Estimating Coverage Holes and Enhancing Coverage in Mixed Sensor Networks

Amitabha Ghosh
Honeywell Technology Solutions Laboratory
151/1 Doraisanipalya, Bannerghatta Road
Bangalore 560076, INDIA
amitabha.ghosh@honeywell.com

Abstract

Area coverage is one of the most fundamental problems in adhoc wireless sensor networks because it directly relates to optimization of resources in a sensing field. Maximizing the coverage area while maintaining a lower cost of deployment has always been a challenge, especially when the monitoring region is unknown and possibly hazardous. In this paper, we present a method to deterministically estimate the exact amount of coverage holes under random deployment using Voronoi diagrams and use the static nodes to collaborate and estimate the number of additional mobile nodes needed to be deployed and relocated to optimal positions to maximize coverage. We follow a two-step deployment process in a mixed sensor network and we argue by simulation and analysis that our collaborative COVerage ENhancing algorithm (COVEN) can achieve a tradeoff between the cost of deployment and percentage of area covered.

1. Introduction

Adhoc wireless sensor networks (AWSN) have been the focus of many research topics in recent years. Due to advancement in MEMS (Micro Electro Mechanical System), a future, where battery powered, resource constrained, small and intelligent sensors instrumenting the environment can be envisioned. A typical AWSN would consist of thousands of sensors deployed over a geographical region, each of the sensors having wireless communication capabilities with other sensors, and accomplishing a task as specified by an application, in a collaborative fashion. A sensor by itself is a resource constrained entity with low-power signal processing, computation and communication capabilities and can sense only a limited portion of the environment; however, when a group of sensors collaborate they can accomplish a task efficiently.

Energy consumption is of critical importance in AWSN because the sensors are typically battery powered and once deployed, they are usually unattended. Hence, each of the sensors is expected to work for months or years without any battery replacement. In certain applications, the locations of the nodes can be pre-determined and nodes can be manually placed; while in other cases, especially in disaster recovery scenarios (wild-fire break out, chemical contamination, earthquake etc), a random deployment strategy is often preferred and feasible. However, random placement does not guarantee full coverage because it is stochastic in nature. It might result in accumulation of nodes in certain areas and very few nodes or no nodes in other areas. Some of the deployment algorithms existing in the literature find out new optimal sensor locations after an initial random placement and then move the sensors to those locations, thus enhancing coverage. Drawback of these algorithms is that they are applicable only to mobile sensor networks, where all the nodes have locomotion capabilities besides sensing and communication capabilities. However, providing mobility to all the nodes might be very expensive and in some cases, it might not be required to move all the nodes to new positions. A deployment comprising balanced number of static and mobile nodes is a preferable solution in these scenarios.

The ratio of static to mobile nodes is dependent on the coverage requirements of the specific application and on the cost of deployment. For instance, a military surveillance application will need a high degree of coverage in high security regions, whereas other kinds of applications might need a framework where it can dynamically configure the degree of coverage. An example of this latter kind of application is an intruder detection system, where restricted regions are usually mon-

Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)
0742-1303/04 $ 20.00 IEEE

itored with a moderate degree of coverage untill a possibility of intrusion takes place. At this point, the network will need to reconfigure and increase the degree of coverage at possible threat locations.

In this paper, we address the problem of enhancing coverage in a mixed sensor network. In the first part of the problem, we find out an exact estimate of the amount of coverage holes (area not covered by any sensor) in a static sensor network. Then, we address the problem of enhancing coverage by deploying an estimated number of additional mobile nodes. We assume that initially a certain number of static nodes are randomly deployed in the sensing field. The static nodes find out coverage holes around their neighborhood using the structure of Voronoi diagrams and then collaborate among each other to estimate the number of additional mobile nodes that needs to be deployed and relocated to the holes' locations to maximize coverage. The static nodes also find out the optimal positions of those mobile nodes based on certain heuristics.

The rest of the paper is organized as follows. Section 2 discusses related works. In section 3, we provide a theoretical background on Voronoi diagrams and formally introduce the problem. In section 4, we discuss an approach to exact estimation of coverage holes in a sensing field under random deployment. Section 5 describes the collaborative coverage enhancing algorithm. In section 6, we list down our evaluation strategy and simulation results. Section 7 gives an analysis of the proposed algorithm and section 8 concludes the paper.

2. Related Works

Area coverage and node connectivity are two fundamental and related problems in AWSN. An array works have been done by researchers that focus on maximizing coverage in a sensing field, while maintaining connectivity across the network.

The authors in [4] use the concept of Voronoi diagrams and Delaunay triangulation to provide polynomial-time worst case and best case algorithms for determining maximal breach path and maximal support path, respectively, in a sensing field. Their approach assumes a centralized computation model and apriori knowledge of the sensor-locations. On similar lines, in [5] and [7], the authors use the concepts of minimal and maximal exposure paths to find out how well an object moving on an arbitrary path can be observed by the sensor network over a period of time. The algorithm in [5] uses certain graph theoretic abstractions and compute minimal exposure path using Djikstra's Single Source Shortest Algorithm or Floyd-Warshal's All Pair Shortest Algorithm. The algorithm in [7] gives a closed form expression using variational calculus, for minimal exposure path in the presence of a single sensor and gives a Voronoi diagram based localized algorithm to find an approximate minimal exposure path in the presence of multiple sensors. The authors also propose a few heuristics based on random path, shortest path, best point and adjusted best point path to approximate the maximal exposure path.

