Self-healing sensor networks with distributed decision making

Xiaojiang (James) Du,* Ming Zhang and Kendall E. Nygard

Department of Computer Science,
North Dakota State University,
Fargo ND 58105, USA
E-mail: Xiaojiang.Du@ndsu.edu
E-mail: Ming.Zhang@ndsu.edu E-mail: Kendall.Nygard@ndsu.edu
*Corresponding author

Sghaier Guizani

Department of Electrical Engineering,
University of Quebec,
Trois-Riviere, QC, Canada
E-mail: guizans@umoncton.ca

Hsiao-Hwa Chen

Institute of Communications Engineering,
National Sun Yat-Sen University,
Kaohsiung City, Taiwan
E-mail: hshwchen@ieee.org

Abstract: Sensor random locations and sensor failures can cause coverage holes, routing voids and disconnections and thus degrade sensor network performance. In this paper, we present a self-healing approach that utilises a few mobile sensors to deal with the problems. A mobile sensor can move to an area with a coverage hole or routing void and significantly improve network performance. We design distributed algorithms to detect coverage holes, estimate hole size and repair holes. A mobile sensor may receive multiple repair requests from different areas and it may not have complete information of the network. We present a fuzzy-logic-based distributed decision-making algorithm for mobile sensors. Extensive simulations demonstrate the good performance of proposed algorithms.

Keywords: sensor networks; self-healing; mobile sensors; decision making; fuzzy logic.

Reference to this paper should be made as follows: Du, X., Zhang, M., Nygard, K.E., Guizani, S. and Chen, H-H. (2007) 'Self-healing sensor networks with distributed decision making', *Int. J. Sensor Networks*, Vol. 2, Nos. 5/6, pp.289–298.

Biographical notes: Xiaojiang (James) Du is an Assistant Professor in the Department of Computer Science, North Dakota State University (NDSU). He received his PhD in Electrical Engineering from University of Maryland, College Park in 2003. His research interests are wireless sensor networks, network security and network management.

Ming Zhang is pursuing his PhD in Computer Science at NDSU. His research interests include the design and optimisation of wireless and sensor networks.

Kendall E. Nygard is a Professor in the Department of Computer Science at North Dakota State University. From 1996 to 2005 he served as department chair. He received his PhD in 1978 from Virginia Polytechnic Institute and State University. He has been Principal Investigator for more than 30 federal, state and private grants and contracts.

Sghaier Guizani received a BS and an MSc in Electrical Engineering from SUNY at Binghamton, NY in 1990 and North Carolina State University in 1992, respectively. Worked with Alcatel Data Network from 1993 to 1999. Joined Nortel Networks, Ottawa, Canada in 1999–2002. He is working towards his PhD at the University of Quebec, Canada.

Hsiao-Hwa Chen is currently a Full-time Professor in National Sun Yat-Sen University, Taiwan. He has authored or co-authored over 170 technical papers in major international journals and conferences and five books in the areas of communications. He served as symposium co-chair of several major international conferences. He served or is serving as an Editorial Board Member or/and Guest Editor for many top journals.

1 Introduction

A wireless sensor network consists of a large number of small sensor nodes that perform sensing tasks and transmit the acquired information to a Base Station (BS). Sensor networks have many applications, such as environmental monitoring, military surveillance and target tracking. Sensor nodes are unreliable small devices with limited energy supply. The effectiveness of a sensor network depends on the sensor coverage on the field.

Previous studies mainly considered sensor networks where either all sensors are static or all sensors are mobile (Wang et al., 2004; Zou and Chakrabarty, 2003). For sensor networks with only static sensors, due to the large number of sensor nodes and possible hostile environment (e.g. battlefield, toxic regions), the exact sensor location may not be controlled by typical deployment methods, for example, distribution from an airplane. With these types of deployment, some geographical areas may not be covered by any sensor because of the randomness of sensor location. The coverage holes or network partitions can degrade network performance. Increasing the density of sensor deployment may not solve the problem, since a coverage hole may be caused by geographic factors, for example, few sensors are located at the top of a hill.

Researchers (Wang et al., 2004; Zou and Chakrabarty, 2003) also have considered sensor networks where all sensors are mobile nodes and studied the problem of how to move mobile sensors to maximise sensing coverage in the network. For example, Zou and Chakrabarty (2003) studied movement assisted sensor deployment, which uses virtual forces to drive the movement of mobile sensors from an initial unbalanced state to a balanced state. However, a mobile sensor has much higher cost than a static sensor with similar sensing capability. Deploying a large number of mobile sensors can be very expensive. Another drawback of these methods is that they have longer deployment time since all sensors need to move in several rounds before settling down.

To provide self-optimising and self-healing capabilities in sensor networks while keeping the cost low, we propose a self-healing deployment model in which a small portion of mobile sensors are purposely deployed in addition to a large number of static sensors. After detecting a coverage hole, mobile nodes can be called to move to the area and repair the network. Mobile sensors can significantly improve network performance by moving to locations where there is a coverage hole or network disconnection. We have designed several centralised schemes (Du and Lin, 2005) that utilise mobile sensors to improve coverage, connectivity and routing performance in sensor networks. These centralised schemes rely on the BS and may not scale well. Due to the limited capabilities of typical sensor nodes, distributed algorithms are preferred and more scalable than centralised algorithms in sensor networks. In Zhang et al. (2005), we have designed distributed algorithms for the static sensors to find coverage holes and disconnected areas.

Following our self-healing sensor deployment model, in this paper we study the decision-making problem for mobile sensors. To reduce sensor cost, only a small number of mobile sensors are deployed in a network. Thus, a mobile sensor may receive multiple requests from different areas. Furthermore, a mobile sensor may not have complete information of the network, since obtaining global information requires excessive communication and energy consumption, which is not suitable for small sensor nodes. In this paper, we propose a fuzzy-logic-based distributed decision-making algorithm that has low computation requirement and is robust to incomplete information. The distributed algorithm can determine a good location to which a mobile sensor should move. The decision depends on several factors, including the distance, the coverage gain and the connectivity gain. A utility function is defined to consider these factors. A mobile sensor moves to the location that maximises the utility function. Extensive simulations are run to demonstrate the self-healing capabilities provided by mobile sensors and the good performance of the distributed decision-making algorithm.

