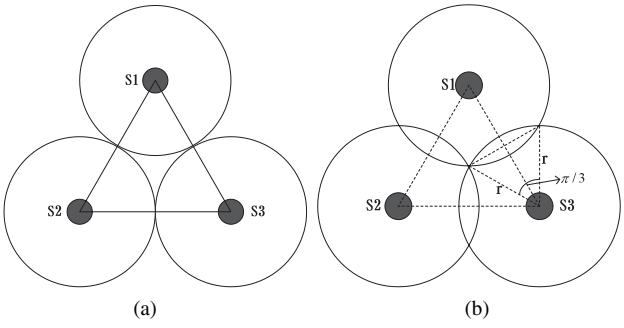


Fig. 11 Non-overlapped and overlapped sensor coverage cases.

getting distance d_m and direction (angle α).

$$\vec{v} = \frac{1}{\left|\vec{C}_i - \vec{C}_j\right|^2}\left(\sum_{j=1}^{k_1}\left(\vec{C}_i - \vec{C}_j\right) + \sum_{j=1}^{k_2}\left(\vec{C}_j - \vec{C}_i\right)\right) \tag{15}$$

$$\tan(\alpha) = \frac{Y(\vec{v})}{X(\vec{v})} \tag{16}$$

The threshold distance d_{th} here is set to a proper value $\sqrt{3}r$ which is proved as follows. We 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. 11. Non-overlapped sensor coverage style

is shown in Fig. 11(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. 11(b) and it ensures that all grid points are covered. Therefore, we adopt the second strategy.

In Fig. 11(b), it is obvious that $\Delta S_1S_2S_3$ 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

$$S_1S_2 = S_2S_3 = S_1S_3 = 2 \times \frac{\sqrt{3}}{2} r = \sqrt{3}r.$$

4. Performance Evaluations

4.1 Performance Evaluation of PSOA

4.1.1 Optimization of Coverage

A. Binary Model Case

The PSO starts with a "swarm" of sensors randomly generated. As shown in Fig. 12 is a randomly deployed sensor network with coverage value 0.4484 calculated using Eq. (3). A linear decreasing inertia weight value from 0.95 to 0.4 is used, decided according to [30]. Acceleration coefficients c_1 and c_2 both are set to 2 as proposed in [30]. For this performance study, we select a large scale deployment of 50×50 square sensor network. 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.

Figure 13 is the coverage optimization results. The coverage improvement verses number of iterations in one run is shown in Fig. 13(a) and the final achieved coverage values for six runs are shown in Fig. 13(b). Compared with the upper bound 0.628, the difference between the average value 0.58 for six runs and upper bound is small.

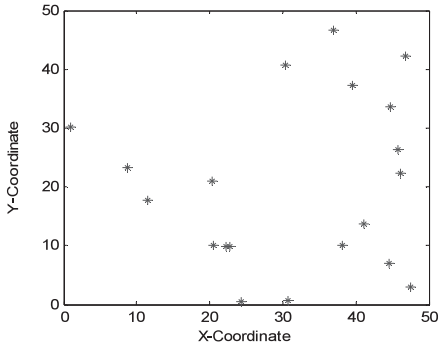
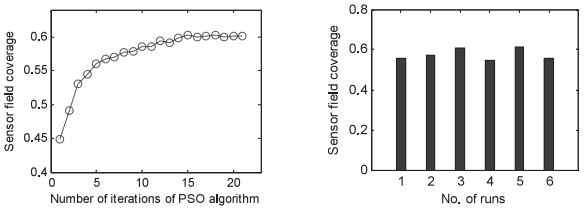


Fig. 12 Randomly deployed sensor network (coverage value=0.4484).

B. Probabilistic Model Case

In probabilistic model case, we use a randomly deployed sensor network as shown in Fig. 14, with coverage value 0.31 calculated by Eq. (4) and approximate method mentioned in Sect. 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 $r = 5$, $r_e = 3$, $\lambda = 0.5, \beta = 0.5$, $c_{th} = 0.7$.

Figure 15(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



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

Fig. 13 Optimal coverage results (binary sensing model).

