PORT SAID ENGINEERING RESEARCH JOURNAL
Faculty of Engineering - Port Said University
Volume. (20) No. 1 March 2016 pp. 109:117

Coverage Improvement for Wireless Mobile Sensor Networks Using Particle Swarm Optimization

Ahmed Sarkhoo[1], Mohamed S. Yasein[2], Rehab F. Abdel-Kader[2]

Abstract

In this paper, an algorithm that aims at improving the coverage of Wireless Sensor Networks (WSN) is proposed. The coverage is improved in random deployment regions by employing some mobile nodes, which can actively move to desired locations. Coverage improvement is attained by incorporating Particle Swarm Optimization (PSO) algorithm tomaximize the sensing coverage area by finding the optimal placement for the mobile sensors that minimizes the overlap between the coverage regions of sensors. The continuous redeployment of the mobile sensor will decrease the uncovered regions and consequently improve the Quality of Service (QoS) of the network. Simulation results demonstrate the effectiveness of the proposed algorithm in increasing the coverage area.The performance of the proposed algorithm is compared with that of the exhaustive search gird based algorithm and results demonstrate that the proposed algorithm is competent for the dynamic deployment in WSNs and has better performance with respect to computation time and effectiveness than the grid quorum based node mobility algorithm.

Keywords: *Wireless Sensor Networks; Coverage; Sensor Deployment; Particle Swarm Optimization.*

1. INTRODUCTION

The coverage problem is one of the most crucial and basic problems in Wireless Sensor Networks (WSNs) [1][2]. High coverage rates lead to high quality of service (QoS) of the WSN. The deployment of the sensors in the field has the greatest impact on the coverage [3][4]. Many optimization and clustering algorithms have been proposed to improve the lifetime and the coverage for the network. WSNshave been successfully employed in many strategic domains such as military, industrial, traffic, health, and business applications. WSN came into prominence with the advancement of modern intelligent systems that are increasingly reliant on the sophisticated applications of sensing, target tracking, classification and control. WSNs consist of numerous, tiny, inexpensive autonomous nodes that are working collaboratively for acquiring information from the environment [5], an example is shown in Figure1. Sensors are multi-functional as they can sense the area of interest, perform data processing, and communicate with each other over limited predefined distances. Each sensor consists of small processor, antenna and small source of energy (battery).

[1] *B.Sc. in Computer Engineering, College of Engineering and Petroleum, Kuwait University, Kuwait 2002*
a_sarkhoo@hotmail.com
[2] *Electrical Engineering Department, Faculty of Engineering Port-Said University, Port Fouad 42523, Port-Said, Egypt.*
myasein@eng.psu.edu.eg, *rehabfarouk@eng.psu.edu.eg*

The design and the topology for the network can reduce the power consumption and increase the lifetime for the network. WSN deployment problem refers to determining positions for sensor nodes (or base stations) such that the desired coverage, connectivity, and energy efficiency can be achieved with as few nodes as possible.

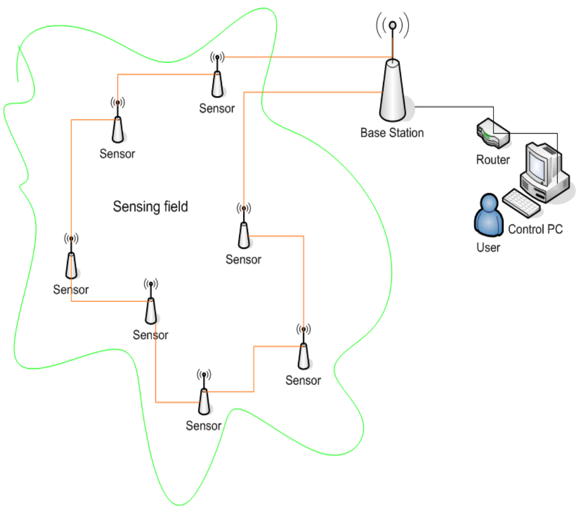

Figure 1: Wireless Sensor Network.

Random deployment of the sensors can't guarantee the best performance evenif the number of sensors is large. The coverage problem and the problem of placing mobile sensors to obtain high coverage have been studied in [6]. Different methods, such as optimization or clustering techniques, can be used to improve the coverage area after the initial deployment. Mobile sensors can be used to improve the coverage but the improvement will be directly dependent on the energy used to move the sensor form one location to another. In [7], the coverage problem for hybrid networks which comprise both static and mobile sensors is investigated. The tradeoff between mobile sensors density and the performance of the network measures was studied. Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) have been widely studied and used to improve the connectivity and coverage problems in WSN [8]. The GA is applied to solve complex design optimization problems because it can handle both discrete and continuous variables and nonlinear objective and constrain functions without requiring gradient information. Particle swarm optimization (PSO) is a simple, effective and computationally efficient optimization algorithm. It has been applied to address WSN issues such as optimal deployment, node localization, clustering and data-aggregation. In [9] PSO was introduced and the issues in WSN applications are discussed.