Two other approaches to the coverage problem have been made using the concepts of potential field in [6] and [3], and virtual forces in [10]. In [6], the authors propose a deployment strategy using mobile-autonomous robots that maximizes the area coverage with the constraint that each of the nodes has atleast K neighbors. They model the sensing field using attractive and repulsive forces exerted on each node by all other nodes and rely on an equilibrium criteria that when the net force on each sensor becomes zero, the network stabilizes. However, their algorithm is computationally expensive because as the network size grows or new node joins, all the nodes need to reconfigure themselves to satisfy the equilibrium criteria. In a similar fashion, the authors in [10] propose a deployment strategy to enhance coverage after an initial random placement of sensors using virtual forces. A cluster head computes the new locations of all the sensors after an initial deployment that would maximize coverage and then nodes reposition themselves to the designated locations. However, the algorithm does not provide any route plan for repositioning to avoid collisions.

In [2], the authors provide a polynomial-time, greedy, iterative algorithm to determine the best placement of one sensor at a time in a grid based scenario, such that each grid is covered with a minimum confidence level. They model the obstacles as static objects and assume that a complete knowledge of the terrain is available.

The work presented in this paper most closely resembles to the work done in [9]. The authors discuss three distributed self deployment algorithms (VEC, VOR and Minimax) for mobile sensors using Voronoi diagrams. After an initial random deployment, the algorithms find out coverage holes in the sensing field and calculate new optimal positions to which the sensors should move from densely populated regions to increase coverage. However, their work does not attempt to estimate the amount of coverage hole existing within each Vornoi polygon. Also, to reduce the coverage holes their algorithms move the sensors close to the farthest Voronoi vertex, which does not always guarantee maximum coverage. The same authors in [8] describe a

protocol, called the bidding protocol, for mixed sensor networks that use both static and mobile sensors to achieve a cost balance. They reduce the problem to the NP-hard set-covering problem and provide heuristics to solve it near optimally. Their algorithm also considers a random initial deployment, where static sensors find out coverage holes based on Voronoi diagrams and place bids for the mobile sensors to move to the farthest Voronoi vertex. The mobile sensors receive all such bids from its neighboring static sensors and chooses the highest bid and move to the new location to heal the coverage hole. The work presented in this paper is different from the afore-mentioned algorithms in the following aspects: 1) it provides a computational geometry approach based on Voronoi diagrams to measure the exact amount of coverage holes, 2) based on the estimated holes, our collaborative algorithm then finds out optimal locations and the number of additional mobile sensors that needs to be deployed to enhance coverage.

3. Theoretical Framework

3.1. Voronoi Diagrams

Voronoi diagrams [1] are well known structures in computational geometry. In 2-dimension, the Voronoi diagram of a set of discrete points divides the plane into a set of convex polygons according to the nearest neighbor rule: all points inside a polygon are closest to only one point. Let S be the set of points (also called sites) in a 2-dimension plane. For two distinct points $p_1, p_2 \in S$, the dominance of p_1 over p_2 is defined as the subset of the plane which is at least as close to p_1 as to p_2. Mathematically, $dom(p_1, p_2) = \{x \in R^2 \mid d(x, p_1) \leq d(x, p_2)\}$, where $d(.)$ is the Euclidean distance function. The function $dom(p_1, p_2)$ is a closed half plane bounded by the perpendiculat bisector of p_1 and p_2, which separates all points in the plane closer to p_1 than those closer to p_2. The region of site p_1 is the portion of the plane that lies in all the dominances of p_1 over the remaining sites in S. Formally, $region(p_1) = \bigcap_{p_2 \in S - \{p_1\}} dom(p_1, p_2)$.

We note that every point on an edge is equidistant from exactly two sites, and each vertex is equidistant from at least three. Consequently, the regions form a convex polygonal partition of the plane. This partition $V(S)$, is called the Voronoi diagram of the point set S. A region of a site p_1 cannot be empty since it contains all points of the plane at least as close to p_1 as to any other sites in S. It follows that $V(S)$ contains exactly N sites if the set S contains N distinct points. Some of these sites are necessarily unbounded and they lie on the boundary of the convex hull of S, because only for those sites there exist points arbitrarily far away but still closest. It should be noted that no vertices occur if the points lie in a single straight line.

In Fig. 1(a), twenty randomly placed points divide the bounded rectangle into twenty convex regions, referred to as Voronoi polygons. A pair of sites p_i and p_j are called Voronoi neighbors if their Voronoi polygons share a common edge. The edges that constitute the polygon for a site p_i are called its Voronoi edges. Note that, the number of edges of a Voronoi polygon of p_i is equal to the number of its Voronoi neighbors.

