

Methods for Increasing Coverage in Wireless Sensor Networks

Sunggu Lee¹ and Younggyu Yang¹

¹Division of Electrical and Computer Engineering, Pohang University of Science and Technology, San 31 Hyoja Dong, Pohang, South Korea
{ slee, zhaoyue } @ postech.ac.kr

Abstract. In a Wireless Sensor Network, it is important to be able to maintain a sufficient level of *coverage* for a given target area. This paper addresses the problem of determining *how many* and *where* to place additional sensor nodes in order to increase the coverage from k to $k + 1$. Several possible candidate solution methods for this problem are proposed and evaluated using theoretical and simulation-based analysis.

Key words: Wireless Sensor Network (WSN), Coverage, Algorithm

1 Introduction

One of the most basic problems that must be addressed when working with Wireless Sensor Networks (WSNs) is the issue of the level of *coverage* provided by a set of WSN nodes. Since a WSN is most often used for “sensing” events of interest in a target area, *coverage* normally refers to *sensing coverage*. Given a target area being monitored by a set of WSN nodes, the area is said to be *covered* if any location within that area is within the sensing range of at least one WSN node.

More generally, a target area can be defined as being *k-covered* if any location within that area is within the sensing range of at least k WSN nodes. Depending on the WSN application, it may be necessary for an area to be *k-covered* with $k > 1$. For example, triangulation-based localization techniques require an object to be sensed by at least three WSN nodes. If localization is required within a 3-dimensional space, then four distance measurements are necessary (implying a coverage level of $k = 4$). Fault-tolerance requirements may dictate the use of a coverage level higher than the minimum for a particular application. Also, since energy conservation is critical for many WSN applications, it may be necessary for a region to be *k-covered* with $k \geq 2$ in order to be able to periodically place sets of nodes in suspended mode.

There are several ways in which WSN nodes could be positioned within a target area such that it is *k-covered* with a prespecified k value. If WSN nodes can be exactly positioned, then specific optimal layout patterns exist for several different types of areas and sensing coverage fields. For instance, if a 2-dimensional rectangular area and circular sensing coverage fields with identical radii are assumed, then a hexagonal beehive-type layout (such as that used for cell phone

networks) is clearly optimal. In many applications, however, WSN nodes cannot be exactly positioned. This could be the case with WSN nodes spread out over rugged terrain or extremely large numbers of WSN nodes deployed over a short time interval. However, given such a set of haphazardly-placed WSN nodes, it may still be possible to *add* WSN nodes at specific locations such that the target area becomes k -covered. Following discussion of related work in Section 2, this problem will be formalized, and several candidate solutions proposed and analyzed, in Section 3. Then, to further assess these candidate solutions, simulation results will be presented and analyzed in Section 4. Finally, conclusions will be drawn in Section 5.

2 Previous Related Work

There are several approaches related to obtaining a *desired* level of coverage for a sensing field. In [1], three general models are used to define the coverage problem. The first model is the *binary model*, where each sensor's coverage region is modeled by a disk. The second is the *probabilistic model*. An event that occurs within the coverage range of a sensor is either detected or not detected by the sensor depending on a probability distribution. The last model considers the coverage problem by considering the movement of targets through the sensing field.

To determine either a given region is k -covered or not, the coverage of each region must be computed. An approximate solution for evaluating the coverage of each region involves dividing the entire sensing field into $1\text{m} \times 1\text{m}$ *patches*. The coverage of a given region is approximated by measuring the number of active sensors that cover the patches within that region. Although this method is simple, managing the coverage values for all patches is a computationally heavy requirement. In [2], an alternative solution is proposed. Rather than determining the coverage of each patch, their solutions are based on checking the *perimeter* of each sensor's sensing range. An advantage of this solution is that it can be easily translated to a distributed protocol.

There are several methods used to obtain a desired level of coverage for a sensing field. [3] proposes a mechanism for step-by-step sequential deployment given a random initial deployment of sensors within the region to be monitored. The strategy consists of deploying a limited number of sensors at a time until the desired minimum exposure level is achieved. Several criteria are used for the efficient deployment of these sensors; however, the level of coverage achieved is not considered. [4] presents protocols that can be used to dynamically configure a network to achieve a guaranteed level of coverage and connectivity assuming that sufficient numbers of sensors have been deployed.

The main difference between this paper and the prior research work discussed above is that this work focuses on determining *how many* and *where* to place additional sensor nodes in order to *increase* the coverage level from k to $k + 1$. According to our survey of the public literature, this work is the first of its kind to address this particular problem.

