Proceedings of the 2009 IEEE International Conference on
Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009

# A Wireless Sensor Network Coverage Optimization Algorithm Based on Particle Swarm Optimization and Voronoi Diagram

Nor Azlina Bt. Ab Aziz
azlina.aziz@mmu.edu.my
Faculty of Engineering and Technology,
Multimedia University,
Jalan Ayer Keroh Lama,
75450 Melaka, Malaysia

Ammar W. Mohemmed
ammar.wmohemmed@mmu.edu.my
Faculty of Engineering and Technology,
Multimedia University,
Jalan Ayer Keroh Lama,
75450 Melaka, Malaysia

Mohammad Yusoff Alias
yusoff@mmu.edu.my
Faculty of Engineering,
Multimedia University,
Jalan Multimedia,
63100 Cyberjaya, Selangor,
Malaysia

*Abstract:* **The coverage problem is a crucial issue in wireless sensor networks (WSN), where a high coverage rate ensures a high quality of service of the WSN. This paper proposes a new algorithm to optimize sensor coverage using particle swarm optimization (PSO) and Voronoi diagram. PSO is used to find the optimal deployment of the sensors that gives the best coverage while Voronoi diagram is used to evaluate the fitness of the solution. The algorithm is evaluated through simulation in different WSN. The simulation results show that the proposed algorithm achieves a good coverage with a better time efficiency.**

## 1. INTRODUCTION

WSN is a group of low-cost, low-power, multifunctional and small size wireless sensor nodes that work together to sense the environment, perform simple data processing and communicate wirelessly over a short distance [1]. Some of these sensor nodes are able to move on their own. The mobility is achieved by mounting the sensors on mobile platforms such as Robomote [2]. With the ability to move independently, these mobile sensors are able to self deploy and self repair, thus adding more to their value [3]. In general the applications of WSN can be divided into military, civilian and health care [4]. Among the military use of WSN are targeting system and battlefield surveillance. For the civilian application, WSN can be applied in traffic monitoring, environmental monitoring, building monitoring and control, wildlife monitoring, security, smart agriculture system and many other applications. In health care, WSN can be used to monitor the vital signs of critical illness patients and also it is very useful in elderly care where in addition to monitoring their vital signs the sensors can be used to track down their locations.

These wireless sensors, however, have several constraints such as restricted sensing and communication range as well as limited battery capacity [5]. The limitations raise several issues including coverage, connectivity, network lifetime, scheduling and data aggregation. In order to prolong the WSN lifetime, energy conservation measures such as scheduling and data aggregation must be taken. Scheduling conserves energy by turning off the sensors whenever possible while data aggregation tries to conserve the energy by reducing the energy used in transmitting the data. Connectivity and coverage problems stem from limited communication and sensing range of the involved sensors. To ensure connectivity, the sensors need to be placed close enough to each other so that they are within the communication range. In the other hand, the coverage problem concerns on how to guarantee that each of the points in the region of interest (ROI) is covered by the sensors. In order to maximize the coverage, the sensors need to be placed not too close to each other so that the sensing capability of the network is fully utilized and at the same time not too far from each other to prevent forming coverage holes (area outside sensing range of sensors).

A sensor's prime function is to sense the environment for any occurrence of the event of interest. Therefore the coverage is one of the major concerns in WSN. In fact, it is used as a key for quality of service (QoS) evaluation in WSN [6].

In this paper, the coverage problem is formulated as an optimization problem and PSO is used to find an optimum or near optimum solution to it. PSO is a metaheuristic optimization algorithm imitating the swarm intelligence of some organisms like birds or fish. It has been successfully used in many applications [7]. In this paper, PSO is used to find the optimal placement of the sensors according to a fitness function that is based on Voronoi diagram. The advantage of using Voronoi diagram over other geometrical structures, for instance the grid, is that its computational complexity is controlled only by one parameter which is the number of sensors in the network.

