Design and evaluation of two distributed methods for sensors placement in Wireless Sensor Networks

Amjad Osmani
Department of Computer Engineering and Information Technology
[1] *Islamic Azad University, Saghez Branch, Saghez, Iran*
a.osmani@{iausaghez.ac.ir, qiau.ac.ir} [1]

Abstract

Adequate coverage is one of the main problems for distributed wireless sensor networks and The effectiveness of that highly depends on the sensor deployment scheme. Given a finite number of sensors, optimizing the sensor deployment will provide sufficient sensor coverage and save power of sensors for movement to target location to adequate coverage. In this paper, we apply fuzzy logic system to optimize the sensor placement after an initial random deployment. Based on Voronoi diagrams and Fuzzy logic, we design two distributed algorithms for controlling the movement of sensors. Simulation results show that our approaches maximiz the sensor coverage.

Keywords: *wireless sensor network (WSN), placement, robotic.*

---

1. INTRODUCTION

WIRELESS sensor networks consist of certain amount of small and energy constrained nodes [1-3]. A typical wireless sensor network consists of thousands of sensor nodes, deployed either randomly or according to some predefined statistical distribution, over a geographic region of interest. A sensor node by itself has severe resource constraints, such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory. However, when a group of sensor nodes collaborate with each other, they can accomplish a much bigger task efficiently. One of the primary advantages of deploying a wireless sensor network is its low deployment cost and freedom from requiring a messy wired communication backbone [1, 4].

For instance, a sensor network can be deployed in a remote island for monitoring wildlife habitat and animal behavior [5, 6], or near the crater of a volcano to measure temperature, pressure, and seismic activities. In many of these applications the environment can be hostile where human intervention is not possible and hence, the sensor nodes will be deployed randomly or sprinkled from air and will remain unattended for months or years without any battery replacement. Therefore, energy consumption or, in general, resource management is of critical importance to these networks. Sensor deployment strategies play a very important role in providing better QoS, which relates to the issue of how well each point in the sensing field is covered. Three types of coverage have been defined by Gage [7]:

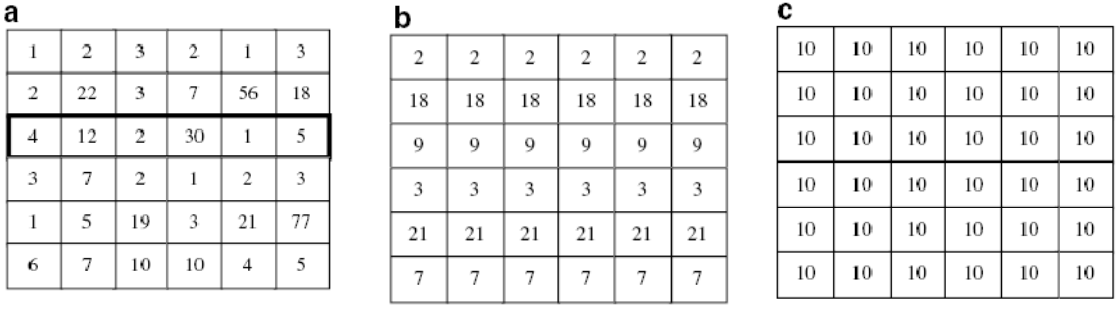
- *Blanket coverage: to achieve a static arrangement of sensor nodes which maximizes the detection rate of targets appearing in the sensing field.*
- *Barrier coverage: to achieve a static arrangement of sensor nodes which minimizes the probability of undetected penetration through the barrier.*
- *Sweep coverage: to move a number of sensor nodes across a sensing field, such that it addresses a specified balance between maximizing the detection rate and minimizing the number of missed detections per unit area.*

We focus mainly on the sweep coverage, where the objective is to deploy sensor nodes in strategic ways such that optimal area coverage is achieved according to the needs of the underlying applications

2. RELATED WORK

Several deployment strategies have been studied for achieving an optimal sensor network architecture which would minimize cost, provides high sensing coverage, be resilient to random node failures, and so on. Random placement does not guarantee full coverage because it is stochastic in nature, hence often resulting in accumulation of nodes at certain areas in the sensing field but leaving other areas deprived of nodes. Some of the deployment algorithms try to find new optimal sensor locations after an initial random placement and move the sensors to those locations, achieving maximum coverage (like our approach). These algorithms are applicable to only mobile sensor networks. Research has also been conducted in mixed-sensor networks, where some of the nodes are mobile and some are static and approaches are also proposed to detect coverage holes after an initial deployment and to try to heal or eliminate those holes by moving sensors. The strategy of sensor placement depends on the application of the distributed sensor network (DNS). For example in [8] the focus is on the grid-based placement and they apply the modified binary PSO algorithm for solving the problem. Considering the existence of many networks with high velocity and computational capabilities of these sensor networks, we can say that they have different applications for example in aviation, military, medical, robot, air forecasting, security and anti terrorism applications and also we can use them in very important infrastructures like power plants ,environmental and natural resource monitoring, and military applications like communication systems, commanding, reconnaissance patrols, looking –out etc [9][1].

