An Automated Sensor Deployment Algorithm Based on Swarm Intelligence for Ubiquitous Environment

KwangEui Lee

Department of Multimedia Engineering, Dongeui University, Busan, Korea

Summary
Sensor deployment is an important topic in wireless sensor network technology. In this paper, we propose a distributed automated sensor deployment algorithm based on swarm intelligence First, we present a basic deployment algorithm and then optimize various parameters including boundary width and number of overlaps.

Key words:
Context awareness, ubiquitous environment, swarm intelligence, sensor deployment.

1. Introduction

Wireless sensor network is one of the core technologies in the area of ubiquitous computing. A sensor network is composed by a set of wireless sensor nodes. Nodes are battery powered and movable smart sensors. Nodes communicate with each other via wireless network. Wireless sensor network can be applied in many application area including traffic control, industrial sensing, environmental monitoring, and intelligent buildings. Almost all application areas are closely related to the ubiquitous computing and context awareness.
In wireless sensor network, node positioning is an important topic and lots of deployment algorithms are proposed. P. Corke et al. proposed a method that repairs the gaps in connectivity of the deployed network [1]. Nojeong Heo et al. presented a distributed energy efficient deployment algorithm. The algorithm constructs an energy efficient node topology for a longer system life time [2]. R. C. Luo et al. proposed a grid deployment algorithm. Without changing previous configuration, the system coverage and uniformity rise rapidly by adding mobile sensors into the monitored environment [3]. Md. Obaidur Rahman et al. proposed an approach to find out least covered regions in a sensor network where further sensor deployment is desirable [4].
In this paper, we present distributed context aware sensor deployment algorithms for wireless sensor network and evaluate the performance of the algorithms. Presented algorithms are based on the swarm intelligence and we find the optimal value for various parameters. The rest of the paper is organized as follows. We introduce swarm intelligence in section 2. In section 3, we present the deployment algorithm. The analysis and simulation result is shown in section 4. Finally, section 5 draws the conclusion.

2. Swarm Intelligence

Swarm intelligence denotes the collective intelligence of groups of simple autonomous agents who behave as a swarm. There is no leader or some global plan in swarm intelligence. But, a complex intelligence emerges from the interaction of the collective simple intelligence. Birds flock is an example of the swarm intelligence. A bird in a flock only adjusts its movements to coordinate with the movement of its neighbors. There is no leader and no global plan. It simply tries to avoid collisions with its neighbors while trying to stay close to them. But, they take several advantages including protection from enemies and searching food.
In swarm intelligence, agents can only interact with their geographical neighbors and local environment. They can only obtain information from local environment and change the local environment [5]. For the sake of completeness, we insert a pseudo code of the swarm intelligence. The code is culled from [6] and slightly modified.

```
For each particle
   Initialize particle
End
Do
   For each particle
      Calculate fitness_value
      If (fitness_value > the best_fitness_value in history)
           set best_fitness_value as the new fitness_value
   End
   For each particle
      Calculate particle velocity according proper equation
      Update particle position according proper equation
   End
While max iterations or min error criteria is not attained
```

Manuscript received December 5, 2007
Manuscript revised December 20, 2007

Swarm intelligence has been applied to solve various optimization problems [7], [8].

3. Deployment Algorithm

Our deployment algorithm runs iteratively. In each round, each sensor counts the number of neighbor sensors that has overlapped sensing area. If any such neighbor sensor exist, sensors make a move to achieve better coverage by reduce the redundancies. We consider two algorithms. One is applied after random positioning of n sensor nodes in target field (initialization algorithm). The other is applied when a new sensor node introduced to stabilized target field (addition algorithm).
Before going further, we define some parameters needed in our model. We assume that there are n movable sensors in m×m target field

- sn[i] : ith sensor node.
- sn[i].x, sn[i].y, sn[i].r are the x and y position and sensing radius of sensor node sn[i]. We assume that the size of sensor node is zero.