In [10] the PSO is used to improve the communication between the nodes to reduce the energy consumption which will increase the lifetime for the network. And in [11] the energy consumption problem is solved by using the Particle Swarm Optimization based Routing protocol (PSOR), where the best optimized path is selected to reduce the energy consumption. In [12] Comparative Research on Particle Swarm Optimization and Genetic Algorithm shows that the PSO performs better than GA in and it is computationally efficient (uses less number of function evaluations) than the GA.

The proposed algorithm, in this paper, aims at improving the coverage of the network by using hybrid network. The mobility is changed depending on percentage of mobility parameter (µ). The Particle Swarm Optimization is used after the initial deployment to improve the overall coverage. The PSO minimizes the overlap between the sensors by using a predefined constrains, using different sensors coverage radiuses and different mobility percentage.The movementsare constrained to different distances depending on the value of the maximum movement mobility (d_{max}).The results of the coverage, using the proposed algorithm, are compared with those of using the Grid based algorithm [13].

2. PROBLEM FORMULATION

Coverage is a vital measurement of the QoS of the WSN sensing function. Therefore sensor positioning is the key factor of the coverage problem. Sensors need to be optimally deployed to ensure that the sensing ability is fully exploited and the coverage area is maximized. Coverage problem can be modeled as a maximization problem. Given a set of sensors, $s = \{s_1, s_2 . . . s_n\}$, in a two dimensional area A. Where n is the number of sensors,

$$\mathbf{A = L.W,} \qquad \mathbf{(1)}$$

where L is the length and W is the width of the region of interest ROI. Area A consists of many sub-regions $A_{sr}$where $1\leq sr \leq L.W$, and each sub region has area of Ʊm2,where Ʊ is predefined, and the total number of sub-regions can be found as:

$$\boldsymbol{A_{sr}} = \frac{\boldsymbol{L.W}}{\mho^{2}} \mathbf{(2)}$$

Each sub-region is characterized by a point $r\_p(x_i, y_j)$ in the center of the sub region, where$0 \leq i \leq L \ and \ 0 \leq j \leq W$.Sensors are deployed and the number of sensors should be less than or equal $n_{\max}$, where $n_{\max}$ is the maximum number of sensors that can be purchased depending onthe budget. Each sensor $s_i$is represented as a point in the area $s(x_p, y_q)$ and that sensor can be static sensors s_s or mobile sensors s_m. The total number of sensors is

$$\boldsymbol{n_{s_{n_{max}}} = n_{s_m} + n_{s_s}} \qquad \mathbf{(3)}$$

The number of mobile sensors is given by

$$\boldsymbol{n_{s_m} = \mu . n_{s_{n_{max}}}} \qquad \mathbf{(4)}$$

Each sensor is located at coordinate $s(x_p, y_q)$inside A. Any point inside the ROI is covered by sensor i (s_i) if the distance between them are less than the s_i's sensing radius R. thus if all the points in the ROI are covered, the ROI is covered [14]. Sensor Coverage region $s_{cr(x_p,y_q)}$ can be described as

$$\left(\left(\mathbf{x} - \mathbf{c}_{\mathbf{x_p}}\right)^{\mathbf{2}} + \left(\mathbf{y} - \mathbf{c}_{\mathbf{y_q}}\right)^{\mathbf{2}}\right) \leq \mathbf{R}, \qquad \mathbf{(5)}$$

where $0 \leq p \leq L \ and \ 0 \leq q \leq W$.$(c_{x_p}, \ c_{y_q})$ is the center points of a sensor.Any location in A is said to be covered by a sensor s_i if it is within s_i sensing range.

Mobile Sensor s_m can move from one point $s_m(x_p, y_q)$ to another point $s_m(x_{p\mp d_m}, y_{q\mp d_m})$ with maximum distance of d_m in x and/or y directions, where $d_m \leq d_{max}$ and d_{max} is representing the energy for moving the sensors from one location to another.

Hence,

$$\mathbf{s_m(x_p, y_q)} \xrightarrow{\text{moves to}} \mathbf{s_m(x_p{}', y_q{}')}$$

, where $\|x_p - x_p{}'\| \le d_m$ and $\|y_q - y_q{}'\| \le d_m$

, and the total coverage is given by

$$\mathbf{C_{tot}} = \sum_{z=1}^{n_{s_m}} \mathbf{s_{m_z}}\left(\mathbf{x_{c_z}}, \mathbf{y_{c_z}}\right) + \sum_{t=1}^{n_{s_s}} \mathbf{s_{s_t}}\left(\mathbf{x_{c_t}}, \mathbf{y_{c_t}}\right), \quad \textbf{(6)}$$