The contributions of the paper include:

1. the self-healing sensor deployment model
2. an Integer Linear Programming (ILP) formulation of the mobile sensor decision-making problem and
3. a distribute decision-making algorithm for mobile sensors.

The rest of this paper is organised as follows. Section 2 describes the self-healing sensor deployment model. Section 3 presents the mobile sensor decision-making problem and the ILP formulation of the problem. Section describes the formulation of utility function based decision making. Section 5 presents the fuzzy logic based distribute decision-making algorithm for mobile sensors. Section 6 shows the experimental results. And Section 7 concludes this paper.

2 Self-healing sensor deployment

In self-healing sensor deployment, a few mobile sensors and a large number of static sensors are deployed in the field. We first present the sensor network model below.

2.1 The sensor network model

Denote the set of all static sensors as $S = \{s_1, s_2, \ldots, s_m\}$ and the set of all of all mobile sensors as $M = \{m_1, m_2, \ldots, m_n\}$ and $|S| >> |M|$. Here $|S|$ and $|M|$ represent the size of the static node and mobile nodes. We assume that each sensor node is aware of its own location. Location awareness is a basic requirement for sensor nodes since most sensing data must be associated with the location where the data is generated. A sensor network can use location services such as discussed in Bruck et al. (2005) and Doherty et al. (2001) to estimate the locations of the individual nodes and no GPS receiver is required at each sensor. For simplicity, we assume that mobile sensors and static sensors have the same sensing range.

2.2 An overview of self-healing deployment

The self-healing deployment scheme has four phases: initialisation, detecting holes, decision and execution. We describe each phase below.

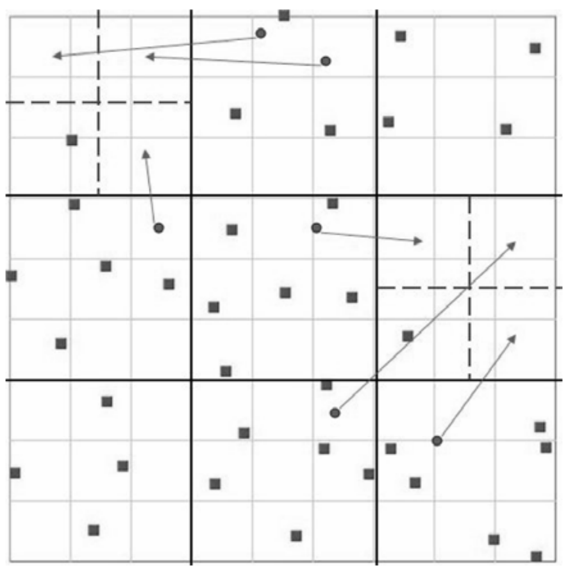
Initialisation: both mobile and static sensors are distributed into the field to be monitored. Initially, the state of mobile sensors is set to waiting and the state of static sensors is set to initialising.

Detecting holes: static sensors exchange location information with their neighbours and use the information to find any coverage holes or disconnected areas in its nearby area. A static sensor may estimate the size of a coverage hole based on the locations of boundary nodes of the hole. A leader is selected from the set of the boundary nodes to coordinate the repair of the hole, including estimating the size of the hole and calling mobile sensors. A detailed algorithm can be found by Zhang et al. (2005).

Decision: in the simple case, a mobile node only receives a help request from one area within a time period, then based on the distance, the mobile node can decide if it should move there or wait for later request. More commonly, the final decision is completed from multiple requests.

As illustrated in Figure 1 (where circle nodes represent mobile sensors and square nodes represent static sensors), mobile sensors can significantly improve network performance by moving to locations where there is a coverage hole or disconnection. Thus, mobile sensors can essentially provide self-healing and self-optimising capabilities for sensor networks. A mobile sensor may receive multiple calls for movement. The mobile sensor then must decide where it should move to achieve the maximum improvement on network performance. We design distributed decision-making algorithms to determine the location where a mobile sensor should move.

Figure 1 Improving performance with mobile sensors



Execution: after making a decision as where to move, a mobile sensor moves to the destination during the execution phase. Since sensor movement consumes energy, we do not consider multiple movements of a mobile sensor in the current research. When the movement is done, the state of the mobile sensor is changed to static and its behaviour will be the same as that of a static sensor.

3 Decision-making algorithms for mobile sensors

In this section, we present the notations and problem formulation of the mobile sensor movement problem. The goal is to find out the best location where each mobile sensor (given the number of mobile sensors) should move and maximise the performance improvement in a sensor network. The global optimisation problem is formulated as an ILP problem.

3.1 Notations

In the following, we list the notations used in the rest of this paper.

- $H = \{h_1, h_2, \ldots, h_k\}$ is the set of coverage holes. A hole h_j is defined by the boundary static sensors of the hole and $h_j \subset S$ is a subset of S.
- $\|h_j\|$: the size of hole of h_j – in terms of the number of required mobile sensors to cover it.
- $d(m_i, h_j)$: the Euclidean distance between a mobile sensor m_i and a hole h_j.
- E: the initial energy of a newly deployed mobile node.
- $g_c(m_i, h_j)$: the coverage gain if mobile sensor m_i moves to hole h_j.
- $g_l(m_i, h_j)$: the connectivity gain if mobile sensor m_i moves to hole h_j.
- $e(m_i, h_j) = \varepsilon \times d(m_i, h_j)$: the required energy for sensor m_i to move to hole h_j, which is proportional to the moving distance. Here ε is a constant.

3.2 The ILP formulation

The optimal sensor movement problem in the entire network can be formulated as an ILP problem. The formulation seeks the global optimal solution of the mobile sensor movement problem. The ILP formulation is presented in the following. We define the variable of the ILP as x_{ij}, where $x_{ij} = 1$, if mobile sensor m_i moves to hole h_j, otherwise $x_{ij} = 0$.