3 Problem Definition and Candidate Solutions

Let us assume a set of WSN nodes $G = \{w_1, w_2, \dots, w_n\}$. The sensing range of a node $w_i \in G$ is modeled by a circular disk with radius r_i . For simplicity, let us also assume that $r_i = r_j$ for all $w_i, w_j \in G$ (uniform circular sensing ranges). However, note that, as described in [2], coverage computation for nonuniform and irregularly-shaped sensing ranges can be achieved using simple extensions to the techniques used for uniform circular sensing ranges. In a like manner, the techniques presented in this paper can be extended to other sensing range models.

Given a uniform circular sensing range model, a set of haphazardly-placed nodes results in a set of regions formed by the intersection of the sensing range circles of nodes $w_i \in G$. Each such region can be labeled with a “coverage” value corresponding to the number of nodes that can detect events within that region. Fig. 1 shows an example of a set of haphazardly-placed nodes and the resulting coverage values.

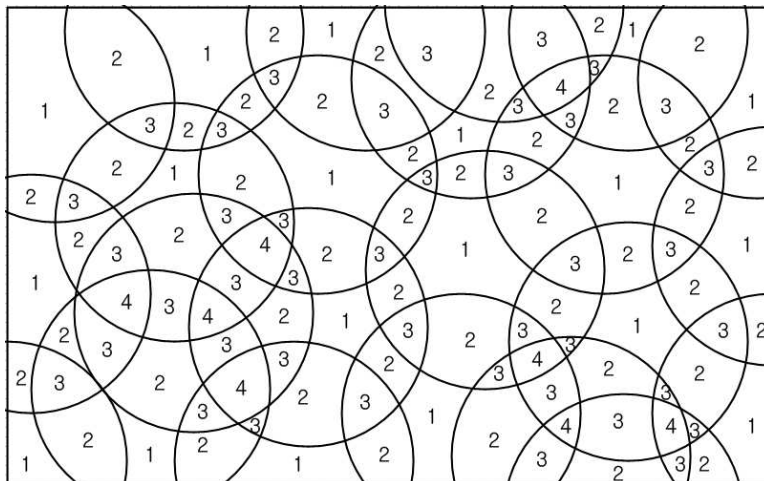


Fig. 1. A set of haphazardly-placed nodes and the resulting coverage values.

Definition 1. A coverage region CR_l is a contiguous area formed by the intersections of sensing range circles and the boundary of the target area A .

Definition 2. A coverage value CV_l is a nonnegative integer corresponding to the number of nodes that can detect events that occur within coverage region CR_l .

Definition 3. A node $w_i \in G$ covers a coverage region CR_l if w_i can sense any event that may occur within CR_l .

Definition 4. The coverage level CL of a set of nodes G is the minimum of the coverage values within the target area A covered by the nodes in G .

Definition 5. The intersection point set IP_l of a coverage region CR_l is the set of points at the boundaries of CR_l where two or more sensing range circles or the boundary of the target area A intersect.

Given the above definitions, let us define the problem to be solved. Let us assume that there are a set of initially-placed WSN nodes G used to sense events of interest within a target area A . There are no limitations on the manner in which these nodes are placed. Thus, in general, we can assume that these nodes are initially deployed in a semi-random haphazard manner. Methods such as those introduced in [2] can be used to compute coverage regions, coverage values, and the overall coverage level given the set of nodes G and the target area A . If the coverage level is sufficient for a particular application, then there is nothing more to be done. However, if the coverage level needs to be increased, then how should additional nodes be deployed in order to reach the desired coverage level? This problem can be stated more formally as follows.

Problem INCC (INcreasing Coverage)

Input: A target area A , a set of initially-placed nodes G , a set of coverage regions (with coverage values), the coverage level CL and the desired coverage level CL^* , with $CL^* > CL$.

Output: A set of additional nodes G^* and their locations within A . $G+G^*$ should have a coverage level of CL^* .

Can the INCC problem be solved in an optimal manner in polynomial time? As discussed in the introduction, if a regular target area is used, the coverage ranges of all nodes are identical and regular and the nodes in G are deployed in a specific layout pattern, then optimal solutions can be found in many cases by using predefined optimal layout patterns. However, in the general case, this is clearly a very difficult problem. More formally, it can be shown that the INCC problem is NP-hard. The proof of this statement follows from the fact that the Knapsack Problem, which is a well-known NP-hard problem [5], can be reduced (in polynomial time) to a restricted form of the INCC problem.

