Improved Approximation Algorithms for Connected Sensor Cover *

Stefan Funke[†] Alex Kesselman[‡] Fabian Kuhn[§] Zvi Lotker[¶]

Michael Segal[||]

April 11, 2005

Abstract

Wireless sensor networks have recently posed many new system building challenges. One of the main problems is energy conservation since most of the sensors are devices with limited battery life and it is infeasible to replenish energy via replacing batteries. An effective approach for energy conservation is scheduling sleep intervals for some sensors, while the remaining sensors stay active providing continuous service. In this paper we consider the problem of selecting a set of active sensors of minimum cardinality so that sensing coverage and network connectivity are maintained. We show that the greedy algorithm that provides complete coverage has an approximation factor no better than $\Omega(\log n)$, where n is the number of sensor nodes. Then we present algorithms that provide approximate coverage while the number of nodes selected is a constant factor far from the optimal solution. Finally, we show how to connect a set of sensors that already provides coverage.

1 Introduction

Technological advances have led to the emergence of small, low-power devices that integrate sensors with limited on-board processing and wireless communication capabilities [4, 13]. Pervasive

*A preliminary version of this paper appeared in *The Proceedings of The Third International Conference on AD-HOC Networks & Wireless (ADHOC-NOW 2004)*.

[†]Max Planck Institut fur Informatik, Saarbrucken, Germany.

[‡]Max Planck Institut fur Informatik, Saarbrucken, Germany. The author has done a part of this work while visiting the Dipartimento di Scienze dell'Informazione at Universita di Roma La Sapienza, Italy. Supported in part by EYES project and MIUR grant FIRB WebMinds.

[§]Computer Engineering an Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland.

[¶]Project Mascotte, INRIA, Sophia Antipolis, 06902, France.

[||]Communication Systems Engineering Dept., Ben-Gurion University of the Negev, Beer-Sheva, Israel.

networks of such sensors open new perspectives for many potential applications, such as surveillance, environment monitoring and biological detection [2, 22]. A sensor network consists of multiple sensor nodes and each sensor can sense certain physical phenomena like light, temperature or vibrations around its location. The purpose of a sensor network is to process some high-level sensing tasks and report the data to the application.

Minimizing energy consumption to prolong the system lifetime is a major design objective for sensor networks since sensors need to operate for a long time on battery power. If all the sensor nodes simultaneously operate in active mode, an excessive amount of energy is wasted and the data collected is highly correlated and redundant. In addition, multiple packet collisions may occur when all the sensors in a certain area try to transmit as a result of a triggering event. Thus, maximizing the network lifetime can be achieved by scheduling some nodes to sleep (a power saving mode) while the remaining active nodes can still provide continuous service.

We study the problem of providing coverage and connectivity, where the goal is to minimize the number of active nodes. We assume that sensors have fixed locations. First we assume that the communication range is twice the sensing range. Then we show how to connect a set of sensors that already provides coverage under a less restricting assumption that the the communication range equals the sensing range.

Our results

We analyze a natural **greedy sector cover** algorithm, which is known to have an approximation factor of $\log m$ (i.e., how many sensors are used in the worst case compared to the optimal solution), where m is the maximal number of sectors covered by a single sensor (the formal definition of a sector is given in Section 3). The greedy sector cover has running time of $O(n^2 \log n)$ and can be applied to *sparse* instances of the problem, where m is a small constant. We demonstrate that despite the geometric nature of the problem, greedy sector cover has an approximation factor no better than $\Omega(\log m)$.

To obtain better approximation factors, we propose a simple **grid placement** algorithm which – while not guaranteeing full coverage of the region of interest – selects at most 6π (≈ 18.84) times the number of sensors of the optimum cover. The area that remains uncovered can be bounded in terms of the chosen grid. For dense sensor distributions we expect this algorithm to perform very well. Another algorithm, which we call **fine grid** algorithm, takes a different approach. It relies on the assumption that the region of interest P is also covered when decreasing the sensing ranges of all nodes by a factor of $(1 - \epsilon)$ — this is reasonable to assume since sensor deployments are typically such that coverage remains stable under small perturbations of node positions or sensing ranges. Under these premises we provide an algorithm which produces a solution with full coverage, and whose cardinality is at most 12 times the size of an optimum solution using the reduced sensing ranges. The theoretical running time of this algorithm unfortunately has an exponential dependency on $1/\epsilon^2$, hence is only feasible for moderately small ϵ. The techniques used in that algorithm extend to covering convex regions with 'fat' objects, which might interesting given that sensing areas in reality are not perfect disks

We also present the **distributed dominating cover** algorithm that has an approximation factor of 18 and provides approximate coverage. The distributed dominating cover algorithm has $O(n)$ time complexity and $O(n \log n)$ message complexity. Distributed dominating cover can be applied to networks without *centralized* control. The coverage provided by distributed dominating cover is less accurate compared to that of the centralized algorithms.

Finally, we extend our model to the case in which sensors locations and points to be covered are grid vertices and sensors locations are not fixed [12]. We derive constant approximation algorithms for different variations of this model, which improves upon the results of [12].

The rest of the paper is organized as follows. Related work appears in Section 2. Section 3 describes our model. Our algorithms and their analysis are presented in Section 4. Section 5 shows how to connect a covering set. We consider some extensions of our model in Section 6 and conclude with final remarks in Section 7.

2 Related Work

Many existing solutions have treated the problems of sensing coverage and network connectivity separately. The former problem has been studied extensively. A protocol that uses a local geometric calculation to preserve the sensing coverage is presented in [27], where if the sensing area of a node is completely covered by its neighbors, it enters sleep mode. A distributed probing-based [31] proposes a density control algorithm for robust sensing coverage called PEAS. In PEAS a sleeping node wakes up occasionally to check if there exist working nodes in its vicinity. If so, it sleeps again, otherwise it enters active mode. Several algorithms that use linear programming techniques to select a minimal set of active nodes for maintaining coverage are designed in [9, 25]. However, all these protocols do not guarantee network connectivity.