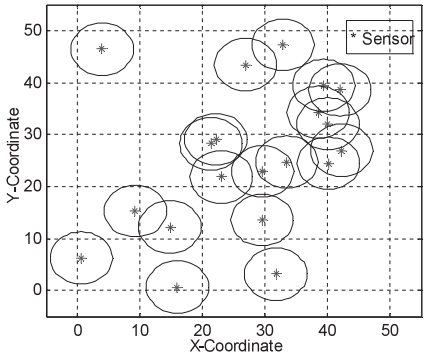
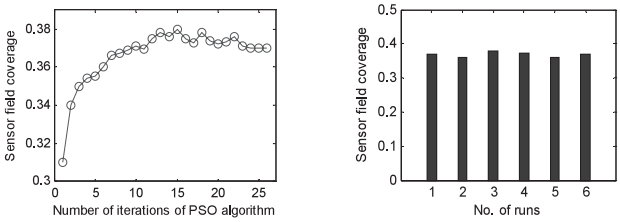


Fig. 14 Randomly deployed sensor network with r=5 (coverage value=0.31).



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

Fig. 15 Optimal coverage achieved using PSO algorithm (probabilistic sensing model).

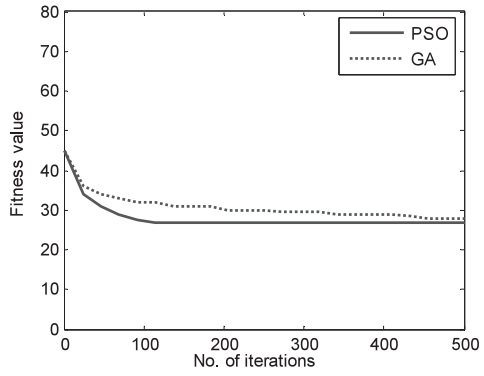


Fig. 16 Comparison of convergence rate between PSO and GA based on Eq. (12).

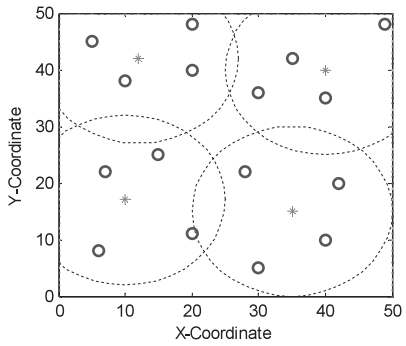


Fig. 17 Final cluster formation by PSO (binary model case).

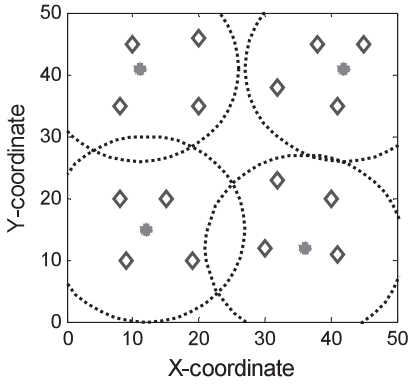


Fig. 18 Final cluster formation by PSO (probabilistic model case).

covered, the overall number of grid points with coverage probability greater than the required level is fewer. We also show the achieved coverage for six runs in Fig. 15(b), and the average is nearly 0.37 which has little difference from the upper bound.

4.1.2 Optimization of Energy Consumption

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). We 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.

We also optimized the placement of CH in the 2-*D* space using GA. We 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 we use 10 chromosomes in the population. The maximum number of generations allowed is 500. In each evolution we 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.

Figure 16 shows the convergence rate of PSO and GA. We ran the algorithm for both approaches six times and in every run PSO converges faster than GA which was used in [20] for coverage and lifetime optimization. The main reason for the fast convergence of PSO is due to the velocity factor of the particle.

Figure 17 and Fig. 18 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 Figs. 12 and 13. The four stars denote CHs, the 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.

4.2 Performance Evaluation of RSBA

The performance of the proposed movement assisted algorithm RSBA is evaluated by simulation. For the convenience of comparison with related work, we set the initial parameters the same as in [15]: 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. 19, the coverage and connectivity of the initial random deployment before running the algorithms are shown. The 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.