The rest of this paper is organized as follows. Section 2 reviews related works to the coverage problem. Section 3 is a technical preliminary where PSO, Voronoi diagram and coverage problem are introduced. The proposed algorithm is presented in section 4 and followed by the simulation result and discussion in the section 5. Finally, in section 6 the paper is concluded and future path of our work is discussed.

## 2. RELATED WORKS

Different strategies have been proposed in the literature for the WSN coverage optimization. Chakrabarty et al. [8] proposed grid coverage strategy for effective surveillance and target positioning using integer linear programming (ILP). The sensor field is represented as grid. With the sensors placed at the grid points, a target can be located easily at any time. Biagioni and Sasaki compared three types of grids – square, triangle and hexagon [9]. Among these grids, triangle grid is the best due to higher survival rate and robustness to single node failure. However, grid based deployment strategies require the sensors to be placed exactly and accurately at the grid points. Therefore, this method is subject to errors such as misalignment and random misplacement [10].

Besides being used as an arrangement for wireless deployment, the grid structure is also used in WSN to measure the coverage percentage which is the ratio of the actual area covered to the area of ROI. Calculating the covered area is not an easy task due to overlapping sensor coverage. Therefore, researchers resort to *sampling* methods where only a set of points inside the ROI is used to evaluate the coverage. As shown by Zou and Chakrabarty, and Shen et al. [11,12] grid is among the commonly used methods for sampling. In this case, the coverage is estimated as the ratio of the number of grid points covered to the total number of the grid points in the ROI. The computation cost of this method is $O(nmk)$ where $n \times m$ is the number of grid points and $k$ is number of sensors deployed [12]. The accuracy of the estimation depends on the size of each grid, the smaller the grid is, the closer the estimation to the actual coverage.

Other approaches for optimizing the coverage in WSN are virtual field and Voronoi diagram. Howard and Poduri [3] proposed the virtual field concept to WSN. It assumed that the sensor nodes and obstacles have potential fields which exert virtual forces. The nodes repel each other until either their sensing fields no longer overlap or they cannot detect each other. Although this method ensures full coverage and full connectivity, it relies highly on the sensor mobility which is a high energy consumption task [2].

In [13], the sensors are initially deployed randomly. Based on the initial placement, they broadcast their locations and construct their own Voronoi diagram. Using the formed diagram, the sensors decide whether to reposition, to eliminate (reduce) the coverage holes, or to stay. Three protocols are suggested: VEC (VECtor based algorithm) – to push sensors out from densely covered area, VOR (VORonoi based algorithm) – a sensor moves to its farthest Voronoi vertex when it detects a coverage hole and Minimax – to cover holes by moving closer to the farthest Voronoi vertex but not as far as VOR. Another work that uses Voronoi diagram is proposed by Wang et al. [14] where a combination of static and mobile sensors is used. The static sensors construct Voronoi diagram which is used to detect coverage holes, while the mobile sensors are used to close these holes.

## 3. BACKGROUND

### 3.1 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based optimization tool inspired by the natural social behavior of certain organisms like bird flocking and fish schooling as developed by Kennedy and Eberhart [15]. This behavior is imitated in PSO where particles (agents) fly over the search domain influenced by their experience and the experience of the surrounding neighbors. The algorithmic flow in PSO starts with a population of particles whose positions, that represent the potential solutions for the studied problem, and velocities, that determine the next move, are randomly initialized in the search space. The search for optimal position (solution) is performed by updating particle velocities ($v_{id}$) and positions ($x_{id}$) by:

$$v_{id} = w*v_{id} + c1*rand()*(p_{id} - x_{id}) + c2*Rand()*(p_{gd} - x_{id}) \quad (1)$$
$$x_{id} = x_{id} + v_{id} \quad (2)$$