Several different heuristic solutions can be proposed for the INCC problem. One simple candidate solution is the random method, in which additional nodes are placed randomly in the target area A until the desired coverage level is reached. However, as will be shown in our simulation results, the random method is close to the worst solution method that can be used. Better solutions should result if coverage regions and coverage values are taken into account. As shown in [6], intersection point sets (Defn. 5) can be useful in determining the WSN node sleep cycles to be used for energy conservation. Thus, taking a hint from their method, let us also consider the use of intersection point sets in determining the placement of additional nodes used to solve the INCC problem. With these considerations, the following candidate solutions are proposed for the INCC problem.

Algorithm Random:

- 0: Let $G^* = \emptyset$.
- 1: Choose a location at random within the target area A .
- 2: Place a new node w_m at that location ($G^* = G^* + w_m$) and recompute CL .
- 3: If $CL \geq CL^*$, then stop. Otherwise, go back to Step 1.

Algorithm LowCoverage:

- 0: Assume $CL > 0$. Let $G^* = \emptyset$, $T = \{CR_i | CV_i = CL\}$.
- 1: Let CR_i = first element of T , $T = T - CR_i$.
- 2: Choose any node $w_i \in G \cup G^* | w_i$ covers CR_i .
- 3: Create a new node w_m and place it at the same location as w_i . Let $G^* = G^* + w_m$.
- 4: If $|T| > 0$, go back to Step 1.
- 5: Recompute the coverage level CL with $G + G^*$. If $CL \geq CL^*$, stop. Otherwise, let $T = \{CR_i | CV_i = CL\}$ and go back to Step 1.

Algorithm MaxIP:

- 0: Let $G^* = \emptyset$, $T = \{CR_i | CV_i = CL\}$.
- 1: Let CR_i = an element of T with the maximum number of intersection points, $T = T - CR_i$.
- 2: Create a new node w_m and place it at the “center of mass” of the intersection points in CR_i ($G^* = G^* + w_m$).
- 3: If $|T| > 0$, go back to Step 1.
- 4: Recompute the coverage level CL with $G + G^*$. If $CL \geq CL^*$, stop. Otherwise, let $T = \{CR_i | CV_i = CL\}$ and go back to Step 1.

Algorithm MaxIPinc:

- 0: Let $G^* = \emptyset$, $T = \{CR_i | CV_i = CL\}$.
- 1: Let CR_i = an element of T with the maximum number of intersection points.
- 2: Create a new node w_m and place it at the “center of mass” of the intersection points in CR_i ($G^* = G^* + w_m$).
- 3: Recompute the coverage level CL with $G + G^*$. If $CL \geq CL^*$, stop. Otherwise, let $T = \{CR_i | CV_i = CL\}$ and go back to Step 1.

Algorithm MaxCV:

- 0: Let $G^* = \emptyset$, $maxCV$ = maximum CV_i value, $T = \{CR_i | CV_i = maxCV\}$.
- 1: Let CR_i = an element of T with the maximum number of intersection points.
- 2: Create a new node w_m and place it at the “center of mass” of the intersection points in CR_i ($G^* = G^* + w_m$).
- 3: Recompute the coverage level CL with $G + G^*$. If $CL \geq CL^*$, stop. Otherwise, let $T = \{CR_i | CV_i = CL\}$ and go back to Step 1.

The motivation behind the above algorithms is as follows. The Random algorithm is the simplest to implement since it uses random new node deployment. The LowCoverage algorithm is a simple method for ensuring that the coverage level is increased in a systematic manner. It has the following property.

Theorem 1. *Suppose $CL = 1$ and $CL^* = 2$. Then Step 5 of the LowCoverage algorithm will be executed only once. In addition, the minimum coverage value will not increase until the last iteration (inner loop) of the LowCoverage algorithm.*

Proof. In Step 1, T is defined as the set of coverage regions CR_l with the same coverage value as CL . By placing one new node at the same location as an existing node that covers a CR_l in T , CV_l will increase by one. This new node will not cover any other coverage region in T since the initial minimum coverage value (which is the same as the initial CL) is 1. Thus, with each inner-loop iteration of the LowCoverage algorithm, exactly one element of T will have its coverage value increased by one. When Step 5 is reached, all elements of T will have been traversed. Thus, the coverage level will increase at this point and the algorithm should terminate since $CL^* = CL + 1$. Finally, as long as elements remain in T , those elements will have their old coverage values. This proves the second part of the theorem. ■

The MaxIP algorithm places a new node at the “center of mass”¹ [7] of the intersection points of a coverage region with the minimum coverage value. This is done for all coverage regions with the minimum coverage value. However, even after all such regions have been covered, it is still possible for new smaller regions to be formed that still have the old minimum coverage value. Thus, this process needs to be repeated until the coverage level increases to the desired value. MaxIPinc is an “incremental” form of the MaxIP algorithm that recomputes the coverage level after each new node has been added.

