

Energy-Efficient Deployment of Intelligent Mobile Sensor Networks

Nojeong Heo and Pramod K. Varshney, *Fellow, IEEE*

Abstract—Many visions of the future include people immersed in an environment surrounded by sensors and intelligent devices, which use smart infrastructures to improve the quality of life and safety in emergency situations. Ubiquitous communication enables these sensors or intelligent devices to communicate with each other and the user or a decision maker by means of ad hoc wireless networking. Organization and optimization of network resources are essential to provide ubiquitous communication for a longer duration in large-scale networks and are helpful to migrate intelligence from higher and remote levels to lower and local levels. In this paper, distributed energy-efficient deployment algorithms for mobile sensors and intelligent devices that form an Ambient Intelligent network are proposed. These algorithms employ a synergistic combination of cluster structuring and a peer-to-peer deployment scheme. An energy-efficient deployment algorithm based on Voronoi diagrams is also proposed here. Performance of our algorithms is evaluated in terms of coverage, uniformity, and time and distance traveled until the algorithm converges. Our algorithms are shown to exhibit excellent performance.

Index Terms—Ambient intelligence, deployment, distributed algorithms, energy-efficiency, mobile wireless networks, wireless sensor networks (WSN).

I. INTRODUCTION

DESIGN and deployment of infrastructured networks, such as a cellular network, has matured over the last two decades. In such networks, mobile users access the network via fixed base stations. Planning and deployment of these networks is carried out based on radio propagation and terrain models, with the goal of maximizing radio coverage. More recently, there has been a great deal of interest in ad hoc networks. These networks employ fixed or mobile nodes and dynamically organize themselves into a network without requiring an infrastructure. In ad hoc networks, each node acts not only as an end node, but also as a router. One important aspect in the design of these networks is the initialization procedure and establishment of the routing structure. In these networks, a new paradigm is considered, where power usage instead of bandwidth is of primary concern. Extending system lifetime and robustness to unpredictable dynamics, rather than optimizing channel throughput or minimizing the number of nodes, is the biggest challenge during the design of these networks. Most research on ad hoc networks has focused on issues such as the

development of routing protocols and quality of service and not on topology and deployment.

Wireless sensor networks (WSN) that employ ad hoc networking have become an area of intense research activity. This is due to the availability of inexpensive sensors for sensing and control and technical advances in sensors, wireless communications and networking, and signal processing. Many applications are envisaged including: environment and habitat monitoring; wild fire detection; inventory tracking; biomedical analysis; pervasive computing; battlefield surveillance; and urban search-and-rescue operations, especially in hazardous situations. WSN operate under limited radio coverage and attempt to conserve bandwidth and battery power. Much research on this issue is underway ranging from the development of power-saving hardware [18] to power-efficient medium access control (MAC) and routing protocols [11], [27], and to the development of collaborative signal processing and power-aware algorithms [14]. Sensor nodes are generally assumed to be fixed and randomly placed. The number of sensors is assumed to be quite large so that coverage of the surveillance area is not an issue. Not much attention has been paid to optimization in terms of the number of nodes or their topology.

One of the key issues in this area is the deployment of mobile sensor nodes in the region of interest (ROI), where interesting events might happen and the corresponding detection mechanism is required. Before a sensor can provide useful data to the system, it must be deployed in a location that is contextually appropriate. Optimum placement of sensors results in the maximum possible utilization of the available sensors [23]. The proper choice for sensor locations based on application requirements is difficult. The deployment of a static network is often either human monitored or random. Though many scenarios adopt random deployment for practical reasons such as deployment cost and time, random deployment may not provide a uniform sensor distribution over the ROI, which is considered to be a desirable distribution in mobile sensor networks. Uneven node topology may lead to a short system lifetime. Self-deployment methods using mobile nodes [10], [25], [28] have been proposed to enhance network coverage and to extend the system lifetime via configuration of uniformly distributed node topologies from random node distributions. Since mobility itself requires energy from its own limited energy source, a deployment scheme should be designed carefully to minimize energy consumption during deployment, as well as to achieve certain goals, such as satisfactory coverage and/or an energy-efficient node topology. Moreover, it is desirable for a distributed sensor network node to have a relatively simple hardware architecture, which requires minimal computing power and memory. Each node should have

Manuscript received October 20, 2003; revised March 9, 2004 and June 21, 2004. This paper was recommended by Guest Editor G. L. Foresti.

The authors are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: nheo@syr.edu; nheo@ieee.org; varshney@syr.edu).

Digital Object Identifier 10.1109/TSMCA.2004.838486

a simple and efficient algorithm for deployment, organization, and management of the network. Even though much research on energy-efficient organization and management for the static node topology [23], [30] has been carried out, there has not been any work on energy efficiency for deployment of mobile nodes to the best of our knowledge.

Deployment process itself is very energy consuming due to the locomotive action as well as computation and communications associated with it. Each node has a limited energy source. Not only minimizing average moving distance, but also reducing the difference of the remaining energy among sensor nodes is essential for a longer system lifetime. Due to the dynamic and distributed nature of deployment, it is a challenging task to obtain full coverage in the ROI and to utilize energy of each sensor in a relatively fair fashion.

Previous research in distributed-sensor networking has largely ignored sensor placement issues. Intelligent sensor deployment strategies are necessary to minimize cost and to provide sufficient sensor coverage. In addition, sensor deployment must take into account the nature of the terrain, redundancy due to the likelihood of sensor failures, and the power needed to transmit between deployed sensors.

The deployment of sensor networks varies with the application considered. It can be predetermined when the environment is sufficiently known, in which case, the sensors can be strategically hand placed [2], [19], [22]. Schwiebert *et al.* [22] restrict their investigation to an important class of WSN, namely biomedical sensor networks, in which the locations of the sensors are fixed and the placement can be predetermined. Biagioni *et al.* [2] present and analyze a variety of regular deployment topologies, including circular and star deployments as well as deployments in square, triangular, and hexagonal grids. There exists a close resemblance between the sensor-placement problem and the traditional art gallery problem (AGP) in computational geometry [19]. The AGP seeks to determine the minimal number of positions for guards or cameras so that every point in a gallery is observed by at least one guard or camera. A deterministic solution can be found for the AGP and it appears to be a possible solution to a variety of sensor-placement problems. Even though there are many solutions to the AGP, all of them assume the availability of a good model of the environment *a priori*. However, it is virtually impossible to have complete information regarding the environment, where a WSN is likely to be deployed. Furthermore, too much communication over long range to obtain global information requires a huge amount of energy. This is an unaffordable burden on a system with limited power supply. Thus, deterministic deployment is impractical for many reasons, such as the harshness of the deployment region that may be remote, and inhospitable and the increased cost and latency due to the large number of nodes deployed [23].

The deployment cannot be determined *a priori* when the environment is unknown or hostile in which case the sensors may be air-dropped from an aircraft [6] or deployed by other means, generally resulting in a random placement [9], [10], [17], [25]. In this paper, the self-deployment of mobile sensor nodes is considered. This is quite similar to problems considered in cooperative mobile robotics [5]. Mobile sensors are often desirable, since they can patrol a wide area, and can be repositioned for

better surveillance [21]. Some researchers have considered the use of mobile robots in sensor networks. A recent work on mobile sensor networks [10] presents a distributed and scalable potential field-based approach for the deployment of mobile sensors. The fields are constructed such that each sensor is repelled by both obstacles and by other sensors, thereby forcing the network to spread itself through the environment. Winfield [25] considered autonomous dispersion of mobile nodes in a scenario where mobility is required to cover the entire region due to a lack of wireless-network connectivity. He used a random diffusion method for node deployment while collecting data over a fixed surveillance region. In the incremental deployment algorithm [9], nodes are added one at a time. The goal is to maximize network coverage under the constraint that nodes maintain line-of-sight with each other. Loo *et al.* considered a system consisting of a number of cooperating mobile nodes that move toward a set of prioritized destinations under sensing and communication constraints [16]. They show how individual agents know when cooperation between agents improves the performance and when they should suspend cooperation.

A related problem to deployment in WSN is spatial localization [4]. In WSN, nodes need to be able to locate themselves in various environments and on different distance scales. Meguerdichian *et al.* have considered the problem of location and deployment of sensors in a WSN from a coverage standpoint [17]. The problems of coverage and deployment are fundamentally interrelated. The authors define the coverage problem from different points of view, including deterministic, statistical, and the worst and best cases. They implicitly assumed fixed wireless sensor nodes. They argued that coverage is a primary performance metric that provides an indication regarding quality-of-service. They combined computational geometry and graph theoretic approaches to develop algorithms for coverage calculations. Coverage in WSN, which is one of the main focuses in this paper, will be discussed later in Section II. Bulusu *et al.*'s work [3] is somewhat similar to the deployment problem that is considered here. They have investigated the problem of adaptive beacon placement for localization in a WSN. They also pointed out the lack of viability and inadequacy of fixed and dense beacon placement in some situations due to node perturbation during deployment, noisy environment, and self-interference. By placing additional beacons incrementally, they achieve empirical adaptation to terrain conditions. Unlike traditional sensor systems, sensor networks depend on dense sensor deployment and physical collocation with their targets to accomplish their goals. Dense deployment allows the use of redundancy [26], can reduce communication costs [20], and provides sufficient number of nodes to allow physical collocation.