The objective function of the ILP is to maximise:

$$\sum_{i \in M} \sum_{j \in H} x_{ij} \times g\left(m_i, h_j\right)$$

where,

$$g\left(m_i, h_j\right) = g_c\left(m_i, h_j\right) + \lambda_1 \times g_l\left(m_i, h_j\right) - \lambda_2 \times e\left(m_i, h_j\right)$$

and λ_1 and λ_2 are normalisation constants.

s.t.

1 The energy spent on movement should be no more than a given portion of the initial energy of a mobile sensor, that is, for each mobile sensor m_i

$$\sum_{j\in H} x_{ij} \times e\left(m_i, h_j\right) \le \sigma E, \quad \text{where } 0 \le \sigma \le 1 \tag{1}$$

2 Each mobile sensor only moves once. For each mobile sensor m_i

$$\sum_{j\in H} x_{ij} = 1 \tag{2}$$

3 The number of moved mobile nodes of the hole should be no more than its size. For each hole h_j

$$\sum_{i\in M} x_{ij} \le \left|h_j\right| \tag{3}$$

Note that constraint (2) may be removed depending on the network design. Constraint (1) includes the case where a mobile sensor moves more than once. The above ILP can be solved by many existing solvers, including fast network optimisation solvers. However, in order to parameterise the model to ensure optimality, excessive communications among all sensor nodes are required. Also the computation requirement of solving the ILP may exceed the capability of typical sensor nodes. Thus, we will not discuss how to solving the ILP. Instead, we propose a fuzzy-logic-based distributed decision-making algorithm that has low computation requirement and can be feasibly computed at sensor nodes. The detail is given in Section 4.

3.3 A distributed decision-making algorithm

In the following, we briefly present the idea of a distributed decision-making algorithm for mobile sensors. The decision of where to move depends on several factors, including the distance, the coverage gain and the connectivity gain. Mobile sensors consume energy while moving (Rahimi et al., 2003). Thus, the distance to a calling area is an important factor to consider. The coverage gain is the coverage improvement from using the mobile sensor and it depends on the size of the coverage hole and the sensing capability of the mobile sensor. The connectivity gain is more complex than the coverage gain and it depends on several factors, such as the decreases in hop distance and delay (to the BS) and the increase of throughput after mobile sensors move in. A utility function is defined to include all of these factors. A mobile sensor should move to the location that maximises the utility function.

Partitioning a sensor network can significantly degrade network performance. Thus, a mobile sensor should always move to a location that can connect a partitioned area. A very large gain value should be assigned to a network partition. The detection of a disconnected area can be based on a change of traffic volume or on active roaming of mobile sensors.

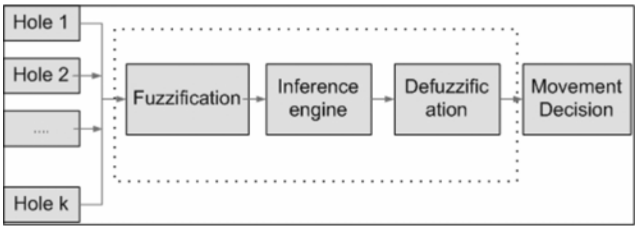
A static sensor can detect a coverage hole or loss of connection in its nearby area and can send out a *Call* message that requests mobile sensors to move to the location. The *Call* message includes the location of the calling node, so that a mobile sensor can calculate the distance. A static sensor can estimate the size of a coverage hole (Zhang et al., 2005) and the size of the hole can be included in the *Call* message. A mobile sensor can then calculate the coverage gain based on the hole size and its sensing capability. The *Call* message can also include the estimated reduction of hop numbers and delay to the BS and the estimated throughput gain after a mobile sensor moves in and this information can be used by a mobile sensor to calculate the connectivity gain. The details of the distributed decision-making algorithms are presented in Section 4.

3.4 Decision making with partial information

A typical sensor network has a large number of nodes and the communication may be unreliable. Obtaining complete network information for the BS or a sensor node could be very expensive or even impossible, since this requires excessive communication and energy consumption. Thus, a mobile sensor may not have complete information of the network. We design schemes for mobile sensors to incorporate uncertainty into decision making with partial information.

Specifically, we utilise fuzzy logic (Driankov et al., 1993) to allow a mobile sensor make good decisions with incomplete information. Fuzzy logic is a problem-solving control system methodology that has implementation in systems ranging from simple, small, embedded microcontrollers to large, networked, multichannel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software or a combination of both. Fuzzy logic deals with imprecision and incompleteness of information by formalising the degree to which observations fit a classification. Decision trees can be built under fuzzy logic for decision making with partial information. This approach provides a straightforward way to arrive at definite conclusions based upon vague, ambiguous, imprecise, noisy or missing input information. Figure 2 shows the block diagram of decision making in each mobile sensor.

Figure 2 Block diagram of fuzzy decision for mobile sensors



The fuzzy decision support approach incorporates simple logic rules (e.g. IF X AND Y THEN Z) to make decisions. Fuzzy logic procedures can be easily implemented in sensor nodes because they require only small amounts of storage and computational resources. To apply fuzzy decision making to mobile sensor movement, we view mobile sensors as agents who have goals to cover holes in the geographical area, take responsibility for their actions and do so with consideration of all the information to which they have access. As they move and increase their knowledge base, the mobile sensors can reason and make local movement choices that seek to optimise their goals. We develop decision rules for fuzzy logic algorithms used in mobile sensor and the details are given in Section 4.

4 Utility function-based decision making

4.1 The utility function

We define a utility function to include the three factors that affect the mobile sensor decision: the movement cost, the coverage gain and the connectivity gain. The three factors are defined below.

Movement cost of mobile sensors: the movement cost of a mobile sensor is in proportion to the distance it travels. For example, in Rahimi et al. (2003), the mobile robot consumes 0.210 J per inch movement. For simplicity, we define the movement cost to be the distance. The actual coefficient will be included in the utility function presented below. The movement cost is defined as: $e(m_i, h_j) = d(m_i, h_j)$.