where $w$ is inertia weight used to control the effect of the previous velocity in the current velocity. A time decreasing inertia weight encourages high exploration at the beginning and fine tuning at the end of the search [16]. $c_1$ and $c_2$ are the learning factors to control the effect of the "best" factors of particles; $p_{id}$ and $p_{gd}$. *rand()* and *Rand()* are two independent random numbers in the range of [0.0,1.0]. The velocity of the particle is influenced directly by two factors; the best position found so far by the particle ($p_{id}$) and the best position found by the neighboring particles ($p_{gd}$). The quality of the solution is evaluated by a fitness function, which is a problem-dependent function. If the current solution is better than the fitness of $p_{id}$ or $p_{gd}$ , the best value will be replaced by current solution accordingly. This update process will continue until stopping criterion is met, usually when either maximum iteration is achieved or target solution is attained. The PSO algorithm is shown in Fig.1.

PSO based algorithms are proposed in [17] and [18] for the WSN coverage optimization, where the square grid structure is used in the fitness functions to evaluate the particles. Besides maximizing the coverage, minimization of energy consumption in cluster based network is also considered in [17]. The energy consumption minimization phase follows a coverage maximization phase. In [18] virtual force is combined with co-evolutionary PSO (CPSO), so that the best location of the sensors is achieved. CPSO is used here to increase the search quality and speed up the convergence rate, while virtual force is introduced into the velocity equation of PSO to direct the particle search.

```
Initialize particles population;

Do{
    Calculate fitness values of each particles using fitness function;
    Update p_id if the current fitness value is better than p_id;
    Determine p_gd : choose the particle position with the best fitness value of
    all the neighbors as the p_gd;
    For each particle {
        Calculate particle velocity according to (1);
        Update particle position according to (2);
    }
} While maximum iteration or ideal fitness is not attained;
```

Fig. 1: PSO Algorithm

### 3.2 Voronoi Diagram

Voronoi diagram is a partition of sites (shown as ◊'s in Fig. 2) in such a way that points inside a polygon are closer to the site inside the polygon than any other sites, thus one of the vertices of the polygon is the farthest point of the polygon to the site inside it. Voronoi diagram can be used as a sampling method in determining WSN coverage; with the sensors act as the sites. If all Voronoi polygons vertices are covered, then the ROI is fully covered otherwise coverage holes exist [10]. The lower bound for the computational complexity of constructing

Voronoi diagram is $\Omega(N \log N)$ [19], where $N$ is the number of sites.

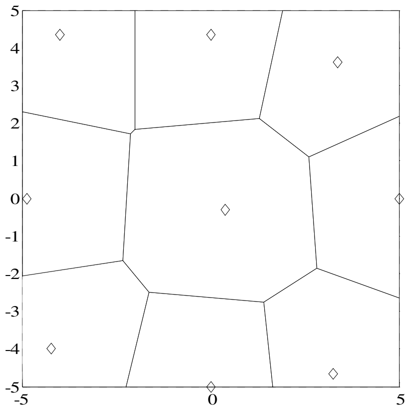



Fig. 2: Voronoi Diagram (9 sites)

Voronoi diagram has been used in a variety of applications such as word extraction from image document [20], robot navigation path planning [21], source to destination optimal path planning [22], study of protein structure [23] and also WSN [13, 14].

### *3.3 The Coverage Problem*

Given a set of $N$ number of sensors, $S = \{s_1, s_2, \ldots, s_N\}$ and a ROI, coverage problem is how to position the sensors in the ROI so that the coverage percentage is maximized and coverage holes is minimized. It can be classified into three classes [6]; area coverage, point coverage and barrier coverage. The classification is based on what is the main concern of the problem, whether it is to cover an area, boundary surveillance or monitoring a set of points of interest. Area coverage, as the name suggests, is on how to cover an area with the sensors, while point coverage deals with the coverage of a set of points of interest. Decreasing the probability of undetected penetration is the main issue in barrier coverage.

This work deals with area coverage, where the objective is to maximize the coverage percentage. This problem can also be seen as a minimization problem [12]. From the minimization point of view, the objective is now how to make sure the total area of the coverage holes in the network is as small as possible.