In this paper, three different deployment methods are proposed. First, a deployment algorithm for mobile nodes is proposed when each node is equally important and a peer-based structure is obtained. In many WSN scenarios, clustering is employed to take advantage of local information and to reduce energy consumption. An intelligent energy-efficient deployment algorithm for cluster-based WSN is proposed. The key idea of the second algorithm is the introduction of local clustering [12], [15] during the deployment process so as to increase the amount of local control over a fraction of the entire ROI. Each node

decides its own mode to be either in a clustering mode or a peer-to-peer mode based on its local environment such as the local density and the remaining energy level in a distributed and adaptive manner. Finally, an energy-efficient deployment algorithm based on Voronoi diagrams (VDs) is proposed.

The goal of the first method is different from prior work on the deployment problem. The main objective of the first deployment algorithm is topology improvement for longer system lifetime by utilizing mobility of sensor nodes. A decision and control mechanism is provided at each sensor during deployment, rather than random diffusion, which is used in Winfield's work [25]. In contrast to Howard *et al.* [9], who use an incremental approach, the nodes in the first algorithm are deployed at the same time and they organize themselves in an adaptive manner. Unlike Loo *et al.* [16], the first algorithm does not require pre-specified destinations to form an energy-efficient topology.

The significance of the second method is to provide a synergistic combination of cluster structuring and peer-to-peer deployment scheme in an intelligent manner in a hostile and unpredictable environment. The goals of our algorithm are the realization of the largest possible coverage area of the network, the formation of an energy-efficient node topology for a longer system lifetime, and the organization of a hierarchical structure for easier management and scalability that supports collaboration among nodes. These goals can be achieved by an adaptive combination of two modes: clustering and peer-to-peer. In a peer-to-peer mode, each node moves itself to a sparse region so that the coverage of the network may increase and/or an energy-efficient node topology may be achieved. In a clustering mode, each node follows the decision of the cluster-head so that each node spends its energy in a balanced way and performs collaborative missions if necessary.

The significance of the third method is to provide an estimate of the lifetime of each node in a distributed fashion by using local VDs. Each node can determine how long it can survive and which action is more useful to its longevity for the current node topology during deployment. The best energy-utilization point is obtained by comparing utility gains for movement to different possible node locations.

All three methods in this paper are based on the same assumptions shown in Section II but have different strengths for possible applications. In practice, sensor capabilities may vary depending on the requirements for a certain task and the available budget. The first method can achieve a quick deployment with simple sensors. The second method can establish clustering structure during deployment. The third method requires more computation, but shows local assessment of the performance and high energy efficiency in mobility. The proper deployment method can be chosen based on the requirements of the applications and resources available.

Our deployment algorithms will be more useful in situations where it is hard to ensure precise initial deployment due to the fact that the deployment area is too dangerous or inaccessible to humans. One can envisage an application involving a hazardous region, where sensors mounted on mobile robots are deployed from an airborne vehicle [6]. These mobile robots then organize themselves using algorithms presented in this paper. Randomly scattered sensors over a battlefield or a hazardous site are not

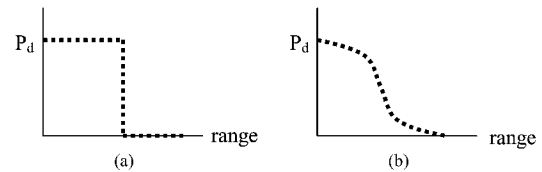


Fig. 1. Sensor coverage models. (a) Binary sensor and (b) stochastic sensor models.

likely to form a uniform distribution and provide desired coverage. Modification of WSN topology in an autonomous and distributed manner using our algorithms can help in improving coverage and also to prolong expected system lifetime. This is essential in time-critical applications. For example, if an area is contaminated by some hazardous material, a properly deployed sensor network can quickly sense and measure the amount of hazardous material such as poisonous gas or nuclear leakage. By fully covering the entire area of interest, the overall condition can be assessed quickly and this information can be used for search and rescue missions, as well as for evacuation-route planning. In some applications, initial sensor distributions may be concentrated at specific points, such as elevators and stairs in a building.

In the next section, the sensor deployment problem is formulated. In Section III, performance metrics for a mobile WSN are discussed. The deployment algorithms are presented in Section IV followed by simulation results in Section V. Some concluding remarks are provided in Section VI.

II. MOBILE-NODE DEPLOYMENT PROBLEM

In a WSN, physical placement or deployment of sensor nodes is needed prior to the initialization of a network for data acquisition and transmission using sensor nodes. The assumptions made in this paper are described and the deployment problem is formulated in this section.

It is assumed that all sensor nodes have identical capabilities for sensing, communication, computation, and mobility. Sensing coverage and communication coverage of each node is assumed to be ideal, which means that both coverage areas have a circular shape without any irregularity.

The coverage of each sensor can be defined either by a binary sensor model [23] or a stochastic sensor model [28] as shown in Fig. 1. In the binary sensor model, the probability of detection of the event of interest is one within the sensing range (sR), otherwise, the probability is zero. So the coverage of a sensor network using the binary sensor model is determined by finding the union of the areas defined by the location of each sensor and its sR. In the stochastic sensor model, the probability of detection of the event of interest follows a decaying function of distance from the sensor. In this paper, the binary sensor model is employed.

Computation capability is required at each node to support a distributed algorithm that includes a reasoning and optimization process for deployment and routing. It is assumed that the initial deployment is random and a distributed-deployment algorithm is executed starting from the initial random topology, using each node's mobility. Another assumption is that every node has the ability to know its own location by some method,

such as the global positioning system or iterative multilateration [24]. This locationing ability is needed by each node, while making a decision regarding its next movement in the deployment process. Also, it is assumed that there are no errors during transmission of data and in the calculation of locations. It is further assumed that each node has only local information from the neighboring nodes within its direct communication range (cR). The cR of each node is defined by the maximum distance at which the signal-to-noise ratio is above a given threshold.

Without loss of generality, the deployment problem for a rectangular ROI with a certain number of nodes that form an ad hoc wireless network is considered. Each node has a limited amount of energy. The goal is to find the positions and movements of nodes to achieve maximum coverage and to form a uniformly distributed wireless network in minimum time and with minimum energy consumption. A suite of heuristic algorithms are developed for this problem and their performances are evaluated in terms of the performance metrics: coverage, uniformity, and the time and distance traveled until convergence. These metrics are described next.

III. PERFORMANCE METRICS IN MOBILE WSN

The selection of suitable measures to compare performances of different approaches and resulting solutions is an important issue in a mobile WSN. Coverage, uniformity, and time and distance traveled prior to convergence are considered as performance metrics in mobile WSN here. Coverage and uniformity are related to the performance of sensor networks after the deployment of sensors is complete. Time and distance traveled prior to convergence are directly related to the performance of the deployment scheme itself.

A. Coverage

Generally, coverage can be considered as the measure of quality of service of a sensor network. The concept of coverage as a paradigm for the system-level functionality of multirobot systems was introduced by Gage [7], [31].

In this paper, coverage [9] is defined as the ratio of the union of areas (in square meters) covered by each node and the area (in square meters) of the entire ROI. Here, the covered area of each node is defined as the circular area within its sensing radius R . Perfect detection of all interesting events in the covered area is assumed

$$C = \frac{\cup_{i=1, \dots, N} A_i}{A}$$

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.

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, i.e., πR^2 , 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. Because of the areas covered by nodes that fall out of the ROI

and the overlap of covered areas between nodes, one needs to use more nodes than simply the ratio of A and the area sensed by a single node.

The overall coverage of a sensor network is composed of the covered regions of each sensor node. Though the coverage of a sensor is expressed by a sensor model which is binary or stochastic, the overall coverage of a sensor network depends on the locations of the sensor nodes in the sensor field. The topology including the locations and spacing of sensor nodes determines the overall coverage of the network as well as the expected lifetime of the network.

sR and cR of a node are distinguished in the paper. In general, they will be different and accordingly sensing coverage and communication coverage will be different. Sensing coverage can be accrued when sensor nodes are connected via wireless links. If the network is separated by any reason, the area covered by the subnets that do not have wireless links to the sink node is lost.

B. Uniformity

Uniformly distributed-sensor nodes spend energy more evenly through the WSN than sensor nodes with an irregular topology. When the distances between nodes become similar, each node can utilize its resources efficiently with the minimum use of its power and an increased throughput, due to reduction of the interferences between nodes. So, uniformity of network topology can be used as a good estimator for the expected system lifetime. Also, fewer nodes are required to cover an ROI when nodes are more evenly distributed.

Uniformity can be defined as the average local standard deviation of the distances between nodes

$$U = \frac{1}{N} \sum_{i=1}^N U_i$$

$$U_i = \left(\frac{1}{K_i} \sum_{j=1}^{K_i} (D_{i,j} - M_i)^2 \right)^{\frac{1}{2}}$$

where

- N is the total number of nodes;
- K_i is the number of neighbors of the i th node;
- $D_{i,j}$ is the distance between i th and j th nodes;
- M_i is the mean of internodal distances between the i th node and its neighbors.

In the calculation of local uniformity U_i at the i th node, only neighboring nodes that reside within its cR are considered. The uniformity measure is a local measure and is computed locally because each node has access to local information only. A smaller value of U means that nodes are more uniformly distributed in the ROI. In uniformly distributed networks, internodal distances are almost the same; the expected energy consumption per communication as well as the expected lifetime of each node is almost the same if the nodes were identical and have the same amount of energy initially. Therefore, it is expected to have full energy utilization at each node and longer system lifetime for uniformly distributed networks.

