International Journal of Wireless Information Networks, Vol. 14, No. 3, September 2007 (© 2007)
DOI: 10.1007/s10776-007-0060-7

Distributed Sensor Networks Deployment Using Fuzzy Logic Systems

Haining Shu[1,3], Qilian Liang[1], and Jean Gao[2]

The effectiveness of distributed wireless sensor networks highly depends on the sensor deployment scheme. Given a finite number of sensors, optimizing the sensor deployment will provide sufficient sensor coverage and ameliorate the quality of communications. In this paper, we apply fuzzy logic systems to optimize the sensor placement after an initial random deployment. We use the outage probability due to co-channel interference to evaluate the communication quality. Fenton–Wilkinson method is applied to approximate the sum of log-normal random variables. Our algorithm is compared against the existing distributed self-spreading algorithm. Simulation results show that our approach achieves faster and stabler deployment and maximizes the sensor coverage with less energy consumption. Outage probability, as a measure of communication quality gets effectively decreased in our algorithm but it was not taken into consideration in the distributed self-spreading algorithm.

KEY WORDS: wireless sensor networks; sensor deployment; fuzzy logic system; Fenton–Wilkson method; outage probability

1. INTRODUCTION

Wireless sensor networks consist of certain amount of small and energy constrained nodes. Sensor nodes are deployed in support of various missions such as environment and habitat monitoring, industrial process control, infrastructure security [1] and automation in the transportation. One networked sensing experiment on Great Duck Island [2] provides a small lens into an expansive future of such applications. The experiment was conducted by a team of computer engineers from the University of California, Berkeley. Up to date, 190 wireless sensors have been deployed on a small island 10 miles off the coast of Maine to study the nesting behaviors of petrels. Biologists are now monitoring the petrels on the island from their offices, browsing data from sensors linked by satellite.

The deployment of sensors varies with different applications. A number of applications require the placement of sensors at desired locations like data collection [3] and infrastructure security [1], where critical area, buildings and facilities are monitored by a network of sensors placed adequately. For such placement-friendly applications, sufficient knowledge of the environment is assumed to be available before deployment is carried out.

In other applications where prior knowledge of the environment can not be obtained, sensors may have to be randomly air-dropped and human intervention after deployment to recharge or replace node batteries may not be feasible. Mobile sensors are practically desirable in this situation because they have the capability to move around and re-adjust their positions for high quality communication and better coverage and surveillance [4]. However mobile sensor deployment itself is an energy consuming process because of the motion and communication between sensors. An efficient sensor re-deployment scheme is a necessity to save energy resources and improve the quality of communications.

[1] Electrical Engineering, University of Texas at Arlington, Arlington, TX, 76019, USA
[2] Computer Science and Engineering, University of Texas at Arlington, Arlington, TX, 76019, USA
[3] E-mail: shu@wcn.uta.edu

Some prior research proposed sensor deployment strategies based on virtual forces for target localization [5–7]. One example of virtual force concept was presented in [7]. The pair-wise interaction between sensor nodes is governed by two kinds of virtual forces – one causes the nodes to repel each other to improve their coverage and the other is an attractive force that prevents the nodes from losing connectivity. Later Cheng et al. [8] formulated a constrained multivariable nonlinear programming problem to determine both the locations of the sensor nodes and data transmission pattern. In [9, 10], Heo and Varshney developed a distributed self-spreading algorithm (DSSA) and an intelligent deployment and clustering algorithm (IDCA) for sensor deployment. Recently, a voronoi diagram (VD)-based deployment algorithm was included in [11]. All the above algorithms have made lots of efforts to formulate the virtual forces, however none of which can well handle the uncertainties with the random move and unpredictable oscillation in sensor deployment. For the purpose of stability and convergence, various parameters or constraints such as oscillation limit, stable status [9–11], and number of neighbors [7] have to be imposed to avoid excessive sensor oscillation.

In this paper, we apply fuzzy logic systems (FLSs) to handle these uncertainties in distributed sensor deployment. Instead of attempting to formulate the virtual forces, we propose to apply FLSs to re-deploy the sensors. Each individual mobile sensor uses a FLS to self-adjust its location. For a single sensor node, neighboring nodes' location is the only information needed to make the movement decision. Therefore the deployment scheme based on FLSs is a fully distributed approach. After applying FLSs, exhaustive move and unpredictable oscillation is efficiently avoided and fast deployment is achieved. As a result, the entire sensor network survives for longer lifetime and the quality of communication in terms of outage probability is greatly ameliorated. A concept of coherence time is introduced for the purpose of synchronization among sensors.

The rest of this paper is organized as follows. In Section 2, we briefly review the basic concept of FLSs. Section 3 details the FLSs design for distributed sensor deployment. Simulation and discussion are presented in Section 4. Section 5 concludes this paper with a summary. Fenton–Wilkinson method to tackle the outage problem is expatiated in Appendix.

2. OVERVIEW OF FLSS

Figure 1 shows the structure of a rule-based type-1 FLS [12]. It contains four components: fuzzifier, rules, inference engine and defuzzifier. When an input is applied to a FLS, the inference engine computes the output set corresponding to each rule. The defuzzifer then computes a crisp output from these rule output sets.

Rules are the heart of a FLS and may be provided by experts or can be extracted from numerical data. In either case, the rules that we are interested in can expressed as a collection of IF-THEN statements, e.g. [13]

IF the total average input rate of real-time voice and video traffic is a *moderate amount*, and the total average input rate of the non-real-time data traffic is *some*, THEN the confidence of accepting the telephone call is a *large amount*.

The IF-part of a rule is its *antecedent* and the THEN-part of a rule is its *consequent*.