Zou and Chakrabarty [10] proposed an algorithm as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors. It is assumed that sensors can move by “virtual force” with the force’s strength determined by node distance. Cao, Wang, La Porta, and Zhang [11] considered the problem of moving some sensors from their initial random placement in order to cover some areas which were not covered by either the nature of randomness or some other effects such as wind. It is also assumed that sensors can move after gathering some information from neighbors. The algorithm proceeds in rounds. In each round, each sensor then subtracts its sensing area from its Voronoi polygon, and moves in the direction of the largest uncovered piece of area. The process repeats until no further improvement is possible. The approach appears suitable when robots, equipped with sensors, are monitoring an area, which can also be monitored by some static sensors. An alternative approach may be to use face routing [12] to estimate the size of a hole, find its centroid, estimate the number of sensors which should move toward the centroid, and provide the best possible information to sensors for their move. Wang, Cao, and La Porta [13] proposed a proxy-based sensor deployment protocol. Instead of moving iteratively, sensors calculate their target locations based on a distributive iterative protocol. Current proxy sensors advertise the service of mobile sensors to their neighborhoods (up to certain parameter distance), searching for a better coverage location. They collect bidding messages and choose the highest bid. Then they delegate the bidder as the new proxy. Actual movement only occurs when sensors determine their final locations. If the bidding process is local, the sensor movement and the area-coverage gains may be restricted. If the bidding process includes neighbors at several hops distance, the communication overhead for bidding becomes significant. Bidding decisions are based on price (number of logical movements made so far) and distance that the moving sensors are physically supposed to move altogether. A procedure to prevent multiple healing is described, which includes some message overhead. The bidding criterion does not include lost area coverage for moving out of the current position. It is not certain whether the described procedure is always loop-free and always converging. The difference between sensing and transmission radii has a direct impact on message complexity. [14] Proposed a scan-based movement-assisted sensor deployment method (SMART) which uses scan and dimension exchange to achieve a balanced state. In SMART, a given rectangular sensor field is first partitioned into a 2-D mesh through clustering. Each cluster corresponds to a square region and has a cluster head which is in charge of bookkeeping and communication with adjacent cluster heads. Clustering is a widely used approach in sensor networks for its support for Design simplification. An example is shown in Fig. 1.



a

1	2	3	2	1	3
2	22	3	7	56	18
4	12	2	30	1	5
3	7	2	1	2	3
1	5	19	3	21	77
6	7	10	10	4	5

b

2	2	2	2	2	2
18	18	18	18	18	18
9	9	9	9	9	9
3	3	3	3	3	3
21	21	21	21	21	21
7	7	7	7	7	7

c

10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10
10	10	10	10	10	10

Fig 1. *Steps for SMART: (a) initial 2D mesh, (b) row scan and (c) column scan.*

Another idea has been described in [15] for the purpose of providing location service. If the network of static sensors is disconnected, then mobile sensors will send one message to each connected component and search several perimeters. In [16] three sensor relocation algorithms were proposed according to the mobility degree of sensor nodes. The first one, PSOA, regards the sensors in the network as a swarm and reorganizes the sensors by PSO, in the full sensor mobility case. The other two, relay shift based algorithm (RSBA) and energy–efficient fuzzy optimization algorithm (EFOA), assume relatively limited sensor mobility, i.e., the movement distance is bounded by a threshold, and to further reduce energy consumption. In [17] for the exposure estimation the whole network is thought to be a Voronoi diagram [18] based network which is formed considering all the cluster head of the network as a single point. The clustering algorithm used here is based on Delaunay triangulated sensor nodes. The key idea of this clustering method is taken from the clustering method used for key frame-based video summarization technique [19]. Some recent work focus on sensors with limited mobility, which is motivated by the DARPA project called Intelligent Mobile Land Mine Units (IMLM) [20]. In IMLM, the mobility system is based on a hopping mechanism. Chellapan et al. [21] studied a special hopping model in which each sensor can flip (or flop) to a new location only once. In addition, the flip distance is bounded. The deployment problem is then formulated as a minimum cost, maximum-flow problem. In [22] and [23], they presented a resource-bounded optimization framework for sensor resource management under the constraints of sufficient grid coverage of the sensor field. In [24], they formulated the sensor placement problem in terms of cost minimization under coverage constraints. In [25] Node placement in heterogeneous WSN is formulated using a generalized node placement optimization problem to minimize the network cost with lifetime constraint, and connectivity. In [26] they formulated and solved the sensor placement problem for efficient target localization in a sensor network. They developed a mathematical framework for the localization of the missile using multiple sensors based on Cramer-Rao Lower Bound (CRLB) analysis. In [27] they presented the practical problem of optimally placing the multiple PTZ cameras to ensure maximum coverage of user defined priority areas with optimum values of parameters like pan, tilt, zoom and the locations of the cameras. Moreover in [28] a heuristic algorithm is proposed based on Simulation Annealing Algorithm to solve this problem considering the coverage and cost limitations. In [29] they have used the Distribution Estimation Algorithms named LAEDA on sensor placement. In Learning Automata Estimation Distributed Algorithm (LAEDA), the independency of genome variables is assumed. In these algorithms a Learning Automata is used for each variable in genome. The number of actions of Learning Automata equals to number of permitted values for the corresponding variable of Learning Automata. For production of each genome sample, the Learning Automata of each variable is asked to select its own suitable action; afterwards, they give a corresponding value of selected action to the corresponding variable. Though, they can calculate the probability of a genome's production $X = (x_1, \dots, x_n)$ based on equation (1).