Figure 20 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

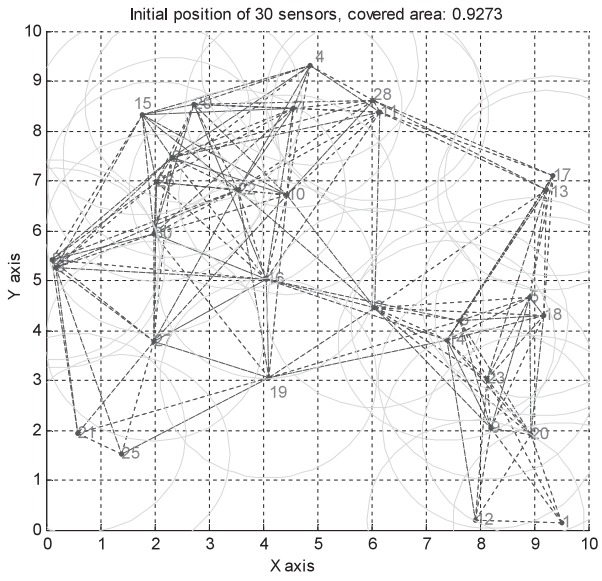


Fig. 19 Initial random deployment with sensing range 2 m and communication range 4 m.

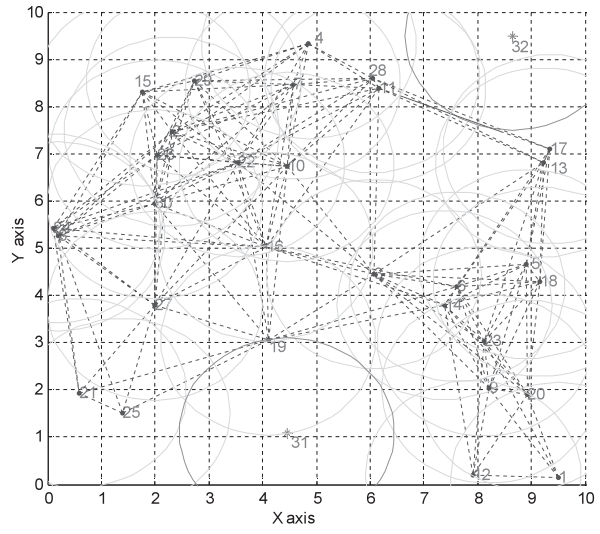


Fig. 20 Determine virtual node point in uncovered area and redundant nodes.

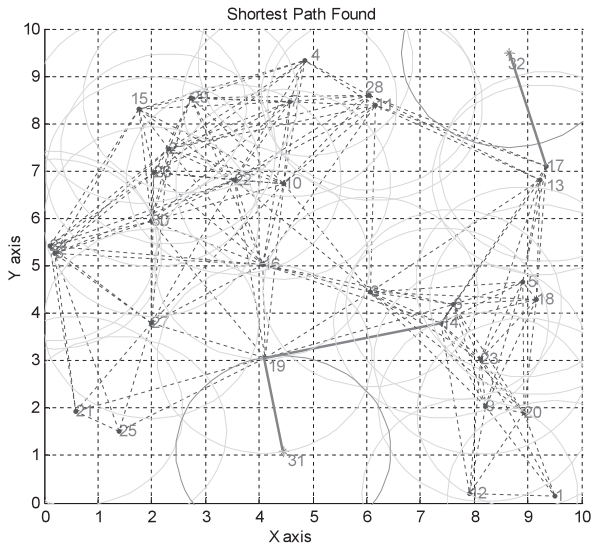


Fig. 21 Find shortest path by A* algorithm from redundant node to virtual node point.