On the other hand, many protocols have been designed to maintain network connectivity. Although a wireless ad hoc network has no physical backbone infrastructure, a virtual backbone can be formed by nodes in a connected dominating set (CDS) of the corresponding unit-disk graph. The most important benefit of virtual backbone-based routing is significant reduction in the protocol overhead, which greatly improves the network throughput. GAF [30] conserves energy by dividing a region using a rectangular grid and electing a leader in each cell while putting the other nodes into sleep. In SPAN [10] a node decides whether it should be active or sleeping based on the connectivity among its neighbors. A different approach is used in ASCENT [8], where to make such a decision each node estimates the number of active neighbors and the per-link data loss rate. However, all the protocols mentioned above do not ensure sensing coverage.

Unfortunately, satisfying only coverage or connectivity alone is not sufficient since nodes may not be able to coordinate effectively or monitor the environment with the required accuracy. As a result, the problem of reducing energy consumption by keeping a minimal number of sensor nodes in active mode while maintaining sensing coverage and connectivity has received significant attention in recent time. The work in [17] designs centralized and distributed approximation $O(\log n)$-approximation algorithms for the connected sensor cover problem, where n is the number of nodes. In [32] it is shown that if the communication range is at least twice the sensing range, a complete coverage of a convex area implies connectivity among the nodes and derive optimality conditions under which a subset of working sensor nodes can be chosen for full coverage. The work in [29] derives the Coverage Configuration Protocol (CCP) that can provide different degrees of connected coverage as well as present a geometric analysis of the relationship between coverage and connectivity.

For energy-efficient monitoring, sensors can be partitioned into covers, which are activated iteratively. This approach takes advantage of the overlap between the sensing areas of individual sensors. The work in [26] considers the problem of maximizing the number of mutually exclusive sets of sensor nodes, where the members of each set together completely cover the monitored area. The work in [1] studies a variation of this problem, where the objective is to partition the sensors into disjoint covers such that the number of covers that include an area, summed over all areas, is maximized. In [3] the problem of finding a monitoring schedule that maximizes the network lifetime is considered.

The work in [6] gives a polynomial time algorithm that provides a constant approximation for the problem of covering points with a minimum number of disks, which have fixed locations. The algorithm of [6] is based on the system of ϵ-nets due to [24]. We note that this algorithm is rather complicated and the approximation guarantee is very large. The work in [19] derives constant-approximation algorithms for dynamic maintenance of the best-case and the worst-case coverage distances.

3 Model Description

Given a set of n sensors $S = \{s_1, \ldots, s_n\}$ distributed on the plane. Each sensor s_i has a location (x_i, y_i). The locations of the sensors are given in advance. Sensor s_i can monitor objects that are within a distance of its *sensing range* R_s^i. This area is called the *sensing region* of s_i and is denoted by A_i (note that A_i is a disk with a radius R_s^i whose center is (x_i, y_i)). We define a *sector* to be a maximal region that is formed by the intersection of a number of sensing regions such that all points within the sector are covered by the same set of sensors. We denote by m the maximal number of sectors covered by a single sensor.

Sensor s_i can communicate to all its neighbors within a distance of its *communication range* R_c^i. The *communication graph* GC of the network is an undirected graph in which nodes are sensors and there is an edge between two nodes if the distance between them is at most the minimum of their communication ranges (i.e., both nodes can talk to each other). For a subset of nodes S', the *communication subgraph* is the subgraph induced by the nodes in S'.

Let P be a region of interest on the plane. A *connected cover* of P is a subset $S' = \{s_{j_1}, \ldots, s_{j_q}\}$ of sensors such that $P \subseteq A_{j_1} \cup \ldots \cup A_{j_q}$ and the communication subgraph induced by S' is connected. An example of a connected cover is presented in Figure 1.

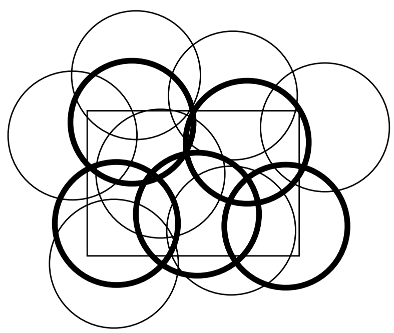

Figure 1: A connected cover example.

Definition 1. *Given a region of interest P, the* connected coverage problem *is to find a connected cover of P that uses the minimum number of sensors. We denote by OPT an optimal connected cover.*

Definition 2. *We say that an algorithm A has an approximation factor of c, if for any instance of the problem σ the size of the solution produced by A is at most $c \cdot |OPT| + a$, where a is a constant independent of σ.*

Definition 3. *We say that A provides* complete coverage *if the set of the selected sensors always covers P provided that there exists a feasible solution. Otherwise, we say that A provides* approximate coverage.

We assume that all sensors are identical and have the same sensing range R_s and communication range R_c. We also make a few simplifying assumptions.

1. We assume that $R_c \geq 2R_s$ unless it is explicitly stated otherwise (i.e., two sensors are able to communicate if their sensing areas intersect).

2. We assume that P is convex.

Theorem 1 ([29, 32]). *Under the above assumptions, complete coverage implies connectivity.*

In Section 5 we show how to replace assumption (1) by a less restricting assumption that $R_c = R_s$.

4 Algorithms for Connected Coverage

First we consider the natural greedy sector cover algorithm. We demonstrate that even for geometric instances of the problem considered here, the approximation factor of greedy sector cover is no better than $\Omega(\log m)$. In order to obtain better approximation factors, we then present an algorithm that always guarantees connectivity, but only provides partial coverage. This simple algorithm is called **grid placement** and selects a set of sensors of size at most 6π (≈ 18.84) times the optimum solution. Furthermore we propose a **fine grid** algorithm that has an approximation factor of 12 and guarantees full coverage provided the coverage of the region of interest P is stable under small perturbations of the sensing ranges, Finally, we present the distributed dominating cover algorithm which achieves an approximation factor of 18.

4.1 Greedy Sector Cover Algorithm

We consider the sectors produced by the sensors as elements to be covered while each sensor represents a set. We establish a tight bound of $\Theta(\log m)$ on the performance of greedy sector cover.

The greedy sector cover algorithm proceeds in two steps. Step 1: we use the algorithm of [5] to check whether the sensors cover the region of interest P and report failure if a feasible solution does not exist. Step 2: we apply greedy set cover to our problem, i.e., at each step we select a sensor that covers the maximal number of uncovered sectors.