$$\mathrm{P}(\mathrm{X} = \mathrm{x}) = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{P}(\mathrm{X_i} = \mathrm{x_i}) = \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{Grad}_\mathrm{i}^\mathrm{j} \qquad (1)$$

Where , $1 \leq \mathrm{j} \leq \mathrm{r_i}$, so $\mathrm{Grad}_\mathrm{i}^\mathrm{j}$ equals to probability of action of corresponding j to value of $\mathrm{x_i}$ by i^{th} Learning Automata. By applying Automata in each stage, a number of N individual genomes are created, which is compatible with the number of population. Then the new population of genomes is evaluated using Evaluation Function, and Se genomes which are considered as the best genomes are chosen from this population. After applying some mechanisms which are dependent on Learning Automata Environment Model, a reinforcement signal vector is created and we apply the Learning process in each Learning Automata. Having accomplished the learning process, a new generation is produced and the above stages will be continued until a terminal condition is satisfied. Another model of probability distribution estimation algorithms is Population Based Incremental Learning [30] that is a technique which combines aspects of Genetic Algorithms and simple competitive learning. Like the GA, PBIL represents the solution set as a population set of solution vectors. In general, each solution vector in the population set, called an individual, is a possible solution of the problem. The population is produced randomly according to the probabilities specified in the probability vector. The population is evaluated and the knowledge about composing of the best individual in the population is acquired and then the probability vector is updated by pushing it towards generating good individuals in the population. After the probability vector being updated, a new generation population is produced according to the updated probability vector, and the cycle is continued until the termination condition is satisfied.

The rest of this article is organized as following. Section 3 is about fuzzy logic system. Section 4 details the proposed algorithms for distributed sensor placement problem. Simulations are presented in Section 5 and section 6 concludes this article through a summary.

3. OVERVIEW OF FUZZY LOGIC SYSTEM

Fig. 2 shows the structure of a rule-based type-1 FLS [31]. 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. [32]. 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 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 [33]. 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 … and x_p is F_p^l THEN y is G^l .l=1,…,M

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

$$\mu_{F_1^l}(x'_1) * \mu_{F_2^l}(x'_2) * ... * \mu_{F_p^l}(x'_p) = \Gamma_{i=}^{p} \mu_{F_i^l}(x'_i) \quad (2)$$

Where * and Γ 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.

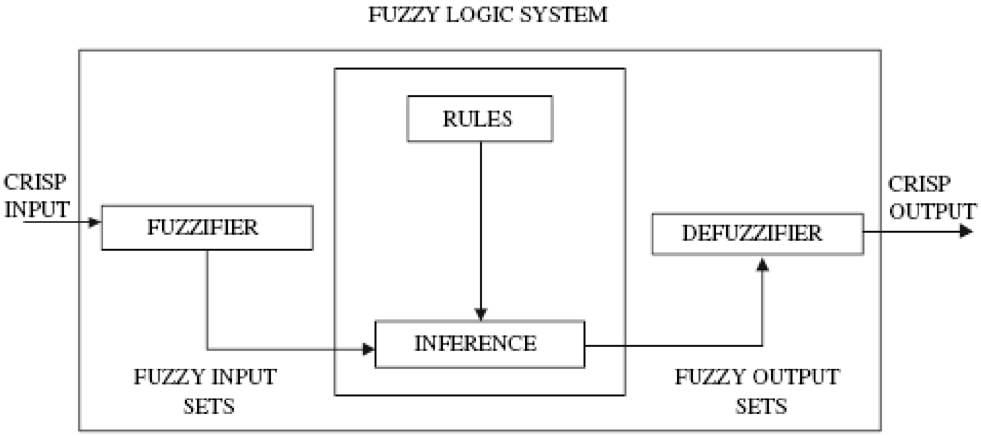


Fig 2. *The structure of a fuzzy logic system*

4. Suggested algorithms

4.1 FReD (Fuzzy based ReDeployment)

We apply FLS to the distributed sensor deployment problem and each sensor makes fully distributed decision on its movement based on FLS. Our algorithm begins 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 airdropping. Because of the randomness in initial deployment, very likely the sensor field will not be fully covered. Our algorithm then intends to re-deploy the sensors such that gains maximum field coverage. An ideal sensor deployment will have uniform distribution for better coverage. Sensor nodes are scattered in a sensing field with varying node densities. Each node has a sensing radius within which can sense data, and a communication radius within which can communicate with another node. One important criterion for being able to deploy an efficient sensor network is to find optimal decision. But in random deployment, coverage uniformity is hardly to achieve initially. In FReD-approach (Fuzzy based ReDeployment), each sensor calculates its voronoi and COVERAGE-FACTOR (the intersection of the polygon (VORONOI) and the sensing circle for sensor.) based on its voronoi as its PRE-STATE (under Fig.3). Then calculates ST-FACTOR (under Formula. 3) as its PRE-ST-FACTOR and target location based on Moving toward center of triangle resulting from current sensor and two neighbors of farthest Voronoi vertex. After that each sensor based on new locations calculates COVERAGE-FACTOR based on its voronoi as its NEXT-STATE, and ST-FACTOR as its NEXT-ST-FACTOR. Each sensor based on consequent of Table.1, if consequent is MOVE, with the probability of mentioned proceeds to moving to target location.