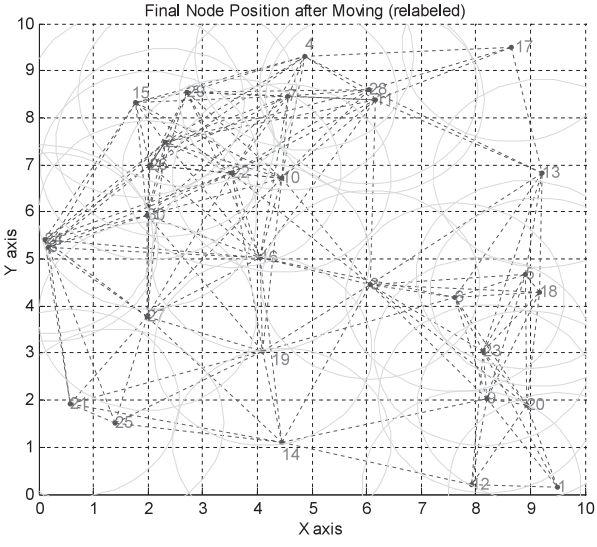


Fig. 22 Final node positions after executing proposed movement-assisted deployment algorithm.

value $T_1 = 1.2$ and $T_2 = r/4$.

Figure 21 shows the two shortest paths found ($14 \rightarrow 19 \rightarrow 31$ and $17 \rightarrow 32$) 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. 19, 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. Figure 22 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.

Next, the performances of RSBA are compared with DSSA, IDCA, and VDDA [15] in terms of coverage, movement distance until convergence, and time. Results are presented in Figs. 23–25. 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.

Figure 23 shows the improvement in coverage area from the initial random deployment for RSBA, DSSA,

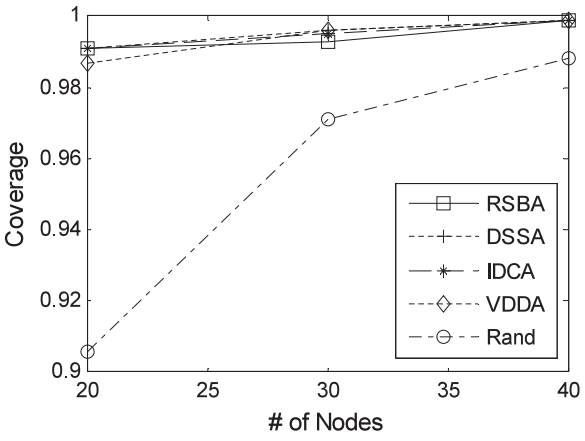


Fig. 23 Coverage comparison.

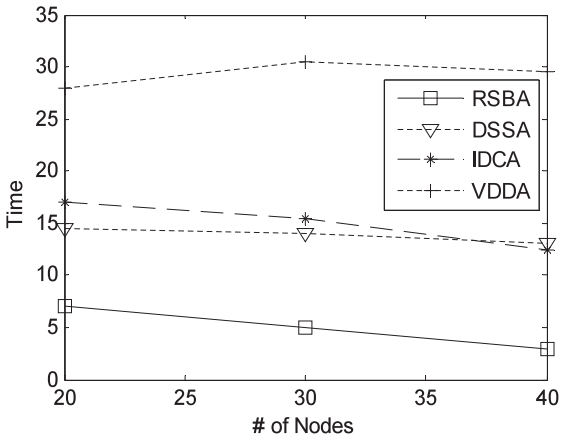


Fig. 25 Termination time comparison.

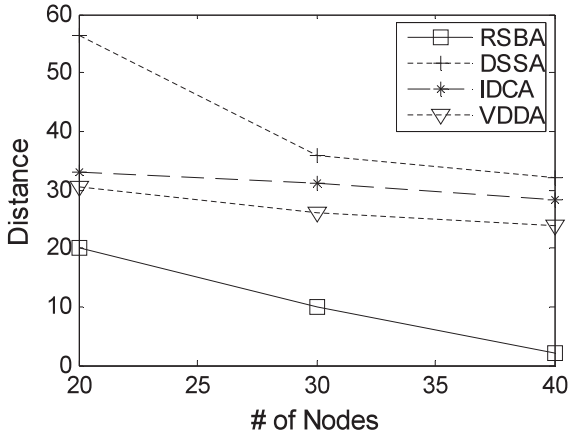


Fig. 24 Total distance traveled comparison.