C. Time

The time spent for deployment [9] is also important in many time-critical applications, such as search-and-rescue and disaster recovery operations. Mostly, the required time depends on the complexity of the reasoning and optimization algorithm and physical time for the movement of nodes. The total time elapsed is defined here as the time elapsed until all the nodes reach their final locations. This paper focuses on the time spent for deployment itself and not on data-transmission delays from a source node to a destination node that is commonly used for network-performance evaluation and its quality of service.

D. Distance

The average distance traveled [9] by each node is related to the energy required for its movement. So, the expected distance traveled is important for the estimation of energy (fuel) required when each node has a limited energy supply. The variance of the distance traveled is also important to determine the fairness of the deployment algorithm and for system energy utilization. If the variance of distance traveled is large, the variance of energy remaining also is large. The nodes that have less energy than other nodes exhaust their energy early. Early dead nodes result in a loss of coverage and the remaining nodes may require an increased transmission range or a longer routing path.

IV. ALGORITHMS

In this paper, three different deployment methods are presented. The first method operates in a peer-to-peer mode, where each node is considered to be equal. The second method is a synergistic combination of the peer-to-peer method with a cluster-based method. Clustering, a hierarchical networking concept, is employed in many WSN scenarios to take advantage of local information and to reduce energy consumption. Finally, an energy-efficient deployment algorithm based on VDs is proposed.

A. Distributed Self-Spreading Algorithm (DSSA)

The peer to peer algorithm which is called the DSSA is inspired by the equilibrium of molecules, which minimizes molecular electronic energy and internuclear repulsion. Each particle determines its own lowest energy point in a distributed manner and its resulting spacing from the other particles is almost the same. While deploying a WSN using mobile nodes, one observes that one has an analogous problem. If sensors are located too close to each other, the gain in coverage from additional sensors is not high. On the contrary, if sensors are located too far from each other, the coverage regions may not overlap and may cause a partitioning of the network. Both situations are similar to internuclear repulsion and attractions between molecules. Optimal spacing between sensors in the sense of coverage can be found by a process similar to the equilibrium of molecules.

To begin with, a specified number of nodes are deployed randomly in a given region, for instance, inside a rectangle. The sR and cR are assumed to be given. Each node can sense or detect an event within its sR and any pair of nodes within their cR can communicate with each other. This communication is needed for finding neighborhoods, obtaining locations of nodes

Procedure Distributed_Self_Spreading_Algorithm

1. Initialization

initial_node_locations p_0 ;

sensing_range sR ;

communication_range cR ;

calculate local_density D ;

calculate expected_density μ ;

While (*Not*(*Oscillation occurred OR In a region of stability*))

2. Partial Force Calculation

calculate partial_force $f_n^{ij}(\mu, D, cR, p_n)$;

update temporary_position p_{n+1}^i ;

3. Oscillation?

If ($|p_{n-1}^i - p_{n+1}^i| < \text{threshold}_i$)

Increase oscillation_count by 1;

If (*oscillation_count* < *oscillation_limit*)

Update next location to the temporary_position;

Update local_density D ;

Else

Move to the centroid of oscillating points;

Update local_density D ;

Stop node i's movement;

Else

Update next location to the temporary_position;

Update local_density D ;

4. Stability?

If ($|p_{n+1}^i - p_n^i| < \text{threshold}_i$)

Increase stability_count by 1;

If (*stability_count* < *stability_limit*)

Go to while loop;

Else

Stop node i's movement;

Else

Go to while loop;

Fig. 2. Pseudocode for the DSSA.

in the neighborhood, and transmitting and forwarding sensed data. The neighborhood of a node is defined here as nodes within its cR . The pseudocode of the algorithm is given in Fig. 2. This distributed algorithm is executed at each node i . The algorithm contains four parts.

1) *Initialization*: In the initialization part, the values of cR, sR, and the initial node locations (p_0) are specified. The cR and sR are assumed to be given. Initial node locations (p_0) are specified in terms of a vector that contains the longitude component and the latitude component of each node location, and is assumed to follow a random distribution. Extension to higher dimensions is possible by adding more components in the position vector. A quantity called expected density, which is a rough estimate of the desired density, is required in the algorithm. This can be calculated by using $\mu(cR) = (N \cdot \pi \cdot cR^2 / A)$, where N is the number of nodes, cR is the cR of each node, and A is the area of the ROI. Thus, expected density is the average number of nodes required to cover the entire area when these nodes are deployed uniformly. Initial local density D_0 of a node is equal to the number of nodes within its cR. These densities will be used when decisions regarding positions of nodes are made.

2) *Partial Force Calculation*: The concept of force is introduced to define the movement of nodes during the deployment process. The force is dependent on not only the distance between the nodes but also the current local density. The force corresponding to high local density is greater than the force corresponding to low local density. The force from a node that is closer is greater than that from a node that is farther, which is similar to the movements of the particles in physics that follow Coulomb's Law.

A force function is defined which satisfies the following conditions.

- i) Inverse relation: $f(d_1) \geq f(d_2)$, when $d_1 \leq d_2$, where d_1 and d_2 are node separations from the origin. The node under consideration is assumed to be at the origin.
- ii) Upper bound: $f(0^+) = f_{max}$.
- iii) Lower bound: $f(d) = 0$, where $d > cR$, d is the node separation and cR is the communication range of each node.

Condition (i) is the same as in physics, but conditions (ii) and (iii) are included to modify the model to incorporate the notion of locality. In other words, a limiting function is applied via conditions (ii) and (iii).

The partial force at time step n on the i th node from the j th node that is in the neighborhood of the i th node is calculated to be a repulsive force as

$$f_n^{i,j} = \frac{D_n^i}{\mu^2} (cR |p_n^i - p_n^j|) \frac{p_n^j - p_n^i}{|p_n^j - p_n^i|} \quad (1)$$

where

- p_n^i stands for the location of the i th node at time step n ;
- D_n^i stands for the local density of the i th node at time step n .

The density factor (D/μ^2), which is defined as the ratio of the local density (D) and the square of the expected density (μ) at each node, is small in sparse regions and is large in dense regions. Its inclusion in the force function expedites the process of node spreading. Also, internodal distance affects the partial force inversely. Closely located nodes impose larger partial forces and nodes that are far apart induce smaller partial forces on each other. The magnitude of the partial force exerted by a

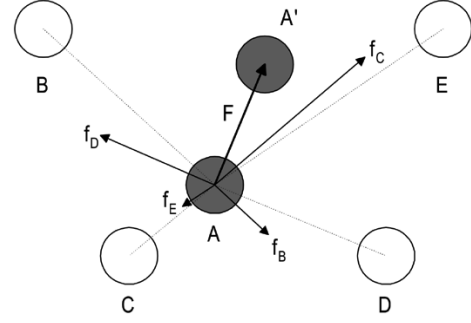


Fig. 3. Illustration of nodal movement.

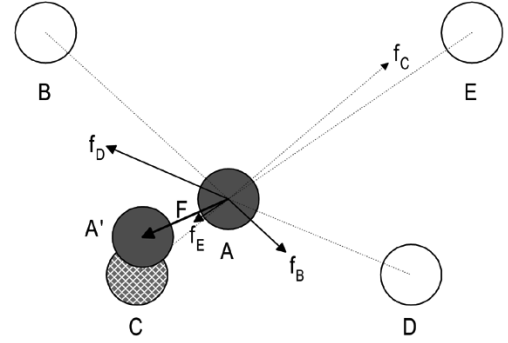


Fig. 4. Illustration of nodal movement in the presence of a failed node.

pair of nodes on each other are the same with the only difference that the directions are opposite to each other.

After adding all the partial forces at the current node location, each node decides its next movement. This process provides a local decision, which includes the consideration of its local situation, such as the locations of the neighboring nodes and dead node(s), if any. The local information is collected from the nodes that are within the cR and that information is used for the calculation of the local density at each node. Each node's movement is decided by the combined force at that node due to nodes in its neighborhood.

3) *Oscillation Check*: An important issue is determining when a node should stop its movement. Two stopping criteria are introduced in the DSSA. If a node moves back and forth between almost the same locations many times, this node is regarded to be in the oscillation state. By examining the history of its movement, each node can determine if oscillations are going on. One counts the number of oscillations and if this oscillation count (O_{count}) is over the oscillation limit (O_{lim}), the movement of that node is stopped at the center of gravity of the oscillating points.

4) *Stability Check*: If a node moves less than $threshold_2$ for the time duration $Stability_limit(S_{lim})$, this node can be considered to have reached the stable status and that node stops its movement. This stopping criterion is useful for stationary nodes because of either exhausted fuel or broken mobile units and also for the nodes that have reached the stable status.

To illustrate the algorithm, examples of nodal movements are shown in Figs. 3 and 4.

In Fig. 3, the next movement of node A is considered. Nodes B–E are its neighboring nodes within its cR. The partial forces f_B, f_C, f_D , and f_E from nodes B–E on A are calculated by (1).

Then the total (resultant) force F on A can be obtained by just adding partial forces on A , i.e., $F = f_B + f_C + f_D + f_E$. Note that this addition includes both magnitude and direction, i.e., it is a vector addition. Node A will move to the new position A' as shown. Node C is the closest node to A among neighboring nodes, so their spacing should be increased. As one can see, partial force f_C dominates the total force F and A moves away from C .

When sensor nodes are deployed in a remote and hostile region, some nodes can be adversely affected during and after deployment. Some nodes can lose their mobility and other nodes can lose their communication functionality. So, a robust deployment method that can overcome these situations is needed. The first algorithm exhibits this kind of robustness. First, when a sensor node loses its mobility, that sensor node does not move and is considered to be an early stopped node. However, this node can still be used as a static node in the sensor network. Neighboring nodes, if they can move, may still improve the irregular topology. Second, when a sensor node loses its communication capability, that sensor node is of no use in a sensor network. In that case, neighboring nodes may move to the locations so that the uncovered region can be covered by them. This case is illustrated in Fig. 4.