,where $c_{X_{z,t}} = \begin{bmatrix} c_{x_1} \\ c_{x_2} \\ \vdots \\ c_{x_l} \end{bmatrix}$ and $c_{Y_{z,t}} = \begin{bmatrix} c_{y_1} \\ c_{y_2} \\ \vdots \\ c_{y_w} \end{bmatrix}$

Coverage percentage *C%* or the ratio of the coverage can be obtainedas

$$\mathbf{C\%} = \frac{\sum_{z=1}^{n_{s_m}} s_{m_z}(x_{c_z}, y_{c_z}) + \sum_{t=1}^{n_{s_s}} s_{s_t}(x_{c_t}, y_{c_t})}{L \times W} \,.\, \mathbf{100} \quad \textbf{(7)}$$

The PSO algorithm is utilized to maximize the coverage percentage C_{tot} by finding the best location which can give the maximum coverage for the mobile sensors, with the minimum number of sensors. Hence

Maximize

$$\mathbf{C_{tot}} = \sum_{z=1}^{n_{s_m}} \mathbf{s_{m_z}}\left(\mathbf{x_{c_z}}, \mathbf{y_{c_z}}\right) + \sum_{t=1}^{n_{s_s}} \mathbf{s_{s_t}}\left(\mathbf{x_{c_t}}, \mathbf{y_{c_t}}\right) \textbf{(8)}$$

Subject to

$\|X_p - X_p{}'\| \le d_m$ **and** $\|Y_q - Y_q{}'\| \le d_m$ **And**

$n_s \le n_{s_{nmax}}$

3. PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a computational method that returns the best result of an optimization problem by repetitively trying to improve a candidate solution with regard to a given measure of quality.Over a number of iterations, a group of variables have their values adjusted closer to the member whose value is closest to the target at any given moment. PSO can be implemented easily and it has a small computational cost compared to other algorithms. The algorithm keeps track of three global variables: Target value or condition, Global best (*gBest*) value indicating which particle's data is currently closest to the Target and Stopping value indicating when the algorithm should stop if the Target isn't found. Each particle consists of three thing:

I. Data representing a possible solution which is in the proposed algorithm is the sensors in the area $A\ s(x_p, y_q)$.

II. A velocity value indicating how much the data can be changed. After many tests and trail the value is set to 5 for the best results as mentioned in many papers such as [14][15]. Higher values such as 10 and 50 has been tested and showed that the effect is very small and will not make a remarkable difference in the results.

III. A personal best (*pBest*) value indicating the closest the particle's data has ever come to the Target. The *pBest* for our algorithm is when the coverage is maximum. Thus when $r\_p(x_i, y_j)$ is fully covered by $s(x_p, y_q)$.

PSO thrives to minimize or maximize the cost (fitness) function. In the proposed algorithm, the fitness function is indicated in Equation (8). The position where the best coverages$s(x_p, y_q)$or particle has its lowest cost is stored as (*pBest*). Besides, $gBest_d$, the position of the best particle. In each iteration *k*, velocity *V* and position *X* are updated using Equations (9) and (10).The update process is iteratively repeated until either an acceptable *gBest* is achieved or a fixed number of iterations $k_{max}$is reached.

$$\mathbf{V_{id}\,(k+1) = \tilde{\omega}V_{id}\,(k) + \phi_1\, r_1\,(k)\,(pBest_{id} - X_{id}) + \phi_2\, r_2\,(k)\,(gBest_{id} - X_{id})} \qquad \textbf{(9)}$$

, where ϕ_1 and ϕ_2 are constants, and $r_1\ (k)$ and $r_2\ (k)$ are random numbers uniformly distributed in [0, 1]; and *k* is the iteration number.

$$\mathbf{X_{id}\,(k+1) = X_{id}(k) + V_{id}(k+1)} \qquad \textbf{(10)}$$

PSO parameters and definitions

- $\tilde{\omega}$: the inertia weight that controls the motion of the particle: If $\tilde{\omega} << 1$, only small motion is kept from the earlier time-step; thus sudden alterations of the path are potential with this condition. Actually, the velocity depends on $\tilde{\omega}$, when $\tilde{\omega}=0$ the velocity completely lost and particle should move without the knowledge of last velocity. On the other hand, if $\tilde{\omega} > 1$ we have a similar result as C_1, C_2 is small, particle can barely change control and turned around, which of executing large area for exploration as well as lack of enthusiasm in finding the optimal solution [15][16].
- Velocity V_{max}: the rate V_{max} concludes the biggest change on particle movement in the search space during loop. In fact, a full search rang of the particle situation will be set as maximum velocity. However, with the use of inertial gravity in velocity will update recipe maximum velocity to some degree and it has become unnecessary. In spite of this fact, greatest velocity can improve the search for the optimal compound.