Finally, in the MaxCV algorithm, we consider the possibility of placing new nodes within the coverage regions with the largest coverage values. The reasoning behind this strategy is that new nodes placed at such locations should have the “biggest” potential impact since they will impact the largest number of other coverage regions. However, simulations with this algorithm revealed that this impact was “negative” in most cases. Thus, the MaxCV algorithm could be considered as a “worst-possible” solution (useful for comparison purposes) for the INCC problem.

4 Simulation Analysis

Computer simulations were used to compare the proposed solutions for the INCC problem. A square target area (X meters on a side) was assumed. An initial set

¹ The “center of mass” of a system of particles is a specific point at which, for many purposes, the system’s mass behaves as if it were concentrated.

of WSN nodes were generated randomly within this area until a specific initial coverage level CL was achieved. However, in order to avoid extreme clustering of WSN nodes (we would also want to avoid such clustering in most practical application scenarios), if a newly generated random node was found to fall within the sensing region of an existing node (or E existing nodes in the more general case), then that newly generated node was discarded and another node generated randomly. The sensing range of all WSN nodes were assumed to be identical and in the shape of a circular disk with radius R . Simulation experiments were conducted with all of the proposed algorithms and various values for the X , CL and R parameters.

Fig. 2 shows the simulation results for all five proposed algorithms given $X = 40\text{m}$ ($40\text{m} \times 40\text{m}$ target area) and $R = 5\text{m}$ (sensing radius). This figure

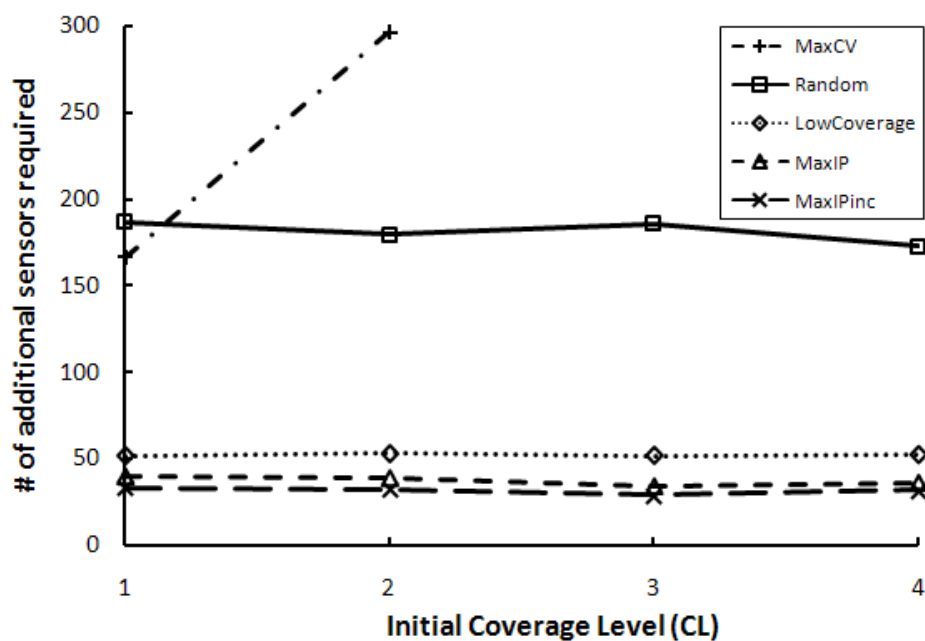


Fig. 2. Number of additional nodes required to increase CL to $CL + 1$ given a $40\text{m} \times 40\text{m}$ target area and WSN nodes with 5m sensing radii.

shows the trends in performance (number of additional nodes required to increase the coverage level by one) of the various algorithms as the initial coverage level CL is increased. The MaxCV algorithm was found to perform so poorly that it was not able to complete in a reasonable amount of time given $CL > 2$. The MaxCV and Random algorithms were found to perform significantly worse than

the LowCoverage, MaxIP and MaxIPinc algorithms. The MaxIPinc algorithm was found to perform the best.