The node positions are the same as in the previous example. Suppose node C is broken during the random deployment period and it cannot use its communication unit. Then, neighboring nodes including node A and disregard this broken node during the calculations of their next movements. The largest partial force f_C in the previous example is now excluded, i.e., $F = f_B + f_D + f_E$. The next position for node A is A' and the lost coverage from the loss of node C can be recovered by node A . Since the movement of each node is only affected by the current status of neighboring nodes, each node adapts to environment changes, such as node failures, various terrain shapes, etc., and changes its position in an autonomous manner to maximize coverage and uniformity.

B. Intelligent Deployment and Clustering Algorithm (IDCA)

In many WSN scenarios, clustering is employed to take advantage of local information and to reduce energy consumption. By introduction of local clustering [12], [15] during the deployment process, it is possible to improve the energy-consumption characteristics of sensor nodes. Each node decides its own mode to be either in a clustering or peer-to-peer mode based on its local environment, such as the local density and the remaining energy level in a distributed and adaptive manner. We call this algorithm the IDCA. The pseudocode of the algorithm is given in Fig. 5. This distributed algorithm is executed at each node i . The IDCA also contains four parts, like the DSSA.

1) *Initialization*: The same as the DSSA algorithm.

2) *Mode Determination and Partial Force Calculation*: Intuitively, sensor nodes in a dense region need to move to a sparse region to improve coverage and connectivity of a sensor network. Node movement and also the corresponding energy consumption is expected in both sparse and dense regions. Nodes in a region with the correct node density do not need to move and spend their energy to improve the performance in terms of

Procedure Intelligent_Deployment_and_Clustering_Algorithm

1. Initialization

initial_node_locations p_n ;

sensing_range sR ;

communication_range cR ;

calculate local_density D ;

calculate expected_density μ ;

While (*Not*(*Oscillation occurred OR In a region of stability*))

2. Mode Determination and Partial Force Calculation

If *local_density* \neq *expected_density*

Mode = *peer_to_peer*;

calculate partial_force $f_n^{ij}(\mu, D, cR, p_n)$;

update temporary_position p_{n+1}^i ;

Else

Mode = *cluster*;

calculate partial_force $f_n^{ij}(\mu, D, cR, p_n)$;

calculate energy_rank r in k neighboring nodes;

multiply energy_factor r/k to *partial_force* $f_n^{ij}(\mu, D, cR, p_n)$;

update temporary_position p_{n+1}^i ;

3. Oscillation?

If ($|p_{n+1}^i - p_n^i| < \text{threshold}_1$)

Increase oscillation_count by 1;

If (*oscillation_count* $<$ *oscillation_limit*)

Update next location to the *temporary_position*;

Update local_density D ;

Else

Move to the centroid of oscillating points;

Update local_density D ;

Stop node i 's movement;

Else

Update next location to the *temporary_position*;

Update local_density D ;

4. Stability?

If ($|p_{n+1}^i - p_n^i| < \text{threshold}_2$)

Increase stability_count by 1;

If (*stability_count* $<$ *stability_limit*)

Go to while loop;

Else

Stop node i 's movement;

Else

Go to while loop;

Fig. 5. Pseudocode for the IDCA.

uniformity. The reason is that frequent movements of neighboring nodes may degrade existing uniformity and the energy spent to improve uniformity is simply wasted. By delaying the movement of sensor nodes in a region with correct node density until nodal movements stabilize to some extent, inefficient energy usage of those sensor nodes can be improved.

Based on the relation between the local density (D) and the expected density (μ), the mode at a node is determined. If D is close to the expected density (μ), the node selects the clustering mode. Nodes in regions that have desired density levels are not expected to move much to improve coverage and/or uniformity of sensor nodes. A sensor node in such regions determines its movement based on its remaining energy level relative to its neighbors. To begin with, partial force in the clustering mode is calculated by using (1) as used in the DSSA algorithm. Then, this partial force is modified by its rank based on its energy level in the neighborhood. The remaining energies of neighboring nodes are rank ordered. If a sensor node has the rank r among k nodes in the neighborhood, the energy factor is r/k and the partial force calculated by (1) is multiplied by this factor. If the remaining energy level is low, the partial force of the node will be smaller than that used in DSSA based on its energy factor in its neighborhood. The node in this situation saves its energy and contributes less to WSN performance improvement. If the remaining energy level is relatively high among the nodes in the neighborhood, the partial force is determined according to its rank in its neighborhood. The node in this situation uses its energy more and contributes more for performance improvement of the WSN. This energy consideration in the clustering mode reduces the variation of the remaining energy among sensor nodes. If D of a sensor node at any time is different than μ at the current location, this node selects the peer-to-peer mode and partial force calculation is done by using (1).

3) *Oscillation and Stability Checks*: Same as the DSSA algorithm. In this paper, the same stopping criteria are used for both modes: peer to peer mode and clustering mode. However, IDCA algorithm may use different local metrics and stopping criteria in a clustering mode.

C. VD-Based Deployment Algorithm (VDDA)

Many researchers have demonstrated the importance and usefulness of VDs in various fields, such as mathematics, computational geometry, biology, chemistry, geography, communications, and coding theory [1], [29]. Given some number of points called generators or sites in the ROI, their corresponding VD divides the region according to the nearest-neighbor rule. Each given point (a site or generator) is associated with the subregion consisting of any points in the ROI that are closest to it. If these points (generators or sites) are assumed to be sensor locations and the subregions defined by the corresponding Voronoi regions are covered by the sensor in the subregion, one has a possible solution to the deployment problem. A deployment algorithm based on the notion of Voronoi regions is developed. In this paper, the goal is to have the Voronoi region corresponding to a sensor to be coincident with the coverage area defined by the sensor model. The Voronoi region corresponding to a sensor at a certain time instance is considered as the desired solution in terms of coverage. At the same time, each sensor has its

coverage defined by the sensor model. If there is any discrepancy between the current sensor coverage and the corresponding Voronoi region, action needs to be taken to align the two by sensor movement and resulting changes in topology. When the two are aligned within predefined tolerance, nodal movements will be halted and the resulting solution will be accepted. We call this the VDDA. The pseudocode of the algorithm is given in Fig. 6. This distributed algorithm is executed at each node i .

1) *Initialization*: In this part, the values of the cR, the sR, and the initial node locations (p_0) are specified. Initial energy for the i th node (E_0^i) is specified and each node is assumed to have the same amount of energy in the beginning. Energy consumption for different activities is specified. Energy consumption for movement of the i th node is specified in terms of M^i defined as the cost for movement per unit distance. The total cost for movement of the i th node is equal to the product of M^i and the distance. Energy consumption per unit time for communication by the i th node is denoted by C^i and is a function of the largest distance between itself and its neighbors. Energy consumption per unit time for sensing and computation by the i th node (S^i) is assumed to have a fixed cost.

In order to determine the degree to which each node achieves coverage in terms of energy efficiency, a node utility metric is considered. The node utility metric (U_n^i) of the i th node at time step n is defined as

$$U_n^i = A_n^i \times T_n^i$$

where

A_n^i represents the effective area covered by the i th node at time step n ;

T_n^i represents the estimated lifetime of the i th node at time step n .

The node utility metric indicates how well the node is utilized to sense over its effective area during its lifetime.

The local VD is used to calculate the effective area of each node. First, the initial local VD (VD_0) is obtained using p_0 and cR. For all the points in the Voronoi region corresponding to the i th node (VD^i), the nearest sensor node is the i th node. The sensing area of the i th node is the circular area centered at p_0^i with the radius sR. Then the initial effective area A_0^i is obtained by the intersection of the Voronoi region of the i th node and the sensing area of the i th node. The effective area of the i th node means the area covered by the i th node based on the nearest node concept. Each node has only local information from the neighboring nodes. The node can calculate its VD with the information of the neighbors. When a different cR is used, the size of the neighborhood is different and the VD is also different.

The estimated lifetime (T_n^i) of i th node at time step n is defined as

$$T_n^i = \frac{E_{n-1}^i - M_n^i \times d_{n-1,n}^i}{C_n^i + S_n^i}$$

where

E_{n-1}^i stands for the energy remaining at the i th node after time step $n - 1$;

M_n^i stands for the energy consumption per unit distance for movement of the i th node at time step n ;

Procedure Voronoi_Diagram_based_Deployment_Algorithm

1. Initialization

initial_node_locations p_0 ;

sensing_range sR ;

communication_range cR ;

initial_energy E_0^i ;

energy_consumption_rate M_0^i, C_0^i, S_0^i ;

calculate initial VD(p_0, cR)

calculate initial_effective_area A_0^i from VD(p_0, cR) and

SensingArea(p_0^i, sR);

calculate initial_estimated_lifetime T_0^i ;

calculate initial_node_utility $U_0^i = A_0^i \times T_0^i$;

While (Not/utility_gain > a predefined threshold)

2. Find the maximal energy utilization point

Set the current solution as the last solution of previous step: $U_{max,n}^i =$

U_{n-1}^i

calculate VD(p_n, cR)

find the centroid of VD: VD_1

{search the maximal energy utilization point among the points linearly spaced between p_1 and VD_1

calculate effective_area $A_{i,n}^i$ from VD(p_n, cR) and SensingArea($p_{i,n}^i, sR$)

calculate energy_consumption_rate $M_{i,n}^i, C_{i,n}^i, S_{i,n}^i$

calculate estimated_lifetime $T_{i,n}^i$

if $U_{i,n}^i > U_{max,n}^i$

$U_{max,n}^i = U_{i,n}^i$

else

keep searching

end

update p_i

}

find the mean of maxVD and minVD: VD_2

{search the maximal energy utilization point among the points linearly spaced between p_1 and VD_2

calculate effective_area $A_{i,n}^i$ from VD(p_n, cR) and SensingArea($p_{i,n}^i, sR$)

calculate energy_consumption_rate $M_{i,n}^i, C_{i,n}^i, S_{i,n}^i$

calculate estimated_lifetime $T_{i,n}^i$

if $U_{i,n}^i > U_{max,n}^i$

$U_{max,n}^i = U_{i,n}^i$

else

keep searching

end

}

update p_i

update $U_n^i = U_{max,n}^i$

update current energy E_n^i

End of while loop

$d_{n-1,n}^i$ stands for the distance moved by the i th node between time step $n - 1$ and time step n ;

C_n^i stands for the energy consumption per unit time due to communication by the i th node;

S_n^i stands for the energy consumption per unit time for sensing and computation by the i th node.