- Swarm size *S*: it is to a certain extent, a shared in the PSO papers to minimize the quantity of particles to the range 20-60. Kenedy [17] showed that there was a slight evolvement of the optimal worth with expanding swarm size; this paper the swarm will be represented as the number of mobile sensors in the area which will be changed from one test to another starting with 50 to 500 depending on the radius r for example if the radius is large then then maximum number of sensors will be less than 500 but if the radius small such as 1 then the maximum number of sensors should be increased to achieve the maximum coverage. 500 sensors will be used for r 5m or larger to save the run time in the simulation.
- The Coefficients C_1 and C_2: Kennedy [17] presented the popular method of selecting for acceleration coefficients C_1 and C_2, where C_1= 1 and C_2 =4-C_1= 3. However, there are other setting for C_1 and C_2 as discussed in other papers and the range is [0, 4]. Ratnaweera [16] investigated the result of varying acceleration coefficient with the time.
- The Fitness Function: is a specific type of aim function that measures the optimality of a solution in PSO. As it was mentioned in the previous section, the objective of this paper is to increase the coverage by relocation or moving the mobile sensors from one location to another to improve the coverage if possible. The fitness function for the coverage is defined in Equation (8).

The coverage in each sub region $r_p(x_i, y_j)$is checked during each iteration and if the coverage C_i is 1then the area is covered and if the C_i is 0 then the sensor will be placed to cover the hole. The popular action of updating the function is to keep the particle toward the target area. This fitness function is useful to each particle in turn through the update of the algorithm.

4. PROPOSED ALGORITHM

The process starts with a number of sensors $s_n$that are randomly distributed in a ROI(A) with a predefined length L and width W; then the algorithm performs the following steps:

1. Checking the initial number of sensorss_{min} in the region and storing the value.
2. Looking to the predefined mobility percentage µ to find the mobile sensors from Equation (4).
3. If the value is zero then PSO will not run. And the coverage can’t be improved with static sensors only. And the number of sensors is incremented by a number of sensors s_{min}.
4. If the mobility percentage is higher than zero, then the PSO will run.
5. PSO seeks to increase the coverage by minimizing the overlap between the sensors. By finding the best location for the mobile sensors.
6. By using Equation (7), the total coverage can be calculated.
7. The algorithm terminates by reaching the full coverage or the maximum number of sensors (n_{max} is reached).

The area A is divided into 10000 sub-region. Each sub-region will be represented as a point in the area $r\_p(x_i, y_j)$.As shown in Figure2.

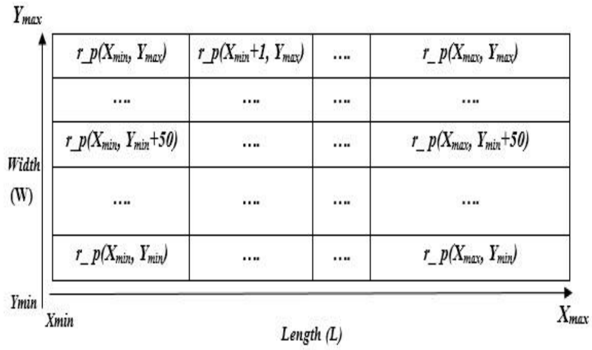

Figure 2:ROI with the Length X_{max} and Width Y_{max}.

Figure3 shows the deployment of Sensors in the area; they are colored to distinguish between the static sensors and mobile sensors.

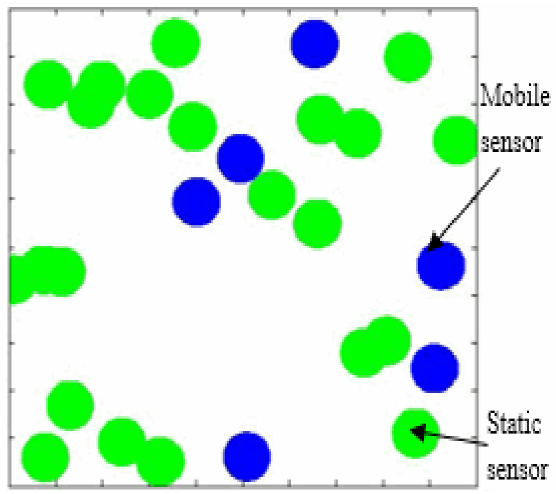

Figure3: Mobile and Static Sensors in ROI

Figure4 showsa flowchart of the proposed algorithm.