The "initialization" algorithm is as follows:

Step 0: activate all sensor nodes.
/* Only active sensor nodes sn[i] performs the following algorithm separately. */
Step 1: counts the number of neighbor nodes sn[j] that satisfies

$$(sn[i].x-sn[j].x)^2 + (sn[i].y-sn[j].y)^2 < (sn[i].r+sn[j].r)^2$$

Let this number be cnt.
Step 2: if cnt is less than some threshold value then stop (deactivate itself)
Step 3: makes a movement according to the strategies.
Step 4: activates all neighbor nodes sn[j] that satisfies the expression of step 1.
Step 5: go to step 1

The addition algorithm does not quite differ from the initialization algorithm. We can use the initialization algorithm as an addition algorithm by activate the newly added sensor node.

4. Experimental Result

This section presents a set of experiments designed to study the characteristics and performance of the proposed algorithm under different input parameters. We analyze the performance of our algorithm from two aspects: coverage and deployment time. Deployment time is determined by the number of rounds We will try to find three optimal values: the first one is the optimal number of nodes for the given target field. The number of nodes means the maximum possible sensing area with given nodes. When a node covers p square meters, q nodes can covers p×q square meters. We call this as TC (theoretical coverage) of given nodes. The second one is BW (boundary width) and the last one is NO (number of overlaps). BW is the allowed minimum distance between the boundary of the field and the center of sensor node. NO is the maximum allowed number of nodes that has same sensing area.
In the following experiments, we assume 100×100 square meter target field. Figure 1 shows a deployment result of presented algorithm. The average round of the algorithm is 1749 and the average coverage is 83.39% of theoretical maximum possible coverage.

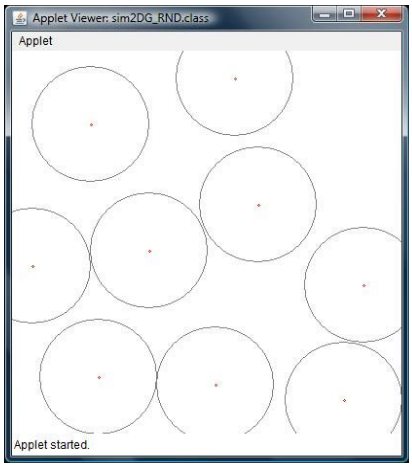

Fig. 1 A deployment result of the proposed algorithm.

Figure 2 shows the convergence speed when the number of nodes is 9 and sensing radius is 16 In this case TC become 71.36% and achieved coverage become 80.45% of TC after average 4462 rounds. As shown in the figure 2, a good deployment is achieved at the beginning of the running of the proposed deployment algorithm.

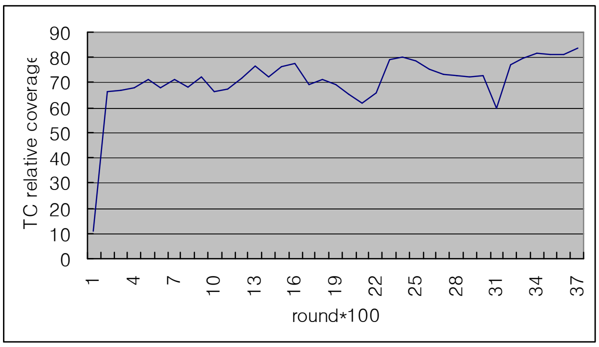

Fig. 2 Convergence speed

41 Experiments on Theoretical Coverage

The performance of the proposed algorithm is depends on the ratio of the area of target field and theoretical coverage of given sensor nodes. In this experiment, we fix the target field size to 100×100 square meter and number of sensors to 9. We investigate the variation of the sensing coverage achieved by the deployed sensors under the variation of sensing radius of sensors.