Basically, the coverage problem in WSN stem from three main factors: not enough sensors to cover the whole ROI, limited sensing range and random deployment. Since the sensors operate under a limited power supply, some of them might die out, resulting in inadequate number of sensors to fully cover the whole ROI thus causing holes to exist. Another reason is the sensor's restricted sensing range, of course this problem can be solved by using sensors with larger sensing range, but this type of sensors are more expensive [8].

One of the appealing aspects of WSN is the ability to be randomly deployed without the need to manual interference. For example, in hostile and unreachable environment such as battlefield and a steep terrain, sensors can be dropped from air. However, random deployment could cause some of the sensors to fall too close to each other while others are too far apart. In both situations coverage problem arises; in the first case, the sensing capabilities of the sensors are wasted and the coverage is not maximized, while in the later case, blind spots are formed.

As stated above, the coverage can be enhanced by using sensors with larger sensing range but this is costly. Thus among the commonly used solutions is to address the problem during deployment phase. Rather than random deployment, the deployment of WSN can be done using a predetermined plan [5]. In predetermined deployment, the WSN coverage is improved by carefully planning the positions of the sensors in the ROI prior to their deployment. Then, the sensors are placed according to the plan either manually or with the help of a mobile robot. However, this method is costly and suitable only for small WSN. As for random deployment, the initial coverage can be enhanced by manipulating the locomotion capability of the sensors or by using incremental deployment after the initial one. In the mobility approach, the mobile sensors are self repositioned after early deployment, to achieve a better arrangement and the coverage is maximized. On the other hand, the incremental deployment method involves analyzing the initial coverage and adding sensors at locations with coverage hole. Overall, for both deployment methods the aim is to solve the coverage problem using sensors' placement.

## 4. Proposed Algorithm

In this paper, a PSO algorithm is proposed to find an optimum (or near optimum) deployment of the WSN to cover the area of ROI. The ROI is assumed to be a two dimensional square area and the WSN is homogeneous; all the sensors have similar sensing radius. In addition, it is assumed that the sensors know their positions and possess locomotion capability; that is they are able to move and change position. The algorithm is to be executed at a base station after an initial random placement. The sensors final optimal positions will be transmitted by the base station to the sensors, based on this information the sensors will move to their optimal positions. Two main issues need to be considered when implementing PSO for a problem: the particle encoding and the fitness function.

### *4.1 Particle Encoding*

A particle encodes a solution for the coverage problem. The final solution represents the optimum positions of the sensors. The position of a sensor $j$ is described by a two coordinates $(x_j, y_j)$. Considering $N$ number of sensor nodes, the particle encoding can be depicted as in Fig. 3. Thus, the dimension of the particle is two times the number of sensors.

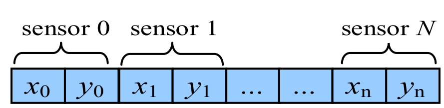



Fig. 3: Particle Encoding

### *4.2 The Fitness Function*

The fitness function evaluates the solution encoded in a particle. Here coverage problem is considered as a minimization problem where the objective is to minimize the total area of coverage holes. The coverage is measured based on Voronoi diagram as follows. Based on the sensors positions encoded in a particle, Voronoi diagram is computed. To measure the coverage holes, a set of points, called interest points, are required to be selected. The interest points set consists of the vertices of the Voronoi polygons – obtained from the computed Voronoi diagram – and a number of points distributed evenly on the boundary of the polygons. These points on the boundary act as pulling forces that prevent the sensors from congregating around a particular point in the ROI. The number of points at the boundary has to be carefully chosen because too many points will pull the sensors too much to the boundary and reduce the coverage and too few points will not prevent the sensors from congregating.