The process of making a crisp input fuzzy is called *fuzzification*. The most widely used

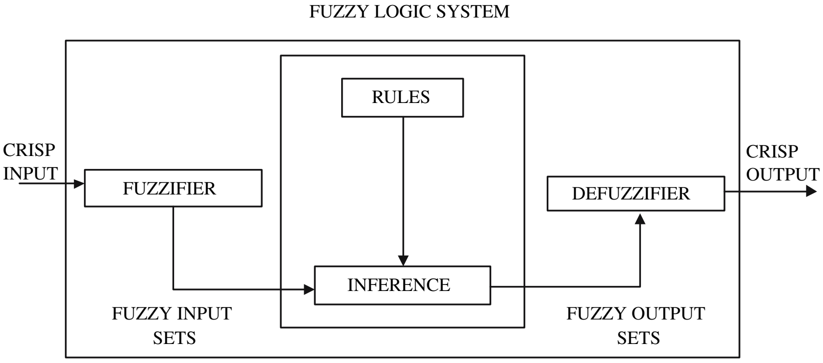

Fig. 1. The structure of a fuzzy logic system.

fuzzification is the *singleton fuzzification*. All fuzziness for a particular fuzzy set is essentially characterized by the *membership functions* (MFs). The shapes used to describe the fuzziness have very few restrictions but with the help of mathematical structure, some standard terms related to the shape of MFs have been developed over the years [14]. The most common forms of MFs are those that are normal and convex.

Consider a type-1 FLS having p inputs and one output. Let us suppose that it has M rules, there the lth rule has the form:

R^l: IF x_1 is F_1^l *and* x_2 is F_2^l *and* $\cdots$ *and* x_p is F_p^l, *THEN* y is G^l. $l = 1, \ldots, M$

Assuming singleton fuzzification is used, when an input $x' = \{x'_1, \ldots, x'_p\}$ is applied, the degree of firing corresponding to the lth rule is computed as

$$\mu_{F_1^l}(x'_1) \star \mu_{F_2^l}(x'_2) \star \cdots \star \mu_{F_p^l}(x'_p) = \mathcal{T}_{i=1}^{p} \mu_{F_i^l}(x'_i) \quad (1)$$

where $\star$ and $\mathcal{T}$ both indicate the chosen t-norm.

The last but not the least process in a FLS is called *defuzzification*. Defuzzification is the conversion of fuzzy output sets to crisp output sets. There are many defuzzification methods including maximum, mean-of-maxima, centroid, center-of-sums, height, modified height and center-of-sets. In this paper, we focus, for illustrative purposes, on the center-of-sets defuzzifier [13]. It computes a crisp output for the FLS by first computing the centroid, c_{G^l}, of every consequent set G^l, and, then computing a weighted average of these centroids. The weight corresponding to the lth rule consequent centroid is the degree of firing associated with the lth rule, $\mathcal{T}_{i=1}^{p} \mu_{F_i^l}(x'_i)$, so that

$$y_{cos}(x') = \frac{\sum_{l=1}^{M} c_G^l \mathcal{T}_{i=1}^{p} \mu_{F_i^l}(x'_i)}{\sum_{l=1}^{M} \mathcal{T}_{\rangle=\infty}{}^{p} \mu_{F_i^l}(x'_i)} \quad (2)$$

where M is the number of rules in the FLS.

In the next section, we will detail the design of the rule-based type-1 FLSs for distributed sensor deployment issue.

3. DESIGN OF FLSS FOR DISTRIBUTED SENSOR DEPLOYMENT

3.1. Assumptions and Notations

In this research, we make several assumptions:

- Sensor field is denoted by a two-dimensional grid. Sensing and communication is modeled as a circle on this grid.
- Coverage discussed in this paper is grid coverage. A grid point is covered when at least one sensor covers this point.
- A sensor can detect or sense any event within its sensing range, denoted by *Rs*. Coverage is determined based on Rs.
- Two sensors within their communication range, denoted by Rc can communicate with each other. Neighbors of a sensor are defined as nodes within its communication range.
- All sensor nodes are assumed peer to peer.
- Sensor nodes have certain mobility and are capable of computing, detection and communication.
- Sensor node can obtain the knowledge of its location.
- Sensors are synchronized by coherence time. One-time move is made within each coherence period.

3.2. FLSs Design for Distributed Sensor Deployment

FLS is well known to be able to handle uncertainty and ambiguity. Practically not all uncertainty is random. Some forms of uncertainty are non-random and hence not suited to treatment or modeling by probability theory. Fuzzy set theory is a marvelous tool for modeling uncertainty associated with vagueness, or with a lack of information regarding a particular element of the problem at hand. Upon concerning the distributed sensor deployment, the moving distance and direction of each sensor are distributed and full of uncertainty which can barely be described by some random distribution. FLS is well known as model free. Their MFs are not based on statistical distributions. Therefore we propose to apply FLS to the distributed sensor deployment problem. Each sensor makes fully distributed decision on its movement based on FLS.

Our algorithm starts with random deployment. Assume a two-dimensional sensor field is the target area of surveillance. In the initial condition, a given number of sensors are randomly deployed such as air-dropping. Because of the randomness in initial deployment, very likely the sensor field will not be fully covered. Part of the sensor field might be over crowded with the sensors. Such unbalanced deployment brings difficulty in target detection and tracking, and increases the interference during communications. Figure 2 gives an example of randomly deployed field. As shown in Figure 2, targets in the

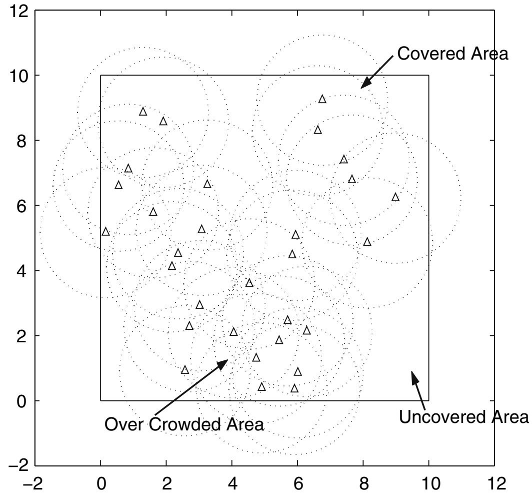

Fig. 2. An example of random deployment: targets in the uncovered area can not be detected; In the over crowded area communication between sensors has lots of interference.

uncovered area can not be detected while in the over crowded area, communication between sensors is corrupted by the interference from neighboring nodes.

Our algorithm then intends to re-deploy the sensors such that maximum field coverage and high quality communication could be achieved. Each individual sensor in the network needs to fine-tune its location such that densely deployed sensors can be evenly spreaded in the field. Two critical procedures are considered in our algorithm:

- Determine the next-step move distance for each sensor.
- Determine the next-step move direction for each sensor.