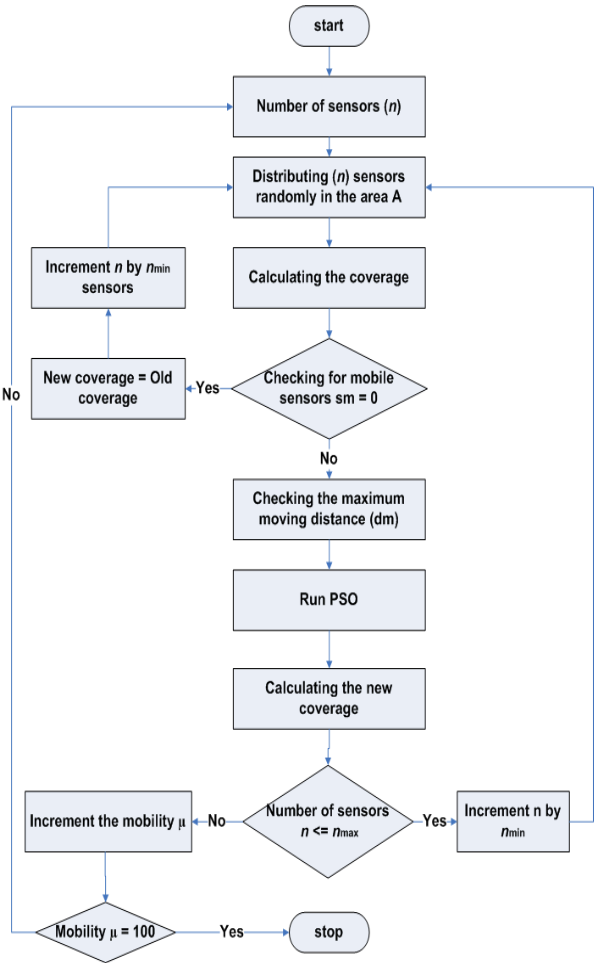

Figure4:Flowchart of the Proposed Algorithm

5. EXPERIMENTAL RESULTS

An area A of 100m length (L) and 100m width (W) is considered for monitoring using wireless sensor network. The network consists of number sensors n. The maximum number of sensors is n_{max}=500, for different radius. The algorithm is performed using MATLAB R2015a on windows 8 platform. The sensors are randomly deployed in the field and these sensors can be static sensors (s_s) or mobile sensors (s_m) or hybrid with predefined percentage of mobility $\mu = 25\%, 50\%, 75\%$ *and 100%*. The radius R will be changed from 1, 3, 5, 7, and 9m. Each sub-region is 1mlength and 1m width.

In the simulation, the number of iteration P_{itr} in the PSO is 4 and the number of particle $n_{particles}$ is 10. The values for the iteration and the particles has been selected based on tests where the runtime and the obtained results has been examined. With higher iteration number, the runtime increases and the resultsimprovement is not noticeable. The starting position for the particles is random $s(x_p, y_q)$, so each iteration will be repeated 4 times with different locations to get the best coverage.

5.1 Coverage Calculation

The coverage is calculated using the pixel difference coverage method. Assuming two sensors $s_i$and $s_j$located in positions (x_i ,y_i) and (x_j , y_j), respectively. The distance between the two sensors is defined by the Euclidian distance measure as

$$d(s_i , s_j) = \sqrt{|x_i - x_j|^2 + |y_i - y_j|^2} \qquad (11)$$

In the Pixel coverage method**,** the coverage is calculated by assuming that a covered region is represented by one’s and the uncovered area is represented byzeros. An example for calculating the coverage of a sensor s_i in the center of a square grid, where the center ispoint p (3, 3) is shown in Figure5. The length of the grid is X_{max} = 7m, and the width isY_{max} = 7m. The Total Coverage is obtained as

$$C_{tot} = \frac{\textbf{the count of ones (area covered)}}{\textbf{total area}} = \frac{13}{7*7}$$

= 0.266%

, which means that 26.6% is covered.

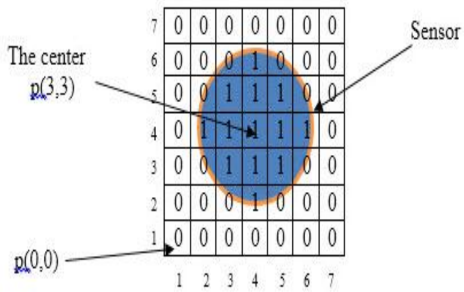

Figure5:A binary image of a sensor coverage area

An illustration of the pixel coverage method and how the PSO algorithm reduces the overlap between the sensors is shown in Figure6.Thered circle shows sensor s_j overlapping sensor s_i (blue) and this is during the first iteration. After the last iteration of the PSO, the overlap is reduced and the coverage improves.

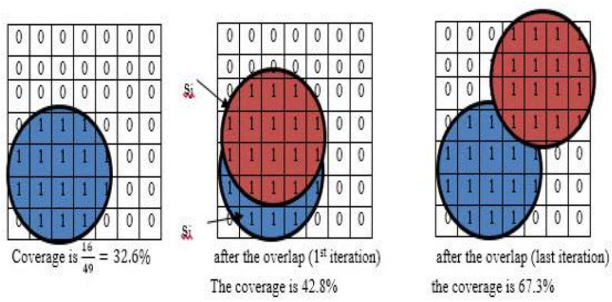

Figure6:Illustration ofpixel coverage method.