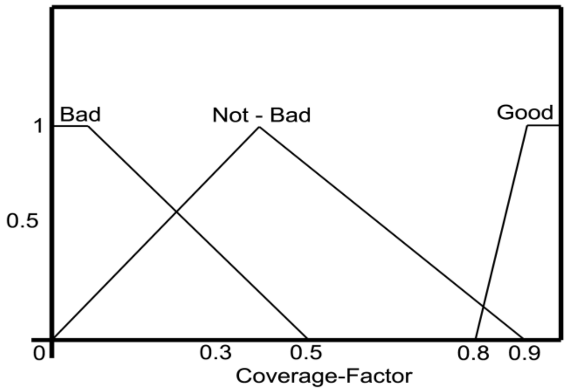


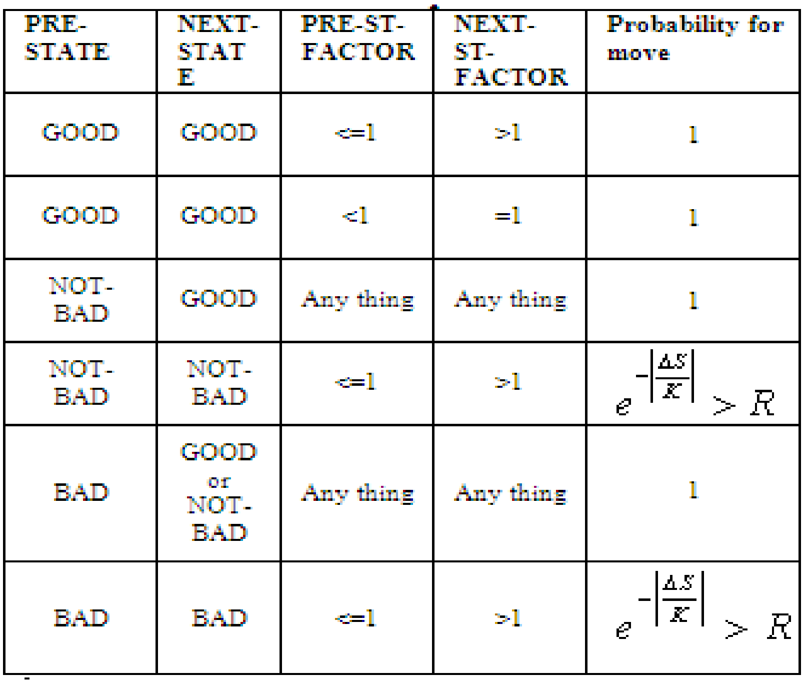
Fig 3. *State membership function.*

Each sensor based on its neighbors information and using the ST-factor and Table .1 decides whether to movement or not, where β is for tuning of ST-factor.

$$ST-factor=\left(\frac{G\_N\_C+((NB\_N\_C)/\beta)}{B\_N\_C}\right) \quad (3)$$

G_N_C : Number of neighbors in GOOD state
NB_N_C : Number of neighbors in NOT - BAD state
B_N_C : Number of neighbors in BAD state

Table 1. *Fuzzy rules for Movement or Not and consequent*



PRE-STATE	NEXT-STATE	PRE-ST-FACTOR	NEXT-ST-FACTOR	Probability for move
GOOD	GOOD	<=1	>1	1
GOOD	GOOD	<1	=1	1
NOT-BAD	GOOD	Any thing	Any thing	1
NOT-BAD	NOT-BAD	<=1	>1	$e^{-\left\lvert\frac{\Delta S}{K}\right\rvert} > R$
BAD	GOOD or NOT-BAD	Any thing	Any thing	1
BAD	BAD	<=1	>1	$e^{-\left\lvert\frac{\Delta S}{K}\right\rvert} > R$

$$e^{-\left|\frac{\Delta S}{K}\right|} > R \begin{cases} K \propto {}^{1}\!/_{\#Neighbors} \\ \Delta S \propto \text{ Difference between} \\ \text{NEXT - STATE \& PRI - STATE} \\ R \propto \text{random number in [0 - 1]} \end{cases}$$

4.2 FSPNS (Fuzzy Sensor Placement based on Neighbors State)

FSPNS uses 3+9 fuzzy rules under Table 2. The input parameters of the fuzzy logic are the linguistic variables of state of sensor and ST-FACTOR value for current sensor and density of neighbors and average distance from neighbors and the output is type of policy for move. The 3 rules are for reaching to the next 9 rules under Table 2.

