

Optimal Deployment with Self-Healing Movement Algorithm for Particular Region in Wireless Sensor Network

Fan ZHU, Hongli LIU, Shugang LIU, Jie ZHAN

College of Electrical and Information Engineering, Hunan University, Changsha, China

E-mail: henry1214ars@yahoo.com.cn

Received July 29, 2009; revised August 15, 2009; accepted August 20, 2009

Abstract

Optimizing deployment of sensors with self-healing ability is an efficient way to solve the problems of coverage, connectivity and the dead nodes in WSNs. This work discusses the particular relationship between the monitoring range and the communication range, and proposes an optimal deployment with self-healing movement algorithm for closed or semi-closed area with irregular shape, which can not only satisfy both coverage and connectivity by using as few nodes as possible, but also compensate the failure of nodes by mobility in WSNs. We compute the maximum efficient range of several neighbor sensors based on the different relationships between monitoring range and communication range with consideration of the complex boundary or obstacles in the region, and combine it with the Euclidean Minimum Spanning Tree (EMST) algorithm to ensure the coverage and communication of Region of Interest (ROI). Besides, we calculate the location of dead nodes by Geometry Algorithm, and move the higher priority nodes to replace them by another Improved Virtual Force Algorithm (IVFA). Eventually, simulation results based-on MATLAB are presented, which do show that this optimal deployment with self-healing movement algorithm can ensure the coverage and communication of an entire region by requiring the least number of nodes and effectively compensate the loss of the networks.

Keywords: Optimal Deployment, Self-Healing Movement, Particular Region, Euclidean Minimum Spanning Tree (EMST), Improved Virtual Force Algorithm (IVFA)

1. Introduction

Sensor networks consist of a large number of small, light-weight, highly power-constrained, and inexpensive wireless nodes called sensors. Sensors are equipped with detectors for intrusion, sensing changes in temperature, humidity, chemicals, or any other characteristic of the environment that needs to be monitored. The data about the environment is constantly observed, consolidated, and sent to a monitor or Base Station (BS). Data transmission from the sensors to the BS can be periodic, event-triggered, or in response to a query from the BS. While each sensor node has limited computation capabilities and usually non-rechargeable battery power, the collaboration among thousands of sensors deployed in a region makes sensor networks a powerful system for observation of the environment [1]. The data sensed by

the sensors is generally highly critical, and may be of scientific or strategic importance. Hence, the coverage provided by sensor networks is a very important criterion of their effectiveness. Special emphasis is placed on coverage especially in tactical applications such as surveillance and reconnaissance. Sensors can easily be used for the perilous and demanding duties of observing landscapes for intrusion detection [2]. In a wireless sensor network, the reasonable deployment of sensors should take both coverage and connectivity into account. Coverage requires that any physical field in a sensing region can be monitored by at least one node. Connectivity requires that each node is under the range of communication of its neighbor sensors. All these nodes can consist of an Ad-hoc network, and also transmit data packets to the BS. On the other hand, as time progresses the sensor nodes may die randomly due to malfunction, energy ex-

haustion or malicious destruction. All these factors result in holes of coverage and connectivity in WSNs, which makes the system unable to meet the performance criterion.

2. Related Works

Recently, a lot of approaches were advanced to solve these problems of coverage, connectivity and dead nodes in WSN. The work in [3,4] discusses how to adjust the locations of the nodes to satisfy the coverage in an open space, but without considering the boundary or obstacle. The grid algorithm in [5,6] is an appropriate way to ensure coverage and connectivity when there are a few nodes in the sensing region, however, with increasing nodes, it would be low efficient. The work in [7] does consider both coverage and communication, but it defaults that the range of sensing and communication are equal to each nodes without discussing varied situations respectively because of the different ranges between sensing and communication. The intersection point of the two dead nodes' neighboring sensors is used to decide where available node moves towards based on the random deployment in [2,8]. The Virtual Force Algorithm (VFA) strategy to enhance the coverage after an initial random placement of sensors is proposed by [9,10], but the shortest distance path among nodes is not rectified during the process of movement. Works [11,12] efficiently adjust the sensor placement after an initial random deployment and apply fuzzy logic theory to handle the uncertainty in sensor deployment problem.

3. Problem Formulation

Assuming a sensing field, the range of communication of each sensor in this region is R_c , within which it can transmit data packets to other sensors. Also, the sensing distance is R_s . The areas of each node's coverage and connectivity are assumed as two ideal circles respectively. K. Kar and S. Banerjee default $R_c=R_s$ in work [7], which satisfy the coverage and connectivity of the region. However, it is not realistic to analyze this issue by defining $R_c=R_s$ simply. We discuss different situations based on different relationships between R_c and R_s in order to adopt an appropriate deployment approach. According to literature [13] D. Pompili the two adjacent sensors, which are separated by no more than $\sqrt{3} R_s$, can ensure effective coverage of the surrounding region. Thus, we can figure out the deployment approaches that respectively regard coverage or connectivity as the first choice in open space. Our reform does not let nodes restrict by any choice, but simultaneity satisfy coverage and connectivity by the least number of nodes. Secondly, another important issue is how to deploy sensors effect-

tively in that region with boundary and obstacle. Because the boundary or the obstacle may limit the distance of sensing and communication, the approach of placing this kind of region is not different from placing in open space. On the other hand, the sensor nodes may die randomly due to malfunction, energy exhaustion or malicious destruction as time progresses. All these factors result in holes of coverage and connectivity in WSN, which makes the system unable to meet the performance criterion. In this paper, we also propose another movement approach to compensate the loss based on the mentioned optimal deployment algorithm. Below, we discuss how to deploy in particular region.