Our approach to an exact calculation of holes is largely dependent on the properties of Voronoi diagrams because they contain proximity information in an explicit and computationally useful manner. The static nodes once deployed, divide the region into Voronoi polygons, such that all points inside a polygon are closest to only one node. Hence, this node is the best candidate to sense all points inside its polygon. To construct the Voronoi polygons, each node needs to know information only about its Voronoi neighbors.

3.2. Problem Statement

Given an environment whose topological information is not available and a certain number N_s of static nodes already randomly deployed in the sensing field, our problem is to find the following.

- An exact estimate of the amount of coverage holes in the sensing region.
- An optimal estimate of the number of additional mobile nodes that needs to be deployed and relocated to reduce or eliminate coverage holes.
- To find out the optimal locations for the mobile nodes and a strategy to deploy them accurately.

We assume that each static node knows their own location and their Voronoi neighbors' locations. We also assume that all the nodes have the same sensing radius. Once the Voronoi diagram is constructed, the complexity of calculating an exact estimate of the coverage holes and determining optimal locations for the additional mobile sensors within each polygon for maximizing coverage, arises because of several factors: 1) the sensing circle of a node might intersect its Voronoi edges at different points, 2) the sensing circle can be superscribed or inscribed inside the Voronoi polygon and 3) since the nodes are deployed randomly, there will be overlapping sensing areas. The calculation of holes should take into consideration such cases and find out the optimal positions of the mobile nodes.

4. An Exact Estimation Of Holes

In this section, we describe a Voronoi diagram based approach for estimating the exact amount of coverage holes in a sensing field. Our method is based on the structure and nearest neighborhood property of Voronoi diagrams. According to this property, all the points inside a Voronoi polygon is closest to only one sensor that lies within that poygon. Hence, if some portion of the polygon is not covered by the sensor lying inside the polygon, it will not be covered by any other sensor, thus contributing to coverage holes. We introduce the following notations, as explained in Table 1, to describe the algorithms that follow.

s_i, m_j	a static and a mobile node, respectively
R_s	uniform sensing radius of all the nodes
$V(s_i)$	Voronoi polygon of node s_i
$d(s_i, s_j)$	Euclidean distance between s_i and s_j
$N(s_i)$	set of Voronoi neighbors of s_i
V_i	a Voronoi vertex
(x_i, y_i)	coordinates of node s_i
l_{ij}	length of the common Voronoi edge between nodes s_i and s_j
L_{ij}	the line extending the Voronoi edge between nodes s_i and s_j
m_{ij}	gradient of line L_{ij}

Table 1. Notations and their meanings

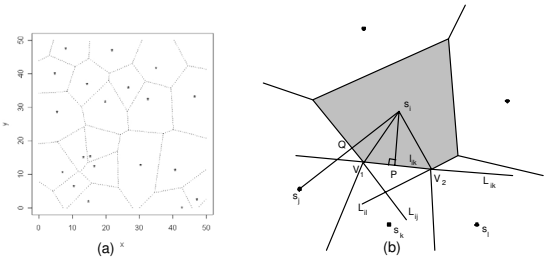


Figure 1. (a) Voronoi diagram formed by 20 randomly placed nodes (b) Voronoi polygon and Voronoi neighbors of node s_i.

4.1. Estimation Of Coverage Holes

To find out the length of the common Voronoi edge between two nodes, we need to know the locations of three Voronoi neighbors. As shown in Fig. 1(b), let us consider a node s_i at location (x_i, y_i) and three of its Voronoi neighbors: s_j at location (x_j, y_j), s_k at location (x_k, y_k) and s_l at location (x_l, y_l). Our goal is to find out the area of the triangle $\Delta s_i V_1 V_2$.

Gradient of line L_{ij}, which is perpendicular to the line connecting s_i and s_j is $-(\frac{x_i - x_j}{y_i - y_j})$, and coordinate of point Q, which lies in the midpoint of the line connecting s_i and s_j is $(\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2})$. Thus, equation of the line L_{ij} is: $L_{ij} : y - \left(\frac{y_i+y_j}{2}\right) = -\left(\frac{x_i-x_j}{y_i-y_j}\right)\left(x - \frac{x_i+x_j}{2}\right)$. In a similar way, we can find the equations of lines L_{ik} and L_{il}.

Solving the equations of L_{ik} and L_{ij} we find the coordinates of V_1 and solving the equations of L_{ik} and L_{il} we find the coordinates of V_2. Then we calculate the length l_{ik} of the line segment V_1V_2. Therefore, the area of the triangle $\Delta s_i V_1 V_2$ is $A_{\Delta s_i V_1 V_2} = \frac{l_{ik} d(s_i, s_k)}{4}$. Note that area of the Voronoi polygon for node s_i is equal to the sum of the triangles contained within the polygon. We term such a triangle as a Voronoi triangle and denote it by Δ_{ik}, when the base of the triangle is formed by the perpendicular bisector of node s_i and s_k. Thus, the area of the Voronoi polygon for node s_i is $A_{V(s_i)} = \sum_{s_k \in N(s_i)} A_{\Delta s_i V_1 V_2}$.

