Worst and best-case coverage in sensor networks

Article *in* IEEE Transactions on Mobile Computing · February 2005

DOI: 10.1109/TMC.2005.15(410) · Source: IEEE Xplore

Worst and Best-Case Coverage in Sensor Networks

Seapahn Megerian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava, *Senior Member*, *IEEE*

Abstract—Wireless ad hoc sensor networks have recently emerged as a premier research topic. They have great long-term economic potential, ability to transform our lives, and pose many new system-building challenges. Sensor networks also pose a number of new conceptual and optimization problems. Here, we address one of the fundamental problems, namely, coverage. Sensor coverage, in general, answers the questions about the quality of service (surveillance) that can be provided by a particular sensor network. We briefly discuss the definition of the coverage problem from several points of view and formally define the worst and best-case coverage in a sensor network. By combining computational geometry and graph theoretic techniques, specifically the Voronoi diagram and graph search algorithms, we establish the main highlight of the paper—an optimal polynomial time worst and average case algorithm for coverage calculation for homogeneous isotropic sensors. We also present several experimental results and analyze potential applications, such as using best and worst-case coverage information as heuristics to deploy sensors to improve coverage.

Index Terms—Sensor networks, coverage, maximal breach, maximal support, best-case coverage, worst-case coverage.

1 Introduction

As our personal computing era evolves into a ubiquitous computing one, there is a need for a world of fully connected devices with inexpensive wireless networks. Improvements in wireless network technology interfacing with emerging microsensors based on MEMs technology [2] is allowing sophisticated, yet inexpensive, sensing, storage, processing, and communication capabilities to be unobtrusively embedded into our everyday physical world. Moreover, embedded Web servers [1], [3] can be used to connect the physical world of sensors and actuators to the virtual world of information, utilities, and services. Consequently, a flurry of research activity has commenced in the sensor networks domain, especially in wireless ad hoc sensor networks. Although many of the sensor technologies are not new, certain physical and technological barriers of performing wireless communications have limited the feasibility of such devices in the past. Some of the benefits of the newer, more capable sensor nodes are their abilities to form large-scale networks, implement sophisticated protocols, reduce the amount of communication (wireless) required to perform tasks by distributed and local computations, and implement complex power saving modes of operation depending on the environment, the application, and the state of the network.

Due to the above-mentioned advances in sensor network technology, more and more practical applications of wireless sensor networks continue to emerge. As an example, consider the millions of acres that are lost around the world, due to forest fires every year. In all fires, early warnings are critical in preventing small harmless brush fires from becoming monstrous infernos. By deploying specialized wireless sensor nodes in strategically selected high-risk areas, the detection time for such disasters can be drastically reduced, increasing the likelihood of success in early extinguishing efforts. Also, since the nodes are self-configuring and do not need constant monitoring, the cost of such a deployment may be minimal compared to the huge losses incurred in large blazes.

In addition to the new applications, wireless sensor networks provide a viable alternative to several existing technologies. For example, large buildings contain hundreds of environmental sensors that are wired to central air conditioning and ventilation systems. The significant wiring costs limit the complexity of current environmental controls and the reconfigurability of these systems. However, replacing the hard-wired monitoring units with ad hoc wireless sensor nodes can improve the quality and energy efficiency of the environmental system while allowing almost unlimited reconfiguration and customization in the future. In many instances, the savings in the wiring costs alone justify the use of the wireless sensor nodes.

One of the fundamental issues that arise naturally in sensor networks is coverage. Due to the large variety of sensors and their applications, sensor coverage is subject to a wide range of interpretations. In general, coverage can be considered as a measure of the quality of service of a sensor network. For instance, in the previous fire detection sensor network example, one may ask how well the network can observe a given area and what the chances are that a fire starting in a specific location will be detected in a given

- *S. Megerian is with the Electrical and Computer Engineering Department, University of Wisconsin at Madison, 3424 Engineering Hall, 1415 Engineering Drive, Madison, WI 53706. E-mail: megerian@ece.wisc.edu.*
- *F. Koushanfar is with the EECS Department, University of California at Berkeley, 545T Cory hall (DOP Center), Berkeley, CA 94720. E-mail: farinaz@eecs.berkeley.edu.*
- *M. Potkonjak is with the Computer Science Department, University of California at Los Angeles, 3532G Boelter Hall, Los Angeles, CA 90095-1596. E-mail: miodrag@cs.ucla.edu.*
- *M.B. Srivastava is with the Electrical Engineering Department, University of California at Los Angeles, MC #951594, 6731-H Boelter Hall, Los Angeles, CA 90095-1594. E-mail: mbs@ee.ucla.edu.*

Manuscript received 8 Aug. 2002; revised 7 Dec. 2003; accepted 7 Jan. 2004; published online 1 Dec. 2004.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 2-082002.

 Published by the IEEE CS, CASS, ComSoc, IES, & SPS

time frame. Furthermore, coverage formulations can try to find weak points in a sensor field and suggest future deployment or reconfiguration schemes for improving the overall quality of service.

Here, we focus our attention to the isotropic class of sensors, deployed in a field to detect certain activities. An example of such a scenario may be seismic or acoustic sensors deployed in a battle field to detect enemy movements. Throughout our discussions, we use the term agent to denote the phenomenon being detected by the sensors (for example, an enemy tank moving in the field). In order to approach the coverage problem here, we formulate the worst and best-case coverage and present algorithms for their calculations. In the worst-case coverage problem, we want to find the closest distance to sensors that an agent traveling on any path in the sensor field must encounter at least once. The main idea here is that the closest distance to sensors is one metric by which sensor coverage of the field can be characterized. The scheme is "worst-case" since we determine the closest distance to sensors even if the agent tries to optimally avoid the sensors.

In the analogous best-case coverage problem, we want to find the farthest distance to sensors that an agent traveling on any path in the sensor field must have from sensors, even if it tries to stay as close to sensors as possible. Clearly, at some points, the agent must move away from sensors in order to be able to traverse the field. Although the two problems seem similar (they are duals in some sense), they solve two problems which have very different physical interpretations.