4. Optimal Deployment with Self-Healing Movement Algorithm for 'Particular Region'

The 'Particular Region' is a closed area or semi-enclosed area with boundary or obstacle which is consisted of unregulated polygons and arches. The optimal deployment algorithm can ensure this region's coverage and connectivity by the least sensors. However, researching on the deployment algorithm in an open space is regarded as the foundation to analyze the different situations in a particular region. This work discusses the deployment in an open space firstly, and then we improve and summarize the specific deployment algorithm in a particular region. All nodes are deployed above the ground about one meter to ensure the most optimal channel.

4.1. Deploying Sensors in an Open Space

Multi-line sensor arrays effectively resolve the issue. Below we study on deploying sensors in an open area without obstacles, and then extend to the deployment method in particular area with boundaries and obstacles.

Firstly, we establish a two-dimensional coordinates without boundaries or obstacles, and deploy lines of sensors, it guarantees the entire coverage of both adjacent nodes and each row. As the adjacent nodes can communicate with each other, if it is required to maintain the whole region's connectivity, we can add some of the sensors between adjacent lines to ensure it.

Case 1: $R_c \leq \sqrt{3} R_s$, the distance of adjacent nodes at each line is set R_c , which guarantees the coverage of adjacent nodes. Because $R_c \leq \sqrt{3} R_s$, the width of belt-like region that covered by a row of sensors is

$$2 \times \sqrt{R_s^2 - \frac{R_c^2}{4}}.$$

The difference value of two adjacent lines of nodes on the Y-axis is $R_s + \sqrt{R_s^2 - \frac{R_c^2}{4}}$, while $\pm \frac{R_c}{2}$ on the

X-axis. The above-mentioned method guarantees the coverage of the entire region. Figures 1–3 show three possible conditions. For $R_c < \sqrt{3} R_s$, what should be paid more attention to is that the method only satisfies communication property between adjacent nodes but not adjacent lines.

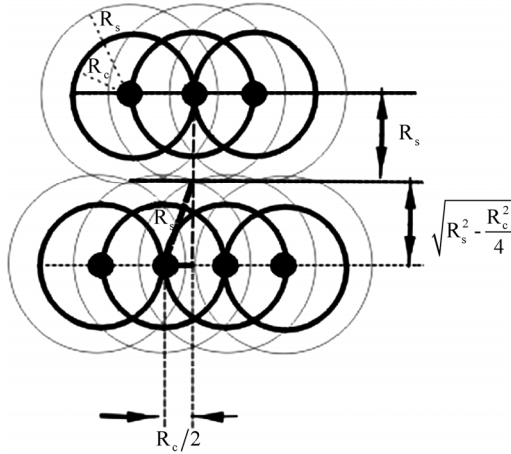


Figure 1. Deployment when $R_s > R_c$.

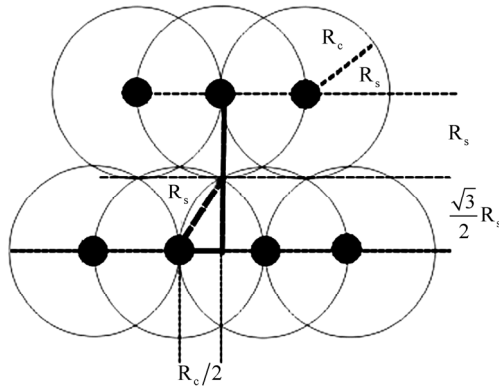


Figure 2. Deployment when $R_s = R_c$.

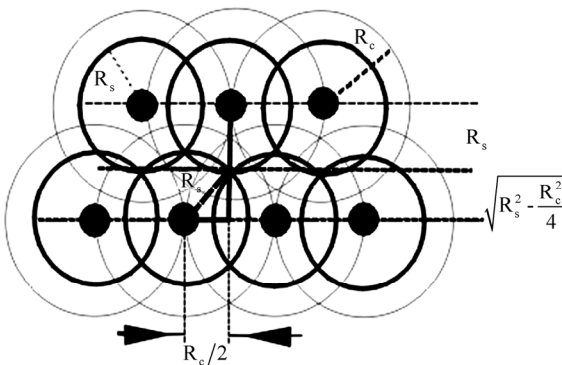


Figure 3. Deployment when $R_s < R_c < \sqrt{3} R_s$.

Case 2: $R_c > \sqrt{3} R_s$, as the smaller R_s , if we continue to adopt above-mentioned methods, which would lead the blind region that is not monitored between two adjacent lines, and also result in a waste of sensor nodes. So here's a typical use of the principle of hexagonal fabric which is more reasonable, and set the two adjacent sensors by R_s as Figure 4 shows. Therefore, it ensures the regional coverage and connectivity.