The node utility metric is used to decide energy-efficient movements by each node.

2) *Finding the Best Energy-Utilization Point:* In this part, the best energy-utilization point of each node is obtained by comparing utility gains for potential movement to different possible node locations. Moving from the current location to different node locations will incur a different energy cost for movement and the resulting node topology may or may not reduce the communication cost. Because the search space is infinite when continuous coordinates are considered, the reduction of search space is necessary to some extent. In this paper, local VDs are used to reduce the search space. Due to the nearest neighbor relation of VDs, the Voronoi region of the i th node will be more likely covered by the i th node than any other nodes. In this sense, the local Voronoi region can be considered as the estimated or desired coverage by the local node. Moving to the centroid of the Voronoi region can be beneficial in terms of coverage and/or uniformity. The centroid of the Voronoi region is obtained by using the *Matlab* function *ployarea*. Also, the center of the Voronoi range, which is defined as the midpoint of maximum and minimum along the x and y coordinates in the Voronoi region, can be used to guide the search. The search space is reduced to several points linearly spaced, starting from the current location to the centroid of the Voronoi region, and from there, to the center of the Voronoi range. For these points, the node utility metric is evaluated and the best action is determined. Because the best solution is kept during the search process, a locally optimal solution is obtained after the search process. After finding the best energy-utilization point for each node, actual movement of each node occurs at a single time to save energy.

V. EXPERIMENTAL RESULTS

The performance of the heuristic algorithms in the paper is evaluated by simulation. In the experiment, 30 randomly placed nodes in a region of size 10×10 are used to run the DSSA, the IDCA, and the VDDA. The sR and cR used in the experiment are 2 and 4 m, respectively.

In Fig. 7, the node locations and coverage of the initial random deployment before running the algorithms are shown. Tiny circles represent the positions of nodes and small (shaded) and large circles are used to show the sR and cR of the nodes, respectively. Sensor information may be collected within the sR and communications between nodes are possible within the cR . Communications are possible between nodes that are connected by a line in the figure. As seen in Fig. 7, some parts of the region cannot be covered by the randomly dispersed nodes, even though there are sufficient nodes in the given ROI. The coverage is obtained by adding up small areas that are considered in the sR of any nodes, not considering the connectivity between the areas. It is possible that the areas are not connected. In that

Fig. 6. Pseudocode for the VDDA.

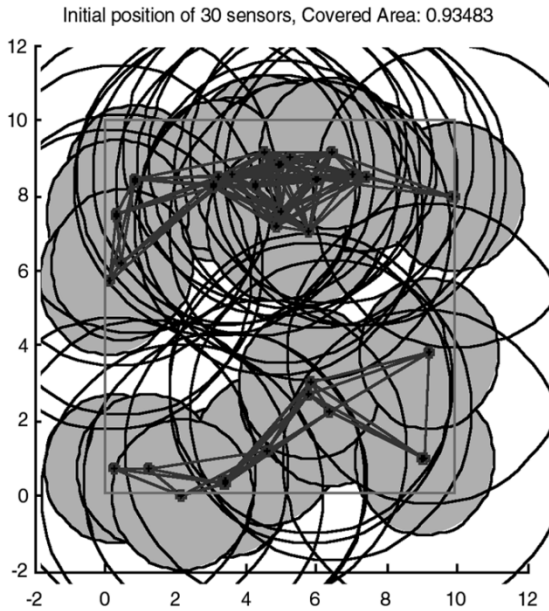


Fig. 7. Initial distribution of sensor nodes.

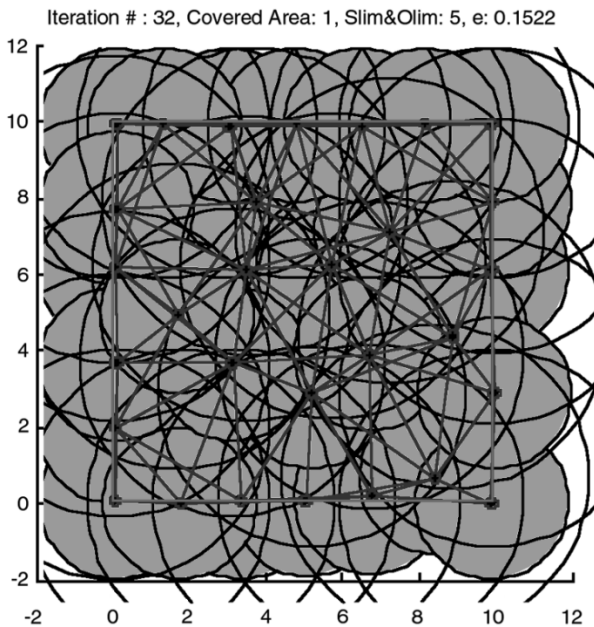


Fig. 8. Final node distribution after running DSSA.

case, the actual coverage that can be reported to the user is the area that is connected to the sink node, not the arithmetic sum of disconnected areas. In this particular example, the network is not fully connected, so the actual coverage is much smaller than just adding the coverages of the disconnected covered regions. Similar situations can occur when random deployment is employed, regardless of the number of sensors used for deployment. The calculated coverage C is more than 85% in Fig. 7, but the actual coverage is well below 50%, because the network is partitioned in two parts. This situation is exactly the case where topology improvement is required.

Fig. 8 shows the node locations and coverage after running the DSSA. The ROI is fully covered after running the algorithm. The parameter values used in this simulation run are:

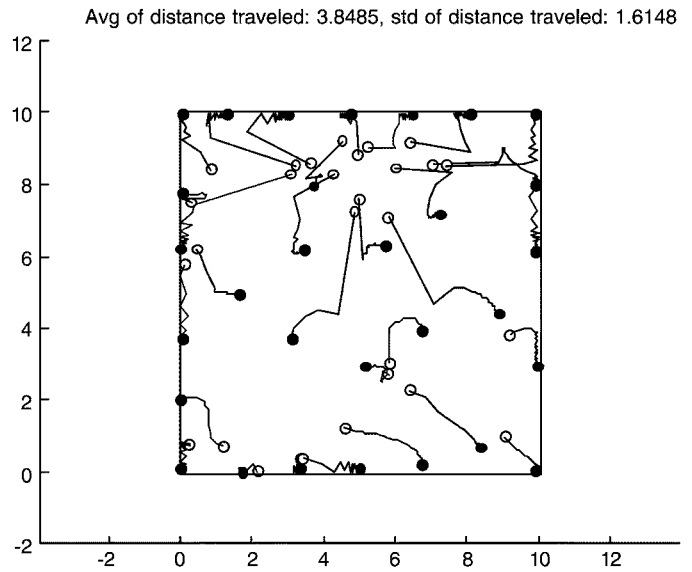


Fig. 9. Sensor-node movements when DSSA is applied.

stable status limit (S_{lim}) = 5, oscillation limit (O_{lim}) = 5, and threshold e for oscillation and stable status = 0.1522. Now, the network is fully connected and also covers the entire ROI. Note that the spatial node distribution is more uniform than the initial random distribution shown in Fig. 7.

Fig. 9 shows the actual paths of individual nodes as they moved from their initial locations to their final locations using DSSA. Blank circles represent the initial locations and filled circles indicate the final locations. For the initial distribution of Fig. 7, each node moves a distance of 3.8485 on average and the standard deviation of distance traveled is 1.6148. 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 full coverage can be expected.