From the conceptual and algorithmic point of view, the main contribution is provably optimal polynomial time algorithm for best and worst-case coverage calculation in a sensor network. As we discuss in Section 5, we combine existing computational geometry techniques and constructs such as the Voronoi diagram, with graph theoretical techniques. The use of Voronoi diagram, efficiently and without loss of optimality, transforms the continuous geometric coverage problem into a discrete graph problem. Furthermore, it enables direct application of search techniques in the resulting graph representation.

1.1 Organization

The remainder of this article is organized as follows: In the next section, we summarize the related work. In Section 3, we survey several key technologies that are fundamental to our study of coverage. Section 4 contains a brief overview of deterministic sensor deployment and coverage. In Section 5, we present formal definitions of the worst (breach) and best-case (support) coverage and propose optimal polynomial-time algorithms for solving each case. Section 6 presents some empirical results followed by a brief discussion of future research directions and the conclusion.

2 Related Work

The increasing trend in research efforts in the areas referred to as smart spaces or pervasive computing are directly related to many problems in sensor networks. Although many researchers in the sensor network area have mentioned the critical notion of coverage in the network, to our knowledge this is the first time that an algorithmic approach combined with computational geometry constructs was adopted in the context of ad hoc sensor networks. Kang and Golay [11] describes a general systematic method for developing an advanced sensor network for monitoring complex systems such as those found in nuclear power plants, but does not present any general coverage algorithms. The Art Gallery Problem [12] deals with determining the number of observers necessary to cover an art gallery room such that every point is seen by at least one observer. It has found several applications in many domains such as for optimal antenna placement problems for wireless communication. The Art Gallery problem was solved optimally in 2D and was shown to be NP-hard in the 3D case. Marengoni et al. [12] proposes heuristics for solving the 3D case using Delaunay triangulations. Sensor coverage for detecting global ocean color where sensors observe the distribution and abundance of oceanic phytoplankton is approached in [7] by assembling and merging data from satellites at different orbits.

Perhaps the most related works are the attempts on coverage of an initially unknown environment for mobile robots [4], [6]. However, when the geometry of the environment is known in advance, coverage becomes a special case of path planning [10]. Both of these problems are solved using generalized Voronoi diagrams.

Radar and sonar coverage also present several related challenges. The radar and sonar netting optimization is of great importance in networking technologies and the optimal distribution of detection and tracking in a surveillance area [15]. Based on the measured radar cross sections and the coverage diagrams for different radars, [16] proposes a method for optimally locating the radars to achieve a satisfactory surveillance area with limited radar resources.

Coverage studies to maintain connectivity in wireless networks have also been the focus of study. For example, [13] and [14] calculate the optimal number of base stations required to achieve a system operator's service objectives. When base stations are present, connectivity is achieved through mobile client attachments to a base station. However, the connectivity coverage is more complex in the case of ad hoc wireless networks since the connections are peer-to-peer. Haas [9] shows the improvement in network coverage due to multihop routing features of ad hoc networks and optimizes the coverage constraint subject to a limited path length.

In the best and worst-case sensor coverage formulations we present here, the distance of an agent to the closest sensor is of importance while in exposure-based methods presented in [19], the detection probability (observability) in the sensor field is represented by a path dependent integral of multiple sensor intensities. It is interesting to note that in both of these schemes, the types of actions that an agent performs impact the coverage metric. For example, the sensor field may have a different coverage level if an agent is traveling west to east as opposed to north to south, or along any other arbitrary paths.

3 Preliminaries

3.1 Topology of the Network and Sensor Model

Generally, wireless sensor networks are targeted to the extremes of miniaturization, availability, accuracy, reliability, and power savings. This requires a networked infrastructure with small physical nodes, low power consumption, and low cost that, in turn, limits communications to the immediate proximity of each node. There are several existing models of sensor behavior each with varying degrees of complexity. However, most models share one aspect in common in that, generally, sensing ability is directly dependant on distance. In all of our subsequent discussions, we assume that sensor sensitivity to a phenomenon decreases as the distance separating the two increases. Furthermore, we restrict our coverage formulation to the relatively broad isotropic class of sensors. As mentioned in Section 2, exposure, a more generic scheme for characterizing and computing coverage, can be used to address more general sensing models [19].

3.2 Location Discovery Techniques and Algorithms

Geographical information is an integral attribute of any physical measurement. Thus, knowledge of node locations is often crucial in proper operation of sensor networks. The ad hoc nature of such networks necessitates that each node determines its location through a location discovery process. The Global Positioning System (GPS) is one method that was designed and is controlled by the United States Department of Defense for this purpose. The GPS system consists of at least 24 satellites in orbit around the earth, with at least four satellites viewable from any point, at a given time, on Earth. They each broadcast time-stamped messages at periodic intervals. Any device that can hear the messages from four or more satellites can estimate its distance from each satellite and, thus, perform trilateration to compute its position.

Although GPS is an elegant solution to the location discovery process, it has several limitations that hinder its use in wireless sensor network applications. First, GPS is costly both in terms of hardware and power requirements. Second, GPS requires line-of-sight between the receiver and the satellites and, thus, does not work well when obstructions such as buildings, trees, and mountains block the direct "view" to the satellites. Consequently, other techniques have been proposed to dynamically compute the locations of the nodes. In several location discovery schemes, the received signal strength indicator (RSSI) of RF communication is used as a measure of distance between nodes. In other schemes, the time difference in arrival of RF and acoustic (ultra-sound) signals are used to approximate node distances. Once nodes have the ability to estimate distances between each other (ranging), they can then compute their locations by trilateration. In order for a trilateration to be successful, a node must have at least three neighbors who already know their locations. This requires that at least a subset of nodes determine their locations through other means such as by using GPS, manual programming, or deterministic deployment (placing nodes at specified coordinates). Savvides et al. [17] provide a detailed discussion on such location discovery techniques and algorithms. In our subsequent discussions, we assume that node locations are known.