Observation 1. *The number of sectors created by intersection of* n *disks is at most* $n(n-1)+1$.

The running time of greedy sector cover is $O(n^2 \log n)$: Step 1 takes $O(n^{1+\epsilon})$ time and Step 2 can be implemented in $O(n^2 \log n)$ time. The next theorem derives an upper bound on the approximation factor of the greedy sector cover algorithm.[1]

Theorem 2. *If the greedy sector cover algorithm terminates successfully, then the returned set is connected and* P *is completely covered. The approximation factor of the greedy sector cover algorithm is at most* $\log m$.

Proof. Step 1 of the algorithm ensures that the union of the sensing regions covers P and in Step 2 all sectors that intersect P are covered. Thus, P is completely covered. According to Theorem 1, the covering set is connected. The approximation factor is a well-known result. □

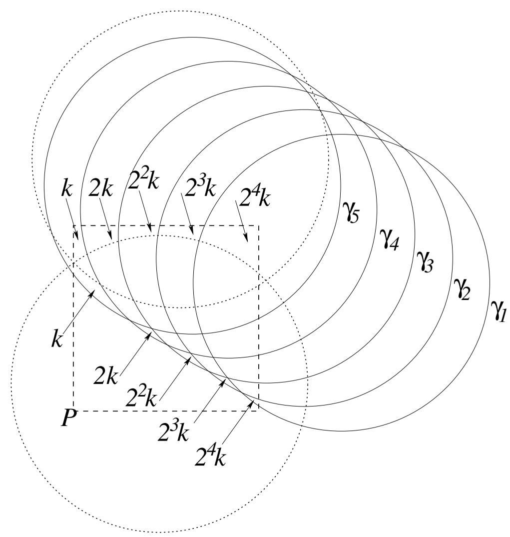

Figure 2: Sketch of the $\Omega(\log m)$ lower bound proof.

In the following, we show that the above upper bound is tight up to a constant factor. Before going into details, we give an intuitive overview. We assume that the region P which has to be covered is a square of side length $s_P \approx R_s$. We place n sensors such that a) P can be covered using only two sensors and b) the greedy sector cover algorithm uses $\Omega(\log n) = \Omega(\log m)$ sensors to cover P, where m is the number of sectors in P. The idea of the sensor placement is sketched in Figure 2. The area P is given by the dotted square. Since P can be covered using the two dashed disks, the size of an optimal cover is 2. For simplicity, assume that we can place an arbitrary

[1] Observe that the analysis of greedy sector cover can be extended to the case of non-uniform sensing radii.

number of sectors on an arbitrarily small area without generating new sectors. We will show how this can be achieved using disks of radius R_s. We generate two exponentially increasing sequences consisting of $t, 2t, 2^2t, \ldots, 2^{k-1}t$ sectors as indicated by the arrows in Figure 2. We also place k disks $\gamma_1, \ldots, \gamma_k$ (corresponding to k sensors) such that disk γ_1 contains the $2^{k-1}t$ sectors of both sector sequences, γ_2 contains the $2^{k-2}t$ and the $2^{k-1}t$ sectors of the upper sequence and the $2^{k-2}t$ sectors of the lower sequence, etc. as showed in Figure 2. If the parameter t is chosen large enough, the number of sectors induced by the disks γ_i and by the two dashed disks becomes negligible. It is not hard to see that the greedy sector cover algorithm will pick all the disks $\gamma_1, \ldots, \gamma_k$ in the given order from right to left. As long as we do not choose t too large (exponential in k is still ok), $k \in \Theta(\log m)$ and hence the number of disks chosen by the greedy algorithm is logarithmic in the number of sectors. We are now coming to a detailed proof of this lower bound.

Theorem 3. *The approximation factor of the greedy sector cover algorithm is at least* $\Omega(\log m)$.

Proof. As mentioned, we construct an example where the greedy algorithm behaves badly.The region P which has to be covered is a square in our example.. We start with some definitions. We say that the set of disks $\{D_i\}_{i=1}^n$ is a *chain* of size n if all the disks intersect and their centers lie on a ray, which defines the ordering of the disks. In the following, we restrict our considerations to only one side of the ray. For all chains which we use in our example configuration, only sectors from one side lie inside P. We denote the intersection point between disk i and disk j by $P_{i,j}$. Figure 3 shows an example of a chain. Note that we can associate a sector with each point $P_{i,j}$. Thus, the number of sectors in a chain is $\binom{n+1}{2}$. Denote by $\Upsilon(P_{1,2}, P_{1,n}, P_{n,n-1})$ all the sectors that are not in the first and the last disks. Observe that we can make the area of $\Upsilon(P_{1,2}, P_{1,n}, P_{n,n-1})$ as small as we like. However, most of the sectors of the chain are in $\Upsilon(P_{1,2}, P_{1,n}, P_{n,n-1})$, only $2n-1$ sectors are outside $\Upsilon(P_{1,2}, P_{1,n}, P_{n,n-1})$. For a chain δ, we denote by $\Upsilon(\delta)$ all the sectors that are not contained in the first and the last disks of δ.

Observation 2. *For any square* S *with side length* ϵ *and for any* $n \in \mathbb{N}$, *there exists a chain* $\delta = \{D_i\}_{i=1}^n$ *s.t.* $\Upsilon(\delta) \subset S$.

We place a sequence of chains $\alpha = \{\alpha_i\}_{i=1}^k$ in such a way that the size of the i-th chain is $2^{k+1-i} \cdot t$ and disks from different chains intersect outside the relevant square (see Figure 4). We also place a sequence of chains $\beta = \{\beta_i\}_{i=1}^k$ in such a way that the size of the i-th chain is $2^{k+1-i} \cdot t$ and it is fully contained in the intersection of all the disks that are in the $(i+1)$-th chain (see Figure 4). Since both sequencers of chains are disjoint, no disk in these sequences can contain more than half of the sectors.