To calculate the exact coverage hole inside a Voronoi polygon, we need to consider the intersection points of the sensing circle of s_i and the sides of each of the Voronoi triangles. The following cases might arise depending on the radius of the sensing circle and whether the triangle is acute or obtuse. Let us denote the lengths of the two sides of a Voronoi trinagle by l_1 and l_2, and the top angle that it makes with the center of the sensing circle by x_{ik}, where x_{ik} is given by the cosine rule, $x_{ik} = \cos^{-1}\left(\frac{l_1^2 + l_2^2 - l_{ik}^2}{2 l_1 l_2}\right)$.

4.1.1. $R_s \leq d(s_i, s_k)/2$: When the sensing radius is less than or equal to half the distance between node s_i and its Voronoi neighbor s_k, the sensing cricle intersects only two sides of the triangle as shown in Fig.

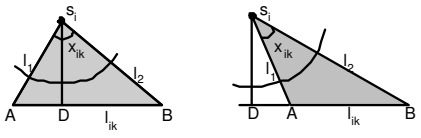


Figure 2. Intersection of sensing circle of s_i and Voronoi triangle $\Delta s_i AB$ when $R_s \leq d(s_i, s_k)/2$, $s_i D = d(s_i, s_k)/2$

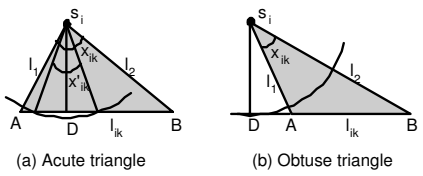


Figure 3. Intersection of sensing circle of s_i and Voronoi triangle $\Delta s_i AB$ when $R_s < l_1, R_s < l_2, R_s > d(s_i, s_k)/2$, $s_iD = d(s_i, s_k)/2$

2. Hence, area of the coverage hole inside this triangle is $A_{C(\Delta_{ik})} = \frac{l_{ik} d(s_i,s_k)}{4} - \frac{x_{ik} R_s^2}{2}$.

4.1.2. $R_s \geq l_1$, $R_s \geq l_2$: When the sensing radius is greater than or equal to both the sides of the Voronoi triangle, the sensing circle circumscribes the triangle. Hence, the coverage hole within the triangle is zero.

4.1.3. $R_s > d(s_i, s_k)/2$, $R_s < l_1$, $R_s < l_2$: In this case, the sensing radius is less than both the sides of the Voronoi triangle but greater than half the distance between node s_i and its neighbor s_k. The sensing circle intersects either two or three edges of the triangle as shown in Fig. 3. In Fig. 3(a), when the angle $\angle s_i AB$ is acute, the coverage hole is given by: $A_{C(\Delta_{ik})} = \frac{l_{ik} d(s_i,s_k)}{4} - \frac{R_s^2 (x_{ik} - x'_{ik})}{2} - \frac{d^2(s_i,s_k)}{4} \tan \frac{x'_{ik}}{2}$, where, $x'_{ik} = 2\cos^{-1} \frac{d(s_i,s_k)}{2R_s}$, and when the angle $\angle s_i AB$ is obtuse as shown in Fig. 3(b), the coverage hole is the same as given by $A_{C(\Delta_{ik})}$ in section 4.1.1.

4.1.4. $l_1 < R_s < l_2$: In this case, the sensing radius is greater than one of the sides of the Voronoi triangle and less than the other one. This is illustrated in Fig. 4(a) and Fig. 4(b) depending on whether the angle $\angle s_i AB$ is acute or obtuse, respectively. The coverage hole is $A_{C(\Delta_{ik})} = \frac{l_{ik} d(s_i,s_k)}{4} - \frac{R_s^2 (x_{ik} - x'_{ik_1} \mp x'_{ik_2})}{2} - \frac{d(s_i,s_k)}{2} \sqrt{l_1^2 + R_s^2 - 2 l_1 R_s \cos(x'_{ik_1} \pm x'_{ik_2})}$, where the

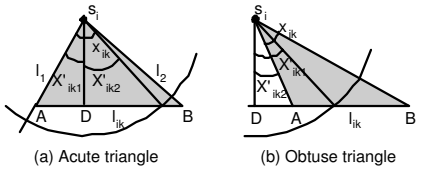


Figure 4. Intersection of sensing circle of s_i and Voronoi triangle $\Delta s_i AB$ when $R_s > l_1, R_s < l_2$, $s_iD = d(s_i, s_k)/2$