3.3 Computational Geometry: Voronoi Diagram and Delaunay Triangulation

The Voronoi diagram has been reinvented, used, and studied in many domains. According to [5], it is believed that the Voronoi diagram is a fundamental construct defined by a discrete set of points. In 2D, the Voronoi diagram of a set of discrete sites (points) partitions the plane into a set of convex polygons such that all points inside a polygon are closest to only one site. This construction effectively produces polygons with edges that are equidistant from neighboring sites. Fig. 1 shows an example of a Voronoi diagram for a set of randomly placed sites. Aurenhammer [5] presents a detailed survey of Voronoi diagrams and their applications.

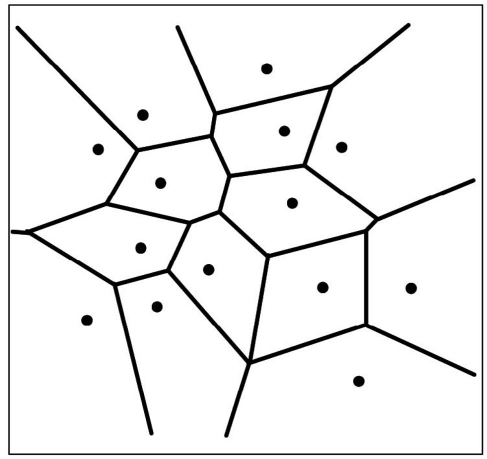

Fig. 1. Voronoi diagram example.

Another structure that is directly related to Voronoi diagrams is the Delaunay triangulation [8]. The Delaunay triangulation can be obtained by connecting the sites in the Voronoi diagram whose polygons share a common edge. It has been shown that among all possible triangulations, the Delaunay triangulation maximizes the smallest angle in each triangle. In addition, a Delaunay triangulation must satisfy the empty circle property, which states that there is a circle containing the end points of a Delaunay edge and no other points (edges). Also, neighborhood information can be extracted from the Delaunay triangulation since sites that are close together are connected. In fact, the Delaunay triangulation can be used to find the two closest sites by considering the shortest edge in the triangulation. We use the properties of the Voronoi diagram and Delaunay triangulation to solve for best and worst-case coverage. In Section 5, we will show how the Voronoi diagram and the Delaunay triangulation serve as the underlying structures to limit our search space for agent "paths" in a sensor field.

The lines at the boundaries of the Voronoi diagram extend to infinity. However, since here we are dealing with a finite area, we must clip the Voronoi diagram to the boundaries of the field. Since traveling along the bounds of the sensor field also constitutes a valid path, we introduce extra edges in the Voronoi diagram corresponding to the bounds. In subsequent discussions, when we refer to the Voronoi diagram, we are actually referring to the bounded diagram.

3.4 Implementation: Centralized versus Distributed

Multihop communication is one of the main enablers in reducing power consumption in ad hoc sensor networks. The energy required for communication between two arbitrary nodes A and B is strongly dependent on the distance d between the two nodes. More precisely, the energy can be modeled as $E = B \cdot d^y$, where $y > 1$ is the path loss exponent depending on the RF environment and B is a proportionality constant describing the overhead per bit. Given this super linear relationship between energy and distance, generally using several short intermediate hops to send a bit is more energy efficient than using one longer hop.

However, an incorrect conclusion would be to use an infinite number of hops over the smallest possible distances. In reality, this is impractical for two reasons:

1. The number of intermediate hops is limited by the number of nodes between A and B.
2. The energy is not limited to the energy radiated through the antenna; there is also energy dissipated in the radio for receiving a bit and readying a bit for retransmission.

For applications such as coverage calculations, the energy of computations per node is also a component of the energy metric. It is important to note that technology scaling will gradually reduce the processing costs, with the transmission cost remaining relatively constant. Using compression techniques, one can reduce the number of transmitted bits, thus reducing the cost of transmission at the expense of more computation. This communication-computation trade off is the core idea behind low energy sensor networks. From this discussion, it is apparent that a good algorithm designed for wireless sensor networks will require a minimal amount of communication. This is in sharp contrast with classical distributed systems [18] where the goal generally is maximizing the speed of execution. This renders the classical distributed algorithm irrelevant for developing wireless sensor networks algorithms.

The most relevant metrics in wireless networks is power. Experimental measurements indicate that the communication cost in wireless ad hoc networks can be two orders of magnitude higher than computation costs in terms of consumed power [22]. Note that the coverage problem presented in this paper is intrinsically global in the sense that lack of knowledge of location of any node may result in the problem not being solved correctly. Therefore, any algorithm which aims to provide the correct solution must inherently use all location data.

Throughout our discussions, we assume a centralized model of computation. Recently, Li et al. [20] proposed a localized approach for solving a variation of the best-case coverage (maximal support) in sensor networks. In addition, a variation of the localized exposure algorithm presented in [21] can be used to solve the worst-case coverage problem locally. However, a detailed treatment of this topic is beyond our scope here.

4 Deterministic Coverage

In order to achieve deterministic coverage, a static network must be deployed according to a predefined shape. The predefined locations of the sensors can be uniform in different areas of the sensor field or can be weighted to compensate for the more critically monitored areas. An example of a uniform deterministic coverage is the grid-based sensor deployment where nodes are located on the intersection points of a grid. In this case, the problem of coverage of the sensor field reduces to the problem of coverage of one cell and its neighborhood due to the symmetric and periodic deployment scheme.

Examples of weighted predefined deployment are the security sensor systems used in museums. The more valuable exhibit items are equipped with more sensors to maximize the coverage of the monitoring scheme. Another deterministic deployment scheme can be found in the 3D Art Gallery Problem heuristics solutions discussed in [12]. Our proposed coverage algorithm can be used in all predefined (deterministic) deployment schemes to determine the coverage in the sensor field.