For the purpose of comparison, a simulated annealing-based algorithm (SABA) was also used for topology improvement. The SABA is known as a good solution of many combinatorial optimization problems. To implement a SABA for topology improvement, four main design issues need to be considered. These are: the definition of the neighborhood, move operator, local energy calculation, and annealing schedule. The definition of neighborhood used here is the same as for DSSA, i.e., it is set equal to the cR. This concept of neighborhood is reasonable because each node can only reach the neighboring node using single hop communication. The move operator is chosen to be a random movement within the neighborhood. Local energy calculation is done by adding up the subforces in the neighborhood just like in our DSSA. An exponential cooling schedule is used as the annealing schedule for efficiency as in [13]. In SABA, if the energy of the proposed solution ($E_{proposed}$) is less than that of the old solution (E_{old}), the proposed solution is accepted as the new solution. Otherwise, the proposed solution is accepted with a certain probability p , which is given by $p = \exp(-(E_{proposed} - E_{old})/T)$, where E is the energy and T

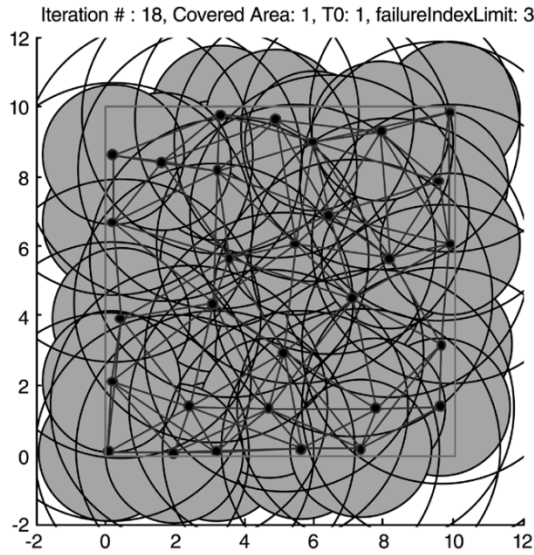


Fig. 10. Final node distribution after running SABA.

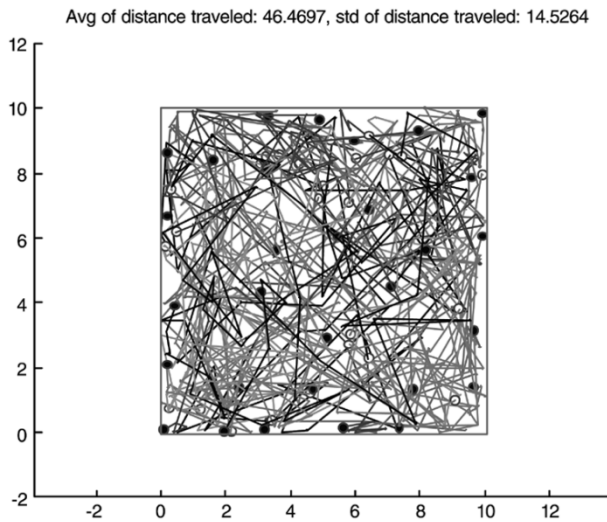


Fig. 11. Node movements when SABA is applied.

is the current temperature. The parameters used are initial temperature $T_0 = 1$ and the stopping criterion is three consecutive failures in achieving the desired acceptance ratio, defined here as the frequency of acceptance to the number of trials. The SA used here is a modified version due to the considerations of time and energy constraints to mobile node deployment as well as the distributed nature of the algorithm. The SABA used in this paper does not necessarily obtain the globally optimal solution. The result after applying SABA is shown in Fig. 10.

Fig. 10 shows that SABA also works well for the initial distribution shown in Fig. 7. The entire area is covered by 30 sensor nodes and these nodes are well spread over the region. Fig. 11 shows how individual nodes move from initial locations to final locations in SABA. For the initial distribution of Fig. 7, each node moved a distance of 46.4697 on an average and the standard deviation of distance traveled is 14.5264. Compared with DSSA, SABA involves more travel distance on an average until convergence and the corresponding energy required is much greater than that of DSSA. Because the standard deviation of

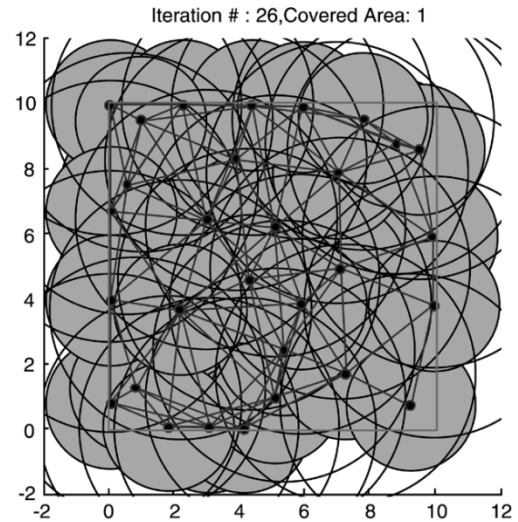


Fig. 12. Final node distribution after running IDCA.

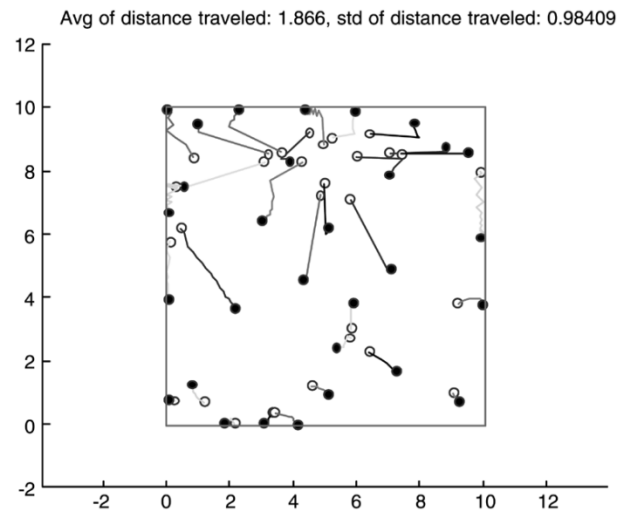


Fig. 13. Sensor node movements when IDCA is applied.

travel distance is also large, the system lifetime with full coverage attained by SABA is expected to be shorter than DSSA.

The result after applying the IDCA is shown in Fig. 12. Fig. 12 shows that IDCA also works well for the initial distribution shown in Fig. 7. The entire area is covered by 30 sensor nodes and these nodes are well spread over the region.

Fig. 13 shows how individual nodes move from their initial locations to final locations in IDCA. For the initial distribution of Fig. 7, each node moved a distance of 1.866 on average and the standard deviation of distance traveled is 0.98409. Compared with DSSA, IDCA involves less travel distance on average until convergence and the corresponding energy required is much less than that of DSSA. Note that the path lengths between starting positions and ending positions in Fig. 13 are shorter than those in Fig. 9. Because the standard deviation of travel distance is also small, the system lifetime with full coverage attained by IDCA is expected to be longer than DSSA.

The result after applying the VDDA can be seen in Fig. 14, which shows that VDDA also works well for the initial distribution shown in Fig. 7. The entire area is covered by 30 sensor nodes and these nodes are well spread over the region.

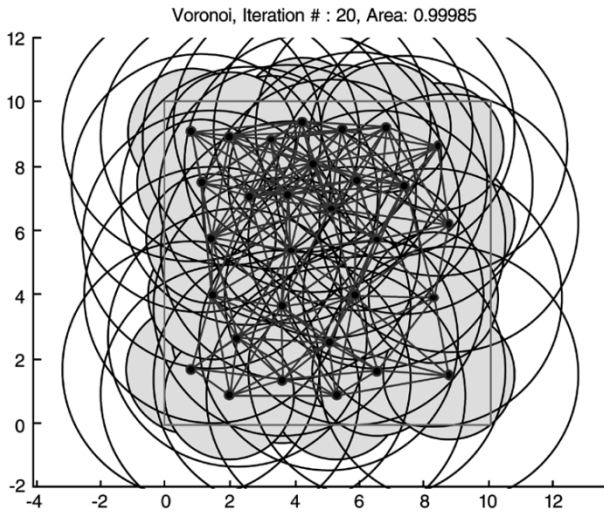


Fig. 14. Final node distribution after running VDDA.

Voronoi: mean of distance travelled = 1.5498, std of distance travelled = 0.67187

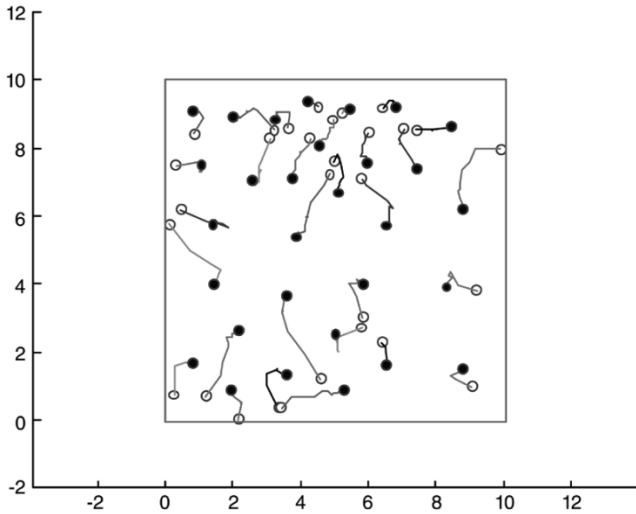


Fig. 15. Sensor node movements when VDDA is applied.

Fig. 15 shows how individual nodes move from their initial locations to final locations in VDDA. For the initial distribution of Fig. 7, each node moved a distance of 1.5498 on an average and the standard deviation of distance traveled is 0.67187. Compared with DSSA and IDCA, VDDA involves less travel distance on average until convergence and the corresponding energy required is much less than those of DSSA and IDCA. Note that the path lengths between starting positions and ending positions in Fig. 15 are shorter than those in Figs. 9 and 13. Because the standard deviation of travel distance is also small, the system lifetime with full coverage attained by VDDA is expected to be longer than DSSA and IDCA.