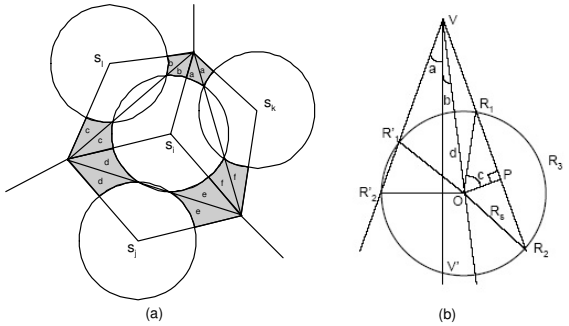


Figure 5. (a) Total coverage hole within s_i's polygon is $(a+b+c+d+e+f)$ (b) Area $R_1 R_2 R'_2 R'_1$ of the circle O lying within an angle $2a$

two angles x'_{ik_1} and x'_{ik_2} are given by the following: $x'_{ik_1} = \cos^{-1} \frac{d(s_i,s_k)}{2R_s}$ and $x'_{ik_2} = \cos^{-1} \frac{d(s_i,s_k)}{2l_1}$. The $\mp$ combination is for the case of acute triangle and the $\pm$ combination is for the case of obtuse triangle.

It should be noted that not all the nodes need to compute its coverage hole using the afore-mentioned methods. A node can also compute its coverage hole using information from its neighbors, thus avoiding redundant computation. For instance, in Fig. 5(a) node s_i has three neighbors s_j, s_k and s_l. By construction of the Voronoi triangles we observe that the total coverage hole within s_i's polygon is equal to the sum of the holes existing within its neighbors' polygons. In the diagram, we illustrate this fact by using the same notations for equal areas. Thus, each sensor node can calculate the exact amount of coverage hole within its Voronoi polygon by summing up the holes in the triangles, and hence in turn, the total amount of coverage hole in the whole sensing field can be calculated by the following sum: $A_{C(S)} = \sum_{i=1}^{N_s} \sum_{k \in N(s_i),\, k \neq i} (A_{\Delta_{ik}})$.

5. Coverage Enhancing Algorithm

In this section, we describe a collaborative algorithm to find out optimal positions and the number of additional mobile nodes needed to enhance coverage in the sensing field. First, we state and prove the following theorem.

Theorem 5.1 *Given an angle and a circle of certain radius at a distance d from the vertex of the angle, such that the circumference of the circle intersects the rays of the angle at four points, to cover maximum area within the angle the circle must lie with its centre on the angle bisector.*

Proof 1 *In Fig. 5, a circle of radius R_s lies at a distance d from the vertex V of an angle $\angle 2a$ and it intersects the rays at points R_1, R_2, R_1' and R_2'. The center of the circle O makes an angle $\angle b$ with the angle bisector. Our claim is that the area of the sector $R_1R_2R_2'R_1'$ is maximum when the center of the circle lies on the angle bisector VV'.*

By construction $OP \perp R_1R_2$. Length of OP is $d\sin(a-b)$ and length of PR_1 is $\left\{R_s^2 - d^2\sin^2(a-b)\right\}^{1/2}$. If we denote the angle between OP and OR_1 by c, then $\angle c = \cos^{-1}\left\{\frac{d}{R_s}\sin(a-b)\right\}$. Area of the crescent shaped sector $R_1R_2R_3$ that lies outside the angle is given by the area of the sector $OR_1R_3R_2$ minus the area of the triangle $\triangle OR_1R_2$. In a similar way, we can find out the area of the crescent shaped sector that lies outside the angle on the left side. Thus, the total area of the circle lying outside the angle:

$$A_c = \frac{R^2}{2}\left[\cos^{-1}\frac{d}{R_s}\sin(a+b) + \cos^{-1}\frac{d}{R_s}\sin(a-b)\right] - \Big[d\sin(a+b)\left\{R_s^2 - d^2\sin^2(a+b)\right\}^{1/2} + d\sin(a-b)\left\{R_s^2 - d^2\sin^2(a-b)\right\}^{1/2}\Big].$$

Differentiating the above equation with respect to b it can be shown that the minimum value of A_c occurs at $b = 0$. In other words, maximum area of the circle will lie within the angle when $b = 0$, i.e, when the center of the circle lies on the angle bisector. This maximum area is given by: $\pi R_s^2 - R_s^2\cos^{-1}\left\{\frac{d}{R_s}\sin a\right\} - 2d\left\{R_s^2 - d^2\sin^2 a\right\}^{1/2}\sin a$.

5.1. The COVerage ENhancing Algoritim

In this section, we give an overview of the collaborative Coven[1] algorithm and in Fig. 6, we detail the algorithm in steps. A static node inside its Voronoi polygon calculates the optimal positions and the number of additional mobile nodes that would be required for maximizing coverage in the following fashion. We claim that since a static node knows about the exact amount of coverage hole within its polygon, it can also thereby estimate the coverage hole sorrounding its Voronoi vertices.