The next-step move distance is hard to determine. Too small or big move distance each step consumes the network more time and energy to get stable deployment. Excessive move and oscillation is unavoidable in previous work with no fuzzy system. In this paper, we design a FLS to determine the next-step move distance for each sensor.

An ideal sensor deployment will have uniform distribution for better coverage. But in random deployment, coverage uniformity is hardly to achieve initially. In sensor network composed of mobile sensors, each sensor detects the number and location of its neighbors and decides its *neighborhood density*. If the sensor has a high density of neighboring nodes, it makes decision using FLSs to shift a certain distance away from the high density area. If the neighborhood density is low, the sensor might stand still or shift a little distance away from the current location.

As illustrated in Figure 2, the *neighborhood density* of a sensor node is determined by two factors: the number of neighbors and the distance between sensor node and its neighbors. The more the neighboring nodes, the higher the neighborhood density. The closer the neighboring nodes, the higher the neighborhood density. Based on this knowledge, we choose two antecedents as follows:

Antecedent 1. Number of neighbors of each sensor.
Antecedent 2. Average Euclidean distance between sensor node and its neighbors.

The linguistic variables to represent the number of neighbors for each sensor are divided into three levels: *high*, *moderate* and *low*; and those to represent the average Euclidean distance between sensor node and its neighbors are divided into three levels: *far*, *moderate* and *near*. The consequent – the shift distance normalized by sensing range *Rs* is divided into three levels: *far*, *moderate* and *near*. Table 1 summaries the rules and consequents.

One example of rules is as follows:

IF the number of neighbors of sensor *i* is *high* and average Euclidean distance between sensor *i* and its neighbors is *moderate*, THEN the normalized shift distance of sensor *i* should be *moderate*.

We setup 9 rules for this FLS because every antecedent has 3 fuzzy sub-sets and there are 2 antecedents. Trapezoidal MFs are used to represent *high*, *low*, *far* and *near* and triangle MFs to represent *moderate*. Two antecedents are normalized to the range [0,10]. We show these MFs in Figures 2 and 3.

Table 1. Fuzzy rules and consequent

Antecedent 1	Antecedent 2	Consequent
Low	Near	Moderate
Low	Moderate	Near
Low	Far	Near
Moderate	Near	Far
Moderate	Moderate	Moderate
Moderate	Far	Near
High	Near	Far
High	Moderate	Moderate
High	Far	Moderate

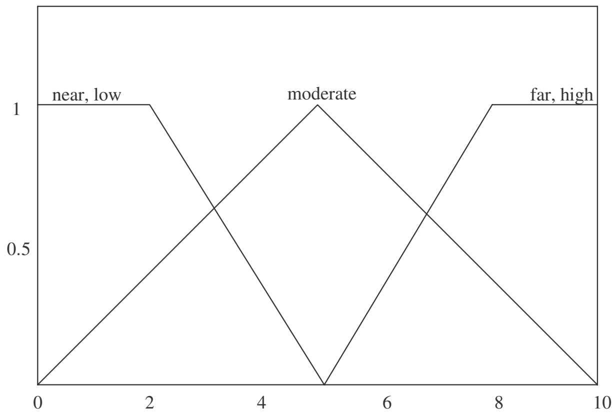

Fig. 3. Antecedent membership functions.

Applying center-of-sets defuzzification [7], for every input (x_1, x_2), the output is computed using

$$y_{(x_1,x_2)} = \frac{\sum_{l=1}^{9} c_G^l \mu_{F_1^l}(x_1)\mu_{F_2^l(x_2)}}{\sum_{l=1}^{9} \mu_{F_1^l}(x_1)\mu_{F_2^l}(x_2)} \qquad (3)$$

Repeating these calculations for $\forall x_i \in [0, 10]$,we obtain a decision surface $y(x_1, x_2)$ as shown in Figure 4.

Generally, the decision surface is time-varying and nonlinear. From Figure 4, we can see that although the number of neighbors for a particular sensor is high, the move distance can be smaller than some sensor with fewer "crowded" neighbors, i.e. very close average Euclidean distance between the sensor and its neighbors. With the assist of decision surface, the next-step move distance can be determined.

Comparing to move distance, the next-step move direction is much easier to decide. Coulomb's law in physics becomes a useful tool to tackle the problem. For instance, assume sensor *i* has 2 neighbors in its communication range as shown in Figure 5.

The coordinate of sensor *i* is denoted as $C_i = (X_i, Y_i)$.

The next-step move direction of sensor *i* could be represented as follows:

$$\vec{v} = \sum_{j=1}^{2} \frac{\vec{C}_j - \vec{C}_i}{|\vec{C}_j - \vec{C}_i|^2} \qquad (4)$$

$$tan(\alpha) = \frac{Y_{(\vec{v})}}{X_{(\vec{v})}} \qquad (5)$$

After getting distance and direction (angle α) , sensor *i* clearly knows his next-step move information. In order to prolong the battery life of each individual sensor, we introduce a coherence time as the duty cycle during which the changes of two antecedents can be ignored. Sensors are put into idle or sleep mode if within the coherence time, the information of neighbors remains unchanged.

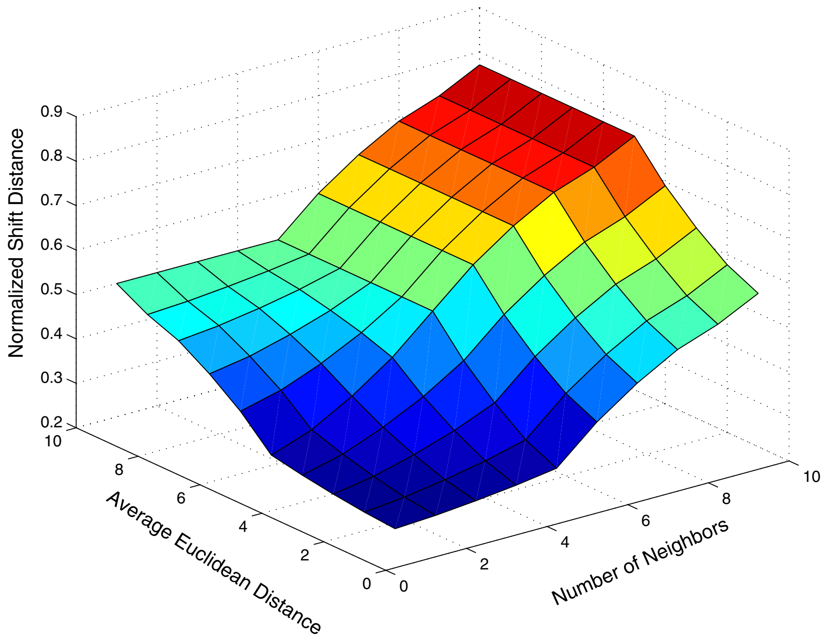

Fig. 4. Control surface of shift distance.