5 Stochastic Coverage

In many situations, deterministic deployment is neither feasible nor practical. Another deployment option is to cover the sensor field with sensors randomly distributed in the environment. The stochastic random distribution model can be uniform, Gaussian, or any other distribution based on the application at hand. In the simulation studies for this paper, we have generally assumed uniform sensor distribution, although our algorithm is applicable to any other deployment scheme of the sensor nodes.

5.1 Worst-Case Coverage and Maximal Breach Path

In order to introduce the worst-case coverage problem, we first formally define *breach* for a path in the sensor field.

Given: A field A instrumented with sensors S, where for each sensor $s_i \in S$, the location (x_i, y_i) is known; areas I and F corresponding to initial (I) and final (F) locations of an agent.

Definition: Breach. *Given a path P connecting areas I and F, breach is defined as the minimum Euclidean distance from P to any sensor in S.*

Thus, the worst-case, breach-based, coverage problem discussed above can formally be stated as:

Problem: Maximal Breach Path. *Identify a Maximal Breach Path P_B, in A, connecting the areas I and F.*

The regions I and F are arbitrary regions determined by the task at hand and may be located anywhere inside or outside the sensor field.

Theorem 1. *At least one Maximal Breach Path must lie on the line segments of the bounded Voronoi diagram formed by the locations of the sensors in S.*

Proof. Since by construction, the line segments of the Voronoi diagram maximize distance from the closest sites, a Maximal Breach Path $\mathrm{P_B}$, must lie on the line segments of the Voronoi diagram corresponding to the sensors in S. If any point p on the path P_B deviates from Voronoi line segments, by definition, it must be closer to at least one sensor in S. □

The following steps outline the algorithm for finding P_B:

1. Generate Voronoi diagram D for S.
2. Apply graph theoretic abstraction by transforming D to a weighted graph.
3. Find P_B using binary-search and breadth-first-search.

The first part of this algorithm, detailed in Algorithm 1, generates the Voronoi diagram corresponding to the sensors in S. The weighted, undirected graph G is constructed by creating a node for each vertex and an edge corresponding to each line segment in the Voronoi diagram. Each edge in graph G is assigned a weight equal to its minimum distance from the closest sensor in S. The algorithm then performs a binary search between the smallest and largest edge weights in G. In each step, breadth-first-search (BFS) is used to check the existence of a path from I to F using only edges with weights that are larger than the search criteria called *breach_weight*. If a path exists, *breach_weight* is increased to further restrict the edges considered in the next search iteration. If a path is not found, *breach_weight* is lowered to relax the constraint on the search. Upon completion, the algorithm has found a Maximal Breach Path, which is a path from I to F with its smallest weighted edge being as large as possible.

```
Generate Bounded Voronoi diagram for S with vertex set U and line segment set L.
Initialize weighted undirected graph G(V,E)
FOR each vertex u_i ∈ U
   Create duplicate vertex v_i in V
FOR each l_i(u_j,u_k) ∈ L
   Create edge e_i(v_j,v_k) in E
   Weight(e_i)=min distance from sensor s_i ∈ S for 1≤i≤|S|
min_weight = min edge weight in G
max_weight = max edge weight in G
range = (max_weight – min_weight) / 2
breach_weight = min_weight + range
WHILE (range > binary_search_tolerance)
   Initialize graph G'(V',E')
   FOR each v_i ∈ V
      Create vertex v_i' in G'
   FOR each e_i ∈ E
      IF Weight(e_i) ≥ breach_weight
         Insert edge e_i' in G'
   range = range / 2
   IF BFS(G',I,F) is Successful
      breach_weight = breach_weight + range
   ELSE
      breach_weight = breach_weight – range
   END IF
```

As an example, consider the random sensor network instance depicted in Fig. 2 with areas I and F shown. Fig. 3 shows the weighted, Voronoi diagram corresponding to this network. Note that each edge in the Voronoi diagram is labeled with its weight in graph G, namely, its closest distance to a sensor. Fig. 4 shows an intermediate step in the search algorithm for finding P_B, where the *breach_weight* threshold is set at 40. Consequently, all edges with weights

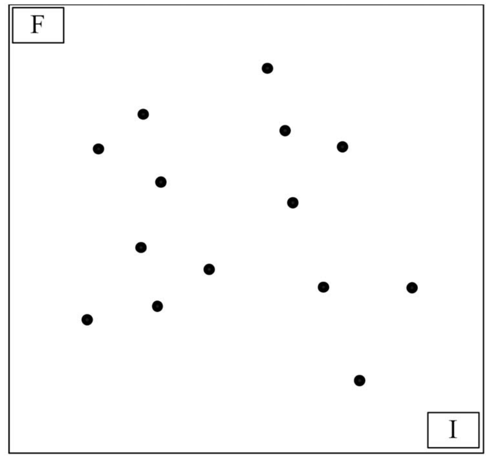

Fig. 2. A random sensor network instance.

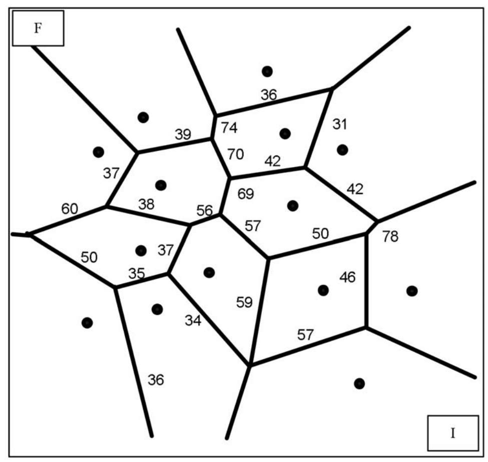

Fig. 3. Weighted Voronoi diagram of the sensor network in Fig. 2.