Let us consider three static nodes A, B, C and their Voronoi polygons as shown in Fig. 7. Suppose node A wants to determine the optimal positions and the number of additional nodes required to enhance coverage inside and sorrounding its Voronoi polygon. Node A knows the coverage holes within each of the triangles that constitute its polygon (as described in section 4.1). Let the area of the holes within the triangles ΔAVP and ΔAVR be $A_{C(\Delta_{AVP})}$ and $A_{C(\Delta_{AVR})}$, respectively. By property of Voronoi diagrams, node A can infer that the same amount of holes will also exist within it's neighbors' polygons. In particular, the hole inside its neighboring node B will be equal to $A_{C(\Delta_{AVP})}$ and that inside the neighboring node C will be equal to $A_{C(\Delta_{AVR})}$. Thus, from local information node A knows that the coverage hole $A_{C(V)}$, sorrounding its Voronoi vertex V is atleast equal to the following: $A_{C(V)} \geq A_{C(\Delta_{AVP})} + A_{C(\Delta_{AVR})} + A_{C(\Delta_{BVP})} + A_{C(\Delta_{CVR})} = 2 * \left(A_{C(\Delta_{AVP})} + A_{C(\Delta_{AVR})}\right)$.

Following this method, node A estimates the coverage hole sorrounding one of its Voronoi vertices and then decides to place an additional mobile sensor nearby that vertex only when the following condition holds: $A_{C(V_i)} \geq \mu\pi R_s^2$, where the value of the heuristic parameter μ is close to one. Next, the static node calculates the optimal position for additional nodes. According to Theorem 5.1, this optimal position should lie on the angle bisector of the corresponding vertex. We claim and prove by simulation results that the exact location should lie at a point p_i, such that the overlap with node A is minimum, as well as the point should lie within the Voronoi polygon of A. In other words, the following conditions should hold.

- p_i lies on the line that bisects the angle formed by V and the adjacent edges of the Voronoi polygon.
- p_i lies within the Voronoi polygon.
- $d(A, p_i) = \min\{2R_s, d(A, V)\}$.

Once a node decides to place an additional sensor nearby one of the vertices, it updates the coverage hole assuming that a hypothetical sensor has been placed in its optimal location. We name such a vertex as a *coverage-enhancing vertex*. The static node then goes on calculating and updating coverage holes nearby other vertices, with the constraint that it does not consider two consecutive vertices if the length of the corresponding Voronoi edge is less than R_s and one of the vertices is a *coverage-enhancing vertex*. The constraint is to eliminate overlap between the mobile nodes, because placing two nodes nearby two vertices whose distance is less than R_s eventually means each node covering both the vertices, and hence not maximizing coverage. A situation of conflict might arise when node A chooses to place mobile nodes nearby particular vertices and one or more of it's neighboring nodes also choose to place additional nodes nearby the same vertices. To avoid this conflicting scenario, our algorithm incorporates a *conflict-resolution scheme*, which is as follows. Once node A finds out the optimal positions

1 Coven - A collection of individuals with similar interests or activities. Here, we use the term symbolically as how the group of static sensors collaborate to estimate and enhance coverage

Notations:
s_i, m_j, R_s, $N(s_i)$, V_i are as defined in Table I. N_s: total number of static nodes deployed randomly
L_s: $\{(id_{s_i}, loc_{s_i}) \mid s_i \in N_s\}$, the list of ids and locations of the static nodes
L_m: $\{(id_{s_i}, loc_{s_i})\}$, the list of ids and locations of the additional mobile nodes
$V_{coven(s_i)}$: the set of coverage-enhancing vertices for s_i, $V_{contention}$: the set of contention vertices
N_{m_i}: number of additional mobile nodes estimated by s_i
$A_{C(\Delta'_{ik})}$: hole lying inside part of Δ_{ik}. In Fig. 7, ΔAVP and ΔBVP are two such triangles and the holes within them are referred as $A_{C(\Delta'_{AB})}$ and $A_{C(\Delta'_{BA})}$ respectively. Note that, by construction $A_{C(\Delta'_{ik})} = A_{C(\Delta'_{ki})}$.
Initialization phase:
Choose a random static sensor as the *anchor* node, which starts the Coven algorithm by being the first member to calculate the hole within its Voronoi polygon and to estimate the number of additional mobile nodes required.
$V_{contention} \leftarrow \Phi$, Initialise the set of contention vertices as the NULL set.
At each static node s_i: (Coverage hole discovery and estimation phase)
$N_{m_i} \leftarrow 0$, $V_{coven(s_i)} \leftarrow \Phi$
Construct the Voronoi polygon $V(s_i)$ from the list L_s.
Calculate the amount of coverage hole $A_{C(\Delta_{ik})}$ existing inside each of the Voronoi triangles Δ_{ik}, $(k = 1, 2, ..., |N(s_i)|)$, that constitute its Voronoi polygon $V(s_i)$, by following the procedure described in section 4.
for $i = 0$ to $|N(s_i)|$ **do**
  if $V_i \notin V_{contention}$ AND $(V_{i+1 mod |N(s_i)|} \notin V_{coven(s_i)}$ OR $l_{ik} \geq R_s)$ **then**
    $A_{C(V_i)} \leftarrow 2 * (A_{C(\Delta'_{ik})} + A_{C(\Delta'_{ik+1})})$, $A_{C(\Delta'_{ik})}$ and $A_{C(\Delta'_{ik+1})}$ are calculated from $A_{C(\Delta_{ik})}$ and $A_{C(\Delta_{ik})}$.
    if $A_{C(V_i)} \geq \mu\pi R_s^2$ **then**
      Decide to place an additional mobile node m_i nearby vertex V_i.
      Calculate the optimal position for m_i at point p_i such that:
        1. p_i lies on the line that bisects the angle formed by V_i and the adjacent edges of $V(s_i)$
        2. p_i lies within $V(s_i)$, 3. $d(s_i, p_i) = \min\{2R_s, d(s_i, V_i)\}$
      $V_{coven(s_i)} \leftarrow V_{coven(s_i)} \cup V_i$, put the vertex V_i in set $V_{coven(s_i)}$
      Update $A_{C(\Delta_{ik})}$ and $A_{C(\Delta_{ik})}$ assuming a hypothetical sensor m_i is placed at p_i.
      if $d(V_i, p_i) \leq \epsilon R_s$ **then**
        $V_{contention} \leftarrow V_{contention} \cup V_i$, mark vertex V_i as a contention vertex.
      end if
      $N_{m_i} \leftarrow N_{m_i} + 1$
    end if
  end if
end for
if $V_{contention} \neq \Phi$ **then**
  Broadcast a message to $N(s_i)$ with the content of $V_{contention}$.
end if
A total $\sum_{i=1}^{N(s)}$ additional mobile nodes are randomly deployed.
At each static node s_i: (Mobile node placement phase)
Broadcast a message which contains all the $A_{C(V_i)}$'s and p_i's, in an attempt to discover N_{m_i} number of mobile sensors in its neighborhood, and a request to move the mobile nodes to locations p_i's.
At each mobile node m_i: (One-time movement phase)
Construct Voronoi polygon considering both the list L_s and L_m.
Calculate the area it is presently covering, call this A_{m_i}.
If not marked as *moved-once*, upon receiving a broadcast message from a static sensor, it finds out to which of the p_i's, it is closest.
if $A_{m_i} < A_{C(V_i)}$ **then**
  Move to point p_i.
  Mark itself as *moved-once*.
end if