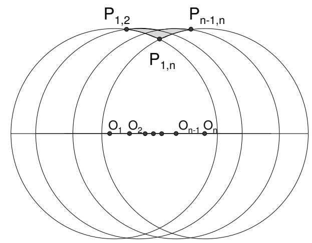

Figure 3: *A chain of unit disk. The point $P_{1,2}$ is the intersection of the first two disks. The point $P_{n-1,n}$ is the intersection of the last two disk and the point $P_{1,n}$ is the intersection of the the first and the last two disks. The gray area is $\curlyvee(P_{1,2}, P_{1,n}, P_{n,n-1})$ contains most of the sectors of the chain.*

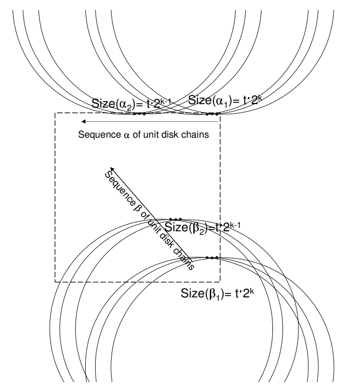

Figure 4: *The dashed square is the area we wish to cover and there are two sequences of chains α and β. The former sequence of chains is close to the upper edge of the square and the latter sequence of chains is close to the diagonal of the square.*

Now we place a chain $\gamma = \{\gamma_i\}_{i=1}^{k}$ so that the greedy sector cover algorithm will select this chain as a cover (see Figure 5). We add this chain in such a way that disk γ_i contains the $\curlyvee(\alpha_1), \curlyvee(\alpha_2), ..., \curlyvee(\alpha_i)$ and $\curlyvee(\beta_i)$ sectors. This means that $\curlyvee(\beta_i) \subset \gamma_j$ iff $i \neq j$ and $\curlyvee(\alpha_i) \subset \gamma_j$ iff $i \leq j$. Note that we can assure that no new sectors are added to chain α. Since disk γ_i does not intersect $\curlyvee(\beta_j)$, the number of new sectors generated by disk γ_i is linear in the number of disks in chain β_j while the number of sectors in $\curlyvee(\beta_j)$ is quadratic in the number of disks in chain β_j. Hence, if t is big enough, the number of new sectors is negligible. We get that the number of sectors contained in disk γ_i is bigger than $2\binom{2^{k+1-i}+1}{2}$.

Finally, we place two more disks δ_1, δ_2. These disks cover all the sectors that are in the square we wish to cover (see Figure 5). Since δ_1 does not contain the β sequence of disk chains and δ_2

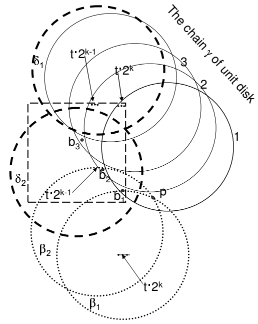

Figure 5: *The dashed disks δ_1, δ_2 form the OPT cover. The solid disks are the disks selected by the greedy sector cover algorithm. Each disk is labeled with the iteration at which it is selected. The point p is the intersection of all k initial disks that form β-chains. The point $b_i \in \gamma_i \setminus (\gamma_{i+1} \cup \gamma_{i-1})$ is the second point we use to define the i-th initial disk.*

does not contain the α sequence of disk chains, we get that the total number of sectors in those disks is less than the number of sectors in γ_1.

We will prove that the greedy sector cover algorithm chooses all γ disks. Let $t = 4^k$.

Lemma 1. *The greedy sector cover algorithm selects disk γ_i at the i-th iteration.*

Proof. The proof is by induction on i. We denote by $num(d)$ the number of sectors covered by disk d.

Basis $i = 1$. Since chain γ does not add new sectors to the α sequence of chains, we obtain that

$$num(\delta_1) \leq \binom{k+1}{2} + \sum_{j=1}^{k} \binom{t \cdot 2^{k+1-j} + 1}{2}.$$

We have that

$$num(\delta_2) < \binom{k+1}{2} + \sum_{j=1}^{k} \binom{t \cdot 2^{k+1-j} + 1}{2} + k \sum_{j=1}^{k} 2t2^{k+1},$$

because chain γ does not intersect $\Upsilon(\alpha_i)$ for all $i = 1, 2, ..., k$. Note that our bound of $num(\delta_2)$ is bigger than the bound on $num(\delta_1)$. A similar calculation shows that the number of sectors covered by each disk in chain β_i is less than $num(\delta_2)$. Now we establish an upper bound on the number of sectors covered by disk γ_i. Since disk γ_i contains the $\Upsilon(\alpha_1), \Upsilon(\alpha_2), ..., \Upsilon(\alpha_i)$ and $\Upsilon(\beta_i)$ sectors, we get that:

$$num(\gamma_i) \leq \sum_{j=1}^{i} \binom{t \cdot 2^{k+1-j} + 1}{2} + \binom{t \cdot 2^{k+1-i} + 1}{2} + \binom{k+1}{2}.$$

We also have:

$$num(\gamma_1) \geq 2\binom{t2^k+1}{2} > \binom{t2^k+1}{2} + \frac{t^2 \cdot 4^k}{2}.$$

That is due to the fact that γ_1 contains $\curlyvee(\alpha_1), \curlyvee(\beta_1)$.

Note that for all $k > 1$:

$$\begin{aligned}
num(\gamma_1) - \binom{t2^k+1}{2} &> \frac{t^2 \cdot 4^k}{2} \\
&> \frac{4}{3}\left(\frac{t^2 4^{k-1}}{2} + \frac{1}{2} \cdot \frac{4^k t^2}{3} + \binom{k+1}{2}\right) \\
&> \frac{4}{3}\left(\frac{(t \cdot 2^{k+1-i})^2}{2} + \sum_{j=2}^{i} \frac{(t \cdot 2^{k+1-j})^2}{2} + \binom{k+1}{2}\right) \\
&> num(\gamma_i) - \binom{t2^k+1}{2}.
\end{aligned}$$

We obtain that $num(\gamma_1) > num(\gamma_i)$ for all $i = 2, 3, ..., k$. On the other hand, from

$$\begin{aligned}
num(\gamma_1) - \binom{t2^k+1}{2} &> \frac{t^2 \cdot 4^k}{2} \\
&> \frac{4}{3}\left(\binom{k+1}{2} + \frac{1}{2} \cdot \frac{4^k t^2}{3} + 2k^2 t 2^{k+1}\right) \\
&> \frac{4}{3}\left(\binom{k+1}{2} + \sum_{j=2}^{i} \frac{(t \cdot 2^{k+1-j})^2}{2} + k\sum_{j=1}^{k} 2t2^{k+1}\right) \\
&> num(\delta_2) - \binom{t2^k+1}{2},
\end{aligned}$$

it follows that $num(\gamma_1) > num(\delta_2)$. Therefore, the greedy sector cover algorithm selects disk γ_1 at the first iteration.