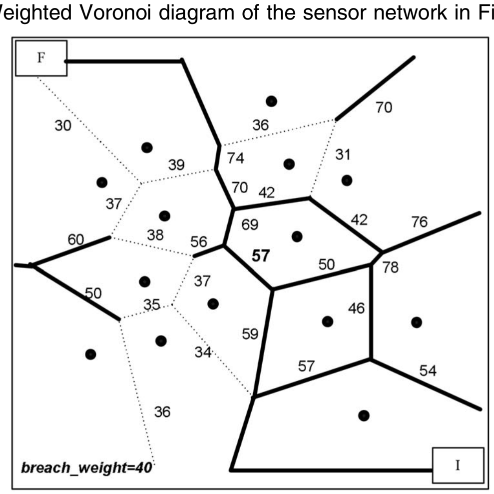

Fig. 4. Maximal Breach Path algorithm iteration: Edges with weights less than 40 are ignored.

less than 40 are ignored in the search and are thus shown as dotted lines. As can be seen, a path can be found from I to F in this case, indicating that the search threshold should be increased. Finally, Fig. 5 shows the result at the termination of the algorithm. The optimal *breach_weight* has been found to be 57. The critical edge is marked in bold and the Maximal Breach Path connecting I and F can be seen as a bold dotted line.

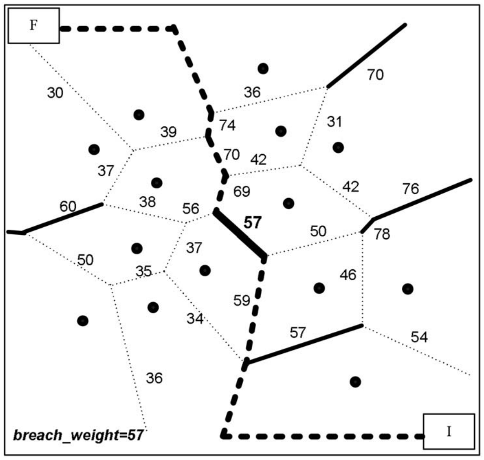

Fig. 5. Maximal Breach Path algorithm final result: optimal breach_weight=57.

We must note here that the Maximal Breach Path is not unique. In fact, in general, there are many paths that can qualify as a Maximal Breach Path. However, they all use edges with weights that are larger or equal to the *breach_weight* determined in the binary search phase of the algorithm, with at least one edge that has a weight equal to *breach_weight*.

The *breach_weight* found by this algorithm is the minimum distance from sensors that an agent traveling on *any* path through the field A, from I to F, must encounter at least once. If new sensors can be deployed or existing sensors moved such that this *breach_weight* is decreased, then the worst-case coverage is improved. Thus, *breach_weight* can be used as a measure of the coverage level provided by a sensor field.

5.2 Best-Case Coverage and Maximal Support Path

Similar to the worst-case (breach) coverage formulation, we first define *support* in order to formulate the best-case support-based coverage problem.

Given. A field A instrumented with sensors S where for each sensor $s_i \in S$, the location (x_i, y_i) is known; areas I and F corresponding to initial (I) and final (F) locations of an agent.

Definition: Support. *Given a path P connecting areas I and F,* support *is defined as the maximum Euclidean distance from the path P to the closest sensor in S.*

Problem. *Identify P_S, the Maximal Support Path in S, starting in I and ending in F.*

Theorem 2. *At least one Maximal Support Path must lie on the edges of the Delaunay triangulation (with the exceptions of the start and end points connecting P_S to I and F).*

We observe that a *Maximal Support Path* can consist of straight line segments connecting sensors. For the proof regarding the use of the Delaunay triangulation, we ignore the details pertaining to connecting the path to I and F. In our algorithm, we connect I and F to their closest sensors, respectively. Also, note that P_S is not unique. The only requirement is that the distance from the farthest point on P_S to the closest sensor is minimized. Hence, we only need to prove that the edge with the point farthest away from sensors in P_S, must lie on the Delaunay triangulation; we are free to use any other line segments in constructing the path, as long as they are closer to sensors than this critical edge.

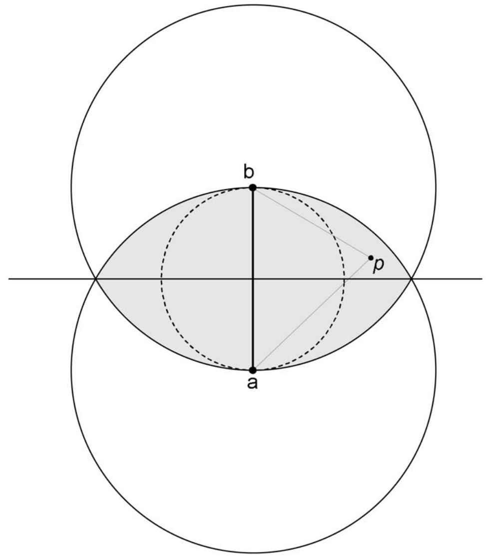

Fig. 6. Proof for Theorem 2.

Proof. Suppose that the edge farthest away from sensors in P_S is not in the Delaunay triangulation. Call the end points (sensors) of this edge a and b. as shown in Fig. 6. The figure also shows the circles with radius equal to the distance between a and b centered at a and at b and their corresponding Voronoi edge shown as a solid line.

There cannot be any sensors in the shaded region in Fig. 6. If there is a point in the shaded region (call it p), then edge ab can be replaced with the two shorter edges ap and pb, implying that P_S is not a Maximal Support Path (ab is not the critical edge with a point farthest from sensors).

If no sensor can exist in the shaded region, then there exists a circle containing both a and b and no other points. Then, a and b must be connected by a Delaunay edge (definition) which contradicts our supposition. □

The algorithm for finding P_S is very similar to the breach algorithm above, with the following exceptions:

1. The Voronoi diagram is replaced by the Delaunay triangulation as the underlying geometric structure.
2. Each edge in graph G is assigned a weight equal to the largest distance from the corresponding line segment in the Delaunay triangulation to the closest sensor.
3. The search parameter *breach_weight* is replaced by the new parameter *support_weight* and the search is conducted in such a way that *support_weight* is minimized.