Figure 6. Detailed description of the collaborative algorithm Coven

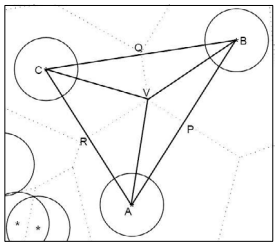


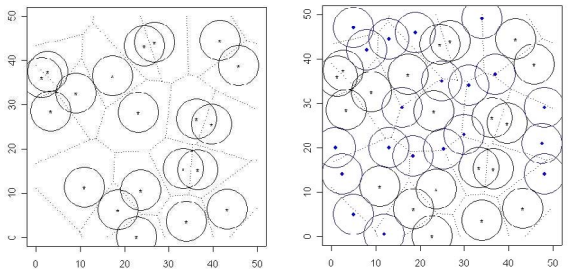
Figure 7. Optimal locations of additional nodes

for the additional nodes, its broadcasts a message to all its Voronoi neighbors when the optimal points lie within a distance ϵR_s from the vertices of contention, where ϵ depends on coverage requirements of the specific application and is close to zero. The neighboring nodes receiving this broadcast message, do not decide to place additional nodes nearby the same vertices.

It should be noted that some static nodes might decide not to place additional nodes nearby any of its Voronoi vertices because those vertices are already tagged as *coverage-enhancing vertices* by its neighboring nodes. Once the estimated number of mobile nodes is randomly deployed, they construct Voronoi polygons considering both the static and mobile nodes and each of them calculates the area that they are presently covering. The static nodes then broadcast a message to discover the estimated number of mobile nodes in their neighborhood and request them to move to the optimal positions. Upon receiving a broadcast message, a mobile node finds out to which of the optimal locations it is closest and then decides to move to the designated location only if the hole it can cover is bigger than its present area of coverage. It also marks itself as *moved-once*, so that on receiving successive broadcast messages it does not move again.

6. Simulation And Results

We used Matlab and the GNU 'R' package for simulating the Coven algorithm. In order to evaluate the percentage coverage by Coven, we compared it with random deployment scheme. We considered a simulation sensing area of $50m \times 50m$ and a uniform sensing radius of $5m$ for all the sensors. We also chose $\mu = 0.8$ and $\epsilon = 0$. In contrast to random deployment which achieves 99% coverage with 80 sensors deployed, Coven can achieve the same amount of coverage using only a combination of 25 static and an estimated number of 22 mobile sensors. The number of mobile nodes additionally deployed is bounded by the number of Voronoi vertices formed by the Voronoi polygons of the static nodes. In Fig. 8 and Fig. 9, we summarize the simulation results by plotting the percentage coverage achieved by deploying different number of static and mobile sensors. Note that, as the number of mobile nodes increases, the coverage increases sharply because of the fact that the sensing field becomes more *flexible* from the point of view of movement of sensors. Also, from Fig. 9 it can be seen that in most of the cases the estimated number of additional mobile nodes is close to 50%, which implies that on average each static node decides to place one additional node within its polygon. Experimenting with different sensing ranges and determining the performance of Coven is part of our future work.