Table 2. *Fuzzy rules for type of Policy and consequent*

State of sensor	ST-FACTOR	Move		
		Density of Neighbors	Average Distance from Neighbors	Policy
Bad	ANY			
NOT-Bad	ANY except (>1)			
Good	< 1	LOW	FAR	P1
		HIGH	FAR	P2
		LOW	NEAR	P3
		HIGH	NEAR	P2
		Low	M	P3
		High	M	P1
		M	Far	P2
		M	Near	P2
		M	M	P1

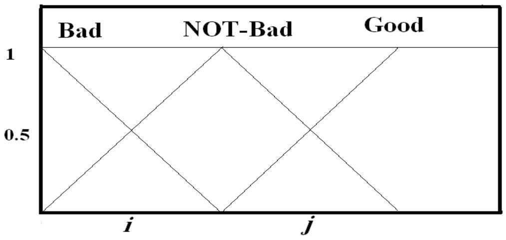


Fig 4. *State membership function.*

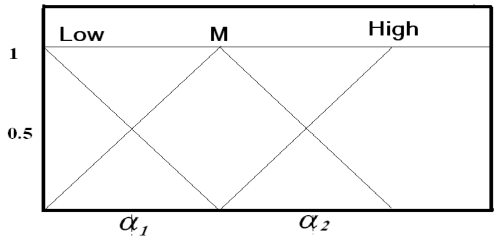


Fig 5. *Density of neighbors membership function.*

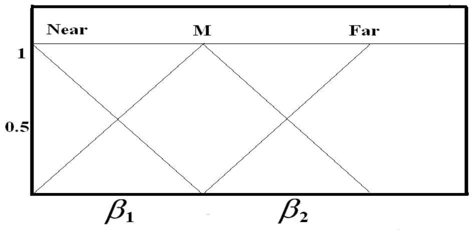


Fig 6. *Average distance membership function*

Some of the deployment algorithms try to find new sensor locations after an initial random placement and move the sensors to those locations, achieving maximum coverage or good coverage (like our approach). Those algorithms are applicable to only mobile sensor networks. Research has also been conducted in mixed-sensor networks, where some of nodes are mobile and some are static; and approaches are also proposed to detect coverage holes after an initial deployment and to try to heal or eliminate those holes by moving sensors. Our algorithm begins 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 airdropping. Because of the randomness in initial deployment, very likely the sensor field will not be fully covered. Our algorithm then intends to re-deploy the sensors such that gains good coverage. Each sensor node has a sensing radius within which can sense data, and a communication radius within which can communicate with another node. In FSPNS (Fuzzy Sensor placement based on neighbors state), each sensor calculates its voronoi and COVERAGE-FACTOR (the intersection of the polygon (VORONOI) and the sensing circle for sensor) based on its voronoi as its STATE (under Fig. 4). In Fig. 4, we use (i, j) that are thresholds between area of bad and not-bad and between not-bad and good. Then Each sensor based on neighbors information STATE, calculate ST-FACTOR and using Table 2 decide about Movement Or Not and if consequent is MOVE then based on neighbors density (under Fig. 5) and average distance from neighbors (under Fig. 6) and using Table 2 decide about type of policy for finding target location. As Fig. 5 shows, we use two thresholds ($\alpha 1$, $\alpha 2$). $\alpha 1$ is used between Low and M and $\alpha 2$ is used between M and High. As Fig.6 shows, we use two thresholds ($\beta 1$, $\beta 2$). $\beta 1$ is used between Near and M and $\beta 2$ is used between M and Far. (Bad, NOT-Bad, Good) and (Low, M, High) and (Near, M, Far) are linguistic variables to map the input fuzzy variable to the output variable. For example if State of sensor is Good and ST-FACTOR is less than one then sensor decide to move. Then if the Density of Neighbors is HIGH and the Average Distance from Neighbors is NEAR then the type of policy is P2. Each sensor using the ST-factor, its state and Table 1 decides whether to movement or not, where β is for tuning of ST-factor.

- *A: Number of neighbors that are in GOOD state.*
- *B: Number of neighbors that are in NOT-BAD state.*
- *C: Number of neighbors that are in BAD state.*

Policies:

- *P1: Moving toward farthest Voronoi vertex.*
- *P2: Moving toward center of triangle resulting from current sensor and two neighbors of farthest Voronoi vertex.*
- *P3: Moving toward* $(\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n})$ *such that n is number of neighbors and (x, y)* is coordinate of points of between current sensor and neighbors.

Policy of P1 in our work is used when the Average Distance from Neighbors is Far or M. Policy of P2 in our work is used when the Average Distance from Neighbors is Far or Near. Policy of P3 in our work is used when the Density of Neighbors is Low.

In this article, we do not consider Position-Clustering [9].But we have an idea for deal with Position-Clustering. In this case (as Fig. 7 shows), sensors located inside the clusters cannot move for several rounds, since their Voronoi polygons are well covered initially. The algorithm in [9] “explodes” the cluster to scatter the sensors apart. Our idea is simple. Here we use an example. As you can see in Fig. 8, the sensors can before running FSPNS-approach check two parameters of Density of Neighbors (DN) and Average Distance (AD) and after detecting that are in center of Cluster use formulas (4 to 8) for calculating new locations of neighbors for temporary solving of cluster problem in that area .