As in [18], the number of sensors required to cover the area A is given by

$$n_{optimal} = A/(3 * \sqrt{3} * R^2/2) \quad (12)$$

In the following sub-section the proposed algorithm is tested and examined for different parameters. The results and analysis are, then, shown and discussed.

The experiments were executed for five different predefined radiuses. Starting from R=1m with an increment of 2m up to R=9m.The maximum movability d_{max}, which represents the power needed for moving a sensor from one location to another, is used with various values. Each experiment was tested for 3 different movability distances starting with d_{max}=2.5m, d_{max}=10m, and finally with a simulated unlimited movability with d_{max}=100m.

5.2 Experiment Setup

In eachof the following experiments, a number of sensors $n_{min}$is deployed in the field randomly with all sensors having the same initial power. All sensors are static sensors, where the mobility is set to 0 and the initial coverage is calculated. Subsequently, additional n_{min} sensors with the same mobility percentage are deployed randomly so the total number of sensors in the field is$2n_{min}$.Then the percentage of mobility is incremented by 25 percent and the experiment is repeated. The proposed algorithm terminates when the maximum coverage is reached or the maximum number of mobile sensors in the field is reached(n_{max}=500) and all mobile sensors are used.

- **Testing the effect of changing mobility percentage, when the radius R =1m**

It can be seen in Figure7 that the maximum coverage is 19.3% with 50% mobility. Which is almost near the optimal solution which is 20%.

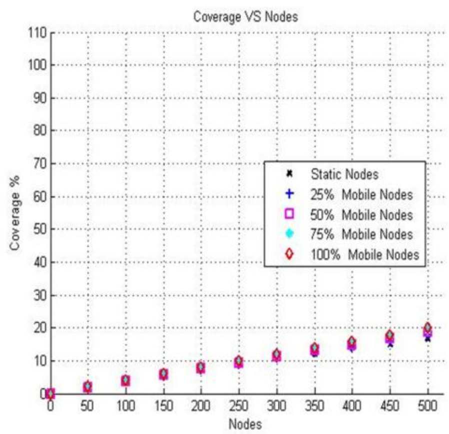

Figure7:Sensor Radius R=1m and d_{max}=100m

- **Testing the effect of changing mobility percentage, when the radius R=3m**

It can be seen in Figure8that the maximum coverage of 98% is achieved with 75% mobility, where the number of mobile sensors is 375 and the static sensors are 125 sensors.

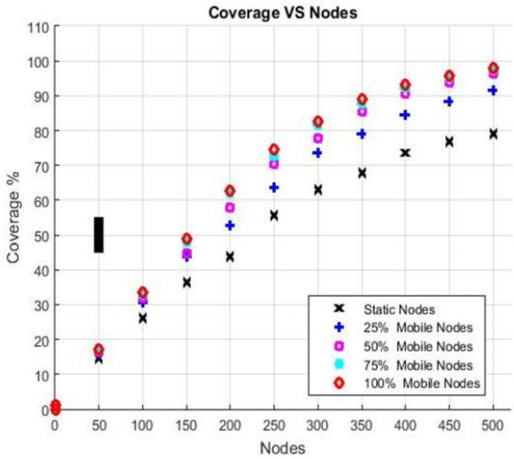

Figure8: Sensor Radius *R*=3m and d_{max}=100m

- **Testing the effect of changing mobility percentage, when the radius R =5m**

In Figure9, it can be seen that the full coverage can be reached with 300 sensors with mobility percentage of 25%. If 200 mobile sensors are used then the full coverage can be achieved but the cost will be bigger than the first choice. And if the full mobility is used with the unlimited movability then the full coverage can be achieved with 200 mobile sensors.

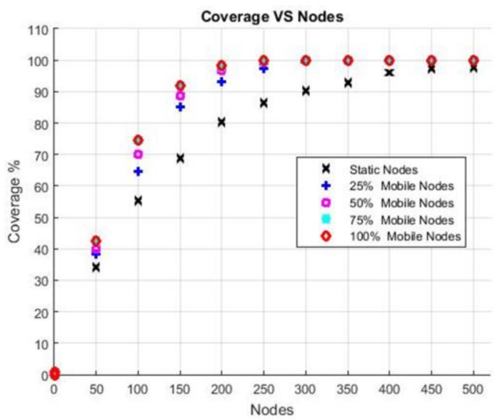

Figure9:Sensor Radius R=5m and d_{max}=100m

- **Testing the effect of changing mobility percentage, when the radius R=7m**

In Figure10, it can be seen that the full coverage can be reached with 200 sensors with mobility percentage of 25%. In other words, to get the full coverage we should use 50 mobile sensors with 150 static sensors. This selection can be used if we want to reduce the cost for the sensors to be deployed. However, if the cost is not important, then the minimum number of sensors is 100 mobile sensors to be distributed in the field.