The total area of coverage holes is determined as follows: The distance of the interest points to their nearest sensors is calculated. If the distance ($d$) is greater than the sensing radius ($r_s$), then a coverage hole exists around the interest point. If the interest point is a Voronoi vertex, the hole area is approximated as the circular area around the vertex not covered by the nearest sensor, while if it is a corner point on the boundary, then it is a quarter of the circle. However, if it is on the boundary then it is a half of the circle. The hole area estimation is shown in Fig. 4. The radius of the hole circle is the difference between $d$ and $r_s$ ($d$-$r_s$). Thus the fitness is the summation of the area of the coverage holes in the ROI. Ideally the fitness value should equal zero, indicating that there is no coverage holes exist. Assuming there are $K$ interest points – boundary points and Voronoi vertices, the fitness function can be computed as in Fig.5.

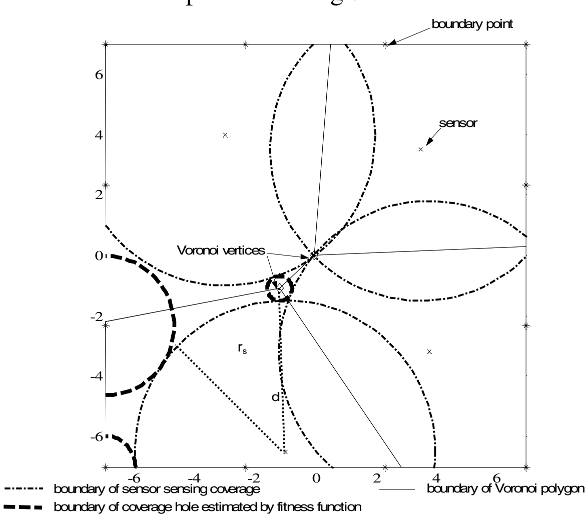



Fig. 4: Coverage Holes Estimation in Fitness Function

The computational complexity of this fitness function is based on the number of Voronoi vertices which is, in turn, proportional to the number of the sensors. This is an advantage over grid structure where the computation complexity and accuracy depends not only on the number of sensors but the size of the grid. Also, there is a need to determine the size of the grid to balance between the accuracy and computation time.

```
Fitness = 0;
Compute Voronoi diagram based on sensors' position;
Interest points = {polygon vertices within the ROI, n evenly distributed points along the boundary};
For each interest point
    Find the distance of current interest point to its nearest sensor;
    If distance > sensing radius
       Hole =  × (distance – sensing radius)^2;
       If current interest point is on ROI's boundary
          If current interest point on ROI's corner
             Fitness += Hole/4;
          Else
             Fitness += Hole/2;
       Else
          Fitness += Hole;
    End
```

Fig. 5: Fitness Function

## 5. RESULTS & DISCUSSION

The performance of the PSO proposed algorithm based in Voronoi diagram (PSO_Voronoi) is investigated through simulation experiments and compared with another PSO algorithm based on grid structure (PSO_Grid) [17]. The objective of the experiments is; to investigate the performance of the proposed algorithm in term of the coverage achieved and its execution time with respect to the WSN parameters including the number of sensors and size of ROI. The algorithm is implemented using MATLAB. The PSO's number of particles is set to 20, and a linearly decreasing inertia weight is adopted in the range [0.5, 2.0]. The learning factors for *c1* and *c2* are both set to 2.0. The particles positions are initialized in the range [0, 50] for test I&II and [0, 30] for test III while the maximum velocity is set to 4. The algorithm runs to a maximum number of iterations of 600. The sensing range of the sensors ($r_s$) is set to 5. The number of points along each of the boundaries is 9 (this number is chosen based on preliminary simulations tests). The average and standard deviation of 30 runs are recorded.

Table 1 presents the results of three tests with different WSN configurations. Test I involves a WSN of 40 sensors and a ROI of dimension 50×50. Tests II and III, use a network consists of 20 sensors with ROI of 50×50, 30×30 respectively. In each test, PSO_Voronoi is compared with two cases of PSO_Grid; grid size of 1 × 1 and 2.5×2.5.