In order to investigate the trends in performance as the size of the target area was increased, simulations were conducted with fixed CL and R and varying X parameter values. Fig. 3 shows these results for the Random, LowCoverage, MaxIP and MaxIPinc algorithms. The MaxCV algorithm was not simulated as

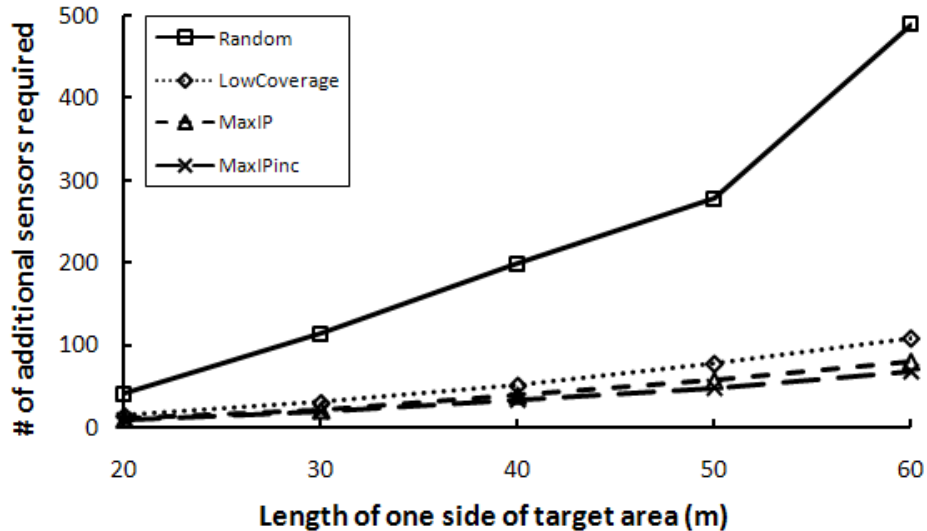


Fig. 3. Number of additional nodes required to increase $CL = 1$ to $CL^* = 2$ (given sensing radius $R = 5\text{m}$) versus X (assuming an $X \times X$ target area).

its performance was found to be significantly poorer than all other algorithms (except the Random algorithm in certain situations), as demonstrated in the plot of Fig. 2. The results of Fig. 3 showed a linear increase in the number of additional nodes required to increase the coverage level by one (versus X , the length of one side of the target area) for the LowCoverage, MaxIP and MaxIPinc algorithms. The relative performance of all algorithms was the same as in Fig. 2.

5 Concluding Remarks

This paper investigated the problem of increasing the minimum sensing coverage value in a target area populated with wireless sensor network (WSN) nodes. By ensuring a minimum sensing coverage value, referred to as the *coverage level* CL , any event that occurs anywhere within the target area can be monitored by at least CL nodes. Alternatively, given $CL > 1$, subsets of the nodes can be placed in sleep mode in order to conserve energy and thereby extend the lifetime of the WSN.

The coverage level issue has been approached by investigating the problem of determining how many and where to deploy additional nodes in order to increase the coverage level to a specific desired value. A system model has been developed and the above problem has been formally defined. Five candidate solution methods have been proposed, including a deterministic method that can guarantee a coverage level increase of one with a fixed number of additional nodes. Simulation studies show that a policy of incrementally deploying additional nodes in regions with large numbers of intersecting sensing range circles results in the best overall performance.

References

1. Wu, S.-L., Tseng, Y.-C.: *Wireless Ad Hoc Networking*, Auerbach Publications (Taylor and Francis), Boca Raton, (2007).
2. Huang, C.-F., Tseng, Y.-C.: The Coverage Problem in a Wireless Sensor Network. Proc. the 2nd ACM international conference on Wireless sensor networks and applications (WSNA), San Diego, California, USA, pp. 115–121 (2003).
3. Clouqueur, T., Phipatanasuphom, V., Ramanathan, P., Saluja, K. K.: Sensor Deployment Strategy for Target Detection. Proc. International Workshop Wireless Sensor Networks and Application, pp. 42–48 (2002).
4. Wang, X., Xing, G., Lu, C., Pless, R., Gill, C.: Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. Proc. the 1st international conference on Embedded networked sensor systems (SenSys), Los Angeles, California, USA, pp. 28–39 (2003).
5. Garey, M. R., Johnson, D. S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman and Co., New York (1979).
6. Huang, C.-F., Lo, L.-C., Tseng, Y.-C., Chen, W.-T.: Decentralized Energy-Conserving and Coverage-Preserving Protocols for Wireless Sensor Networks. Proc. IEEE International Symposium on Circuits and Systems (ISCAS), Vol. 1, pp. 640–643 (2005).
7. Wikipedia, <http://en.wikipedia.org/>