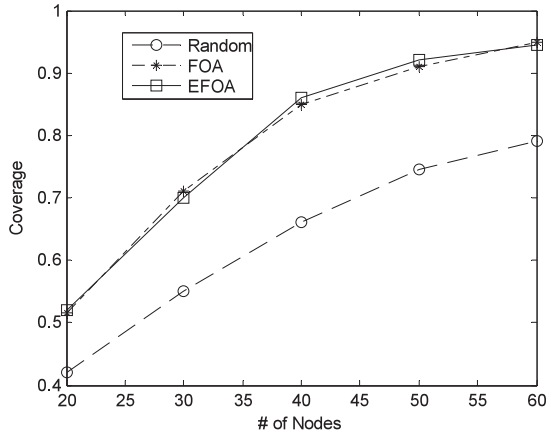


Fig. 26 Coverage vs. number of nodes (R=2, r=1).

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.

Figure 24 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. Figure 25 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.

4.3 Performance Evaluation of EFOA

For the convenience of comparison between EFOA and related work FOA [21], we set the initial parameters the same as in [21]: various number of sensors deployed in a field of 10×10 square area are investigated; the r and R used in the experiment are 1 m and 2 m (2 m and 4 m) respectively. So d_n should be ranged as 0–2 (0–4), not 0–10 as set by [21]. We 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.

Figure 26 shows that the coverage of the initial random deployment, FOA and EFOA when r=1 m and R=2 m. The FOA and EFOA algorithms have similar results that both of them can improve the network coverage by 20–30% in average.

Figure 27 gives the results when r=2 m and R=4 m, 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

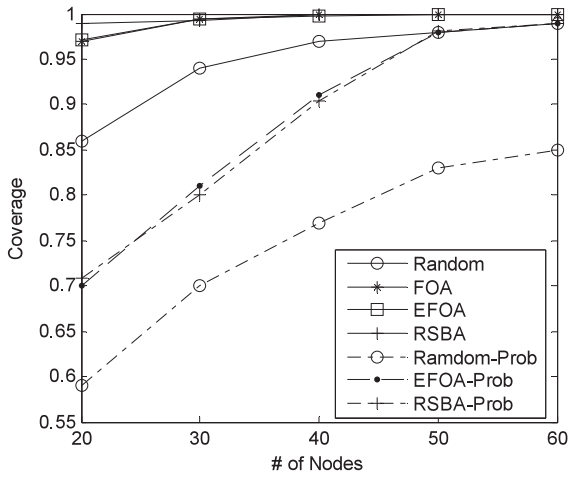


Fig. 27 Coverage vs. number of nodes (R=4, r=2).

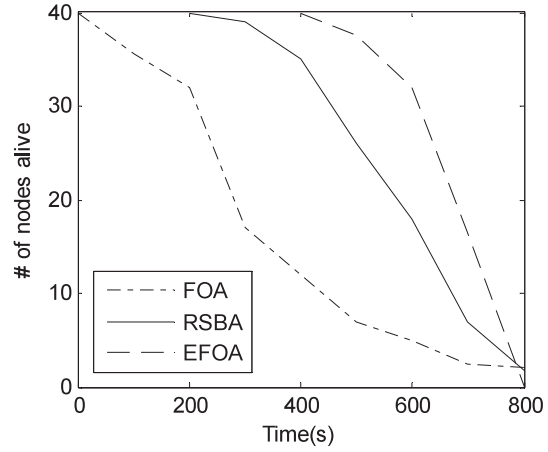


Fig. 28 # of nodes alive over time where each node begins with 2 J of energy (R=4, r=2).

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.

Figure 28 shows the total number of nodes that remain alive over time where each node begins with 2 J 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

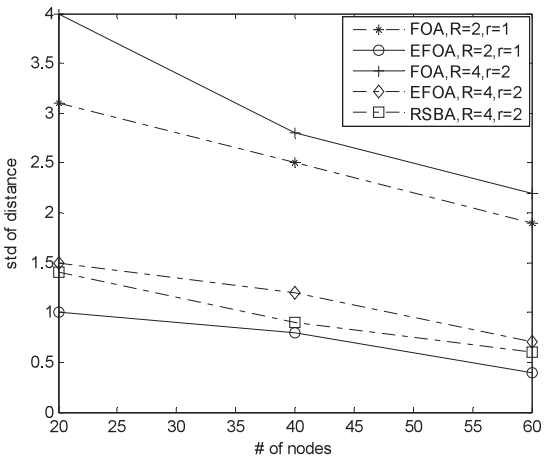


Fig. 29 Standard deviation of distance traveled verses number of nodes.

prolonged in EFOA compared with FOA.