4.2. Deploying Sensors in Particular Area

For placing in particular regions, we can sum up the rules of deployment in two-dimensional coordinates from analyzing on a large area. First, establishing the two-dimensional coordinates, and assuming an initial node $S_{(0,0)} = (x_0, y_0)$ which is nearest to the origin than other nodes. According to above conclusion, the node $S_{(1,0)} = (x_0 + R_c, y_0)$, which is deployed next to the initial node along the positive x-axis direction. While the node $S_{(0,1)} = (x_0 + R_c/2, y_0 + R_s + \sqrt{R_s^2 - R_c^2/4})$ deployed first on the second row that is close to initial row along the positive y-axis direction.

$$S_{(2,2)} = (x_0 + 2R_c, y_0 + 2R_s + 2\sqrt{R_s^2 - R_c^2/4}).$$

By the same token any node's position placed by the above deployment algorithm in the two-dimensional coordinates can be calculated. (Note: The odd and even lines are different):

$$S_{(n',2n)} = (x_0 + n'R_c, y_0 + 2nR_s + 2n\sqrt{R_s^2 - R_c^2/4}), \quad (1)$$

$$S_{(n',2n+1)} = (x_0 + R_c/2 + n'R_c, y_0 + (2n+1)R_s + (2n+1)\sqrt{R_s^2 - R_c^2/4}), \quad (2)$$

$$n' = (0, 1, 2, \dots, \infty), n = (0, 1, 2, \dots, \infty).$$

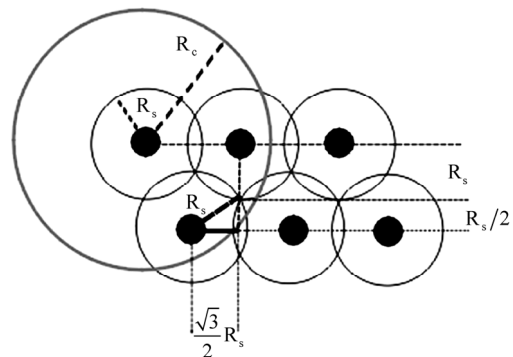


Figure 4. Deployment when $R_c > \sqrt{3} R_s$.

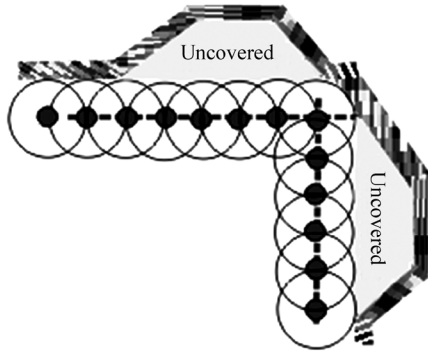


Figure 5. This approach leads to uncovered area.

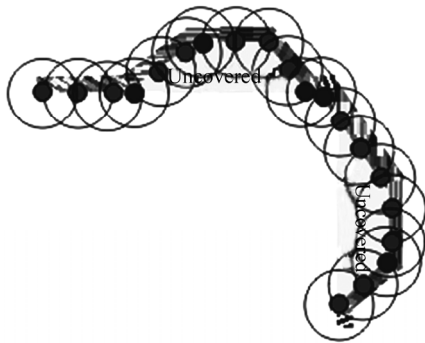


Figure 6. Deploying along the boundary (nodes in the corners are redundant).

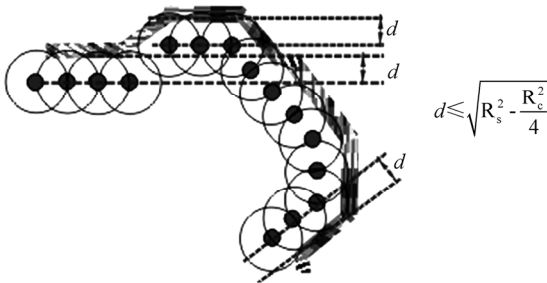


Figure 7. Improved deployment.

Of course, the node expression derived would change if we set different initial nodes. We need to be flexible in establishing the appropriate two-dimensional coordinates depending on the different shape of the area. However, the above-mentioned connectivity, which is limited that only ensures sensing data exchange between adjacent nodes at the same line, is limited. The following work focusing on the deployment of particular area will discuss how to ensure entire network's connectivity.

Assuming several nodes in a particular area with boundaries and obstacles are deployed as Figure 5, it results in uncovered area. The approach as Figure 6 shows meets the whole coverage and connectivity, but in

order to ensure the communication of adjacent nodes not to be blocked by obstacles, there are extra nodes to be added at the corner of the boundary, so it certainly wastes sensors. The deployment method in particular area can be improved on the basis of the above research as Figure 7 shows.

Assuming d is the width of uncovered area:

Case 1: $d \leq \sqrt{R_s^2 - \frac{R_c^2}{4}}$, and $R_c \leq \sqrt{3} R_s$, the distance

between nodes and boundary is set $\sqrt{R_s^2 - \frac{R_c^2}{4}}$, and two adjacent nodes are separated by R_c .

Case 2: $d \leq \sqrt{R_s^2 - \frac{R_c^2}{4}}$, and $R_c > \sqrt{3} R_s$, the distance

between nodes and boundary is set $\sqrt{R_s^2 - \frac{R_c^2}{4}}$, and two adjacent nodes are separated by $\sqrt{3} R_s$. This deployment method can satisfy coverage and connectivity.