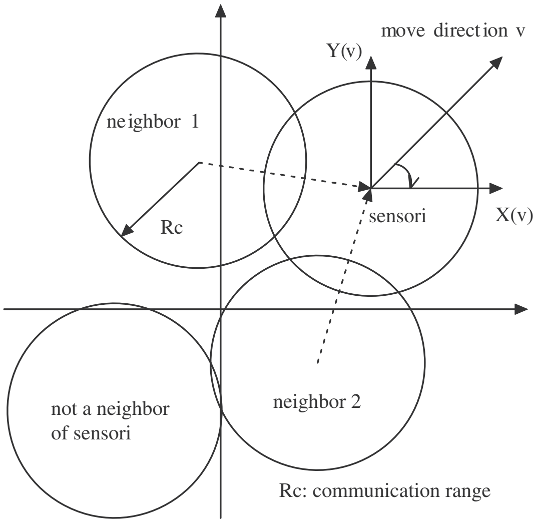

Fig. 5. Example of next step move direction for sensor having two neighbors.

4. SIMULATION AND DISCUSSION

We investigate various number of sensors deployed in a field of 250 × 250 m^2 area. We assume each sensor is equipped with an omni-directional antenna to carry out the task of detection and communication. Evaluation of our scheme follows three criteria: field coverage, converging speed, mean travel distance per node and outage probability. Results are averaged over 200 Monte Carlo simulations.

We compare the performance of our algorithm with the Distributed Self Spreading Algorithm (DSSA) proposed in [9]. DSSA is known as a good solution in the self-deployment of mobile sensor nodes. The main idea of DSSA is to define a partial force for the movement of sensors during the deployment process. The force a node receives from a closer neighbor node is greater than that from a farther neighbor. For N sensor nodes deployed in a square field with area A, DSSA formulates the partial force sensor node i receives from neighbor node j as:

$$f_n^{i,j} = \frac{D}{\mu^2}(R_c - |p_n^i - p_n^j|)\frac{p_n^i - p_n^j}{|p_n^i - p_n^j|} \tag{6}$$

where R_c stands for communication range, $\mu = N \cdot \pi \cdot R_c^2/A$ is called the expected density while D is the local density, and p_n^i stands for the location of node i at time step n. Each node makes decision to move by adding up all partial forces from its neighboring nodes. DSSA sets up two criteria: stable status limit ($S_{\lim}$) and oscillation limit ($O_{\lim}$) to stop a sensor's movement.

Figure 6 shows at 50 m sensing range (R_s = 50 m) and 100 m communication range (R_c = 100 m), the coverage of the initial random deployment, the coverage after DSSA is implemented and the coverage after using FLSs. We ran three iterations for all three schemes. When 20 sensors are deployed, the coverage after random deployment was initially around 85% and the DSSA increased it to 93%. After FLSs were used, the coverage reached approximate 98% after three iterations.

Figure 7 gives the results when 10 iterations are completed for the three deployment schemes. We observe that the performance of DSSA gets closer to our FLSs after 10 iterations. Both FLSs and DSSA can dramatically increase the network coverage in the low density network. Figures 6 and 7 also indicates that instead of deploying large amount of sensors, the desired field coverage could be achieved with fewer sensors. Comparing Figures 6 and 7, we noticed that our FLSs increases the network coverage faster than DSSA in terms of iteration times.

Figures 8–10 are the real pictures of 20 sensors from random deployment, after implementing FLS and DSSA respectively. Both FLS and DSSA can spread the densely deployed sensors but the deployment after using FLS demonstrates more uniformity than the one using DSSA.

We then simulated two cases when 30 sensor nodes and 60 sensor nodes are deployed respectively. Network coverage according to these two cases are presented in Figurs 11 and 12.

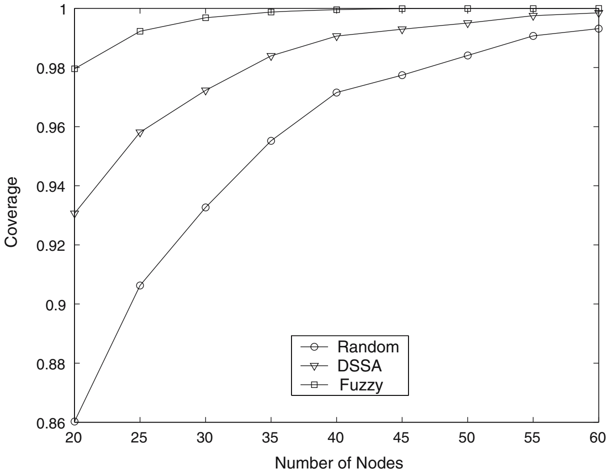

Fig. 6. Coverage vs. number of sensors deployed (after three iterations, Rc = 100 m).

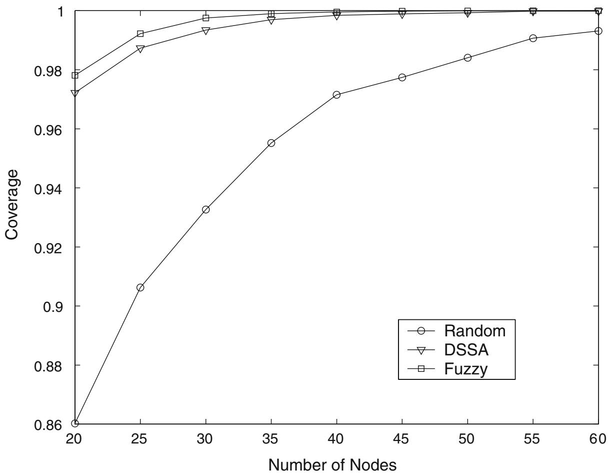

Fig. 7. Coverage vs. number of sensors deployed (after 10 iterations, Rc = 100 m).