Induction Step $i \rightarrow i+1$. We assume that the hypothesis holds for the the i-th iteration and show it also holds for the $(i+1)$-th iteration. Using similar counting arguments, we get the following equations:

$$num(\gamma_m) \leq \begin{cases} 0 & m \leq i \\ \sum_{j=i+1}^{m} \binom{t \cdot 2^{k+1-j}+1}{2} + \binom{t \cdot 2^{k-i}+1}{2} + \binom{k+1}{2} & m > i; \end{cases}$$

$$num(\delta_1) \leq \binom{k+1-i}{2} + \sum_{j=i+1}^{k} \binom{t \cdot 2^{k+1-j}+1}{2};$$

$$num(\delta_2) \leq \binom{k+1-i}{2} + \sum_{j=i+1}^{k} \binom{t \cdot 2^{k+1-j}+1}{2} + (k-i)\sum_{j=i+1}^{k} 2t2^{k+1-i};$$

$$num(\gamma_{i+1}) \geq 2\binom{t2^{k-i}+1}{2} > \binom{t2^{k-i}+1}{2} + \frac{t^2 \cdot 4^{k-i}}{2}.$$

We obtain that for all $k > 1$:

$$\begin{aligned} num(\gamma_{i+1}) - \binom{t2^{k-i}+1}{2} &> \frac{t^2 \cdot 4^{k-i}}{2} \\ &> \frac{4}{3}\left(\frac{t^2 4^{k-1-i}}{2} + \frac{1}{2} \cdot \frac{4^{k-i}t^2}{3} + \binom{k+1}{2}\right) \\ &> num(\gamma_j) - \binom{t2^{k-i}+1}{2}. \end{aligned}$$

We get that $num(\gamma_{i+1}) > num(\gamma_j)$ for all $k \geq j > i+1$. Finally from

$$\begin{aligned} num(\gamma_{i+1}) - \binom{t2^{k-i}+1}{2} &> \frac{t^2 \cdot 4^{k-i}}{2} \\ &> \frac{4}{3}\left(\binom{k+1}{2} + \frac{1}{2} \cdot \frac{4^{k-i}t^2}{3} + 2(k-i)^2 t 2^{k+1}\right) \\ &> num(\delta_2) - \binom{t2^{k-i}+1}{2}, \end{aligned}$$

it follows that $num(\gamma_1) > num(\delta_2)$. Therefore, greedy sector cover selects disk γ_{i+1} at the $i+1$-th iteration. □

Now we will elaborate more on the placement of the disks. The idea is to place the γ chain first in such a way that the ray of γ is parallel to the diagonal, and $\Upsilon(\gamma)$ is close to the upper right corner of the square. By Observation 2, we can place the α sequence of chains as claimed. Placement of the β sequence of chains is a little bit more tricky. Note that two points on the boundary completely define a unit disk. The idea is to put k initial disks as follows. We pick one point p outside the square near the lower right corner. We have that p is on the boundary of all the initial disks. The second point of the i-th disk is a point in $\gamma_i \setminus (\gamma_{i+1} \cup \gamma_{i-1})$.

The theorem follows by Lemma 1 and the fact that the total number of disks is at most 8^k. □

4.2 Grid Placement Algorithm

To overcome the potentially bad performance of the greedy algorithm, we propose in the following a simple grid-based algorithm that selects a set of sensors which form a connected subgraph of the communication graph, and whose cardinality is at most a constant factor larger than the optimal connected cover. While we cannot guarantee complete coverage, our algorithm bounds the size of the uncovered region in terms of the chosen grid. In particular for densely distributed sensors we expect our algorithm to perform very well.

We put a grid with cells of size $R_s/\sqrt{2} \times R_s/\sqrt{2}$ over the region of interest P. For simplicity, we assume that the number of grid cells intersecting the boundary of P is negligible compared to the total number of cells and we will ignore it in the analysis. A specific instance of a grid is defined by its position. We choose exactly one sensor in each cell to be in the covering set. Finally, we add extra sensors to make the covering set connected.

Observation 3. *The selection of the cell size implies that each sensor covers its cell completely and sensors in neighboring cells are able to communicate with each other.*

Observe that depending on the position of the grid, some cells may be empty. To obtain better coverage, we aim to minimize the number of such cells. The work in [7] gives an algorithm for solving the grid placement problem and minimizes the number of grid cells containing no point with running time of $O(n \log n)$. We use the algorithm of [7] to optimize the grid placement.

The grid placement algorithm and the MST connection algorithm are presented in Figure 6 and Figure 7, respectively. An example of the grid placement algorithm can be found in Figure 8. The running time of the MST connection algorithm is $O(n^3)$: Step 1 takes $O(n^3)$ time and Step 2 takes $O(n^2)$ time. The running time of the grid cover is dominated by that of MST connection.

1. Define a grid with the cell size of $R_s/\sqrt{2} \times R_s/\sqrt{2}$ covering P.
2. Apply the algorithm of [7] to find the grid placement that minimizes the number of empty cells.
3. Select an arbitrary sensor in each non-empty cell intersecting P and add it to the covering set (the *basic* cover).
4. Run the MST connection algorithm and add the returned disks to the final set (the *extended* cover).

Figure 6: The grid placement algorithm.

Before we analyze the proposed scheme we want to point out the simplicity of our scheme based on the MST construction which makes it quite attractive for use in practice.

The following theorem states the performance of the grid placement algorithm.