From Table 1, it could be seen that the average coverage of the proposed algorithm is very close to PSO_Grid (1×1) and always better than PSO_Grid (2.5×2.5). At the same time and for the three tests, the execution time of the proposed algorithm is much more superior to PSO_Grid. The only case when PSO_Grid is faster is when the size of the grid is (2.5×2.5) in test III, this is due to less number of grid points. However, this comes with a compromise in the coverage.

Because the computation complexity of the fitness function of PSO_Voronoi is based only on the number of the sensors hence despite the ROIs are of different sizes in test II and tests III, the execution time of PSO_Voronoi remains constant at •27sec. This is in contrast to PSO_Grid, where the execution time increases with the size of ROI and grid points.

The standard deviation of coverage for the proposed algorithm is fairly small compared with PSO_Grid; this indicates that the proposed algorithm has consistent performance with different network parameters.

Fig. 6 compares the performance of PSO_Voronoi with PSO_Grid in terms of coverage and execution time for different runs in test III. This figure shows that PSO_Voronoi gives a much more stable performance in both coverage and time than PSO_Grid.

TABLE 1: Results

| | | | |
|---|---|---|---|
| **Test I: 40 sensors in 50x50 ROI** | | | |
| Ideal coverage* | 100% | | |
| | PSO_Grid (1×1) | PSO_Grid (2.5×2.5) | PSO_Voronoi |
| Average of coverage | 91.74% | 86.51% | 89.83% |
| Standard Deviation of coverage | 0.0121 | 0.0163 | 0.0112 |
| Average execution time (seconds) | 645.3572 | 100.4937 | 61.2844 |
| **Test II: 20 sensors over 50x50 ROI** | | | |
| Ideal coverage* | 62.83% | | |
| | PSO_Grid (1×1) | PSO_Grid (2.5×2.5) | PSO_Voronoi |
| Average of coverage | 61.07% | 57.77% | 59.24% |
| Standard Deviation of coverage | 0.0036 | 0.0074 | 0.0058 |
| Average execution time (seconds) | 362.4401 | 53.6447 | 27.6683 |
| **Test III: 20 sensors in 30x30 ROI** | | | |
| Ideal coverage* | 100% | | |
| | PSO_Grid (1×1) | PSO_Grid (2.5×2.5) | PSO_Voronoi |
| Average of coverage | 97.89% | 92.15% | 97.96% |
| Standard Deviation of coverage | 0.0203 | 0.0399 | 0.0054 |
| Average execution time (seconds) | 125.1811 | 22.1166 | 27.1712 |

## 6. CONCLUSION & FUTURE WORK

This paper presents a PSO/Voronoi Diagram algorithm (PSO_Voronoi) that can be used for optimizing the coverage problem in WSN. Simulation results show that the algorithm proposed here provides good coverage within a reasonable computational time. The execution time is not influence by other factors rather than the number of sensors in the network.

From the test results it is suggested that the proposed algorithm to be used when there is a need for a large network in a large ROI, while the grid method is used only either when the network is small or when the execution time is not important. If grid structure is chosen the network designer has to ensure that the grid size is small enough so that the quality of the solution is not compromise.

As this is an on going research project, in the future we will focus more in doing more tests for different conditions and optimizing other problems in WSN.

* Ideally coverage is 100% as long as number of sensors is bigger than the minimum number of circles (*N*) – with radius; *r* – required to cover an area (*A*): $N = A /( 3*\bullet 3*r^2/2)$

## REFERENCES


[1] Zhao, J., Wen, Y., Shang, R. and Wang, G. Optimizing Sensor Node Distribution with Genetic Algorithm in Wireless Sensor Network, Advance in Neural Network, Springer Berlin / Heidelberg, 2004.

[2] Dantu, K., Rahimi, M., Shah, H., Babel, S., Dhariwal, A., and Sukhatme, G. Robomote: Enabling Mobility In Sensor Networks IEEE/ACM International Conference Information Processing in Sensor Networks, Apr. 2005.