Coverage: each sensor node has a physical sensing range. The sensing area of a mobile sensor m_i is represented by a circle with radius r and a centre point (x_i, y_i), that is, the set of points covered by m_i is,

$$\left\{(x,y) \mid (x-x_i)^2 + (y-y_i)^2 \le r^2\right\}$$

Coverage gain: given two sensor nodes s and m_i, the *overlapping coverage* between sensor s and m_i is denoted as $v(s \cap m_i)$ and it is given below:

If $d \triangleq d(s, m_i) \le 2r$

$$v(s \cap m_i) = 2r^2 \arccos\frac{d}{2r} - d\sqrt{r^2 - \frac{d^2}{4}}$$

Otherwise, $v(s \cap m_i) = 0$.

The coverage gain of a mobile sensor m_i to a hole h_j is defined as:

$$g_c(m_i, h_j) = \pi r^2 - \bigcup_{s \in h_j} v(s \cap m_i)$$

That is, $g_c(m_i, h_j)$ is the net area covered by the mobile sensor m_i (after it moves in).

Connectivity gain: the connectivity gain is defined as the fraction of new links (after a mobile sensor moves in) to the existing links. That is, the number of new links (after the mobile node m_i moves to the hole h_j) divided by the total number of links among existing sensors.

For simplicity, assume the transmission ranges of all static sensors and all mobile sensors are the same. Firstly, we denote the link between two sensor nodes m and s as:

$$l_{m,s} = \begin{bmatrix} 1 & d(m,s) \le r \\ 0 & otherwise \end{bmatrix}$$

where r is the transmission range of a sensor node. Then we denote $l_{m_i} = \sum_{s \in h_j} l_{m_i,s}$ as the total number of new links between a mobile sensor m_i and boundary nodes of a hole h_j. The number of new links can be determined by using the location information of the mobile sensor (after movement) and each boundary node of the hole. Now we define the connection gain as:

$$g_l(m_i, h_j) = \frac{l_{m_i}}{\sum_{s \in h_j} l_s}$$

where l_n is the number of links of node n.

The utility function: for a given mobile sensor m_i and hole h_j, the utility function is:

$$g(m_i, h_j) = g_l(m_i, h_j) + \lambda_1 \times g_c(m_i, h_j) \\ -\lambda_2 \times c(m_i, h_j) \quad (*)$$

where λ_1 and λ_2 are normalisation constants.

4.2 The distributed algorithm in each mobile sensor

When a mobile sensor receives the first movement request from a boundary node of a hole, the mobile sensor sets a timer T and waits for possible more requests. When the timer expires, the mobile sensor uses the fuzzy logic algorithm to determine the best location to which it should move. In each mobile sensor, a threshold-based algorithm is used, that is, when the value of the utility function is less than a predefined threshold, the mobile sensor rejects the request and waits for new requests. The threshold-based algorithm is used to avoid unnecessary movement of a mobile sensor, especially over a long distance. The distributed algorithm running in each mobile sensor is presented in Figure 3.

Figure 3 The distributed algorithm in each mobile sensor

1. A mobile sensor *m* receives the first request from a hole. And *m* sets a timer T.
2. **While** time t < T:
3. Wait for new requests.
4. **Calculate** the movement cost (based on the distance), the coverage gain and connection gain.
5. **If** the utility function <= the threshold
6. Reject the request and wait for new requests;
7. **else**
8. Mobile sensor stores the request.
9. **End if**
10. **End While**.
11. Apply fuzzy logic algorithm and find out the best location.
12. Move to the location.

5 Distributed decision making using fuzzy logic

A mobile sensor uses a fuzzy controller to decide its action. The fuzzy controller has three inputs: distance, coverage gain and connectivity gain which is calculated based on the *Call* message from the hole. We define the fuzzy rule in Table 1. Based on the fuzzy rules in Table 1, a mobile sensor can make the movement decision. The fuzzy controller has four possible outputs: positive move, move, negative move and not move. We first define the fuzzy variables as follows. Let $w_m^h = (b_m^h, \mu_m^h)$ represent the decision of a mobile node m decision on the request from a hole h. Where b_m^h, μ_m^h denote the belief of moving to the hole and waiting for further request, respectively. The beliefs satisfy $b_m^h + \mu_m^h = 1$.

Table 1 Fuzzy rule

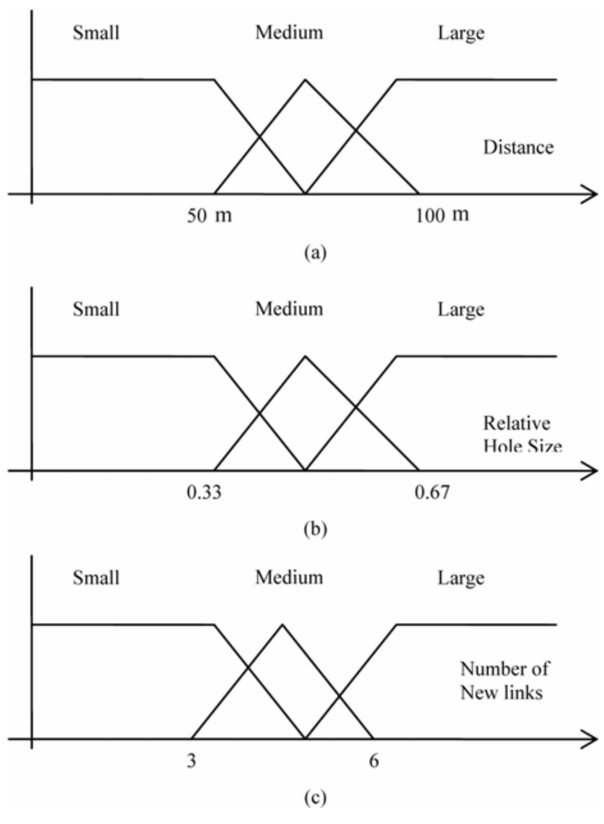
Distance	*Size*	*Number of new connections*	*Decision*
L	–	–	N
M	M/S	S	N
M	M/S	L	Y–
M	L	L	Y
S	L	S	Y+
S	M	S	Y
S	S	S	N
S	L	M	Y+
S	M	M	Y+
S	S	M	Y
S	L/M	L	Y
S	S	L	Y–

Note: L – Large; M – Medium; S – Small; N – No and Y – Yes.