Theorem 4. *The covering set found by the grid placement algorithm is connected and the uncovered area of P is bounded by the union of the empty cells. The approximation factor of the grid placement algorithm is at most 6π.*

1. Create a weighted graph G in which nodes are the connected components in the communication subgraph induced by the nodes from the basic cover. We call a node in G a *super-node*. We say that a node s is *directly reachable* from a super-node U if there exists an edge in GC (communication graph) between a node $v \in U$ and s. We add an edge between two super-nodes U and V in G if there exists a path between them in GC that does not contain any node directly reachable from another super-node W but not directly reachable from either U or V. The weight of the edge equals to the number of regular nodes that are not included in the basic cover on a shortest path satisfying the above condition.

2. Compute a minimum weight spanning tree (MST) of G, T.

3. Return the set of nodes that lie on the shortest paths corresponding to the edges of T.

Figure 7: The MST connection algorithm.

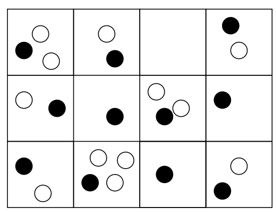

Figure 8: An example of the grid placement algorithm.

Proof. First we consider the MST connection algorithm. We argue that if there exists a path between two super-nodes U and V in GC, then there also exists a path between them in G. If G contains edge (U, V), we are done. Otherwise, by our construction, U can reach V in G through another super-node W. Therefore, the MST connection algorithm returns a connected set. By Observation 3, all non-empty cells are covered. Hence, the uncovered area of P is bounded by the union of the empty cells.

Let k and l be the number of nodes in the basic and the extended cover, respectively. The area covered by a single sensor from the basic cover is $R_s^2/2$. On the other hand, any sensor in OPT can cover the area of at most πR_s^2. Therefore, $k \leq 2\pi|OPT|$, since P is covered by OPT. We will show that the number of sensors in the extended cover is at most $2(k-1)$. Clearly, the number of nodes in G is at most k and thus the number of edges in the MST is bounded by $k-1$. We claim that the weight of any edge in G is at most two. We say that a super-node in G *covers* a cell if it includes a node from the basic cover that is located in this cell. Suppose towards a contradiction that the weight of an edge between super-nodes U and V is greater than two. We have that at least one intermediate node on the shortest path between U to V must lie in a cell $\mathcal{C}$ that is not covered by the super-nodes U and V. Let W be the super-node covering $\mathcal{C}$. We obtain that the node from $\mathcal{C}$ on the path between U and V is directly reachable from W but not directly reachable from U or V, which contradicts our construction. Therefore, we get that $l \leq 2(k-1) < 4\pi|OPT|$, which establishes the theorem. □

4.3 Fine Grid Algorithm

While the grid placement algorithm could guarantee connectedness and bounded size of the selected sensors, there was no formal guarantee on the coverage of the chosen sensors. In particular, sparse sensor distributions might lead to a large number of empty cells in the grid and hence a considerable portion of uncovered area. In the following we present a grid-based algorithm that guarantees full coverage of the region of interest P, provided we have a somewhat 'stable' problem instance, where even after decreasing the sensing radii to $R'_s = (1 - \epsilon/\sqrt{2})R_s$ – we call this the *reduced sensing radius* –, a full cover of the region of interest exists. We guarantee that the size of our solution (which uses sensing radius R_s) is at most a constant times larger than the optimum solution for the reduced sensing radius R'_s.

We first place a coarse grid with cells of size $2R_s \times 2R_s$ over the domain and put over this grid another fine grid which further subdivides each cell of the coarse grid into small cells of size $\epsilon R'_s \times \epsilon R'_s$. Then we consider the set of disks $\{A_1, \ldots, A_n\}$ and snap the center of each disk to the closest vertex of the fine grid and set its radius to R_s.

Observation 4. *The selection of the cell sizes implies that (i) each sensor with reduced sensing range R'_s has its covered area completely contained in the respective aligned disk with sensing range R_s (ii) each aligned disk intersects at most four cells of the coarse grid, and (iii) each cell of the coarse grid is intersected by at most $20/\epsilon^2$ aligned disks for reasonably small ϵ.*

Using property (iii) of the observation, we can compute individually for each cell of the coarse grid a minimum cover using the aligned disks of radius R_s by brute-force.

The fine grid algorithm itself is described in Figure 9 and Figure 10 provides an illustration to our claims. The running time of the fine grid algorithm is dominated by Step 2 and is $O(q \cdot 2^{20/\epsilon^2})$, where q is the number of cells in the coarse grid. We want to emphasize, though, that this is a worst-case running time, and in practice, we do not expect such a large number of aligned disks overlapping a single cell of the coarse grid.

As in the previous algorithm, the connection step is based on MST construction. Let us now prove the approximation guarantee of the produced solution.

1. Construct a grid of cell width $2R_s \times 2R_s$ over the domain and impose a fine grid with cells of width $\epsilon R'_s \times \epsilon R'_s$.
2. For each cell $\mathcal{C}$ of the coarse grid that is contained in P, consider all subsets of aligned disks (to the fine grid) intersecting $\mathcal{C}$ and find the smallest set that covers $\mathcal{C}$ Add these disks to the covering set (the *basic* cover).
3. Run the MST connection algorithm and add the returned disks to the final set (the *extended* cover).

Figure 9: The fine grid algorithm.

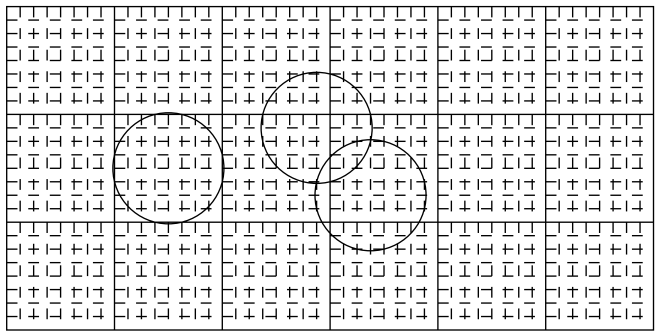

Figure 10: Coarse and fine grids with aligned disks.