In this case, the maximal support path may also not be unique. However, the *support_weight* found in the search phase of the algorithm is indicative of the best-case coverage of the network. Here, *support_weight* is the maximum distance from the closest sensors that an agent traveling on

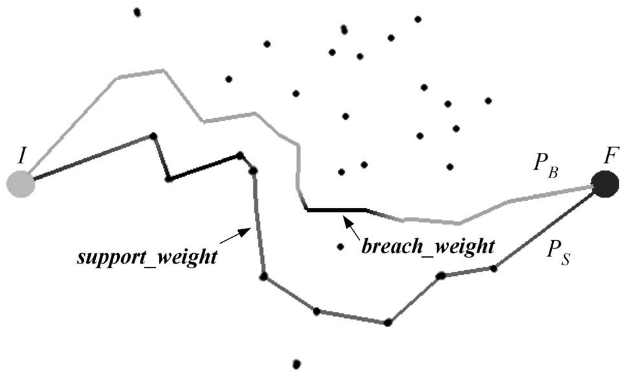

Fig. 7. Sensor field with Maximal Breach Path (P_B) and Maximal Support Path (P_S).

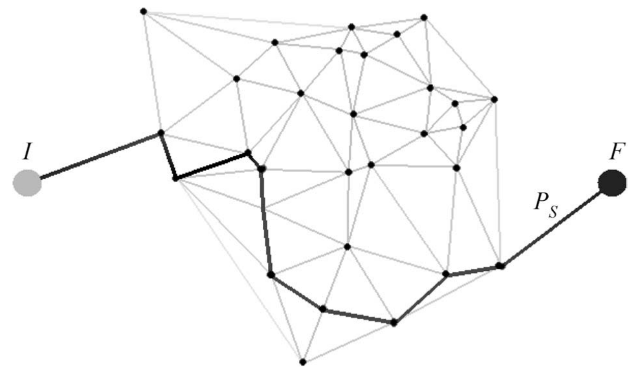

Fig. 9. Sensor field with Delaunay triangulation and a Maximal Support Path (P_S).

any path through the field A, from I to F, must encounter at least once. If additional sensors can be deployed or existing sensors moved such that *support_weight* is decreased, then the best-case coverage is improved.

5.3 Complexity

Given n sensors, the best known algorithms for the generation of the Voronoi diagram have $O(n \log n)$ complexities. In 2D, Voronoi diagrams are essentially linear complexity in terms of vertices and edges. So, for n points, $|V|$ and $|E|$ (vertices and edges) in the Voronoi graph are both $O(n)$. So, the resulting graph used later in the search phase of the algorithm is $O(n)$ in terms of the edges. Thus, the BFS and binary search phase has a complexity of $O(n \log range)$, where $range$ is the difference between highest and lowest weighted edge in the Voronoi graph. In practice, the complexity of the algorithm is dominated by the Voronoi diagram generation procedure which has a large constant factor in its complexity.

6 Experimental Results

6.1 Experimentation Platform

The coverage algorithms presented here have been implemented and used in several studies as stand-alone C packages. In this section, we present several results and try to provide an overview and analysis of the applications.

Fig. 7 shows an instance of the coverage problem where 30 sensors are deployed at random. The Maximal Breach Path (P_B) and the corresponding edge with *breach_weight* depicts where the breach takes place in the field. The Maximal Support Path (P_S) and the corresponding edge with *support_weight* are also shown.

Fig. 8 shows the underlying bounded Voronoi diagram for the same problem instance depicted in Fig. 7. Extra edges with 0 weight are used to connect the I and F regions to the structure so that all possible paths can be considered in the search algorithm. The 0-weight edges are drawn between all points where Voronoi edges intersect the boundary of the field and the corresponding point (I or F). Fig. 9 shows the corresponding Delaunay triangulation. In this case, only two extra edges are introduced to connect I and F to the closest sensors in the structure.

6.2 Sensor Deployment Heuristics

The edges corresponding to *breach_weight* described in Section 5 can be used as a guide for future sensor deployments. Since *breach_weight* corresponds to the edge in the breach path where P_B is closest to the sensors, deploying additional sensors along that edge can be used as a heuristic to improve overall coverage.

Fig. 10 shows the average improvement in breach coverage when up to four additional sensors are introduced successively in the network, according to the heuristic described above. Note that after each additional sensor deployment, the algorithm was repeated to find the new breach region. The results represent average improvements over 100 random

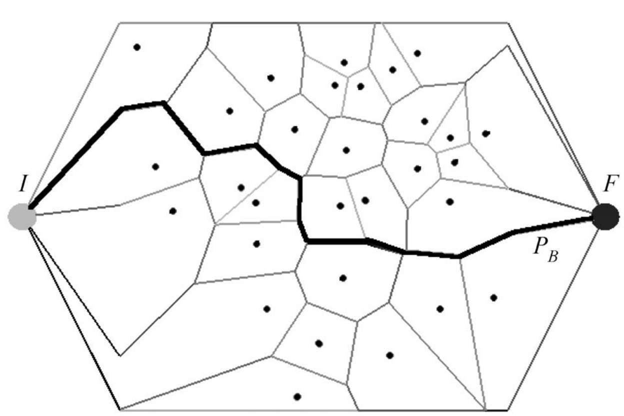

Fig. 8. Sensor field with Voronoi Diagram and a Maximal Breach Path.

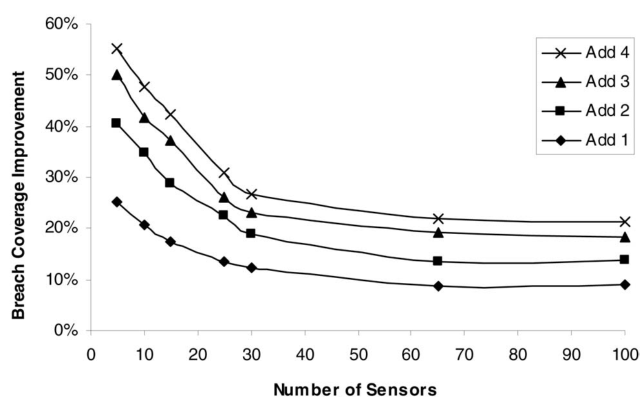

Fig. 10. Average worst-case coverage (breach) improvement by additional sensor deployments.