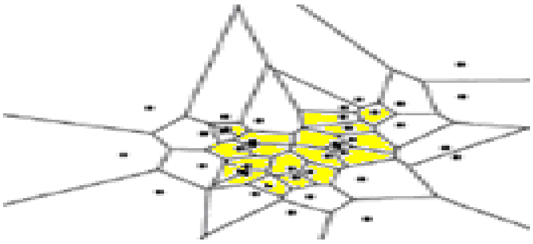


Fig 7. *A Cluster problem*

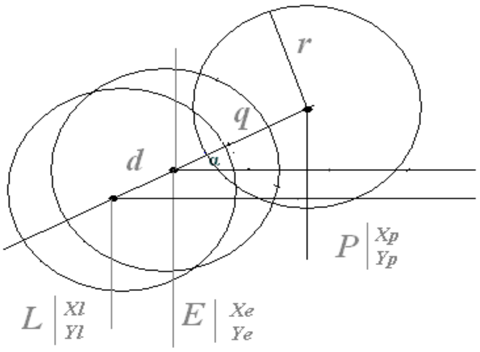


Fig 8. *An example for solving Cluster problem*

$$m = (Y_e - Y_l)/(X_e - X_l) \tag{4}$$

$$\widehat{\alpha} = \arctan(m) \tag{5}$$

$$q = 2r - (d + f) \tag{6}$$

$$X_p = X_e + q\cos(\widehat{\alpha}) \tag{7}$$

$$Y_p = Y_e + q\sin(\widehat{\alpha}) \tag{8}$$

d is current distance between two sensors and q is the final distance and f is for tuning the formula.” The sensor that are in center of Cluster” means DN is greater than a DN-threshold and AD is less that an AD-threshold. The sensor that are in center of Cluster after calculating new locations of neighbors sends a one hop packet (that are any-cast) that are containing new locations for some neighbors . We say some neighbors that are needed for movement to removing cluster or exploding of that. Note that our algorithm in this article does not consider Cluster problem but we offer our idea and in our other works (next article) we extend this problem and extend our idea for solving this problem.

5. Simulation Results

5.1 FReD

We investigate various number of sensors deployed in a field of 200 × 200 m×m area. We distribute seven different numbers of sensors, ranging from 20 to 80, in increments of 10 sensors. The initial deployment follows the random distribution .To evaluate under different parameter settings, we run 10 experiments based on different initial distribution and calculate the average results and set β to 2. Evaluation of our scheme follows two criteria: mean coverage, and mean travel distance. We compare the performance of our algorithm With VEC-algorithm [9], at 40 m sensing range (Rs = 40 m) and 80 m communication range (Rc = 80 m).Sensing range means the circle with radius of Rs that sensors can sense .Communication range means the circle with radius of Rc that sensors can communicate with each other (send and receive). Fig.9 shows the coverage of the initial random deployment, the coverage after VEC and the coverage after using FReD. We ran three iterations for all three schemes. After FReD was used, the coverage reached approximate 93%. Fig.9 shows that after three iterations, coverage is always above 93% and with the number of sensors between 40 and 60 the coverage reaches to 100%. The key point behind of FReD is that algorithm tries to movement of sensors as in layers and outer layers are maximum movement and inner layers try to minimum movement or not. Fig.10 shows different numbers of sensors that the coverage increases vastly within the first 3 to 4 iterations. After that coverage approaches stable for all different number of sensors. Fig.11 shows Average movement (distance) against number of sensors deployed (after variable Iterations), which is decreasing by increasing iterations and sensors, because fuzzy rules are used to deciding about movement and sensors move when their states will be better than before.

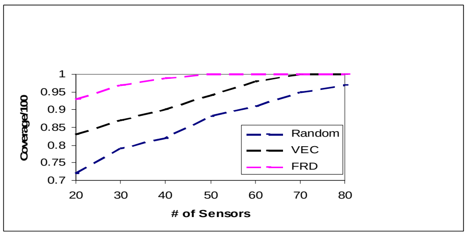


Fig 9. *Coverage vs. number of sensors deployed (after three Iterations)*

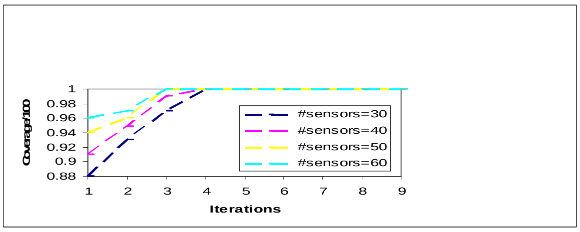


Fig 10. *Coverage vs. Iteration*

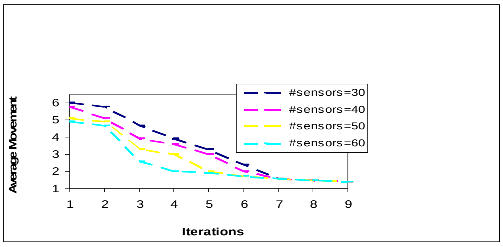


Fig 11. *Average movement vs. number of sensors deployed (after variable Iterations)*