Case 3: $d > \sqrt{R_s^2 - \frac{R_c^2}{4}}$, no matter what relationship

between R_c and R_s is, the method is as well as the deployment approach in large area.

In this way, both the coverage of whole area and the connectivity of adjacent nodes are guaranteed by the least number of nodes.

However, only satisfying connectivity of adjacent nodes on the same line is not enough to make all the nodes form an Ad-hoc network. In this paper, the EMST (Euclidean Minimum Spanning Tree) algorithm is introduced to estimate the communication location of the longest boundary, and also it combines with geometric analysis to solve the entire network problem of connectivity.

Assuming T area, and Choose any point S that can correspond to a leaf node of T , and all $\{S\}$ are defined as subsets to C in T , set $C \leftarrow \{S\}$, Also set $K=0, K \rightarrow K+1$. Choose any $S' \in C$. The R_c -disk which is chosen as D_k centered at S' . Move any points in C which are covered by D_k . Set I_k as the point of intersection by D_k and the boundary of T . For each point $S'' \in I_k$, including $S'' \in C$, if $S'' \notin D_1 \cup D_2 \cup D_3 \cup \dots \cup D_{k-1}$. So the path from initial S to S'' in T is covered by $D_1 \cup D_2 \cup D_3 \cup \dots \cup D_{k-1} \cup D_k$ completely. The deployment of specific path can be regarded as the geometric issues. The straight-line distance between two lines of

nodes is $R_s + \sqrt{R_s^2 - \frac{R_c^2}{4}}$. According to the parallelogram principle, two diagonal d_1, d_2 respectively is:

$$d_1 = \left(R_s + \sqrt{R_s^2 - \frac{R_c^2}{4}} \right) + \frac{R_c^2}{4}; \quad (3)$$

$$d_2 = \left(R_s + \sqrt{R_s^2 - \frac{R_c^2}{4}} \right) + \frac{9R_c^2}{4}; \quad (4)$$

We choose an appropriate diagonal depending on the different shape of particular area, and calculate the number of complementary sensors: $\frac{d_1}{R_c}$ or $\frac{d_2}{R_c}$. These added sensors are separated by the equal distance on the diagonal as shown in Figure 8. To sum up, the communication path among all lines is established, and all sensors ensure the connectivity in entire network.

4.3. Self-Healing Movement

Even if the applications of above-mentioned optimal deployment algorithm can satisfy both connectivity and coverage, the sensor nodes may die randomly due to malfunction, energy exhaustion or malicious destruction as time progresses. All these factors result in holes of coverage and connectivity in WSN, which makes the system unable to meet the performance criterion. In this paper, we propose another movement approach to compensate the loss based on the mentioned optimal deployment algorithm. After the initialization of network, we assume that every node is equipped with the capability of movement, and acquires their location and communication neighbors respectively by localization protocol as [14,15] referred. Also communication neighbors will detect when any node dies. Then these nodes' neighbors broadcast a packet containing its location to next one-hop node which continues to transmit to another until all the nodes get the message of hole. The following section presents our movement algorithm:

According to above-mentioned deployment algorithm, in order to satisfy the whole connectivity and coverage in networks, it is inevitable to produce some edge nodes

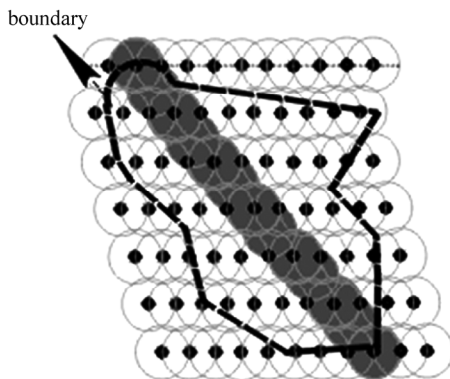


Figure 8. Deployment by EMST.

whose real coverage areas are smaller than other's like the R nodes as the Figure 9 shows. In our approach, we need to make full use of these edge nodes to compensate the holes of coverage and connectivity in WSN. Hence we divide those nodes which were deployed near to the boundary or obstacles into three categories on the basis of different relationship between R_s and R_c :

Case 1: When $R_c \leq \sqrt{3} R_s$:

- 1) The vertical distance between node and boundary or obstacles is $d < \sqrt{R_s^2 - \frac{R_c^2}{4}}$,
- 2) The vertical distance between node and boundary or obstacles is $\sqrt{R_s^2 - \frac{R_c^2}{4}} \leq d < R_s$,
- 3) The vertical distance between node and boundary or obstacles is $d \geq R_s$.

Also three types of nodes are set in different priority classes to move. The nodes ($d < \sqrt{R_s^2 - \frac{R_c^2}{4}}$) get the top priority, while the nodes ($\sqrt{R_s^2 - \frac{R_c^2}{4}} \leq d < R_s$) is mid.

Case 2: When $R_c > \sqrt{3} R_s$:

- 1) The vertical distance between node and boundary or obstacles is $d < \frac{R_s}{2}$,
- 2) The vertical distance between node and boundary or obstacles is $\frac{R_s}{2} \leq d < R_s$,
- 3) The vertical distance between node and boundary or obstacles is $d \geq R_s$.