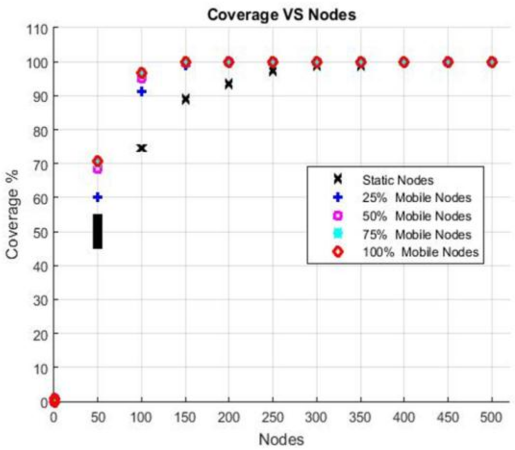

Figure10: Sensor Radius R=7m and d_{max} =100m

- **Testing the effect of changing mobility percentage, when the radius R=9m**

In Figure11, it can be seen that the full coverage can be reached with 150 sensors with mobility percentage of 25%. In other words, to get the full coverage we should use 38 mobile sensors with 112 static sensors. This selection can be used if we want to reduce the cost for the sensors to be deployed but the coverage is related to the initial coverage for the static sensors in the field. The minimum number of sensors will be 100 mobile sensors and it should be less than 100 but because of the step increment of 50 sensors we reached the 100 mobile sensors. Another test is done and the result shows that the minimum number of sensors required to get the full coverage is 48 mobile sensors.

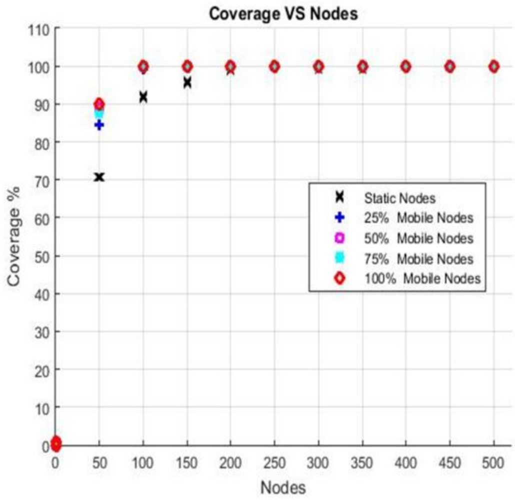

Figure11: Sensor Radius R=9m and d_{max}=100m

5.3 Analysis

From Figures 7 to 11, we can find the best number of sensors that can achieve the best coverage of each sensor radius. This helps in selecting the best radius for the sensors to be used, for each particular application. When the radius is equal to 5m and maximum distance to move is from 2.5m to 100m we can see that the full coverage can be achieved with 500 sensors and the full coverage can’t be guaranteed if the mobility is low, such as 25%. If the limitation in the maximum movement is set to 10 meters which is more flexible, we can get the full coverage with 400 sensors or less depending on the mobility percentage. In addition, if the limitation in the maximum movement is set to 100 meters, where the power of moving the sensor from one location to another is neglected, then the minimum number of sensors is 200 mobile sensors. In addition when the radius is equal to 7m, we can see that the full coverage can be achieved with 225 static sensors and 75 mobile sensors. In addition, if all sensors are mobile then the full coverage can be achieved with 225 mobile sensors. If the limitation in the maximum movement is set to 10m, which is more realistic, we can get the full coverage with 275 sensors or less depending on the mobility percentage or 150 mobile sensors. With 100m maximum movement limitation, where the power of moving the sensor from one location to another is neglected, then the minimum number of sensors is 100 mobile sensors.

Finally, when the radius is set to 9m**,**we can see that when the movement limitation is equal to 2.5m the full coverage can be achieved with 75 static sensors and 75 mobile sensors. In addition, if all sensors are mobile then the full coverage can be achieved with 150 mobile sensors, which is much costly compared to 50% mobility. If the maximum limitation in the movement is set to 10m, we can get the full coverage with 50 static sensors and 50 mobile sensors. If the maximum movement limit is set to 100m, then minimum required number of sensors is 100 sensors, where 50 static sensors and 50 mobile sensors are used to achieve the full coverage.

The simulation results demonstrate that if sufficient energy for the sensor movement is available, then the coverage can be improved significantly and full coverage can be achieved with the minimum number of sensors. However, if the energy is limited and the movement is constrained to small movability distances then using mobile sensors will cost more and the coverage improvement will not be significant compared to the static sensors.

By comparing the proposed algorithm with grid quorum based on node mobilityalgorithm [13], it can be noticed that the assumptions used in the proposed algorithm, are more realistic as the coverage in the proposed algorithmis considered for circle areas, while in [13], assumption is that each sensor can sense a square area, which is unrealistic. Moreover, another assumption, in [13], that the sensors can move without any restrictionis hard to implement.