Figure 29 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. Conclusion and Future Work

In this paper, we firstly introduced a comprehensive taxonomy for WSN self-deployment in which 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, 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. We also indicate in the diagram that in the zero mobility case static topology control and scheduling schemes can be used such as optimal cluster formation. Simulation results show that our approaches greatly improve the network coverage as well as energy efficiency compared with related works.

However in this paper, only the discrete mobility metric cases were discussed. In the future work, we plan to study the abstraction of mobility degree in which both the physical factors (such as the friction of movement) and environmental factors (such as obstacles) will be included, so that the continuous mobility metric can be generated. Based on this metric we plan to design general deployment scheme. In addition, according to different situations, different types of initial deployment distribution such as Gaussian distributions can be further studied.

Acknowledgments

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-(C1090-0602-0002)).

References

[1] X. Wu, H. Heo, R.A. Shaikh, J. Cho, O. Chae, and S. Lee, "Individual contour extraction for robust wide area target tracking in visual sensor networks," Proc. 9th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing, pp.179–185, Gyungju, Korea, April 2006.

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

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

[4] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K.K. Saluja, "Sensor deployment strategy for detection of targets traversing a region," ACM Mobile Networks and Applications, vol.8, no.4, pp.453–461, Aug. 2003.

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

[6] A. Howard, M.J. Mataric, and G.S. Sukhatme, "An incremental self-deployment algorithm for mobile sensor networks," Autonomous Robots, vol.13, no.2, pp.113–126, Sept. 2002.

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

[8] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," Proc. IEEE Conference on Computer Communications (INFOCOM), vol.2, pp.1073–1083, San Francisco, March/April 2003.

[9] 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), pp.1380–1387, Anchorage, Alaska, USA, April 2001.

[10] S. Megerian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava, "Worst and best-case coverage in sensor networks," IEEE Trans. Mobile Computing, vol.4, no.1, pp.84–92, Jan./Feb. 2005.

[11] 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), pp.475–483, Fort Lauderdale, Florida, USA, Oct. 2004.

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

[13] G. Wang, G. Cao, and T. La Porta, "Movement-assisted sensor deployment," IEEE Trans. Mobile Computing, vol.5, no.6, pp.640–652, June 2006.

[14] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," ACM Trans. Embedded Computing Systems, vol.3, no.1, pp.61–91, Feb. 2004.

[15] N. Heo and P.K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," IEEE Trans. Syst. Man Cybern. A, Syst. Humans, vol.35, no.1, pp.78–92, 2005.

[16] X. Wu, S. Lei, Y. Jie, X. Hui, J. Cho, and S. Lee, "Swarm based sensor deployment optimization in ad hoc sensor networks," Proc. 2nd International Conf. on Embedded Software and Systems (LNCS 3820), pp.533–541, Xi'an, China, 2005.

[17] 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, pp.300–308, Urbana-Champaign, Illinois, USA, May 2005.

[18] G. Wang, G. Cao, and T. La Porta, "Proxy-based sensor deployment for mobile sensor networks," Proc. 1st IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS'04), pp.493–502, Fort Lauderdale, Florida, USA, Oct. 2004.

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

[20] D.B. Jourdan and O.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), vol.5, pp.2466–2470, Milan, Italy, May 2004.

[21] H. Shu and Q. Liang, "Fuzzy optimization for distributed sensor deployment," Proc. IEEE Wireless Communications and Networking Conference, pp.1903–1907, New Orleans, USA, 2005.

[22] X. Wu, Y. Niu, L. Shu, J. Cho, Y.-K. Lee, and S. Lee, "Relay shift based self-deployment for mobility limited sensor networks," Proc. 3rd International Conf. on Ubiquitous Intelligence and Computing (LNCS 4159), pp.556–564, Wuhan and Three Gorges, China, Sept. 2006.