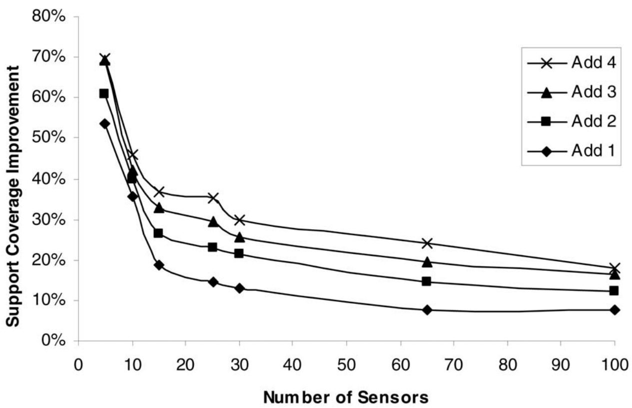

Fig. 11. Average best-case coverage (support) improvement by additional sensor deployments.

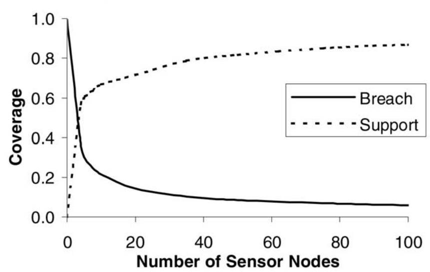

Fig. 12. Normalized worst-case (breach_weight) and best-case (1-support_weight) coverage as a function of the number of sensors.

deployments. It is interesting that, even after deploying 100 sensors, breach coverage can be improved by about 10 percent by deploying just one more sensor.

Similarly, the *support_weight* and the midpoint of the corresponding edge in the Delaunay triangulation can be used as a heuristic for deploying additional sensors to improve support coverage. As shown in Fig. 11, on average, a 50 percent improvement can be achieved in support coverage by adding one additional sensor when five nodes have already been randomly deployed. After deploying 100 random sensors, on average, a 10 percent support coverage improvement can be expected by using the heuristic to deploy one more sensor.

6.3 Stochastic Deployment—Asymptotic Behavior

The graph in Fig. 12 shows how the coverage of randomly placed (uniform) sensor nodes in a field varies as the number of sensors is changed. The results shown in the graph represent the average field breach and support coverages for 1,000 random sensor placements. For each placement, two uniform random variables X and Y are used to determine the coordinates (x_i, y_i) of each sensor s_i in a unit square field. Fig. 12 demonstrates the asymptotic nature of these metrics from the sensor field operator's point of view who wants to minimize breach and maximize support (minimize *support_weight*). Thus, for clarity, the figure shows a normalized plot of *breach_weight* (values closer to 0 preferred) and *1-support_weight* (values closer to 1 preferred) as a function of the number of sensors.

Given the unit square field and using the distance-based sensor model described earlier, on average, after deploying about 100 sensors, additional random sensors do not improve coverage very significantly. This asymptotic nature of breach and support coverage suggests that by analyzing a given field and selecting the proper number of sensor nodes, certain levels of coverage can be expected even if sensor deployment cannot be performed according to an exact plan.

7 Future Research Directions

Although our algorithm was developed for a wireless ad hoc sensor network, we have assumed a centralized computation model. A natural course of study would be to compare the centralized coverage algorithm to localized ones in terms of power consumption, cost, performance, and accuracy. Here, we have assumed identical sensor sensitivity models where the coverage depends only on the Euclidean distances from sensors. In practice, other factors influence coverage such as obstacles, environmental conditions, and noise. In addition to nonhomogeneous sensors, other possible sensor models can deal with nonisotropic sensor sensitivities, where sensors have different sensitivities in different directions. The integration of multiple types of sensors such as seismic, acoustic, optical, etc., in one network platform and the study of the overall coverage of the system also presents several interesting challenges.

In addition, the general problem of where to deploy additional sensors to improve the coverage remains open. Although we have introduced heuristics based on this coverage model that may perform well for single-sensor deployment, it is interesting to investigate methods of optimally deploying multiple sensors at a time.

8 Conclusion

We presented best and worst-case formulations for isotropic sensor coverage in wireless ad hoc sensor networks. An optimal polynomial time algorithm that uses graph theoretic and computational geometry constructs was proposed for solving for best and worst-case coverages. Experimental results show some applications of the theoretic coverage formulations and algorithms specifically for solving for Maximal Breach (worst-case coverage), Maximal Support (best-case coverage), additional sensor deployment heuristics to improve coverage, and stochastic field coverage.

Acknowledgment

The authors would like to thank John Agre at Rockwell Science Center for providing the initial insights regarding Voronoi diagrams and coverage in sensor networks. His comments were key enabling factors that lead to the development of the optimal breach and support path-based coverage algorithms. This material is based upon work partially supported by the US National Science Foundation under Grant No. NI-0085773 and the US Defense Advanced Research Projects Agency and Air Force Research Laboratory under Contract No. F30602-99-C-0128.

References

[1] D. Tennenhouse, "Proactive Computing," *Comm. ACM*, vol. 43, pp. 43-50, May 2000.