In figure 3 and 4, x coordinate is the sensing radius of the sensor nodes. We vary the sensing radius from 8 to 19 unit length. Figure 3 shows the DC sensing coverage of the sensors deployed by the algorithm We call this as DC (Deployment Coverage)

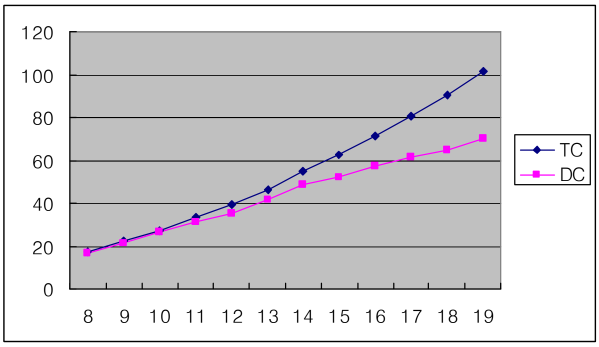

Fig. 3 Variation of DC for different values of TC.

Figure 4 shows the variation of time for the same experiments. As the figure shows, when the TC is over 80%, the running time is increased very fast.

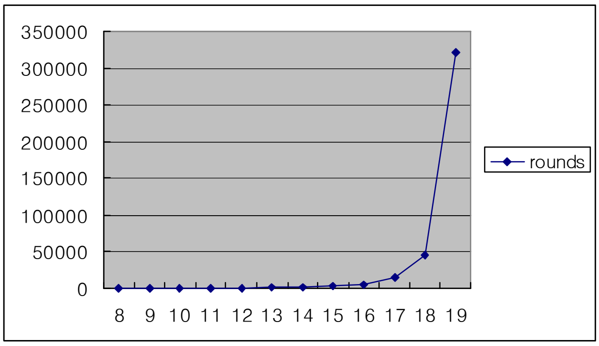

Fig. 4 Variation of the running time for different values of TC.

42 Experiments on Boundary Width

When a sensor is placed close to the border, we have loss at coverage. In the case of all sensors are on the boundary, coverage is become half of TC. Therefore, by managing sensors keep optimal distance from boundary, we can get more coverage. In this experiments, we set the target field size to 100×100 square meters, number of sensors to 9, and sensing radius of sensors to 16.

In figure 5, x coordinate is the minimum distance between sensor nodes and the boundary. Figure 5 shows the variation of the relative sensing coverage of deployed sensors to the theoretical coverage. As shown in this figure, the coverage is increased according to the minimum distance is increased.

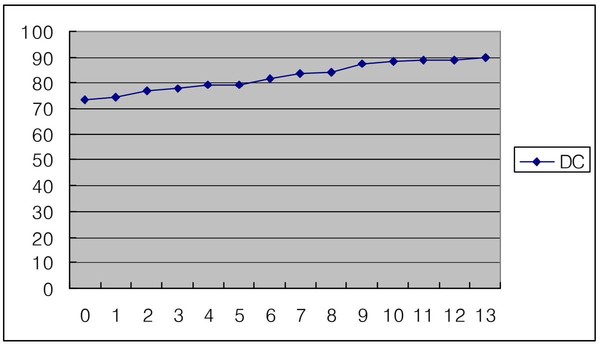

Fig. 5 Variation of sensing coverage for different values of BW

Figure 6 shows the variation of time for the same experiment. The number of round after 8 is too big so we didn't draw it (display as same to the case of 8).

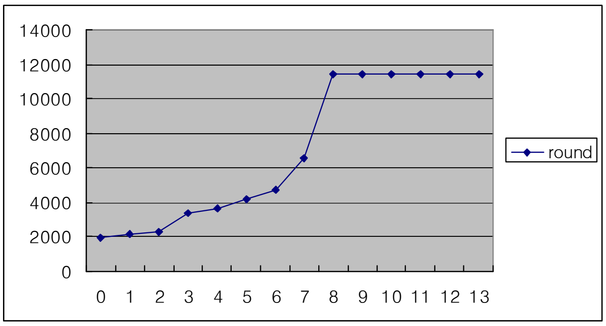

Fig. 6 Variation of running time for different values of BW.

43 Experiments on Number of Overlaps

To increase the sensing coverage we consider the number of overlaps of sensing area. By minimizing the number of overlaps, we can expect the coverage will be maximized. Here we investigate the relation of running time and sensing coverage according to the number of overlaps. In this experiments, we set the target field size to 100×100

square meters, number of sensors to 9, and sensing radius of sensor to 16 meters.
Figure 7 shows the variation of running time of the proposed algorithm according to the number of overlaps. We can see that the running time is dramatically reduced even when we allow only one overlap.