(a) Initial random deployment of 20 static sensors

(b) After an estimated 20 additional mobile nodes are placed according to Coven

Figure 8. Simulation results - I

7. Analysis

COVEN follows a two-step deployment process and provides a strategy for the static sensors to collaborate and estimate the additional number of mobile nodes needed to be placed in optimal locations to maximize coverage in a mixed sensor network. The following two interesting cases might arise: 1) a node is very close to the boundary of the sensing field and 2) a node does not have any neighbor. In the first case, since we assume that a node knows its own location, it can also find out the location of the boundary, and considering the line representing the boundary as an edge of its Voronoi polygon, it can calculate the coverage within its polygon. Thus, in other words, COVEN constructs a bounded Voronoi diagram in the first phase. The algorithm does not scale very well when the network is too sparse, which can give rise to isolated nodes. In such a situation, we argue that after the mobile nodes are deployed, some of them volunteer to "roam" around and find out the isolated nodes and then requests the application to deploy additional nodes at those locations.

The algorithms described in [8] and [9] resemble to

our work, but there, the number of mobile nodes is fixed and they are placed one per polygon, towards the farthest Voronoi vertex. On the contrary, in Coven, the number of mobile nodes to be placed within each Voronoi polygon is decided by the static sensors dynamically. Also, since the locations p_i of the additional nodes are governed by the three criteria mentioned in section 5.1, it guarantees maximum coverage. Therefore, this heuristic provides a better coverage as the application can deploy additional mobile nodes depending on the topology formed by the static nodes.

8. Conclusions and Future Work

In this paper, we considered a sensor network comprising both static and mobile nodes and provided a Voronoi diagram based approach to estimate the exact amount of coverage holes in a sensing field. We also proposed a collaborative algorithm (Coven) to estimate the number of additional mobile nodes to be deployed and relocated to the location of the holes to enhance overall coverage. Coven follows a two-step deployment process, where initially randomly deployed static nodes find out holes in the sensing field and then estimate the number of additional nodes required and their optimal positions to maximize coverage. Simulation results show that the collaborative algorithm can achieve significant performance improvement over random deployment. Instead of limiting the number of mobile nodes by a cost function, we rather let the static nodes decide depending on their locations and estimate the number of additional nodes needed to be deployed. Note that the additional number of mobile nodes is bounded by the number of Voronoi vertices.

As part of our future work, we would like to estimate the exact amount of coverage holes in case of nodes with different sensing ranges. We will also extend our collaborative algorithm to incorporate obstacle modelling and study cost analysis from an optimization point of view. We believe collaboration in sensor network is essential for any distributed algorithm to perform efficiently and applying it in a mixed sensor network would provide many significant challenges and open up new research areas.

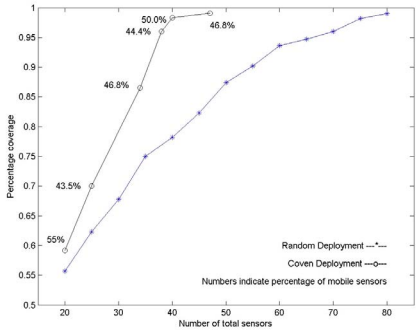


Figure 9. Simulation Results - II. Plot of percentage coverage against the number of sensors.

References

[1] F. Aurenhammer. Voronoi diagrams a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

[2] S. S. Dhilon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *Proc. of International Conference on Information Fusion (FUSION'02)*, pages 1581–1587, Annapolis Maryland, USA, July 2002.

[3] A. Howard, M. Mataric, and G. Sukhatme. Mobile sensor network deployment using potential fields: A distributed scalable solution to the area coverage problem. In *Proc. of the 6th International Conference on Distributed Autonomous Robotic Systems (DARS'02)*, pages '299–308, Fukuoka, Japan, June 2002.

[4] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proc. of IEEE Infocom (INFOCOM'01)*, pages 1380–1387, Anchorage, Alaska, USA, Apr. 2001.

[5] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Proc. of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom'01)*, pages 139–150, Rome, Italy, July 2001.

[6] S. Poduri and G. S. Sukhatme. Constrained coverage in mobile sensor networks. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA'04)*, pages 40–50, New Orleans, LA, USA, Apr./May 2004.

[7] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak. Minimal and maximal exposure path algorithms for wireless embedded sensor networks. In *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03)*, pages 40–50, Los Angeles, California, USA, Nov. 2003.

[8] G. W, G. Cao, and T. LaPorta. A bidding protocol for deploying mobile sensors. In *Proc. of the 11th IEEE International Conference on Network Protocols (ICNP'03)*, pages 315–324, Atlanta, Georgia, USA, 2003.

[9] G. Wang, G. Cao, and T. LaPorta. Movement-assisted sensor deployment. In *Proc. of IEEE Infocom (INFOCOM'04)*, pages 80–91, Hong Kong, China, Mar. 2004.

[10] Y. Zou and K. Chakrabarty. Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems*, 3(1):61–91, Feb. 2004.