Also three types of nodes are set in different priority classes to move. The nodes ($d < \frac{R_s}{2}$) get the top priority, while the nodes ($\frac{R_s}{2} \leq d < R_s$) is mid.

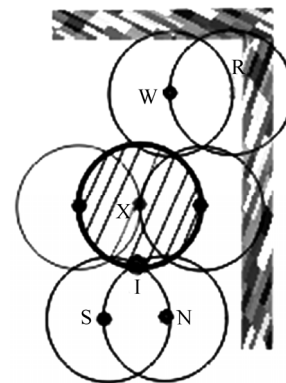


Figure 9. X is a dead node, R is the node with top priority.

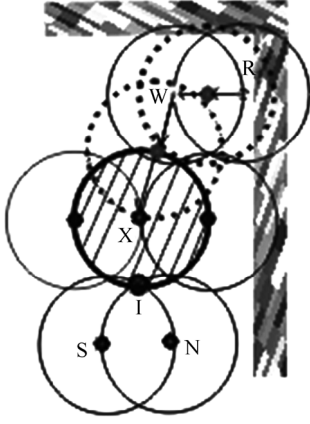


Figure 10. The process of movement.

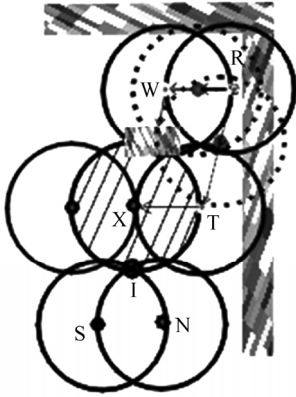


Figure 11. Changing another routing movement.

Though the following research and proof are based on $R_c \leq \sqrt{3} R_s$, we adopt the same principle methods to move the nodes on the condition of $R_c > \sqrt{3} R_s$ except only the constant of slope and location of nodes changed by above analyses. When the message is received by the nodes with top priority, they can calculate the distance from the dead node to its location $d_{dt} = \sqrt{(x_t - x_d)^2 + (y_t - y_d)^2}$, (i.e. (x_t, y_t) (x_d, y_d) are the location of top priority node and dead node in two-dimensional coordinates respectively) and broadcast to the other top priority nodes. The radio energy dissipation model as [16] referred:

$$E_T(l, d) = \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2 & \text{if } d \leq 0 \\ lE_{elec} + l\varepsilon_{mp}d^4 & \text{if } d > 0 \end{cases} \quad (5)$$

It presents that more and more energy would be wasted with the increasing of distance. So we choose the minimum distance d_{dt} from all the top movement priority nodes. In order to avoid energy depletion caused by excessive movement, we propose the new type move-

ment like routing to counterpoise the node's energy consumption in movement by calculation of routing movement distance as shown in Figure 10. When the available node R with top priority is chosen to replace the dead node X, R will broadcast available nodes W, S, N and then these nodes produce the virtual force to make W move to the location of X, and simultaneously R is forced to the W location. On the other hand, if there is an obstacle caused failure of movement on the path from W to X, W will broadcast back to R, and the routing of movement change immediately as the Figure 11 shows.

4.4. Calculation of Routing Movement Distance

Assuming the neighbor W of the dead node X. Set the co-ordinates of X be (x_0, y_0) , and those of W be (x_w, y_w) . Consider another two neighbors of node X, S and N which located at (x_s, y_s) and (x_n, y_n) respectively. The circle of coverage of nodes S and N intersect at the point I by the co-ordinates. Our algorithm makes node W move towards X such that the area that was earlier sensed by X can now be covered by node W.

Step 1: The co-ordinates of intersection node I and the distance d_{sn} between S and N can be derived as follows:

$$\begin{cases} (x - x_s)^2 + (y - y_s)^2 = R_s^2 \\ (x - x_n)^2 + (y - y_n)^2 = R_s^2 \end{cases} \quad (6)$$

$$d_{sn} = \sqrt{(x_s - x_n)^2 + (y_s - y_n)^2} \quad (7)$$

The solution of equation group:

$$\begin{cases} x = (x_s + x_n)/2 \pm (y_s - y_n)\sqrt{d^2(4r_c^2 - d^2)}/2d^2 \\ y = (y_s + y_n)/2 \mp (x_s - x_n)\sqrt{d^2(4r_c^2 - d^2)}/2d^2 \end{cases} \quad (8)$$

The solutions (x_i, y_i) which is closer to dead node X is the required answer.

Step 2: As above-mentioned optimal deployment algorithm, adjacent nodes have a fixed line slope in different relationship between R_s and R_c .

$$\text{When } R_c \leq \sqrt{3} R_s: \quad tg\alpha = (R_s + \sqrt{R_s^2 - \frac{R_c^2}{4}}) / \frac{R_c}{2}$$

$$\text{When } R_c > \sqrt{3} R_s: \quad tg\alpha = \sqrt{3}$$

Step 3: set $X' = (x', y')$ as the point node W move towards.

$$\begin{cases} (x' - x_t)^2 + (y' - y_t)^2 = R_s^2 \\ (y_w - y') / (x_w - x') = tg\alpha \end{cases} \quad (9)$$

So we can prove that the node W move towards (x', y') which was the location of dead node X.