5.2 FSPNS

We investigate various number of sensors deployed in a field of 300 × 300 m×m area. We distribute eight different numbers of sensors, ranging from 25 to 60, in increments of 5 sensors. The initial deployment follows the random distribution .To evaluate our algorithm, we set β to 2. We simulate our algorithm at 50 m sensing range (Rs = 50 m) and 100 m communication range (Rc = 100 m).Sensing range means the circle with radius of Rs that sensors can sense .Communication range means the circle with radius of Rc that sensors can communicate with each other (send and receive). $(\alpha 2-\alpha 1)$ is equal to 7 and $(\beta 2-\beta 1)$ is 8m. Fig. 12 shows the coverage of the initial random deployment and the coverage after using FSPNS. Fig. 12 shows that after three iterations, with the number of sensors between 25 and 60 the coverage reaches to 100%. Random placement does not guarantee full coverage because it is stochastic in nature, hence often resulting in accumulation of nodes at certain areas in the sensing field but leaving other areas deprived of nodes. Fig. 13 shows different number of sensors that cover network area duration of one to sex iterations. The coverage increases vastly within the first 3 to 5 iterations. After that coverage approaches stable for different number of sensors. Fig. 14 shows coverage against parameter of (j-i). As you see, by increasing the parameter, coverage increase too, and reach to 100%. The parameter is difference thresholds between areas of bad and not-bad and between not-bad and good.

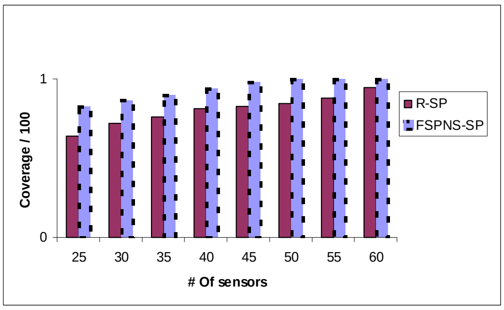


Fig 12. *Coverage vs. number of sensors*

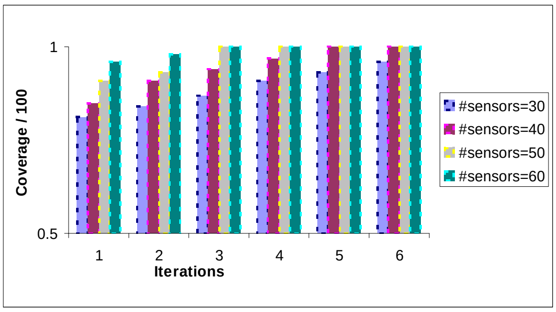


Fig 13. *Coverage vs. Iteration*

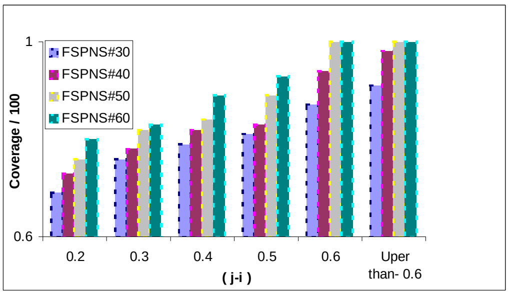


Fig 14. *Coverage vs. (j-i)*

6. Conclusion and Future Works

In this paper, our algorithms after random deployment of sensors at the beginning of network setup start to redeployment of sensors to increase the coverage on the sensor field. FSPNS after deciding for movement uses one of three policies depend on fuzzy rules based on neighbors density and average distance from them. FSPNS is simple and simulation shows coverage enhancement related to random coverage. FReD uses neighbor's information about location and state to decide on movement. Since sensor placement in the Wireless Sensor Network (WSN) is important, we should find better intelligent algorithms which use less energy which is very important.

References

[1] C. Y. Chong, S. P Kumar, "*Sensor networks: evolution, opportunities, and challenges"*, Proceedings of the IEEE, Vol. 91, No. 8, pp. 1247–1256, 2003.

[2] G.J.Pottie,"*Wireless sensor networks*", Proceedings of Information Theory Workshop, pp.139–140, June 1998.

[3] G.J.Pottie, W.Caiser," *Wireless sensor networks*", Proceedings of the Commun .ACM,Vol 43,No.5, 2000.

[4] I.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci,"*Wireless sensor networks A survey"*, Proceedings of the Comput . Networks, Vol 38, No.2, pp.393–422, 2002,

[5] A. Mainwaring, J.Polastre, R.Szewczyk, D.Culler, J.Anderson," *Wireless sensor networks for habit at monitoring"*, Proceedings of the 1st ACMInt.Workshop on Wireless Sensor Networks and Applications (WSNA’02), pp.88–97, 2002.

[6] H. Qi, S. S. Iyengar, K. Chakrabarty," *Distributed sensor fusion – a review of recent research"*, Proceedings of the Journal of the Franklin Institute, Vol. 338, pp. 655–668, 2001.

[7] D.W.Gage,*"Command control for many-robot systems",* Proceedings of the 19th Annual AUVS Technical Symp. Reprinted in Unmanned Syst.Mag.Vol 10, No.4, pp.28–34, 1992.

[8] Khezri, S., Faez, K., Osmani, A., "*Modified Discrete Binary PSO Based Sensor Placement in WSN*", International Conference on Computational Intelligence and Communication Networks, (IEEE) CICN, India, 2010.

[9] I.F. Akyildiz, W. Su, .Sankarasubramaniam, E. Cayirci, "*Wireless Sensor Networks: a survey*", Proceedings of the Computer Networks, Published by Elsevier Science, Vol. 38, No. 4, pp. 393-422, 2002.