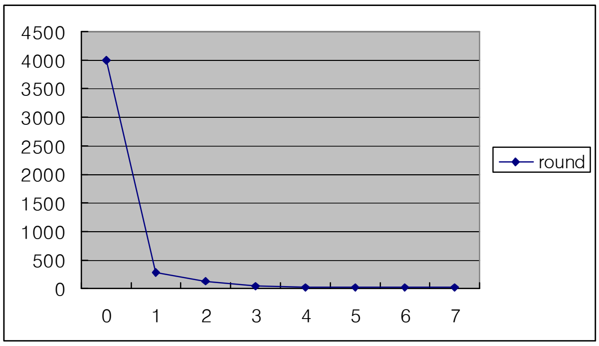

Fig. 7 Variation of the running time for different values of NO

Figure 8 shows the variation of the coverage for the same experiments. The coverage is changed linearly to the number of overlaps.

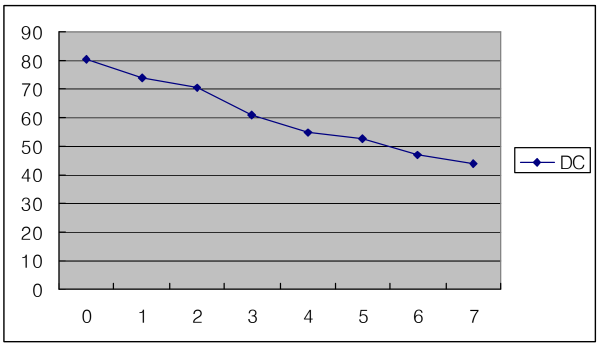

Fig. 8 Variation of DC for different values of NO

By setting the number of overlap to 1, we can reduce the running time dramatically without loss of sensing coverage of resulting deployment.

5. Conclusion

In this paper, we propose distributed automated sensor deployment algorithms and find optimal values for the various parameters of deployment. This deployment algorithm can be used in the case of non rectangular shape of target field and easily adapted to the 3D case. As a result, we show that the swarm intelligence can be perfectly applied to the sensor deployment problem of sensor network.

Acknowledgments

This work was supported by Dong-eui University Grant (2005AA186)

References

[1] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme. “Deployment and Connectivity Repair of a Sensor Net with a Flying Robot,” ISER 2004, 2004.
[2] Nojeong Heo, Varshney, P. K., “Energy-efficient deployment of Intelligent Mobile sensor networks,” IEEE Transctions on Systems, Man and Cybernetics, vol. 35, pp. 78-92, Jan. 2005.
[3] R. C. Luo, L. C. Tu, O. Chen, “Auto-deployment of Mobile Nodes in Wireless Sensor Networks Using Grid Method.” IEEE International Conference on Industrial Technology, pp. 359–364, 2005
[4] Md. Obaidur Rahman, Md. Abdur Razzaque and Choong Seon Hong, “Probabilistic Sensor Deployment in Wireless Sensor Network: A New Approach,” Proceedings of the 9th IEEE ICACT 2007, Volume II, pp. 1419-1422, Feb. 2007
[5] J. Kennedy, R. C. Eberhart, Y. Shi. Swarm intelligence. Morgan Kaufmann Publishers, 2001.
[6] www.swarmintelligence.org
[7] Secrest, B. R. and Lamont, G. B., “Visualizing particle swarm optimization – gaussian particle swarm optimization,” Proceedings of the IEEE Swarm Intelligence Symposium, pp. 198-204, 2003
[8] Clerc, M., "Discrete Particle Swarm Optimization," New Optimization Techniques in Engineering Springer-Verlag, 2004.

KwangEui Lee received his B.S., M.S. and the Ph.D. degrees from Sogang University, Seoul, Korea in 1990, 1992, and 1997, respectively. From 1997 to 2001, he joined ETRI as a senior research member. Since 2001, He has been an associate professor of Dongeui University. His research interests include computation theory, artificial life, context awareness and their applications