5.1 The member function

The member functions of the fuzzy logic algorithm are presented in Figure 4. The outputs of the member functions are classified into three categories: small, medium and large, depending on the distance, the size of the hole and the number of new links created by the mobile sensor, respectively.

Figure 4 The membership functions of the fuzzy logic algorithm (a) input is the distance; (b) input is the hole size and (c) input is the number of new links



5.2 Fuzzy rules

We define fuzzy rules based on the following principles: short distance has higher priority; large hole has higher priority; large increase of new connections has higher priority. The fuzzy rules are listed in Table 1.

5.3 Defuzzification

We use the Centre of Area (COA) method to do the defuzzification. It calculates the centre of gravity of the resultant fuzzy set.

6 Performance evaluation

In Section 5, we presented the fuzzy-logical-based distributed decision-making algorithm. With the help of the fuzzy algorithm, a mobile sensor uses the fuzzy controller to decide its action. The fuzzy controller has three inputs: distance, coverage gain and connectivity gain. Based on the fuzzy rules in Table 1, the mobile sensor can determine the moving decision. The fuzzy controller has four possible outputs: positive move, move, negative move and not move.

We design simulation experiments to evaluate the performance of the fuzzy-logical-based distributed decision-making algorithm. The performance of the fuzzy algorithm is compared with two heuristic algorithms – First Call First Go (FCFG) algorithm and Weighted Sum (WS) algorithm. The two heuristic algorithms is presented below.

FCFG algorithm: once a mobile sensor receives a request from a hole, it moves to the hole immediately. WS algorithm: a mobile sensor waits for requests for a predefined time and then moves to the hole that has the maximum WS of three factors: distance, coverage gain and connectivity gain. The weight of each factor is given (obtained from extensive simulation experiments). The sets of weights used for the following experiments are: Distance: Coverage: Connectivity = 4:3:3 and 2:5:3.

6.1 Simulation setting

We design simulation experiments to evaluate the performance of the fuzzy-logical-based distributed decision-making algorithm. In the experiments, sensor nodes are randomly deployed in a square area. The side length of the square varies from 400 m to 1000 m with an increasing step of 100 m. The schemes in Zhang et al. (2005) are used to detect coverage holes and estimate the size of holes, the coverage gain and connectivity gain. In each experiment, the total number of required mobile sensors (i.e. the total area of uncovered holes) is set to 60 and 30 mobile nodes are deployed at random locations in the network. The maximal acceptable moving distance of a mobile sensor is set to 100 m. The experiment is repeated ten times for each configuration and the presented result is the average of the ten experiments.

We evaluate the utility-based self-healing deployment algorithm using two performance metrics:

1 the value of the utility function, which includes the moving distance, coverage gain and connectivity gain and

2 sensor network lifetime.

The network lifetime is the most relevant and critical metric for real sensor applications. Many sensor networks are expected to run for run for a long period of time, For example, several months or even years (Culler and Mulder, 2004; Polastre et al., 2004a,b). Thus, it is very important to prolong sensor network lifetime.

We design three sets of experiments to evaluate the performance of the fuzzy-logic-based distributed decision-making algorithm. The first two sets of experiments measure and compare the utility function and the third set of experiments compare the sensor network lifetime. In the first set, the total area of coverage holes and the number of mobile nodes are fixed. The network size varies from 400 m × 400 m to 1000 m × 1000 m. In the second set of experiments, the network size and the number of mobile nodes are fixed, while the total area of coverage holes changes. The results of the three sets of experiments are presented in Sections 6.2–6.4, respectively.

6.2 *Results from experiment set one*

In the experiments, sensor nodes are randomly deployed in a square. The side length of the square varies from 400 m to 1000 m with an increasing step of 100 m. The schemes in Zhang et al. (2005) are used to detect coverage holes and to estimate the size of holes, the coverage gain and connectivity gain. In each experiment, the total number of required mobile sensors is set to 60 and 30 mobile nodes are deployed in random location in the network. The maximal acceptable moving distance of a mobile sensor is set to 100 m. The experiment is repeated ten times for each configuration and the presented result is the average of the ten experiments.

The experimental results are shown in Figures 5–7, which plot the total moving distances, the total coverage gains and the total connectivity gains of mobile sensors under different algorithms, when the network size varies from 400 m × 400 m to 1000 m × 1000 m. From Figure 5, we can see that FCFG algorithm always has shorter moving distances than other algorithms. The main reason is that mobile sensors using FCFG always move to the first calling hole, which usually is a close hole. However, Figures 6 and 7 show that FCFG has less coverage and connectivity gains than WS and fuzzy algorithms. From Figures 6 and 7, we find out that WS algorithm and fuzzy logic algorithm have similar coverage and connectivity gains for all the tested network sizes. In addition, Figure 5 shows that mobile sensors under fuzzy logical algorithm have shorter moving distance than those under WS algorithm.

We also calculate the utility function (given in equation (*)), for different λ_1 and λ_2. Of course the utility function depends on the values of λ_1 and λ_2. Determining the proper values for λ_1 and λ_2 is not a trivial task. One approach is to run extensive simulations with different values of λ_1 and λ_2 and compare the utility function with the actual lifetime of the sensor network. The λ_1 and λ_2 that provide a good match between the utility function and the network lifetime are proper choices for λ_1 and λ_2. The simulation-based approach will be in our future work. In the current work, we compute the utility function for several different pairs of λ_1 and λ_2. Most results show that the fuzzy-logic-based distributed algorithm achieves the largest utility function among all the algorithms. We plot the utility function for two pairs of λ_1 and λ_2 in Figure 8, where (λ_1, λ_2) are (0.7, 0.02) and (0.8, 0.01) in Figure 8(a) and (b), respectively.