[10] Y.Zou, K.Chakrabarty,"Sensor deployment and target localization in distributed Sensor networks ", Proceedings of the Trans .IEEE Embedded Comput .Syst. Vol 3, No.1, 2004.

[11] G. Wang, G. Cao, T. La Porta, *"Movement-assisted sensor deployment*", Proceedings of the IEEE Transactions On Mobile Computing, Vol.5, No.6, 2006.

[12] P.Bose, P.Morin, I.Stojmenovic, J, Urrutia,*"Routing with guaranteed delivery in Adhoc wirelesss networks"*, Proceedings of the Wireless Networks, Vol 7, No.6, 2001.

[13] G. Wang, G. Cao, T. La Porta,"*Proxy-based sensor deployment for mobile sensor networks",* Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04), 2004.

[14] Sh. Yang, M. Li, J. Wu, *"Scan-Based Movement-Assisted Sensor Deployment Methods in Wireless Sensor Networks"*, Proceedings of the IEEE Transactions On Parallel And Distributed Systems, Vol. 18, No. 8, 2007.

[15] I.Stojmenovic,*" A Scalable Quorum Based Location Update Scheme for Routing In Ad Hoc Wireless Networks"*. Technical Report TR-99-09, SITE, University of Ottawa, 1999.

[16] X.WU, J. CHO, B. J.D'AURIOL, S. LEE,*" Mobility-Assisted Relocation for Self-Deployment in Wireless Sensor Networks",* Proceedings of the IEICE TRANS.COMMUN., Vol E90–B, NO.8, 2007.

[17] Md. O. Rahman, Md. A. Razzaque and Ch. S. Hong,*" Probabilistic Sensor Deployment in Wireless Sensor Network: A New Approach"*, Proceedings of the ICACT, pp. 12-14,2007.

[18] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M. B. Srivastava,*" Coverage Problems in Wireless Ad-hoc SensorNetworks"*, Proceedings of the International Conference on Mobile Computing and Networking, 2001.

[19] M. Padmavathi, R. Yong, Y. Yelena,*" Key frame-based Video Summarization using Delanunay Clustering"*, Proceedings of the International Journal on Digital Libraries, Vol 6, pp. 219 - 232, 2006.

[20] http://www.darpa.mil/ato/progarms/shm/index.html, 2007.

[21] S. Chellappan, X. Bai, B. Ma, D. Xuan, "*Mobility Limited Flip-Based Sensor Networks Deployment,"* Proceedings of the IEEE Mobile Ad Hoc and Sensor Systems (MASS), 2005.

[22] S.S. Dhillon, K**.** Chakrabarty, S.S**.** Iyengar, "*Sensor Placement for Grid Coverage under Imprecise Detections*," Proceedings of the Fifth International Conference on Information Fusion**,** , Vol. 2, No.3, pp. 1581-1587, July 2002.

[23] S.S. Dhillon, K. Chakrabarty, "*Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks,"* Proceedings of the lEEE WCNC, Vol. 3, pp. 1609-1614, 2003.

[24] K. Chakrabarty, S.S**.** Iyengar, H. Qi, E. Cho, *"Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks,*" Proceedings of the IEEE Trans. on Computers, Vol. 51, No. 12, 2002.

[25] P. Sasikumar, S. K. Vasudevan, C. Vivek , V. Subashri, "*Heuristic Approaches with Probabilistic Management for Node Placement in Wireless Sensor Networks"*, Proceedings of the International Journal of Recent Trends in Engineering, Vol. 2, No. 4, 2009.

[26] R. Rajagopalan, R. Niu, Ch. K. Mohan, P. K. Varshney, A.L. Drozd, "*Sensor placement algorithms for target localization in sensor networks",* Proceedings of the IEEE Radar Conference, pp. 1-6, 2008.

[27] S. Indu, S. Chaudhury, N. R. Mittal, A. Bhattacharyya, "*Optimal Sensor Placement for Surveillance of Large Spaces",* Proceedings of the Third International Conference on Digital Object Identifier, pp. 1-8, 2009.

[28] F. Y. S. Lin and P. L. Chiu, "*A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks*", Proceedings of the IEEE Communication letters, Vol. 9, No. 1, 2005.

[29] S. Khezri, A. Osmani, M. Gholami, "*Estimation of Distribution Algorithm Based on Learning Automata for Sensors Placement in Wireless Sensor Networks*", Proceedings of the 3nd National Conference on Computer/Electrical and IT Engineering (CEIC), Iran,2011.

[30] S. Khezri, M. Meybodi, A. Osmani, "*Fuzzy Adaptive PBIL based Sensor Placement in WSN*", Proceedings of the International Symposium on Computer Networks and Distributed Systems, (IEEE)CNDS, Iran, 2011.

[31] J. M. Mendel, "*Fuzzy logic systems for engeneering: a tutorial*", Proceedings of the IEEE, Vol. 83, No. 3, pp. 345–377, 1995.

[32] J. M. Mendel, "*Uncertain Rule-Based Fuzzy Logic Systems*", Prentice-Hall, Upper Saddle River, NJ, 2001.

[33] T. J. Ross, "*Fuzzy Logic with Engineering Applications*", McGraw-Hill International Editions, Singapore, 1997.