The comparison of the results of the proposed algorithm with those of grid quorum based on node mobility algorithm [13], as shown in Table 1, shows that theproposed algorithm outperforms the Grid Quorum based algorithm at the scenarios that have largemaximum movement mobility (d_{max}), which is demonstrated by the lower number of required sensors.

Table 1. PSO and Grid Quorum

	d_{max}=2.5m	d_{max}=10m	d_{max}=100m
Grid Quorum	**350 sensors**	**355 sensors**	**360 sensors**
PSO	**350 sensors**	**225 sensors**	**200 sensors**

6. CONCLUSION

The proposed algorithm targets the efficient distribution for the sensors to achieve the best coverage for the region of interest. The PSO algorithm has been utilized to improve coverage of sensors by finding the best placement for the mobile sensor nodes in the WSN. In addition, it maximizes the coverage by reducing sensors overlap after the initial random distribution in the network. Different constraints on the maximum allowed movability distance of the mobile sensors were imposed. This regulates how far the mobile sensors can be moved from the old location to the new location. This value is directly proportional to the energy required for sensor movement. The performance of the proposed algorithm has been tested using different sensor radiuses, different maximum movability distances, and different mixtures of mobile and static sensor networks. The performance of the proposed algorithm has been compared with that of the exhaustive search gird based algorithm and results have demonstrated that the proposed algorithm is competent for the dynamic deployment in WSNs and has better performance with respect to computation time and effectiveness than the grid quorum based node mobility algorithm

REFERENCES

[1] Li, Wei, and Yuwei Wu. "Tree-based Coverage Hole Detection and Healing Method in Wireless Sensor Networks." Computer Networks (2016).

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks: A Survey”, Elsevier Journal on Computer Networks, Vol. 38, No. 4, Mar. 2002, pp. 393^122.

[3] Chong, Chee-Yee, and Srikanta P. Kumar. "Sensor networks: evolution, opportunities, and challenges." *Proceedings of the IEEE* 91.8 (2003): 1247-1256.

[4] Kim, Donghyun, et al. "Maximum lifetime dependable barrier-coverage in wireless sensor networks." *Ad Hoc Networks* 36 (2016): 296-307.

[5] Abbasi, Ameer Ahmed, and Mohamed Younis. "A survey on clustering algorithms for wireless sensor networks." *Computer communications* 30.14 (2007): 2826-2841. 136.

[6] Del Valle, Yamille, et al. "Particle swarm optimization: basic concepts, variants and applications in power systems." *Evolutionary Computation, IEEE Transactions on* 12.2 (2008): 171-195.

[7] Saipulla, Anwar, et al. "Barrier coverage with sensors of limited mobility."*Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2010.

[8] Ge, Ji-ke, et al. "Summary of genetic algorithms research." *Application Research of Computers* 25.10 (2008): 2911-2916.

[9] Poli, Riccardo. "An analysis of publications on particle swarm optimization applications." *Essex, UK: Department of Computer Science, University of Essex* (2007)..

[10] Hojjatoleslami, Samaneh, Vahe Aghazarian, and Ali Aliabadi. "DE based node placement optimization for wireless sensor networks." *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*. IEEE, 2011..

[11] Sarangi, Snehal, and Biju Thankchan. "A novel routing algorithm for wireless sensor network using particle swarm optimization." *IOSR Journal of Computer Engineering (IOSRJCE)* 4.1 (2012): 26-30.

[12] Li, Zhijie, et al. "Comparative research on particle swarm optimization and genetic algorithm." *Computer and Information Science* 3.1 (2010): 120.

[13] Mathur, Prateek, et al. "Coverage improvement for wireless sensor networks using grid quorum based node mobility." *Networking and Electronic Commerce Research Conference 2012*.

[14] Chakrabarty, Krishnendu, et al. "Grid coverage for surveillance and target location in distributed sensor networks." *Computers, IEEE Transactions on*51.12 (2002): 1448-1453.

[15] Quintão, Frederico Paiva, Fabíola Guerra Nakamura, and Geraldo Robson Mateus. "Evolutionary algorithm for the dynamic coverage problem applied to wireless sensor networks design." *Evolutionary Computation, 2005. The 2005 IEEE Congress on*. Vol. 2. IEEE, 2005.

[16] Ratnaweera, Asanga, Saman K. Halgamuge, and Harry C. Watson. "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients." *Evolutionary Computation, IEEE Transactions on*8.3 (2004): 240-255.

[17] Kennedy J & Eberhart R C. (1995). *Particle Swarm Optimization.* Proceedings of the IEEE International Conference on Neural Networks IV, IEEE computer society, Washington DC. P.1942-1948.

[18]http://stackoverflow.com/questions/7716460/fully-cover-a-rectangle-with-minimum-amount-of-fixed-radius-circles# last seen in (2016)