4.5. Improved Virtual Forces Algorithm

In the process of movement, we also combined with the VFA model as [10] presents:

$$\bar{F}_{ij} = \begin{cases} (\omega_A(d_{ij} - d_{th}), \alpha_{ij}) & \text{if } d_{ij} > d_{th} \\ 0, & \text{if } d_{ij} = d_{th} \\ (\omega_R \frac{1}{d_{ij}}, \alpha_{ij} + \pi), & \text{if otherwise} \end{cases} \quad (10)$$

where d_{ij} is the Euclidean distance between sensor s_i and s_j , d_{th} is the threshold on the distance between s_i and s_j , α_{ij} is the orientation (angle) of a line segment from s_i to s_j , and $w_A(w_R)$ is a measure of the attractive (repulsive) force. The threshold distance d_{th} controls how close sensors get to each other. We assume that the neighbors of the dead nodes will produce the ‘‘attractive’’ force to the predetermined movement nodes. In order to reduce the total moving distance of the sensors, we determine whether s_i can move toward p_j at every period (namely round) as follows:

Step 1: The dead nodes p_j is detected by its neighbors, and its location is obtained by above geometry algorithm.

Step 2: Calculating $d_{p_j s_1}, d_{p_j s_2}, \dots, d_{p_j s_n}$. When the shortest $d_{p_j s_i}$ is found, the s_i decide to move toward p_j with a threshold λ , λ is the maximal distance a sensor can move forward at every round. Then the $s_i(x_{s_i}, y_{s_i})$ is updated with $s'_i(x'_{s_i}, y'_{s_i})$ which can be calculated by the Equations (1) and (2).

As Figure 12 shows, the linear equation of the line passes through the sensor s_i and the predetermined location p_j is $(y - y_{p_j})(y_{s_i} - y_{p_j}) = (x - x_{p_j})(x_{s_i} - x_{p_j})$. We can obtain

$$x'_{s_i} = \lambda(x_{s_i} + x_{p_j}) / d_{p_j s_i} + x_{p_j}$$

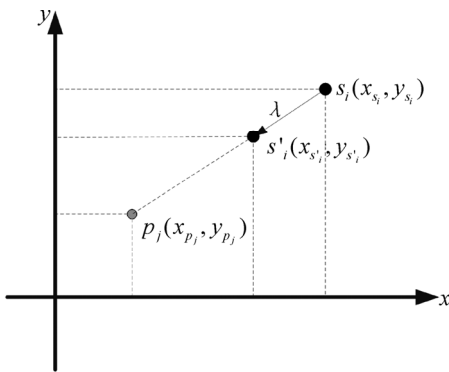


Figure 12. The coordinate of s_i is updated after moving a λ threshold.

and $y'_{s_i} = \lambda(y_{s_i} + y_{p_j}) / d_{p_j s_i} + y_{p_j}$.

So we can summarize an improved VFA with above analyses, if the final force of the dead node’s neighbors is calculated, the sensor with priority moves towards the dead node’s location according to the magnitude and direction. The updated move can be calculated:

$$\begin{cases} x(i)_{new} = x(i)_{old} + sign(\bar{F}_{ix}) \left| \frac{\bar{F}_{ix}}{\bar{F}_i} \right| \times \lambda \\ y(i)_{new} = y(i)_{old} + sign(\bar{F}_{iy}) \left| \frac{\bar{F}_{iy}}{\bar{F}_i} \right| \times \lambda \end{cases} \quad (11)$$

To sum up, with combination geometry and improved VFA, we prove the feasibility of our movement algorithm. If all these top priority nodes already compensate the loss in the network, the mid priority nodes will continue to move to the lower and dead ones. Note that the crucial Euclidean leaf nodes which ensure the whole connectivity of network need to be recovered first if any one doesn’t works.

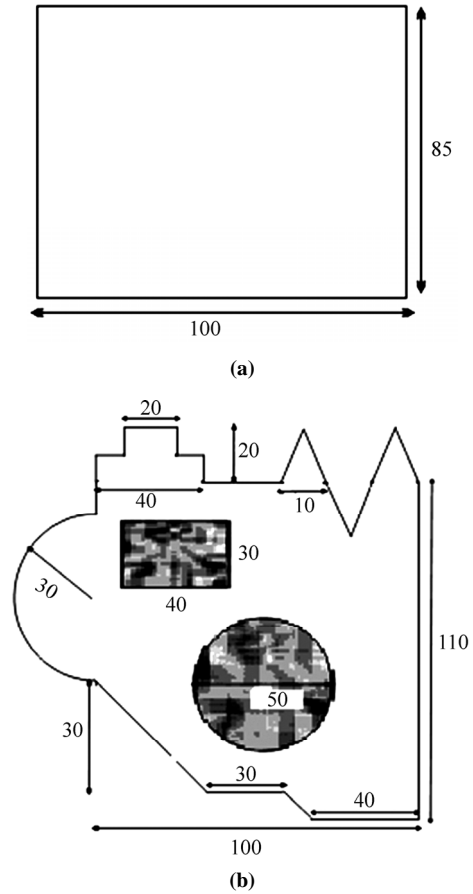


Figure 13. (a) 100*85 rectangle, (b) Particular areas: a complex area with kinds of boundaries and obstacles (shadow).