Figure 5 The total moving distance under different algorithms

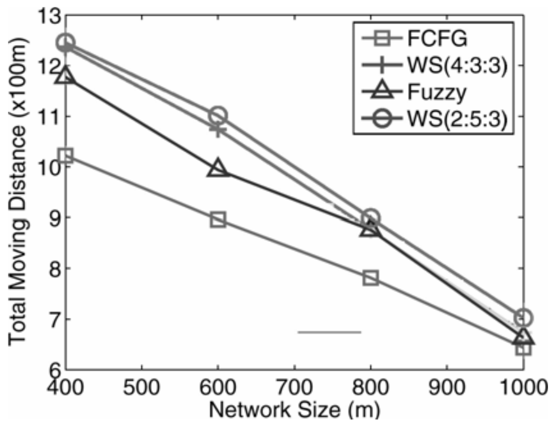


Figure 6 The total coverage gain under different algorithms

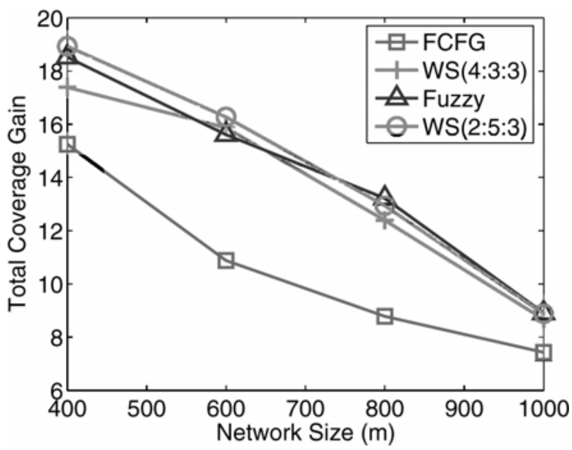


Figure 7 The total connectivity gain under different algorithms

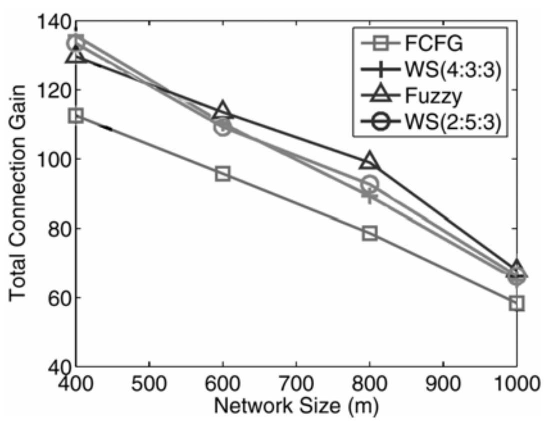
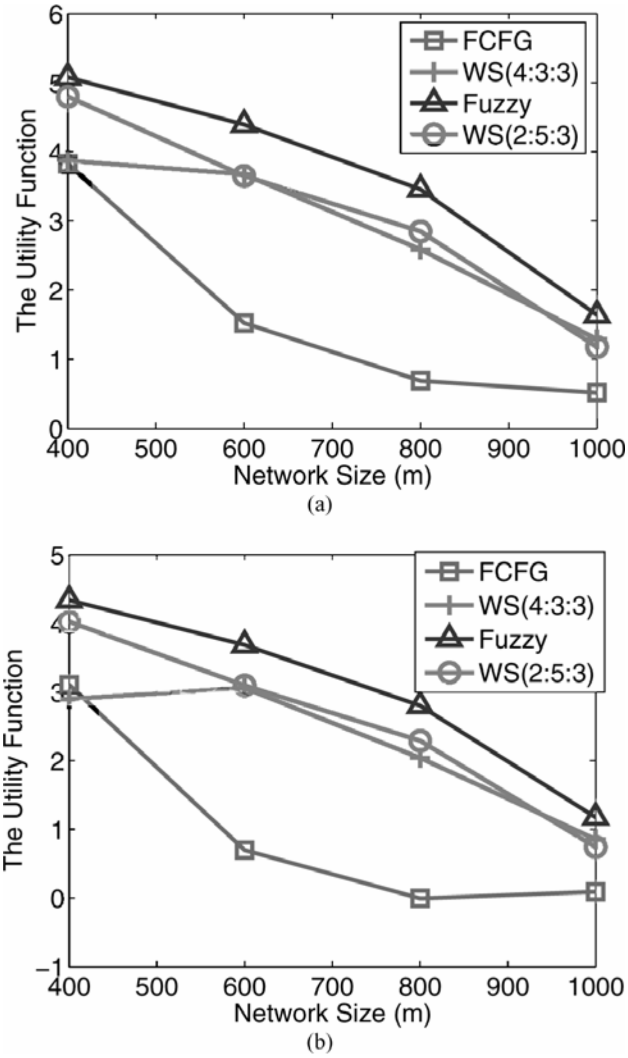


Figure 8 The utility function under different algorithms, (a) $(\lambda_1, \lambda_2) = (0.7, 0.02)$ and (b) $(\lambda_1, \lambda_2) = (0.8, 0.01)$



6.3 Results from experiment set two

In the second set of experiments, we fix the network size to be 1000 m × 1000 m and 200 mobile sensors are randomly deployed in the network. The maximal acceptable moving distance for mobile node is set to 100 m. The number of holes varies from 20 to 100 in the tests.

We measure the total moving distances, the total coverage gains and the total connectivity gains of mobile sensors under different algorithms, which are plotted in Figures 9–12. From Figure 9, we can see that FCFG algorithm always has shorter total moving distance than the other two algorithms. The reason is already stated in Section 5.1. Changing the weight of different factors have impact on the decision. In Figure 10, the WS algorithm with coverage weight 0.5 has better coverage gain than other algorithms. Figure 10 also shows that fuzzy logic algorithm has a good coverage gain. FCFG algorithm has the lowest coverage gain and connectivity gain. On the other hand, the response time of FCFG algorithm is much shorter than other algorithms. Figure 11 shows that the WS algorithm and fuzzy logic algorithm have similar results. When adding the three factors together, the fuzzy logic algorithm has the largest value of the utility function among all the tested algorithms. We also compute the utility function for different values of λ_1 and λ_2. Most results show that the fuzzy-logic-based distributed algorithm achieves the largest utility function among all the algorithms. The utility functions for two pairs of λ_1 and λ_2 are shown in Figure 12, where λ_1 and λ_2 are (0.5, 0.2) and (0.8, 0.1) in Figure 12(a) and (b), respectively.