Next, the performances of DSSA, SABA, IDCA, and VDDA are evaluated in terms of the metrics presented in Section III. Coverage, uniformity, time, and distance until convergence for different algorithms are compared here. Results are presented in Figs. 16–19. 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

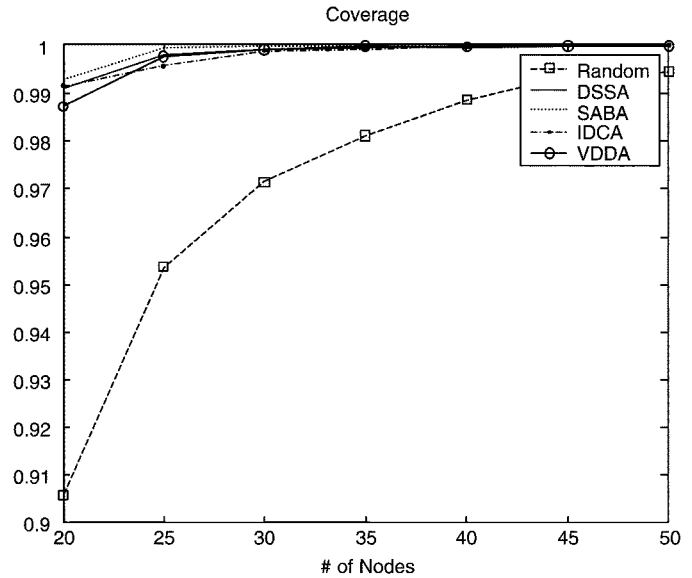


Fig. 16. Coverage versus network size.

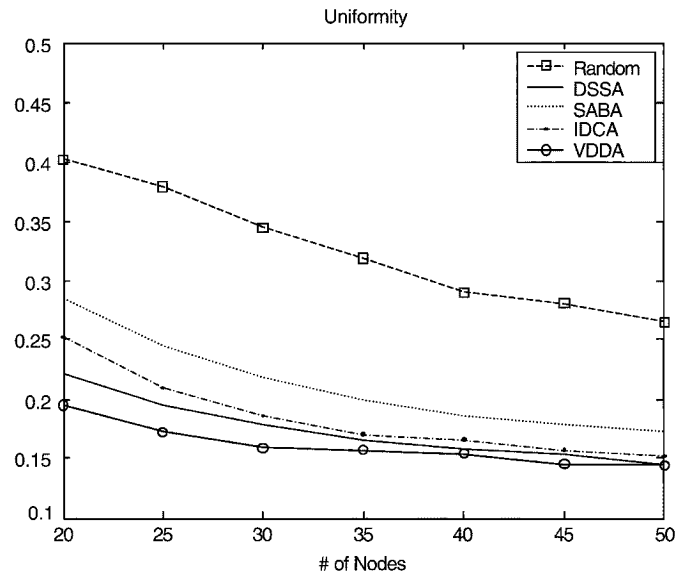


Fig. 17. Uniformity versus network size.

varies from 20 to 50 and results are averaged over 100 runs (initial random distributions) for each node density.

Fig. 16 shows the improvement in coverage area from the initial random deployment for SABA, DSSA, IDCA, and VDDA. All four algorithms exhibit a similar performance over different network sizes. The coverage achieved by all the algorithms increases as the network size goes up. As the number of nodes increases, the improvement in coverage diminishes. Even though the average coverage of random dispersion reaches about 99% at high node density and this number may appear satisfactory for many application requirements, random deployment may not guarantee the intended goal of all the applications. Moreover, even if random deployment can cover 99% of the ROI, there is a possibility of improvement in the uniformity of internodal distance to improve the lifetime of a sensor network. As indicated earlier, the standard deviation of internodal distances is employed as the metric for uniformity of the networks.

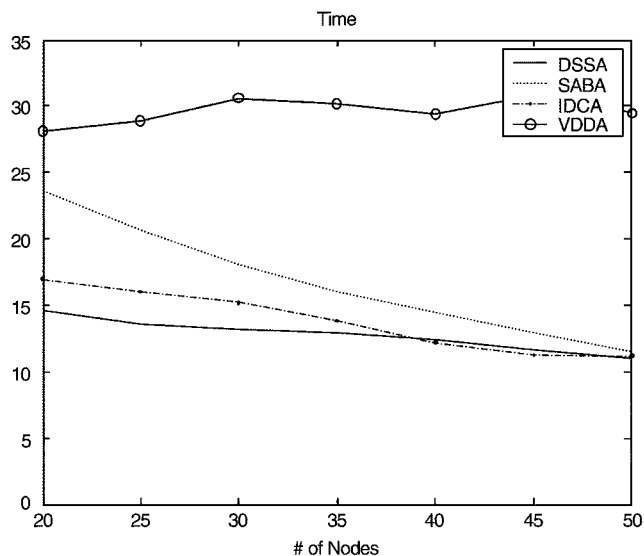


Fig. 18. Termination time versus network size.

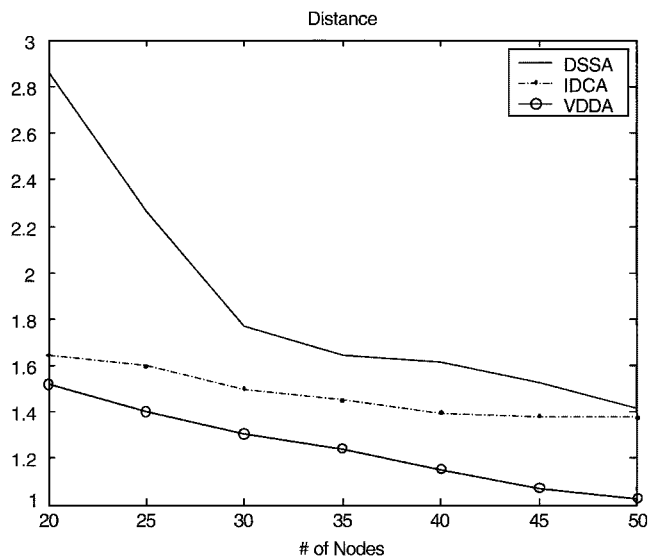


Fig. 20. Distance traveled versus network size (zoomed in).

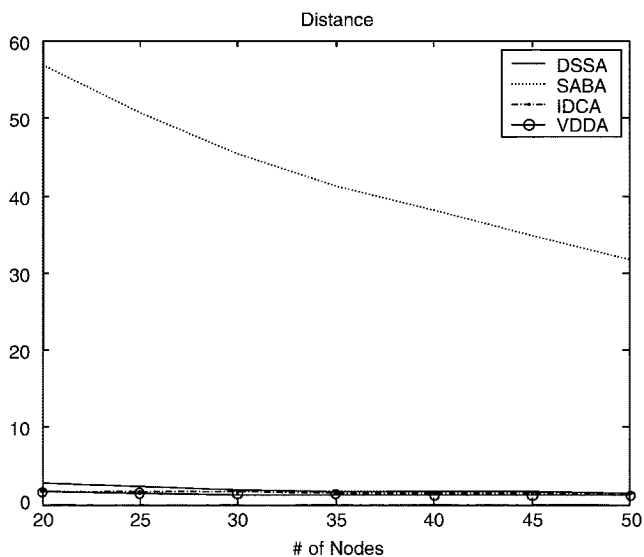


Fig. 19. Distance traveled versus network size.

Fig. 17 shows the reduction in the standard deviation from the initial random deployment case. DSSA, IDCA, and VDDA obtain better uniformity than the initial one and DSSA outperforms IDCA slightly. Though SABA also obtains better uniformity than the initial one, DSSA, IDCA, and VDDA still outperform it. The improvement in uniformity saturates as network density increases.

Fig. 18 shows that IDCA leads to faster deployment than the DSSA at high node densities on average. Termination time is measured in the number of iterations until the algorithms stop. Both DSSA and IDCA outperform SABA on average. Also, the variation in termination times of DSSA and IDCA is less than that of SABA over a wide range of number of nodes. This means that both algorithms are less sensitive to the number of nodes, i.e., network density in terms of termination time for deployment than SABA. VDDA takes much longer to terminate than the other algorithm.

Fig. 19 shows the mean distance traveled to reach the final locations for deployment. DSSA and IDCA require much less travel distance than SABA. The performance of VDDA is the best among the four algorithms in terms of distance traveled. This distance is related to the required energy (fuel) for deployment.

A zoomed-in version of the mean distance traveled for DSSA, IDCA, and VDDA is shown in Fig. 20. VDDA requires less mean distance traveled than DSSA and IDCA, so VDDA saves more energy for the movement of sensors than the other algorithms. Though DSSA and VDDA have similar coverages as seen in Fig. 16, VDDA needs a longer time to converge as seen in Fig. 18, VDDA requires less energy for the movement of nodes than DSSA as shown in Fig. 20, and VDDA can obtain more uniformly distributed node topology than DSSA as seen in Fig. 17. Therefore, VDDA is more energy efficient than DSSA in general. Also, it is observed that the required energy as well as the distance traveled at different node densities is almost constant, especially in IDCA. Thus, the required energy (fuel) is quite insensitive to network density in IDCA. This can make the planning of energy consumption during deployment easier over a wide range of network densities.

As seen in Figs. 16–20, 25–40 nodes are required to attain acceptable performance for the problem considered here. When too few nodes are used, it is unable to obtain full coverage over the ROI. When too many nodes are used, there is not much gain in coverage improvement because of the diminishing marginal gain in terms of coverage, though more uniform distribution still can be obtained. With the number of nodes in this range, the required time to converge is almost the same for all the algorithms. Because the variation in time required to converge and the travel distance is smaller over this range of node densities, it is easier to estimate the required energy for deployment. Extensive experiments on nonhomogeneous nodes, nonrectangular regions, realization of nonuniform target distribution, and robustness of the algorithms are conducted and results are available in [8].