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

[24] S. Chellappan, X. Bai, B. Ma, and D. Xuan, "Sensor networks deployment using flip-based sensors," Proc. IEEE International Conference MASS, pp.291–298, Washington, DC, Nov. 2005.

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

[26] I.N. Kassabalidis, M.A. El-Sharkawi, R.J. Marks, II, L.S. Moulin, and A.P. Alves da Silva, "Dynamic security border identification using enhanced particle swarm optimization," IEEE Trans. Power Syst., vol.17, no.3, pp.723–729, Aug. 2002.

[27] G.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, pp.1200–1206, Seattle, WA, USA, Oct. 2004.

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

[29] 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. World Automation Congress, pp.31–36, Spain, 2004.

[30] Y. Shi and R.C. Eberhart, "Empirical study of particle swarm optimization," Proc. 1999 Congress on Evolutionary Computation, vol.3, pp.1948–1950, Washington, DC, 1999.

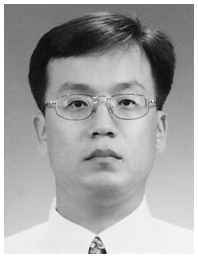
[31] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," IEEE Trans. Wireless Communications, vol.1, no.4, pp.660–670, 2002.

[32] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to algorithms, 2nd ed. MIT Press and McGraw-Hill, 2001.

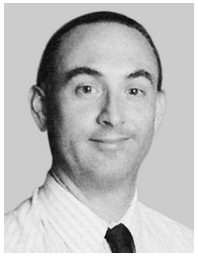
[33] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," Proc. 3rd Annual Communication Networks and Services Research Conf., pp.255–260, Halifax, Canada, May 2005.

[34] The MathWorks, Inc., Fuzzy Logic Toolbox 2.2.3, "http://www.mathworks.com/products/fuzzylogic/," 2006.

Xiaoling Wu received her B.S. and M.S. degrees in electrical engineering and power engineering from Harbin Institute of Technology (HIT), China, in 2001 and 2003, respectively. Currently, she is a Ph.D. student of Department of Computer Engineering at Kyung Hee University, Korea. Her research interests include ubiquitous and intelligent computing, especially sensor networks.



Jinsung Cho received his B.S., M.S., and Ph.D. degrees in computer engineering from Seoul National University, Korea, in 1992, 1994, and 2000, respectively. He was a visiting researcher at IBM T.J. Watson Research Center in 1998 and a research staff at SAMSUNG Electronics in 1998–2003. Currently, he is an assistant professor of School of Electronics & Information at Kyung Hee University, Korea. His research interests include mobile networking & computing, embedded systems & software.



Brian J. d'Auriol received the BSc (CS) and Ph.D. degrees from the University of New Brunswick in 1988 and 1995, respectively. He joined the faculty of computer engineering at Kyung Hee University, Republic of Korea, in 2006 as a research and teaching faculty. Previously, he had been a researcher at the Ohio Supercomputer Center, USA and assistant professor at several American and Canadian universities, including, The University of Texas at El Paso, The University of Akron, Wright State University and The University of Manitoba. He has organized and chaired the International Conference on Communications in Computing (CIC) 2000–2008 and the 11th Annual International Symposium on High Performance Computing Systems (HPCS'97) in 1997. His research includes information and data visualization, optical-communication parallel computing models, communication and computation modeling, and recently, ubiquitous sensor networks. He is a member of the ACM and IEEE (Computer Society).

Sungyoung Lee received his B.S. from Korea University, Seoul, Korea. He got his M.S. and Ph.D. degrees in Computer Science from Illinois Institute of Technology (IIT), Chicago, Illinois, USA in 1987 and 1991 respectively. He has been a professor in the Department of Computer Engineering, Kyung Hee University, Korea since 2001. He is also the director of Neo Medical u-life Care Information Technology Research Center. Before joining Kyung Hee University as an assistant professor in 1993, he was an assistant professor in the Department of Computer Science, Governors State University, Illinois, USA. His current research interests include Ubiquitous Computing Middlewares, Operating Systems, Real-Time Systems and Embedded Systems. He is a member of the ACM and IEEE.