[3] Howard, A. and Poduri, S. Potential Field Methods for Mobile-Sensor-Network Deployment in Bulusu, N. Jha, S. Wireless Sensor Networks A System Perspective Artech House, London, 2005.

[4] Wu, J. Sensor Networks in Wu, J. Handbook on Theoretical & Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, Auerbach Publication, USA, 2006.

[5] Ghosh, A. and Das, S.K, Coverage and Connectivity Issues in Wireless Sensor Networks in Shorey, R., Ananda, A.L, Chan, M.C. and Ooi, W.T, Mobile, Wireless, and Sensor Networks: Technology, Applications and Future Directions, John Wiley & Sons, Inc., 2006.

[6] Cardei, M. and Wu, J. Coverage in Wireless Sensor Networks in Ilyas, M. and Mahgoub, I. Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems , CRC Press, USA, 2005.

[7] Xiaohui Hu; Yuhui Shi; Eberhart, R., Recent advances in particle swarm. Evolutionary Computation Congress on Volume 1, 19-23 June 2004.

[8] Chakrabarty, K., Iyengar, S.S., Qi, H. and Cho, E. Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks, IEEE Transactions on Computers Vol 51, No. 12, 2002.

[9] Biagioni, E.S. and Sasaki, G., Wireless Sensor Placement for Reliable and Efficient Data Collection Proceeding of the 36th Hawaii International Conference on System Sciences, 2003.

[10] Xu, K., Takahara, G. and Hassanein, H., On the Robustness of Grid-Based Deployment in Wireless Sensor Networks IWCMC, 2006.

[11] Zou, Y. and Krishnendu Chakrabarty, Sensor deployment and target localization based on virtual forces, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Volume 2, 30 March-3 April 2003.

[12] Shen, X., Chen, J., Wang, Zhi. And Sun, Y. Grid Scan: A Simple and Effective Approach for Coverage Issue in Wireless Sensor Networks. IEEE International Conference Communications, June 2006.

[13] Wang, G., Cao, G. and Porta, T.L Movement-Assisted Sensor Deployment, IEEE INFOCOM 2004.

[14] Wang, G., Cao, G. and Porta, T.L A Bidding Protocol for Deploying Mobile Sensors Network Protocols, Proceedings. 11th IEEE International Conference on Network Protocols. 2003.

[15] Kennedy, J., and Eberhart, R. C. Particle swarm optimization. Proc. IEEE International Conference on Neural Networks,1995.

[16] Shi, Y. H., Eberhart, R. C., A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation, Anchorage, Alaska, May 4-9, 1998.

[17] Wu Xiaoling, Shu Lei, Yang Jie, Xu Hui, Jinsung Cho, and Sungyoung Lee, Swarm Based Sensor Deployment Optimization in Ad hoc Sensor Networks, ICESS (LNCS), Xi'an, China, Dec 16-18, 2005.

[18] Wang, X., Wang, S. and Ma, J.J. An Improved Co-evolutionary Particle Swarm Optimization for Wireless Sensor Networks with Dynamic Deployment, Sensors, Issue 3 Vol 7, March 2007.