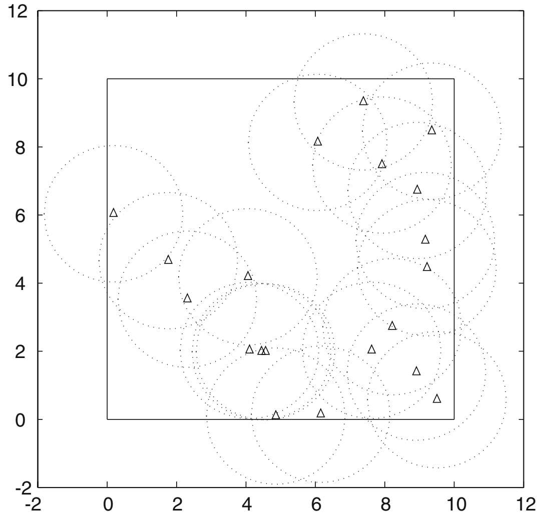

Fig. 8. Random deployment with 20 sensors (Rc = 100 m).

It is fairly clear in Figures 11 and 12 that our FLSs increase the network coverage much faster than the DSSA. For instance, when 30 sensor nodes were deployed, our FLSs boost the network coverage from initial 93% to around 98.5% in only 1 iteration whereas the DSSA takes 6 iterations to reach the same coverage.

The average distance traveled by each sensor node is also important in energy saving problem. For energy constrained wireless sensor nodes, less travel distance leads to less energy consumption. Our goal is to adjust sensors' positions appropriately such that the maximum coverage is achieved with minimum energy dissipation in deployment. We calculated the average distance traveled by each sensor node for our FLS and compared it against the DSSA as both reach the same network coverage. Results in Figure 13 indicate that for the FLS scheme, each sensor node travels less average distance than that in DSSA. Furthermore, in FLS scheme, the average travel distance by each node varies little when the number of sensors changes which implies that the energy consumed in deployment is nearly independent of network density.

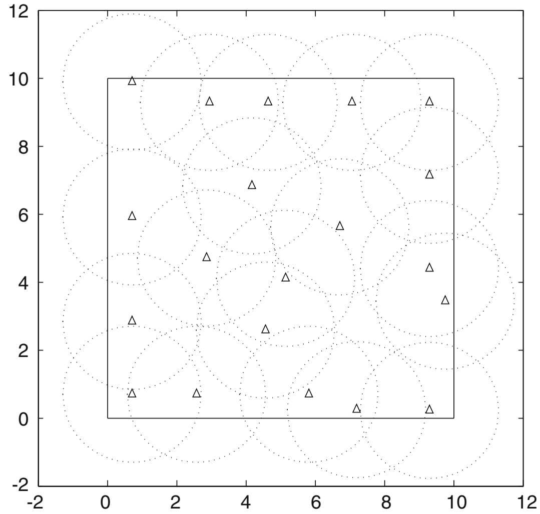

Fig. 9. Deployment with 20 sensors after implementing FLS (after three iterations, Rc = 100 m).

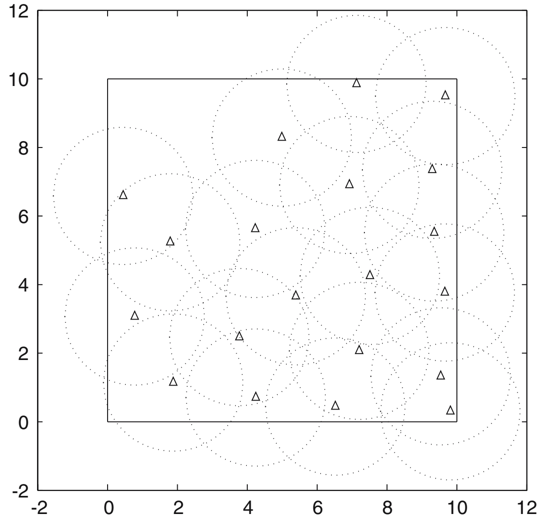

Fig. 10. Deployment with 20 sensors after implementing DSSA (after three iterations, Rc = 100 m).

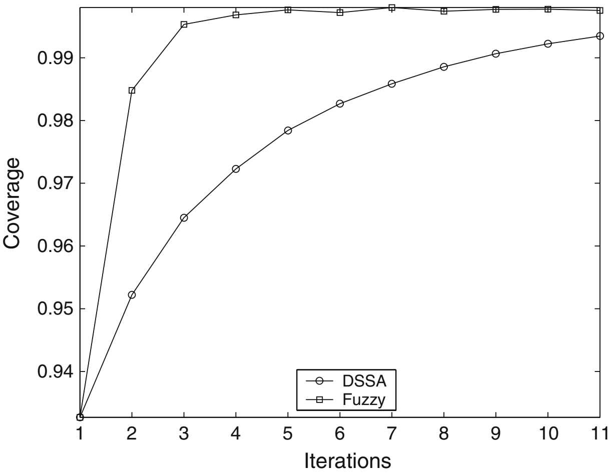

Fig. 11. Coverage vs. number of iterations (30 nodes deployed, Rc = 100 m).

In wireless sensor networks, the radio link performance is usually limited by interference rather than noise, therefore, the probability of outage due to co-channel interference is of primary concern. Measurements [16] have shown that at any value of $d_{i,j}$ (the Euclidean distance between sensor i and sensor j), the path loss $PL(d_{i,j})$ is random and distributed log-normally (normal in dB) about the mean distance dependent value. That is:

$$PL(d_{i,j})[\mathrm{dB}] = \overline{PL}(d_{i,j}) + X_\sigma = \overline{PL}(d_0) + 10n\log\left(\frac{d_{i,j}}{d_0}\right) + X_\sigma \tag{7}$$

and

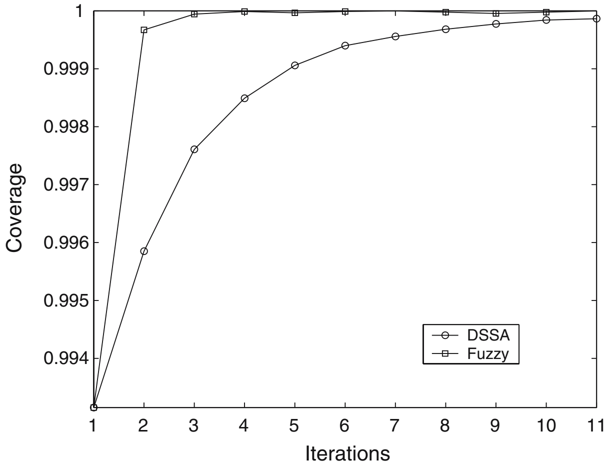

Fig. 12. Coverage vs. number of iterations (60 nodes deployed, Rc = 100 m).