Figure 9 The total moving distance under different algorithms

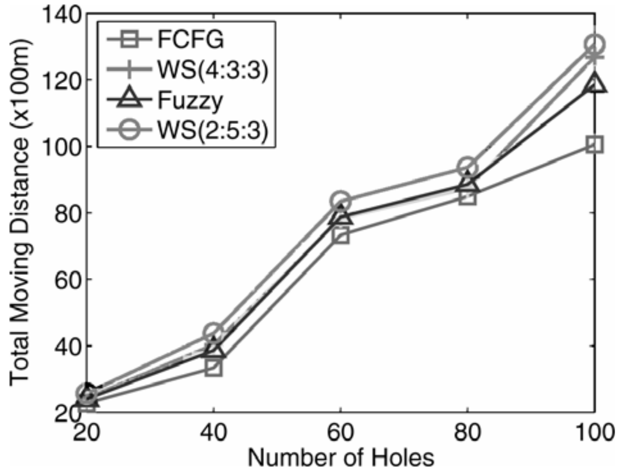


Figure 10 The total coverage gain under different algorithms

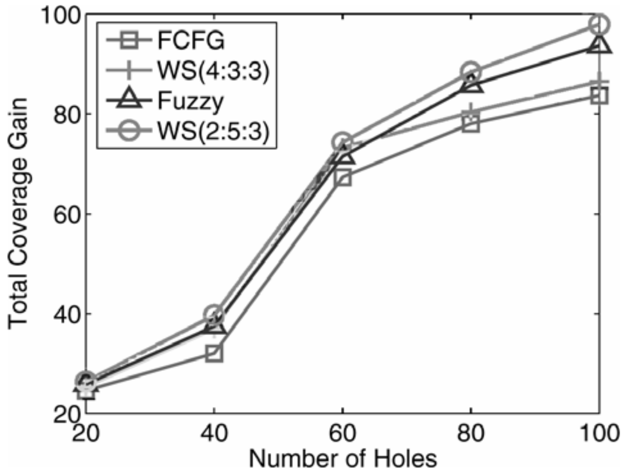


Figure 11 The total connectivity gain under different algorithms

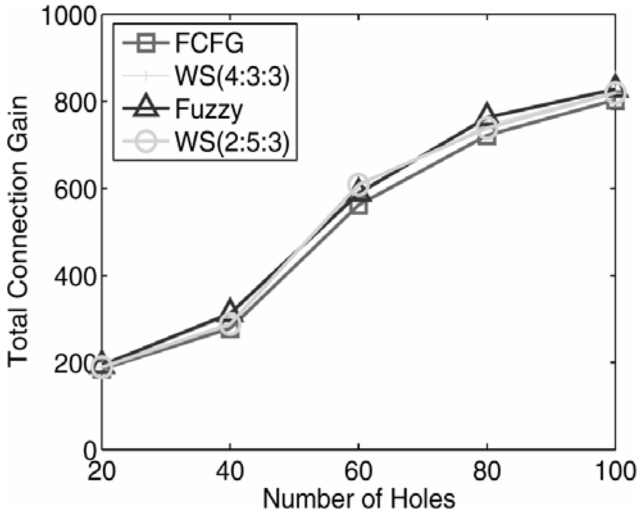
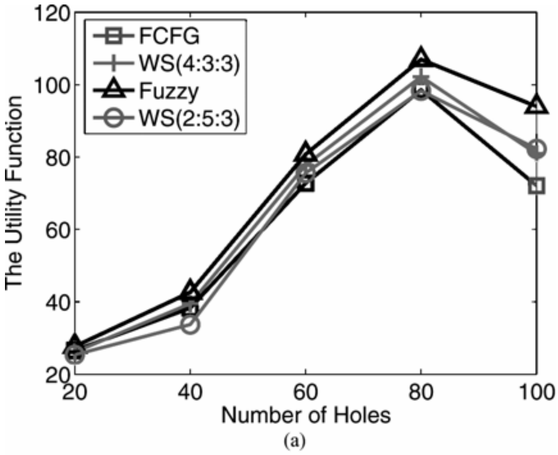
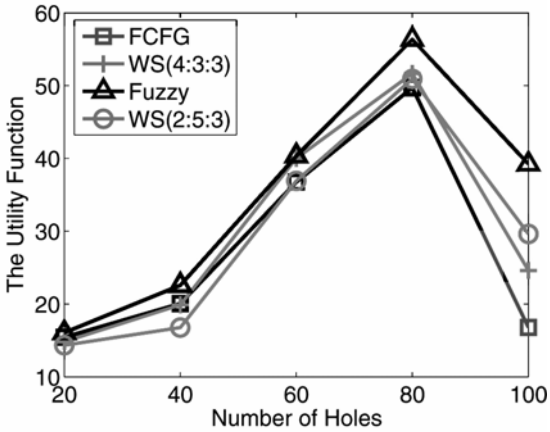


Figure 12 The utility function under different algorithms (a) $(\lambda_1, \lambda_2) = (0.5, 0.2)$ and (b) $(\lambda_1, \lambda_2) = (0.8, 0.1)$



(a)



(b)

6.4 *Comparison of network lifetime*

We study the impact of self-healing deployment on sensor network lifetime. In this subsection, we present the related simulation results. The network lifetime is measured as the time when 30% of all sensors die. We compare sensor network lifetimes when different algorithms are used by mobile sensors to make decisions about movement. In the simulation, each sensor sends a number of packets to the BS. We use Prowler (http://www.isis.vanderbilt.edu/projects/nest/prowler/) as the simulator, which is an improvement of the Rmase (http://www2.parc.com/spl/projects/era/nest/Rmase/) simulator. In the Prowler simulator, the energy consumption parameters are set according to the MICA2 Mote datasheet (www.xbow.com). The energy consumed to receive a packet is E_{rx} = 32 mW and the transmitter energy consumption is E_{tx} = 81 mW. The idle power consumption is P_s = 32 mW.