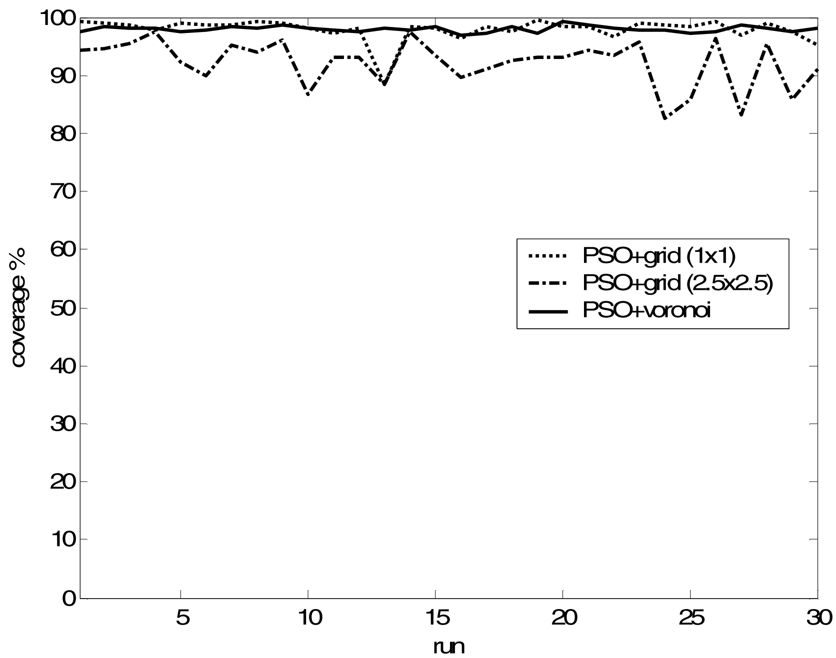
[19] Aurenhammer, F. and Klein, R Voronoi diagrams. In J. Sack and G. Urrutia, editors, Handbook of Computational Geometry,. Elsevier Science Publishing, 2000.

[20] Wang, Z., Lu, Y., and Tan, C.L. Word Extraction Using Area Voronoi Diagram, Conference on Computer Vision and Pattern Recognition Workshop - Volume 3, 2003.

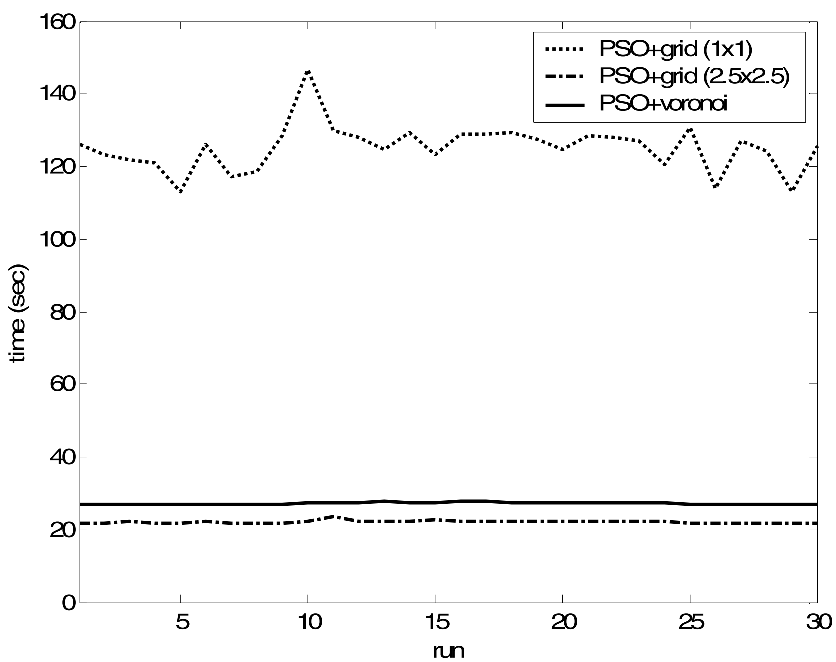
[21] Nagatani, K.; Choset, H.; Thrun, S.; Towards exact localization without explicit localization with the generalized Voronoi graph, IEEE International Conference on Robotics and Automation Volume 1, 16-20 May 1998.

[22] Bhattacharya, P.; Gavrilova, M.L.; Voronoi diagram in optimal path planning, 4th International Symposium on Voronoi Diagrams in Science and Engineering, 9-11 July 2007.

[23] Poupon, A. Voronoi and Vorornoi Related Tessellations in Studies of




(a)




(b)

Fig. 6: Test III: (a) Percentage of Coverage (b) Execution Time (s)