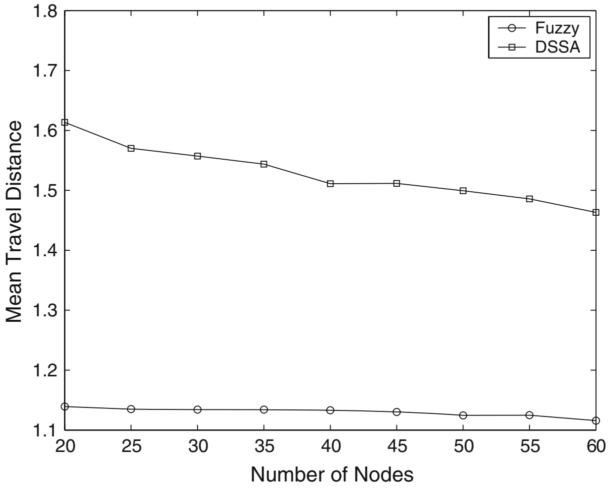

Fig. 13. Travel distance vs. number of sensors deployed (10 iterations).

$$P_r(d_{i,j})[\mathrm{dB}m] = P_t[\mathrm{dB}m] - PL(d_{i,j})[\mathrm{dB}] \tag{8}$$

where X_σ is a zero-mean Gaussian distribution random variable (in dB) with standard deviation σ (also in dB).

The log-normal distribution describes the random *shadowing* effects on the propagation path which implies that measured signal levels at certain distance have a Gaussian (normal) distribution about the distance-dependent mean and standard deviation σ. Since $PL(d_{i,j})$ follows normal distribution, so is $P_r(d_{i,j})$, and the Q function may be used to determine the probability that the received signal level will exceed (or fall below) a particular level.

The probability that the received signal level will exceed a certain value γ can be calculated from the cumulative density function as

$$P_r[P_r(d_{i,j}) > \gamma] = Q\left(\frac{\gamma - \overline{P_r}(d_{i,j})}{\sigma}\right) \tag{9}$$

For sensor i with N neighbors, if sensor i acts as the destination node during one communication, the signal to interference ratio (SIR) is represented as:

$$\mathrm{SIR}(i) = \frac{P_r(d_{i,j})}{\sum_{k=1}^{N} P_r(d_{i,k})}, k \neq j \tag{10}$$

The denominator denoting the effect of co-channel interference is a sum of $N-1$ log-normal signals. Evaluating the outage probability requires the probability distribution of the interference power. There is no known exact expression for the probability distribution for the sum of log-normal random variables, but various authors have derived several approaches which approximate the sum of log-normal random variables by another log-normal random variable.

In this paper, we used Fenton–Wilkinson method [17]. The co-channel interference can now be approximated by one log-normal random variable. SIR(in dB) as a result follows log-normal distribution as well. We expatiate the Fenton–Wilkinson method in the Appendix. Results of outage probability are presented in Figure 14.

Observe Figure 14, the FLSs scheme successfully reduced the outage probability by nearly 15% comparing to DSSA when the number of sensors is 60, which implies a higher probability that the received signal level will exceed the SIR threshold using our FLSs scheme. The DSSA did not perform well considering the outage probability because it did not take the outage probability into performance evaluation [9].

We have introduced earlier that DSSA stops a sensor's movement by two criteria: stable status limit ($S_{\lim}$) and oscillation limit ($O_{\lim}$). Ref. [9] shows that it takes more than 10 times iteration to termination. Our fuzzy approach gains a distinct advantage over DSSA by converging in around three iterations. Thus stop criteria is not required in our fuzzy approach. These facts indicate that our FLS scheme is much faster and simpler to implement comparing to DSSA and more significantly, the FLS scheme maximizes the network coverage with less energy consumption in deployment.

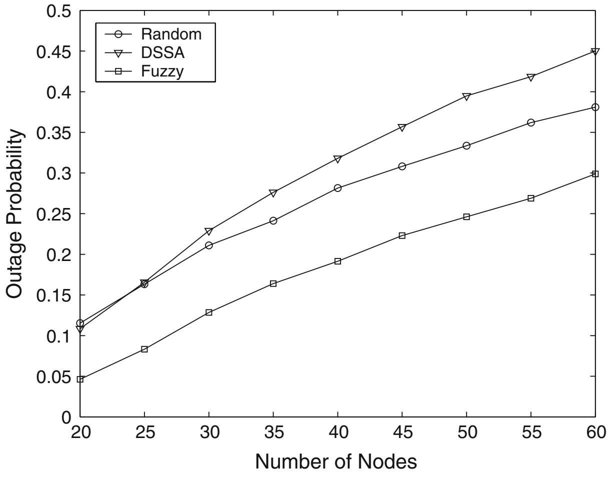

Fig. 14. Outage probability vs. number of sensors (Rc = 100 m).

5. CONCLUSIONS

In this paper, we proposed a sensor deployment strategy based on FLS. Our approach has a great advantage to deal with the uncertainty in distributed sensor deployment which is particularly useful when emergency rescue or redeployment over hostile situation is needed. We believe that in an energy constrained wireless sensor network, fast and efficient deployment strategy is a necessity to save energy and extend network lifetime. Our FLSs scheme is capable to model all distributed sensor deployment with a FLS. The network coverage and quality of communication in term of outage probability are greatly improved as a result. Moreover, the FLSs scheme brings the whole network to a stable and optimal deployment very soon which will significantly reduce the energy consumption. Our future work will focus on modeling the random deployment with some existing pattern so that the energy consumption can be further studied in the deployment problem.