Theorem 5. *The covering set found by the fine grid algorithm is connected and for sensing radius R_s covers the whole region P. The size of the computed set is at most 12 times the size of the optimal solution for sensing radius $R'_s = (1 - \epsilon/\sqrt{2})R_s$.*

Proof. Full coverage is trivially achieved since in each cell we separately determine a full cover. This cover is also minimal for that cell with respect to the aligned disks of sensing radius R_s. On the other hand, by Observation 4, in the optimum solution with sensing radius R'_s, a (unaligned) disk covers only parts of at most 4 cells. Therefore, the size of the basic cover is at most four times the size of OPT that uses the unaligned disks of reduced sensing radius R'_s to cover P. The approximation factor follows since the size of the extended cover is bounded by twice the size of the basic cover. If it is not the case, the center of at least one disk is not covered, which contradicts the construction of the basic cover (see the proof of Theorem 4). □

The fine grid algorithm also extens to the more general problem of covering a convex area with fat geometric objects. Suppose that we wish to cover a convex area using the minimum number of disks, squares or triangles. We put a coarse and a fine grid over this area so that each object intersects a constant number of coarse grid cells and each coarse grid cell is intersected by a constant number of different objects located at the vertices of the fine grid. Then we cover each cell of the coarse grid using the minimum number of objects by running exhaustive search over the sets of objects intersecting it. Finally, we return the union of the sets that cover all cells of the coarse grid. In real applications sensing areas are rarely perfect disks, but still have a non-skinny shape. Our algorithm still remains applicable for such cases.

4.4 Distributed Dominating Cover Algorithm

A connected dominating set (CDS) of GC is a subset $S' \subseteq S$ such that each node in $S \setminus S'$ is adjacent to some node in S' and the communication subgraph induced by S' is connected. It has been shown that the problem of finding a minimum CDS for unit-disk graphs is NP-hard [11]. The work in [28] gives an 8-approximation algorithm with $O(n)$ time complexity and $O(n \log n)$ message complexity. The recent work by Funke et al [15] improves the approximation factor to 6.91.

We assume that each sensor has the communication range of $R_c = 2R_s/3$ and the communication graph GC is connected. Let GD be a unit disk graph in which each sensor corresponds to a disk of radius $R_s/3$ and two nodes are connected by an edge if the corresponding disks intersect. The dominating cover algorithm just computes a connected dominating set in GD using the algorithm of [28]. The following observation is useful to demonstrate the coverage property.

Observation 5. *If a disk d is adjacent to another disk d' in GD, the sensor corresponding to d completely covers d'.*

The next theorem analyzes the performance of the dominating cover algorithm.

Theorem 6. *The covering set found by the dominating cover algorithm is connected and the uncovered area of P is bounded by the part of P that is not covered by the set of disks in GD. The approximation factor of the dominating cover algorithm algorithm is at most* 18*.*

Proof. Obviously, dominating cover returns a connected set since GD is connected. The coverage property follows by Observation 5. We have that the area covered by a single sensor from a maximal *independent set* IS computed by the CDS algorithm [28] is $\pi R_s^2/9$. On the other hand, any sensor in OPT can cover the area of at most πR_s^2. The approximation factor of 18 is due to the fact that in [28] the size of the final connected dominating set is at most twice the size of IS. □

5 Connectivity

In this section we show how to convert a *complete* covering set B into a connected covering set under a realistic assumption that $R_c = R_s$.[1] We analyze two algorithms based on MST and Steiner Tree techniques. If the basic covering set is small, it is worth to use the former algorithm. Otherwise, it is preferable to use the latter algorithm.

Observation 6. *For each sensor from the basic cover, there is a sensor in OPT at distance of at most R_s.*

The observation follows from the fact that the sensing range is R_s and if there is no such an OPT sensor, the location of at least one sensor is not covered by OPT. Observation 6 implies the following lemma.

Lemma 2. *Adding nodes in OPT makes the basic cover connected.*

We consider the MST connection algorithm (see Figure 7). The proof of the next theorem is similar to that of Theorem 4.

Theorem 7. *The size of the connecting set returned by the MST connection algorithm is at most $2(|B| - 1)$.*

In the steiner tree connection algorithm, we apply the algorithm of [16] for the node-weighted steiner tree problem with unit weights on the communication graph GC while the basic cover B represents the set of terminal nodes. Note that Lemma 2 implies that the cost of an optimal steiner tree is bounded by $|OPT|$.

Theorem 8. *The size of the connecting set returned by the steiner tree connection algorithm is at most $O(\ln |B|) \cdot |OPT|$.*

6 Extensions

In this section we extend our model to the discrete case in which sensors locations and points to be covered are grid vertices and sensors locations are not fixed [12]. We derive constant approximation algorithms for the model with and without obstacles. We are given a $N \times N$ grid and the goal is to cover a subset of grid vertices using the minimum number of sensors (the covering set

[1]We note that even if B does not provide complete coverage, our algorithms would still return a connected set, but the approximation factor could be worse.

must not be necessarily connected). The restriction is that sensors must be placed only at the grid vertices. There may be some *obstacles* on the plain, and thus some parts of the sensing region of a sensor can be obscured. We assume that each cell has size of $a \times a$. We present two constant-factor approximation algorithms for the model with and without obstacles. We improve upon the results of [12], where they give algorithms with running time $O(N^4)$ providing no worst-case performance guarantees. In contrast to [12], we focus on deterministic and not probabilistic coverage.

Our first algorithm for the model *with obstacles* is based on greedy set cover. The greedy point cover algorithm proceeds as follows. We consider N^2 sensors located at the grid vertices and sensor at each location covers the points in its sensing region that are not obscured by the obstacles. Then we apply greedy set cover: until all points of interest are covered, at each iteration we select a sensor that covers the maximal number of uncovered points. The running time of greedy point cover is $O(N^2 \log N)$.

Theorem 9. *The approximation factor of the greedy point cover algorithm is at most* $\log(4\pi R_s^2/a^2)$.

The theorem is due to the fact that each sensor covers at most $4\pi R_s^2/a^2$ points.

The second algorithm is for the model *without obstacles*. The PTAS point cover algorithm proceeds as follows. We apply the PTAS of [14] for covering points with unit disks (see [18]) to find a coverage of the points of interest. Now we need to adjust the locations of the sensors to be at the grid vertices. We replace each sensor that is located not at a grid vertex by a minimal set of sensors located at the grid vertices of its cell that cover the same set of points. Obviously, we replace each sensor by at most four sensors. PTAS point cover has running time of $O(N^2)$.