5. Simulation Results

In this section, we present two groups of experimental results to prove the effectiveness of our optimal deployment with self-healing movement algorithm in different fields. We choose one simple and representative sensing fields, and then design a more complex particular area as shown in Figures 13(a), (b). We consider four groups of cases $(R_c, R_s)=(4,6)$; $(5,5)$; $(6,4)$; $(8,4)$ to reflect the relationships as above-mentioned: $R_s > R_c$; $R_s = R_c$; $R_s < R_c < \sqrt{3} R_s$; $R_c > \sqrt{3} R_s$ respectively. All nodes are deployed above the ground about one meter to ensure the most optimal channel. We compare the number of sensor being deployed as comparison metric in four different methods including ours optimal algorithm, coverage-first algorithm, connect-first algorithm, grid algorithm discussed in Section 3. Then we make some nodes die randomly on purpose, and compare the coverage of this network with the other one which already healed it.

The number of sensors in four different relationships of R_s and R_c under particular area is shown in Figure 14. Most sensors are used in the grid algorithm, because all adjacent nodes are separated by the minimum of R_s and R_c . Sensors are placed horizontally by separation of R_c under the connect-first algorithm, thus it results in wasting many overlapping nodes in coverage when $R_s > R_c$. On the contrary, for it needs so many extra nodes to maintain the connectivity between adjacent sensors, the coverage-first method uses the most sensors except under grid algorithm when $R_s < R_c < \sqrt{3} R_s$. Also when $R_c > \sqrt{3} R_s$, the coverage-first method works the same as our optimal algorithm because of enough communication distance. To sum up, our optimal algorithm uses the least number of nodes to satisfy both coverage and connectivity in all four different situations.

The ratio of coverage in Region of Interest (ROI) is defined in [4] as shown in Equation (12).

$$C_r = \frac{\sum_{i=1..n} A_i}{A} \tag{12}$$

A_i is the area covered by the i^{th} node; N is the total number of nodes; A stands for the area of the ROI, which is simulated under our optimal deployment. In our simulation, we set $R_c=4$, $R_s=6$ and $R_c=8$, $R_s=4$ to reflect the different relationship of Case 1: $R_c \leq \sqrt{3} R_s$ and Case 2: $R_c > \sqrt{3} R_s$ respectively. We assume that the rest of nodes exception the ones with top and mid priority die firstly. As the above-mentioned analyses because the number of nodes with top and mid priority is less than the half of all nodes in ROI, and the rest of nodes is meaningless in moving to increase the coverage, we set the maximum number of dead nodes are 120 and 150 respectively. The comparison of the coverage ratio between the network with self-healing and the other one without movement is shown in Figure 15. The red lines are the simulation in Case 1 and the blue lines represent that in Case 2. We can find our algorithm with

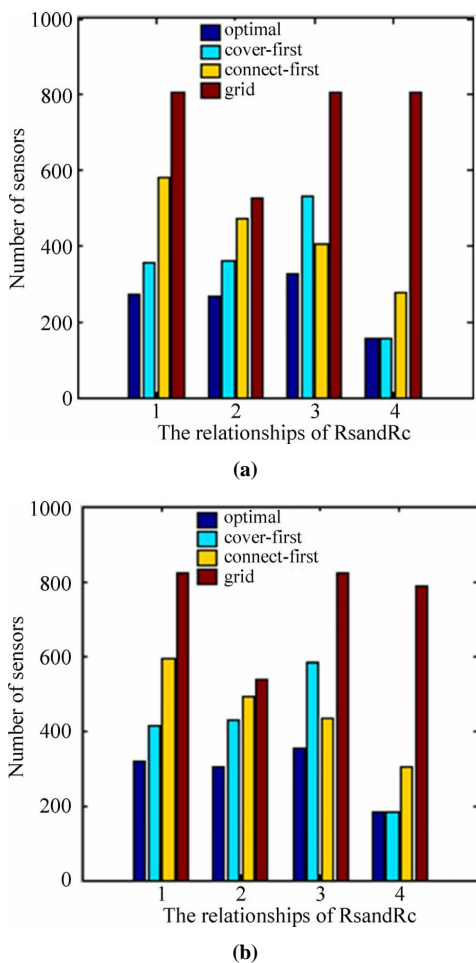
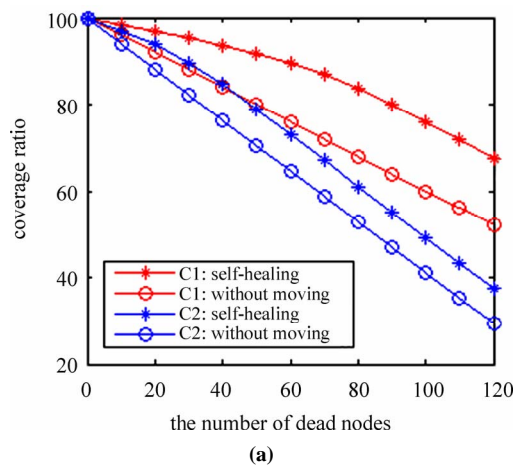


Figure 14. (a) The number of sensors used in 100*85 rectangle, (b) The number of sensors used in particular sensing area.