Acknowledgments This work was supported by the Office of Naval Research (ONR) Young Investigator Award under Grant N00014-03-1-0466, "Energy Efficient Wireless Sensor Networks for Future Combat System Using Fuzzy Logic".

Appendix Multiple Log-Normal Interferers Consider the sum of N_I log-normal random variables

$$I = \sum_{k=1}^{N_I} \Omega_k = \sum_{k=1}^{N_I} 10^{\Omega_{k(\mathrm{dBm})}/10} \tag{11}$$

where the $\Omega_{k(\mathrm{dBm})}$ are Gaussian random variables with mean $\mu_{\Omega_{k(\mathrm{dBm})}}$ and variance $\sigma^2_{\Omega_k}$, and the $\Omega_k = 10^{\Omega_{k(\mathrm{dBm})}/10}$ are the log-normal random variables. Unfortunately, there is no known closed form expression for the probability density function (pdf) of the sum of multiple ($N_I \geq 2$) log-normal random variables. However, there is a general consensus that the sum of independent log-normal random variables can be approximated by another log-normal random variable with appropriately chosen parameters. That is,

$$I = \sum_{k=1}^{N_I} 10^{\Omega_{k(\mathrm{dBm})}/10} \approx 10^{Z_{(\mathrm{dBm})}/10} = \hat{I} \tag{12}$$

where $Z_{(\mathrm{dBm})}$ is a Gaussian random variable with mean $\mu_{Z(\mathrm{dBm})}$ and variance σ^2_Z. The problem is to determine $\mu_{Z(\mathrm{dBm})}$ and variance σ^2_Z in terms of the $\mu_{\Omega_{k(\mathrm{dBm})}}$ and variance $\sigma^2_{\Omega_k}$, $k = 1, \ldots, N_I$. Several methods have been suggested in the literature to solve this problem including those by Fenton, Schwartz and Yen, and Farley. Each of these methods provides varying degrees of accuracy over specified ranges of the shadow standard deviation σ_Ω, the sum I, and the number of interferes N_I.

Fenton–Wilkinson Method With the Fenton–Wilkinson method, the mean $\mu_{Z(\mathrm{dBm})}$ and variance σ^2_Z of $Z_{(\mathrm{dBm})}$ are obtained by matching the first two moments of the sum I with the first two moments of the approximation $\hat{I}$. To derive the appropriate moments, it is convenient to use natural logarithms. We write

$$\Omega_k = 10^{\Omega_{k(\mathrm{dBm})}/10} = e^{\epsilon\Omega_{k(\mathrm{dBm})}} = e^{\hat{\Omega}_k} \tag{13}$$

where $\epsilon = (\ln 10)/10 = 0.23026$ and $\hat{\Omega}_k = \epsilon\Omega_{k(\mathrm{dBm})}$. Note that $\mu_{\hat{\Omega}_k} = \epsilon\mu_{\Omega_{k(\mathrm{dBm})}}$ and $\sigma^2_{\hat{\Omega}_k} = \epsilon^2\sigma^2_{\Omega_k}$. The nth moment of the log-normal random variable Ω_k can be obtained from the moment generating function of the Gaussian random variables $\hat{\Omega}_k$ as

$$E[\Omega_k^n] = E[e^{n\hat{\Omega}_k}] = e^{n\mu_{\hat{\Omega}_k} + (1/2)n^2\sigma^2_{\hat{\Omega}_k}} \tag{14}$$

To find the appropriate moments for the log-normal approximation we can use (14) and equate the first two moments on both sides of the equation

$$I = \sum_{k=1}^{N_I} e^{\hat{\Omega}_k} \approx e^{\hat{Z}} = \hat{I} \tag{15}$$

where $\hat{Z} = \epsilon Z_{(\mathrm{dBm})}$. For example, suppose that $\hat{\Omega}_k$, $k = 1, \ldots, N_I$ have mean $\mu_{\hat{\Omega}_k}$, $k = 1, \ldots N_I$ and identical variances $\sigma^2_{\hat{\Omega}}$. Identical variances are often assumed because the standard deviation of log-normal shadowing is largely independent of the radio path length. Equating the means on both sides of (15)

$$\mu_I = E[I] = \sum_{k=1}^{N_I} E[e^{\hat{\Omega}_k}] = E[e^{\hat{Z}}] = E[\hat{I}] = \mu_{\hat{I}} \tag{16}$$

gives the result

$$\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}}\right) e^{(1/2)\sigma^2_{\hat{\Omega}}} = e^{\mu_{\hat{Z}} + (1/2)\sigma^2_{\hat{Z}}} \tag{17}$$

Likewise, we can equate the variances on both sides of (15) under the assumption that the $\hat{\Omega}_k$, $k = 1, \ldots, N_I$ are independent

$$\sigma^2_I = E[I^2] - \mu^2_I = E[\hat{I}^2] = \sigma^2_{\hat{I}} \tag{18}$$

giving the result

$$\left(\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}\right) e^{\sigma^2_{\hat{\Omega}}}\left(e^{\sigma^2_{\hat{\Omega}}} - 1\right) = e^{2\mu_{\hat{Z}}} e^{\sigma^2_{\hat{Z}}}\left(e^{\sigma^2_{\hat{Z}}} - 1\right) \tag{19}$$

By squaring each side of (17) and dividing each side of resulting equation by the respective side of (19) We can solve for $\sigma^2_{\hat{Z}}$ in terms of the known values of $\mu_{\hat{\Omega}_k}$, $k = 1, \ldots, N_I$ and $\sigma^2_{\hat{\Omega}}$. Afterwards, $\mu_{\hat{Z}}$ can be obtained from (17). This procedure yields the following solution:

$$\mu_{\hat{Z}} = \frac{\sigma^2_{\hat{\Omega}} - \sigma^2_{\hat{Z}}}{2} + \ln\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}}\right) \tag{20}$$

$$\sigma^2_{\hat{Z}} = \ln\left(\left(e^{\sigma^2_{\hat{\Omega}}} - 1\right)\frac{\sum_{k=1}^{N_I} e^{2\mu_{\hat{\Omega}_k}}}{\left(\sum_{k=1}^{N_I} e^{\mu_{\hat{\Omega}_k}}\right)^2} + 1\right) \tag{21}$$