[2] G.J. Pottie and W.J. Kaiser, "Wireless Integrated Network Sensors," *Comm. ACM,* vol. 43, pp. 51-58, May 2000.
[3] G. Borriello and R. Want, "Embedded Computation Meets the World Wide Web," *Comm. ACM,* vol. 43, pp. 59-66, May 2000.
[4] G.S. Sukhatme and M.J. Mataric, "Embedding Robots into the Internet," *Comm. ACM,* vol. 43, pp. 67-73, May 2000.
[5] F. Aurenhammer, "Voronoi Diagrams—A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys 23,* pp. 345-405, 1991.
[6] Z.J. Butler, A.A. Rizzi, and R.L. Hollis, "Contact Sensor-Based Coverage of Rectilinear Environments," *Proc. IEEE Int'l Symp. Intelligent Control Intelligent Systems and Semiotics,* pp. 266-271, Sept. 1999.
[7] W.W. Gregg, W.E. Esaias, G.C. Feldman, R. Frouin, S.B. Hooker, C.R. McClain, and R.H. Woodward, "Coverage Opportunities for Global Ocean Color in a Multimission Era," *IEEE Trans. Geoscience and Remote Sensing,* vol. 36, pp. 1620-1627, Sept. 1998.
[8] K. Mulmuley, *Computational Geometry: An Introduction through Randomized Algorithms.* Prentice-Hall, 1994.
[9] Z.J. Haas, "On The Relaying Capability of the Reconfigurable Wireless Networks," *Proc. IEEE 47th Vehicular Technology Conf.,* vol. 2, pp. 1148-1152, May 1997.
[10] C. Hofner and G. Schmidt, "Path Planning and Guidance Techniques for an Autonomous Mobile Cleaning Robot," *Robotics and Autonomous Systems,* vol. 14, pp. 199-212, May 1998.
[11] C.W. Kang and M.W. Golay, "An Integrated Method for Comprehensive Sensor Network Developement in Complex Power Plant Systems," *Reliability Eng. & System Safety,* vol. 67, pp. 17-27, Jan. 2000.
[12] M. Marengoni, B.A. Draper, A. Hanson, and R.A. Sitaraman, "System to Place Observers on a Polyhedral Terrain in Polynomial Time," *Image and Vision Computing,* vol. 18, pp. 773-780, Dec. 1996.
[13] A. Molina, G.E. Athanasiadou, and A.R. Nix, "The Automatic Location of Base-Stations for Optimised Cellular Coverage: A New Combinatorial Approach," *Proc. IEEE 49th Vehicular Technology Conf.,* vol. 1, pp. 606-610, May 1999.
[14] K. Lieska, E. Laitinen, and J. Lahteenmaki, "Radio Coverage Optimization With Genetic Algorithms," *Proc. IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm.,* vol. 1, pp. 318-322, Sept. 1998.
[15] G.F. Van Blaricum, "Tactical Air Surveillance Radar Netting Simulator/Emulator," AD_A110382, 1982.
[16] Q. Weiyan, P. Yingning, L. Dajin, and H. Xiuying, "Approach to Radar Netting," *J. Tsinghua Univ. (Science and Technology),* vol. 37, pp. 45-48, Apr. 1997.
[17] A. Savvides, C.C. Han, and M.B. Srivastava, "Dynamic Fine-Grained Localization in Ad Hoc Networks of Sensors," *Proc. Seventh Ann. Int'l Conf. Mobile Computing and Networking (MobiCOM 2001),* July 2001.
[18] N. Lynch, *Distributed Algorithms.* Morgan Kaufman Publishers, 1996.
[19] S. Meguerdichian et. al., "Exposure in Wireless Ad Hoc Sensor Networks," *Proc. Seventh Ann. Int'l Conf. Mobile Computing and Networking (MobiCOM 2001),* pp. 139-150, July 2001.
[20] X.-Y. Li, P.-J. Wan, and O. Frieder, "Revisiting Coverage Problems in Wireless Ad Hoc Sensor Networks," *Proc. Int'l Conf. Comm.,* Apr. 2002.
[21] S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak, "Localized Algorithms in Wireless Ad Hoc Networks: Location Discovery and Sensor Exposure," *Proc. MobiHOC 2001*, pp. 106-116, Oct. 2001.
[22] V. Raghunathan, C. Schurgers, S. Park, and M.B. Srivastava, "Energy-Aware Wireless Microsensor Networks," *IEEE Signal Processing Magazine,* vol. 19, no. 2, pp. 40-50, Mar. 2002.

Seapahn Megerian received the BS degree in computer science and engineering in 1998 and the MS degree in computer science in 1999 from UCLA. He received the PhD degree in computer science from the University of California at Los Angeles in 2004. He is an assistant professor in the ECE Department at the University of Wisconsin at Madison. His main research areas are distributed embedded systems and wireless sensor networks. In addition, his research includes design automation of high performance communication architectures, computational security, and intellectual property protection.

Farinaz Koushanfar received the BS degree from Sharif University of Technology (1999) and MS from University of California at Los Angeles (2001) in electrical engineering. She is currently pursuing the PhD degree in the Electrical Engineering and Computer Science Department at the University of California, Berkeley. Her research interests include various aspects of wireless sensor and information networks (in particular, location discovery, software sensor appliances and network infrastructure architecture, modeling, and distributed algorithms and optimizations), embedded systems, practical large-scale optimization techniques, and intellectual property protection.

Miodrag Potkonjak received the PhD degree in electrical engineering and computer science from the University of California, Berkeley, in 1991. He is a professor in the Computer Science Department at the University of California, Los Angeles (UCLA). In 1991, he joined C&C Research Laboratories, NEC USA in Princeton, New Jersey. Since 1995, he has been with UCLA. He has received the US National Science Foundation CAREER award, OKAWA foundation award, UCLA TRW SEAS Excellence in Teaching Award, and a number of best paper awards. His research interests include complex distributed systems, communication systems design, embedded systems, computational security, practical optimization techniques, and intellectual property protection.

Mani B. Srivastava received the BTech degree in electrical engineering from IIT Kanpur in India and the MS and PhD degrees from the University of California, Berkeley, in 1987 and 1992, respectively. He is an associate professor in the Electrical Engineering Department at the University of California, Los Angeles. From 1992 to 1996, he was a member of the technical staff at Bell Laboratories in networked computing research. His current research interests are in mobile and wireless networked computing systems, low power systems, and sensor networks. He received the Us National Science Foundation CAREER award in 1997 and the President of India Gold Medal in 1985. He is a senior member of the IEEE.