We run experiments ten times for each setting and the average is presented. The simulation results are shown in Figures 13 and 14. We compare the network lifetime improvements under four different mobile sensor movement algorithms – (FCFG) algorithm, WS algorithm with two different sets of parameters and the fuzzy-logic-based decision-making algorithm. For simulations in Figure 13, the parameters are the same as in Section 6.1, that is, in each experiment, the total number of required mobile sensors is set to 60 and 30 mobile nodes are deployed in random location in the network. The maximal acceptable moving distance of a mobile sensor is set to 100 m. Figure 13 shows that fuzzy algorithm has better performance than the other three algorithms for most cases. Figure 13 also shows that the network lifetime improvement decreases as network size becomes large. Since the number of mobile sensors and the total area of holes are fixed, the average distance between a hole and a mobile sensor increases as network size increases. Thus, some holes may not be able to find any mobile sensors nearby.

Figure 13 Lifetime increase for different network sizes

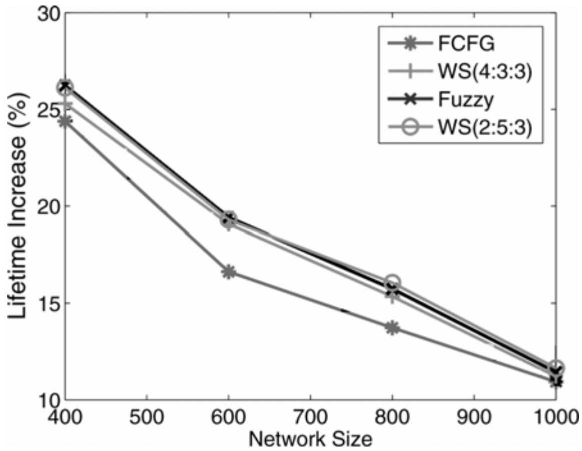
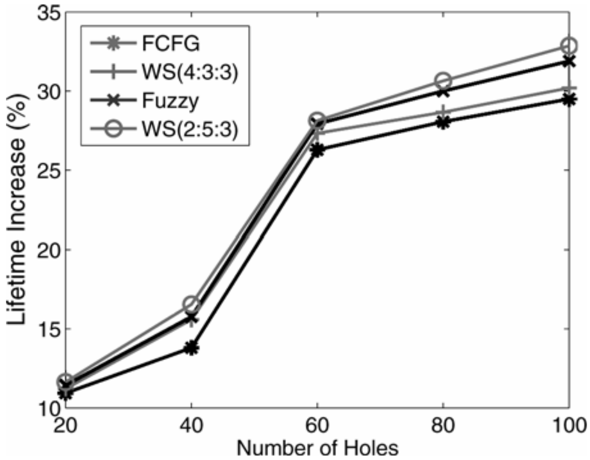


Figure 14 Lifetime increase for different number of holes



For the simulations in Figure 14, the simulation setting is the same as that in Section 6.2, that is, the network size is fixed to 1000 m × 1000 m and 200 mobile sensors are randomly deployed in the network. The maximal acceptable moving distance for mobile node is set to 100 m. The number of holes varies from 20 to 100. Figure 14 shows that more coverage improvement is achieved when the number of holes increases. Both the Figures 13 and 14 demonstrate that sensor network lifetime can be increased by moving mobile sensors to coverage (or connectivity) holes.

7 Conclusion

To provide self-healing capabilities for sensor networks, we propose to deploy a few mobile sensors in addition to a large number of static sensors in a sensor network.

Mobile sensors can move to coverage or connectivity holes and improve network performance. A mobile sensor may receive multiple requests from different areas. In this paper, we formulated the optimal sensor movement problem as an ILP problem. The computation requirement of solving the ILP problem exceeds the capability of typical sensor nodes. Thus, we proposed a fuzzy-logic-based distribute decision-making algorithm that has low computation requirements and is robust to incomplete information. Our simulations demonstrate that the fuzzy logic algorithm performs well and mobile sensors can significantly increase sensor network lifetime.

References

Bruck, J., Gao, J. and Jiang, A. (2005) 'Localization and routing in sensor networks by local angle information', *Proceedings of the Sixth ACM MobiHoc*, UIUC, May.

Culler, D. and Mulder, H. (2004) 'Smart sensors to network the world', *Scientific American*, June.

Doherty, L., Ghaoui, L.E. and Pister, K.S.J. (2001) 'Convex position estimation in wireless sensor networks', *Proceedings of IEEE INFOCOM*, April.

Driankov, D., Hellendorf, H. and Reinfrank, M. (1993) *An Introduction to Fuzzy Control*, Springer-Verlag.

Du, X. and Lin, F. (2005) 'Improving sensor network performance by deploying mobile sensors', *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC)*, April.

Polastre, J., Mainwaring, R.A. and Culler, D. (2004a) 'Lessons from a sensor network expedition', *Proceedings of First European Workshop on Wireless Sensor Networks (EWSN)*, January, pp.307–322.

Polastre, J., Szewcyk, R., Mainwaring, A., Culler, D. and Anderson, J. (2004b) 'Analysis of wireless sensor networks for habitat monitoring', *Wireless Sensor Networks*, Kluwer Academic Pub, pp.399–423.

Rahimi, M., et al. (2003) 'Studying the feasibility of energy harvesting in a mobile sensor network', *Proceedings of the IEEE International Conference on Robotics and Automation*, May.

Wang, G., Cao, G. and Porta, T.L. (2004) 'Movement-assisted sensor deployment', *Proceedings of the IEEE INFOCOM*, April.

Zhang, M., Du, X. and Nygard, K. (2005) 'Improving coverage performance in sensor networks by using mobile sensors', *Proceedings of the IEEE Military Communication (MILCOM)*, October.

Zou, Y. and Chakrabarty, K. (2003) 'Sensor deployment and target localization based on virtual forces', *Proceedings of the IEEE INFOCOM*, April.