Finally, $\mu_{Z(\mathrm{dBm})} = \epsilon^{-1}\mu_{\hat{Z}}$ and $\sigma^2_Z = \epsilon^{-2}\sigma^2_{\hat{Z}}$.
The accuracy of this log-normal approximation can be measured in terms of how accurately the first two moments of $I_{(dB)} = 10\log_{10} I$ are estimated, and how well the cumulative distribution function (cdf) of $I_{(\mathrm{dB})}$ is described by a Gaussian cdf. In problems relating to the co-channel interference outage in cellular radio systems, we are usually interested in the tails of both the complementary distribution function (cdfc) $F_I^C = P(I \geq x)$ and the cdf $F_I(x) = 1 - F_I^C = P(I < x)$. In this case, we are interested in the accuracy of the approximation

$$F_I(x) \approx P(e^{\hat{Z}} \geq x) = Q\left(\frac{\ln x - \mu_{\hat{Z}}}{\sigma_{\hat{Z}}}\right) \tag{22}$$

for large and small values of x. It will be shown later that the Fenton–Wilkinson method can approximate the tails of the cdf and cdfc functions with good accuracy.

REFERENCES

1. C. Y. Chong, and S. P Kumar, Sensor networks: evolution, opportunities, and challenges, *Proceedings of the IEEE,* Vol. 91, No. 8, pp. 1247–1256, 2003.
2. A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, Wireless sensor networks for habitat monitoring, *Proceedings of the WSNA*'02, Atlanta, Georgia, Sep 28, 2002.
3. E. Biagioni and G. Sasaki, Wireless sensor placement for reliable and efficient data collection. *Proc. Hawaii Int. Conf. Syst. Sci.*, Jan., 2003.

4. H. Qi, S. S. Iyengar, and K. Chakrabarty, Distributed sensor fusion – a review of recent research, *Journal of the Franklin Institute,* Vol. 338, pp. 655–668, 2001.
5. Y. Zhou, and K. Chakrabarty, Sensor deployment and target localization based on virtual forces, *IEEE Twenty-Second Annual Joint Conference of the Computer and Communications Societies,* Vol. 2, pp. 1293–1303, 2003.
6. T. Wong, T. Tsuchiya, and T. Kikuno, A self-organizing technique for sensor placement in wireless micro-sensor networks, *18th International Conference on Advanced Information Networking and Applications,* Vol. 1, pp. 78–83, AINA 2004.
7. S. Poduri, and G. S. Sukhatme, Constrained coverage for mobile sensor networks, *IEEE International Conference on Robotics and Automation,* Vol. 1, pp. 165–171, 2004.
8. P. Cheng, C. Chuah, and X. Liu, Energy-aware node placement in wireless sensor networks, *IEEE Global Telecommunications Conference,* Vol. 5, pp. 3210–3214, 2004.
9. N. Heo, and P. K. Varshney, A distributed self spreading algorithm for mobile wireless sensor networks, *IEEE International Conference on Wireless Communications and Networking,* Vol. 3, pp. 1597–1602, New Orleans, LA 2003.
10. N. Heo, and P. K. Varshney, An intelligent deployment and clustering algorithm for a distributed mobile sensor network, *IEEE International Conference on Systems, Man and Cybernetics,* Vol. 5, pp. 4576–4581, 2003.
11. N. Heo, and P. K. Varshney, Energy-efficient deployment of Intelligent Mobile sensor networks, *IEEE Transactions on Systems, Man and Cybernetics, Part A,* Vol. 35, No. 1, pp. 78–92, 2005.
12. J. M. Mendel, Fuzzy logic systems for engeneering: a tutorial. *Proceedings of the IEEE*, Vol. 83, No. 3, pp. 345–377, 1995.
13. J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems*,Prentice-Hall, Upper Saddle River, NJ, 2001.
14. T. J. Ross, *Fuzzy Logic with Engineering Applications*,McGraw-Hill International Editions, Singapore, 1997.
15. E. H. Mamdani, Applications of fuzzy logic to approximate reasoning using linguistic systems, *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 26, No. 12, pp. 1182–1191, 1977.
16. T. S. Rappaport, *Wireless Communications: Principles and Practice*,Prentice-Hall, Upper Saddle River, NJ, 2001.
17. G. L. Stuber, *Principles of Mobile Communication*,Kluwer Academic Publishers, Boston, 2001.

Haining Shu received the B.S. degree in Electronics and Information Systems from Peking University, China in 1996 and M.S. degree in Electrical Engineering from the University of Texas at Dallas in 2002. Prior to the master program, she was a System Engineer at Telecom Planning and Research Institute in Beijing, China. She is currently working toward the Ph.D degree in Electrical Engineering at the University of Texas at Arlington. Her research interests include fuzzy logic systems and applications, distributed source coding, sensor networks and collaborative radar system.

Qilian Liang received the B.S. degree from Wuhan University, China, in 1993, M.S. degree from Beijing University of Posts and Telecommunications in 1996, and Ph.D degree from University of Southern California (USC) in May 2000, all in Electrical Engineering.Dr. Liang joined the faculty of the University of Texas at Arlington in August 2002. Prior to that he was a Member of Technical Staff in Hughes Network Systems Inc at San Diego, California. His research interests include Sensor networks (energy efficiency, cross layer design, optimal sensor deployment, etc), wireless communications, wireless networks, communication system and communication theory, signal processing for communications, fuzzy logic systems and applications, multimedia network traffic modeling and classification, collaborative and distributed signal processing.

Dr. Liang has published more than 100 journal and conference papers, 5 book chapters, and has 6 U.S. patents pending. He received 2002 IEEE Transactions on Fuzzy Systems Outstanding Paper Award, 2003 Office of Naval Research (ONR) Young Investigator Award and 2005 UTA College of Engineering Outstanding Young Faculty Award.

Jean Gao is an Assitant Professor in the Computer Science and Engineering Department at University of Texas at Arlington. She is the recipient of prestigious CAREER Award from National Science Foundation. She received her Ph.D. in electrical engineering from Purdue University, and her M.S. and B.S. in biomedical engineering from Rose-Hulman Institute of Technology and Shanghai Medical University, respectively. Her research interests in computer vision and pattern recognition are object motion estimation, shape classification, multi-dimensional multi-object tracking, and reconstruction.