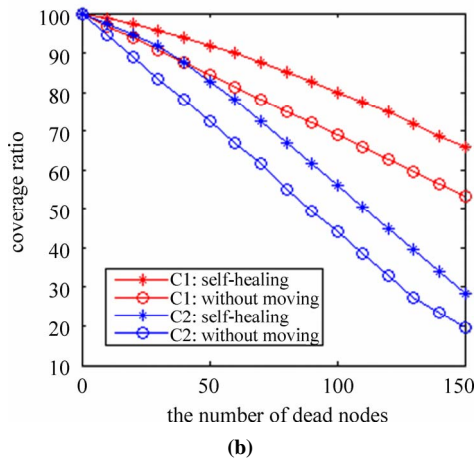


Figure 15. (a) The coverage ratio of 100*85 rectangle in two Cases, (b) The coverage ratio of particular region in two Cases.

self-healing maintains much higher coverage ratio by contrast with the original network without movement when the nodes die continuously. The lines with our algorithm decline slowly at first and then get faster as all top priority nodes used up while the mid priority nodes start moving. When all the nodes with top and mid priority have already relocated, the slope of the line is the same as another ratio line of original network without moving. However, when most nodes have already died, the slope of original network's coverage ratio line will rise slowly as a part of top and mid priority nodes that occupy smaller area can not be available. Because of less nodes in Case 2, the coverage ratio decline faster than the ratio in Case 1.

6. Conclusions

In this work, the optimal deployment with self-healing movement algorithm has been proposed to ensure the coverage and connectivity of particular area by fewer sensors as compared to other three methods. Besides, with the capacity of self-healing the coverage of the entire particular area is obviously enhanced by contrast with another network without movement when some nodes are already dead. Thus, this method can be applied in closed sensing area or semi-enclosed sensing area with boundaries or obstacles which are modeled by irregular polygons or arches. Furthermore, with a combination of programs and protocols, the Ad-hoc network can be built.

7. References

- [1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communica-*

tions Magazine, Vol. 40, No. 8, pp. 102–114, August 2002.

- [2] A. Sekhar, B. S. Manoj, and C. S. R. Murthy, "Dynamic coverage maintenance algorithms for sensor networks with limited mobility," *Pervasive Computing and Communications, PerCom 2005, Third IEEE International Conference*, pp. 51–60, March 8–12, 2005.
- [3] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," *Proceedings of the 24th International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05) Miami, FL, March 2005*.
- [4] N. Heo and P. K. Varshney, "Energy-efficient deployment of intelligent mobile sensor networks," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, Vol. 35, No. 1, pp. 78–92, 2005.
- [5] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," *Proceedings of IEEE Infocom, San Francisco, March 2003*.
- [6] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," *In Proceedings of the Fifth International Conference on Information Fusion*, pp. 1580–1588, 2002.
- [7] K. Kar and S. Banerjee, "Node placement for connected coverage in sensor networks," *Proceedings of the Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt' 03), Sophia Antipolis, France, 2003*.
- [8] S. Ganeriwal, A. Kansal, and M. B. Srivastava, "Self aware actuation for fault repair in sensor networks," *Robotics and Automation, Proceedings, ICRA'04, IEEE International Conference*, Vol. 5, pp. 5244–5249, April 26–May 1, 2004.
- [9] J. Chen, S. Li, and Y. Sun, "Novel deployment schemes for mobile sensor networks," *Sensors*, No. 7, pp. 2907–2919, 2007.
- [10] Y. Zou and C. Krishnendu, "Sensor deployment and target localization based on virtual forces," *INFOCOM 2003, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE*, Vol. 2, pp. 1293–1303, March 30–April 3, 2003.
- [11] H. N. Shu and Q. L. Liang, "Fuzzy optimization for distributed sensor deployment," *Wireless Communications and Networking Conference, IEEE*, Vol. 3, pp. 1903–1908, March 13–17, 2005.
- [12] X. L. Wu, J. S. Cho, B. J. d'Auriol, and S. Y. Lee, "Optimal deployment of mobile sensor networks and its maintenance strategy," *Computer Science, Advances in Grid and Pervasive Computing, Springer, Berlin*, pp. 112–123, June 21, 2007.
- [13] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks," *Proceedings of the ACM International Workshop on Under-Water Networks (WUWNet), Los Angeles, CA, September 2006*.
- [14] A. Savvides, C. C. Han, and M. B. Srivastava, "Dynamic

- fine-grained localization in ad-hoc networks of sensors,” ACM MobiCom, Rome, Italy, pp. 166–179, July 2001.
- [15] S. Borbash and M. McGlynn, “Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks,” ACM MobiHoc, Long Beach, USA, 2001.
- [16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application specific protocol architecture for wireless microsensor networks,” IEEE Transactions on Wireless Communications, Vol. 1, No. 4, pp. 660–670, 2002.
- [17] M. Younis and K. Akkaya, “Strategies and techniques for node placement in wireless sensor networks,” A Survey in ELSEVIER Ad Hoc Networks, No. 6, pp. 621–655, 2008.
- [18] Y. Zou, “Coverage-driven sensor deployment and energy-efficient information processing in wireless sensor network,” Duke University, 2004.