The performance of the algorithms in this paper is not compared with those in [10], [25], and [28] because the assump-

tions are different. In [10], additional equipment is used to detect other nodes and obstacles. Moreover, nodes are initially deployed over a compact region and then are spread out. In [25], it is assumed that there are insufficient robots in a physically bounded region. In [28], it is assumed that global information regarding other nodes is available.

Experiments using different values for sR and cR were conducted and similar performances are observed. The algorithms were simulated in environments that are not rectangular and the performances are similar to those of rectangular regions. These results are not included in this paper due to length considerations. Results are available in [8].

VI. SUMMARY

The deployment problem for mobile WSN is considered in this paper. A ROI needs to be covered by a given number of nodes with limited sensing and cR . A random distribution of nodes over the ROI is assumed as the initial node distribution. Though many scenarios adopt random deployment for practical reasons, such as deployment cost and time, random deployment may not provide a uniform distribution, which is desirable for a longer system lifetime over the ROI. In this paper, a number of distributed algorithms for the deployment of mobile nodes are proposed to improve an irregular initial deployment of nodes. A peer-to-peer algorithm analogous to the equilibrium of molecules and an enhanced intelligent energy-efficient deployment algorithm for cluster-based WSN by a synergistic combination of cluster structuring and peer-to-peer deployment scheme are proposed. A distributed algorithm using VDs based on local computation is also proposed. After employing these algorithms, the ROI is covered by more uniformly distributed nodes. While developing these algorithms, one should consider factors such as density of nodes, memory constraints, localization errors, and scalability of mobile nodes (network size). Through mobility and locationing ability of nodes, these algorithms provide a way to avoid expensive redeployment process. This postdeployment idea is quite useful for many situations, especially when a large fraction of nodes are destroyed or broken during deployment or operation in a hostile situation, or where initial distribution is quite uneven and when human intervention for redeployment is too costly or too risky. The performance of these algorithms is determined by the percentage of region covered, computational/deployment time, the mean distance traveled required for deployment, and uniformity of the networks. Simulation results show that the proposed algorithms successfully obtain a more uniform distribution from initial uneven distributions in an energy-efficient manner.

In this paper, only one-hop neighbors were included while making the decision regarding next nodal movement. However, better solutions in terms of energy efficiency may be found when a wider neighborhood is used. Computation cost, time delay, energy consumption, and tradeoffs between them for different neighborhood sizes will be the main issues to be considered in this direction. This will require the inclusion of multihop neighbors. Also, the relation between cluster-size and neighborhood size will need to be established for energy saving during deployment.

In practice, a WSN is deployed over large regions. The ROI can be divided into multiple sub-ROIs for easy deployment, organization, and management. Hierarchical deployment schemes may be investigated to handle the scalability issue in WSNs. The determination of the size of sub-ROI and their corresponding density and edge effects due to the division of the ROI are worth pursuing. The effect of uncertainty in sensor-node locations on the performance of our algorithms is another area for further investigation.

REFERENCES

- [1] F. Aurenhammer, "Voronoi diagrams—A survey of a fundamental geometric data structure," in *ACM Comput. Survey*, vol. 23, 1991, pp. 345–405.
- [2] E. Biagioni and G. Sasaki, "Wireless sensor placement for reliable and efficient data collection," in *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan. 2003, p. 127b.
- [3] N. Bulusu, J. Heidemann, and D. Estrin, "Adaptive beacon placement," in *Proc. 21st Int. Conf. Distributed Comput. Syst.*, Apr. 2001, pp. 489–498.
- [4] —, "GPS-less low-cost outdoor localization for very small devices," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 28–34, Oct. 2000.
- [5] Y. U. Cao, A. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, pp. 1–23, 1997.
- [6] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2003, pp. 1609–1614.
- [7] D. W. Gage, "Command control for many-robot systems," presented at the 19th Annu. AUVS Tech. Symp, Huntsville, AL, June 22–24, 1992.
- [8] N. Heo, "Distributed deployment algorithms for mobile wireless sensor networks," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Syracuse Univ., Syracuse, NY, 2004.
- [9] 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, 2002.
- [10] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. 6th Int. Conf. Distributed Autonomous Robotic Syst.*, Fukuoka, Japan, 2002, pp. 299–308.
- [11] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Netw.*, vol. 7, no. 4, pp. 343–358, 2001.
- [12] V. Kawadia and P. R. Kumar, "Power control and clustering in ad hoc networks," in *Proc. IEEE INFOCOM Conf.*, 2003, pp. 459–469.
- [13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [14] S. Kumar, F. Zhao, and D. Shepherd, "Collaborative signal and information processing in microsensors networks," *IEEE Signal Process. Mag.*, vol. 19, no. 2, pp. 13–14, Mar. 2002.
- [15] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Sel. Areas Commun.*, vol. 15, no. 7, pp. 1265–1275, Sep. 1997.
- [16] L. Loo, E. Lin, M. Kam, and P. K. Varshney, "Cooperative multi-agent constellation formation under sensing and communication constraints," in *Cooperative Control and Optimization*. Norwell, MA: Kluwer, 2002, pp. 143–170.
- [17] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad hoc sensor networks," in *Proc. IEEE INFOCOM Conf.*, 2001, pp. 1380–1387.
- [18] R. Min *et al.*, "Energy-centric enabling technologies for wireless sensor networks," *IEEE Wireless Commun.*, vol. 9, no. 4, pp. 28–39, Aug. 2002.
- [19] J. O'Rourke, *Art Gallery Theorem and Algorithms*. New York, NY: Oxford University Press, 1987.
- [20] G. J. Pottie and W. J. Kaiser, "Embedding the internet: Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000.
- [21] H. Qi, S. S. Iyengar, and K. Chakrabarty, "Distributed sensor fusion—a review of recent research," *J. Franklin Inst.*, vol. 338, pp. 655–668, 2001.
- [22] L. Schwiebert, S. K. S. Gupta, and J. Weinmann, "Research challenges in wireless networks of biomedical sensors," in *Proc. ACM/IEEE Conf. Mobile Comput. Netw.*, 2001, pp. 151–165.
- [23] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 2, 2001, pp. 472–476.

- [24] K. Sohrabi, B. Manriquez, and G. Pottie, "Near-ground wideband channel measurements," in *Proc. 49th Veh. Technol. Conf.*, 1999, pp. 571–574.
- [25] A. F. T. Winfield, "Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots," in *Distributed Autonomous Robotic Systems 4*, L. E. Parker, G. Bekey, and J. Barhen, Eds. New York: Springer-Verlag, 2000, pp. 273–282.
- [26] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proc. ACM/IEEE Int. Conf. Mobile Comput. Netw.*, July 2001, pp. 70–84.
- [27] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. IEEE INFOCOM Conf.*, vol. 3, 2002, pp. 1567–1576.
- [28] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. IEEE INFOCOM Conf.*, vol. 2, 2003, pp. 1293–1303.
- [29] Voronoi [Online]. Available: <http://www.voronoi.com/>
- [30] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Algorithms for energy-efficient multicasting in static ad hoc wireless networks," *Mobile Netw. Applicat.*, vol. 6, no. 3, pp. 251–263, 2001.
- [31] D. W. Gage, "Command control for many-robot systems," *Unmanned Syst. Mag.*, vol. 10, no. 4, pp. 28–34, 1992.



Nojeong Heo received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 1996 and the M.S. and Ph.D. degrees in electrical engineering from Syracuse University, Syracuse, NY, in 1999 and 2004, respectively.

He is currently a Senior Engineer at Samsung Electronics Co., Ltd. His research interests include wireless communication, ad hoc networks, sensor networks, and next-generation mobile networks.



Pramod K. Varshney (S'72–M'77–SM'82–F'97) was born in Allahabad, India, on July 1, 1952. He received the B.S. degree in electrical engineering and computer science (with highest honors) and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois, Urbana, in 1972, 1974, and 1976, respectively.

He is currently Research Director of The New York State Center for Advanced Technology in Computer Applications and Software Engineering (CASE). During 1972–1976, he held teaching and research assistantships at the University of Illinois. Since 1976, he has been with the Department of Electrical and Computer Engineering, Syracuse University, Syracuse, NY, where he is currently Professor of electrical engineering and computer science. He has served as the Associate Chairman of the department from 1993 to 1996. His current research interests are in distributed sensor networks and data fusion, detection and estimation theory, wireless communications, image processing, remote sensing, radar signal processing, and parallel algorithms. He has supervised 34 Ph.D. dissertations, authored or coauthored over 85 journal papers and over 250 conference papers. He is the author of *Distributed Detection and Data Fusion* (New York: Springer-Verlag, 1997). He has consulted for General Electric, Hughes, Booz-Allen and Hamilton, SCEE, Kaman Sciences Corp., Andro Computing Solutions, ITT, and Digicomp Research.

Dr. Varshney is a member of Tau Beta Pi and is the recipient of the 1981 ASEE Dow Outstanding Young Faculty Award. He was elected to the grade of Fellow of the IEEE in 1997 for his contributions in the area of distributed detection and data fusion. In 2000, he received the Third Millennium Medal from the IEEE and Chancellor's Citation for Exceptional Academic Achievement at Syracuse University. He was the Guest Editor of the Special Issue on Data Fusion of the PROCEEDINGS OF THE IEEE, January 1997. He is on the editorial board of *Cluster Computing Information Fusion*. He is a Distinguished Lecturer for the IEEE AES Society. He was the President of the International Society of Information Fusion in 2001. While at the University of Illinois, he was a James Scholar, a Bronze Tablet Senior, and a Fellow.