Theorem 10. *The approximation factor of the PTAS point cover algorithm is at most* $4(1+\epsilon)$.

7 Concluding Remarks

In this paper we investigate an important problem of maintaining coverage and connectivity in wireless sensor networks. The goal is to keep the minimum number of sensor nodes in active mode, thus maximizing the network lifetime. We present approximation algorithms with provable worst-case guarantees. There is a trade-off between the ease of implementation and the accuracy of the proposed algorithms, which allows one to select the proper algorithm for the specific needs. An open problem is whether it is possible to obtain a distributed constant approximation algorithm that provides full coverage.

Acknowledgment. We would like to thank two anonymous referees for many helpful suggestions.

References

[1] Z. Abrams, A. Goel, and S. Plotkin, "Set K-Cover Algorithms for Energy Efficient Monitoring in Wireless Sensor Networks." *Proc. of Third International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.

[2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, March 2002.

[3] P. Berman, G. Calinescu, C. Shah and A. Zelikovsky, ”Power Efficient Monitoring Management in Sensor Networks," *IEEE Wireless Communication and Networking Conference (WCNC'04)*, Atlanta, March 2004.

[4] B. Badrinath, M.Srivastava, K.Mills, J.S Holtz and K. Sollins, editors, "Special Issue on Smart Spaces and Environments," *IEEE Personal Communications*, 2000.

[5] B. Ben-Moshe, M. J. Katz and M. Segal, "Obnoxious Facility Location: Complete Service with Minimal Harm," *International Journal of Computational Geometry and Applications*, Vol. 10, pp. 581-592, 2000.

[6] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite VC-dimension," *Discrete and Computational Geometry*, Vol. 14, pp. 463-479, 1995.

[7] P. Bose, A. Maheshwari, P Morin, and J. Morrison, ”The grid placement problem," *Proceedings of the 7th Annual Workshop on Algorithms and Data Structures*, Vol. 2125 of LNCS, pp. 180-191, 2001.

[8] A. Cerpa and D. Estrin, ”ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," *Proc. of INFOCOM '02*, New York, NY, USA, June 2002.

[9] K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho, ”Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, Vol. 51(12), pp. 1448-1453, December 2002.

[10] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, ”Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks," *ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 2001)*, Rome, Italy, July 2001.

[11] B. Clark, C. Colbourn and D. Johnson, “Unit Disk Graphs”, *Discrete Mathematics*, Vol. 86, pp. 165-177, 1990.

[12] S. S. Dhillon and K. Chakrabarty, ”Sensor placement for effective coverage and surveillance in distributed sensor networks,” *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 1609-1614, 2003.

[13] D. Estrin, R. Govindan and J. Heidemann, editors, “Special Issue on Embedding the Internet,” *Communications of the ACM*, Vol. 43, 2000.

[14] M. Franceschetti, M. Cook, and J. Bruck, ”A Geometric Theorem for Approximate Disk Covering Algorithms,” *Technical Report ETR035*, Caltech University, January 2001.

[15] S. Funke, A. Kesselman, U. Meyer and M. Segal, “A Simple Improved Distributed Algorithm for Minimum CDS in Unit Disk Graphs”,it *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications’2005*, to appear.

[16] S. Guha and S. Khuller, “Improved Methods for Approximating Node Weighted Steiner Trees and Connected Dominating Sets,” *Inf. Comput.*, Vol. 150(1), pp. 57-74, 1999.

[17] H. Gupta, S. R. Das and Quinyi Gu, “Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution,” *Proc. of MobiHoc 03*, Annapolis, Maryland, June 2003.

[18] D. S. Hochbaum and W. Maas, “Approximation schemes for covering and packing problems in image processing and vlsi,” *J. ACM*, Vol. 32, pp. 130-136, 1985.

[19] H. Huang, A. Richa and M. Segal, “Dynamic Coverage and Related Problems in Ad Hoc Sensor Networks,” *ACM Mobile Networks and Applications: Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks*, 2003.

[20] D. S. Johnson, “Approximation algorithms for combinatorial problems,” *J. Comput. System Sci.* Vol. 9, pp. 256-278, 1974.

[21] D. T. Lee, M. Sarrafzadeh and Y. F. Wu, “Minimum cut for circular-arc graphs”, *SIAM J. Computing* Vol. 19(6), pp. 1041-1050, 1990.

[22] A. Mainwaring, J. Polastre, R. Szewczyk, and D. Culler, “Wireless sensor networks for habitat monitoring,” *First ACM International Workshop on Wireless Workshop in Wireless Sensor Networks and Applications*, August 2002.

[23] S. Masuda and K. Nakajima, "An optimal algorithm for finding a maximum independent set of a circular-arc graph," *SIAM Journal on Computing*, Vol. 17(1), pp. 41-52, 1988.

[24] J. Matousek, R. Seidel and E. Welzl, "How to net a lot with little: small ϵ-nets for disks and halfspaces," *Proc. of 6th Annu. ACM Sympos. Comput. Geom.*, pp. 16-22, 1990.

[25] S. Meguerdichian and M. Potkonjak, "Low Power 0/1 Coverage and Scheduling Techniques in Sensor Networks," *UCLA Technical Report 030001*, January 2003.

[26] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," *Proc. of IEEE International Conference on Communications (ICC)*, pp. 472-476, 2001.

[27] D. Tian and N.D. Georganas, "A Coverage-preserved Node Scheduling scheme for Large Wireless Sensor Networks," *Proc. of First International Workshop on Wireless Sensor Networks and Applications*, Atlanta, USA, September 2002.

[28] P. J. Wan, K. Alzoubi and O. Frieder, "Distributed Construction of Connected Dominating Set in Wireless ad hoc networks," *Proc. of INFOCOM 2002*.

[29] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. D. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks," *Proc. of SenSys '03*.

[30] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *ACM/IEEE International Conference on Mobile Computing and Networking (Proc. of MobiCom 2001)*, Rome, Italy, July 2001.

[31] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," *Proc. of The 23nd International Conference on Distributed Computing Systems (ICDCS)*, 2003.

[32] H. Zhang and J. C. Hou, "Maintaining Sensing Coverage and Connectivity in Large Sensor Networks," *Technical Report UIUC*, UIUCDCS-R-2003-2351, 2003.