Improved Algorithms for the Connected Sensor Cover Problem

December 10, 2003

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, i.e., maintain sensing coverage and network connectivity. In this paper we consider the problem of selecting a set of active sensors of minimum cardinality satisfying the above requirements. We develop constant-factor approximation algorithms for fixed and variable sensor locations. Our algorithms are based on different techniques, which allows to find the desired trade-off between the complexity and the quality of approximation. We also present an improved 6-approximation algorithm for the minimum connected dominating set problem in unit disk graphs. Finally, we propose algorithms for connecting a covering set.

1 Introduction

Recent 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, 15]. Pervasive networks of such sensors open new perspectives for many potential applications, such as surveillance, environment monitoring and biological detection [1, 27]. 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. Prolonging the network lifetime can be achieved by scheduling some nodes to sleep (a power saving mode) while the remaining active nodes provide continuous service. Note that as long as coverage and connectivity are maintained, a sensor network still functions properly even if some sensors die much earlier than others.

Many existing solutions have treated the problems of sensing coverage and network connectivity separately. The problem of sensing coverage has been studied extensively. A protocol that uses a local geometric calculation to preserve the sensing coverage is presented in [31]. In this protocol if the sensing area of a node is completely covered by its neighbors, it enters sleep mode. A distributed probing-based density control algorithm for robust sensing coverage PEAS has been designed in [35]. 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 have been proposed [8, 29]. However, 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 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 [34] conserves energy by dividing a region using a rectangular grid and electing a leader in each cell while putting all the other nodes into sleep. In SPAN [9] a node decides whether it should be active or sleeping based on the connectivity among its neighbors. A different approach is used in ASCENT [7], where to make the decision each node estimates the number of active neighbors and the per-link data loss rate.

In general, a *connected dominating set* (CDS) of a graph $G = (V, E)$ is a subset $V' \subseteq V$ such that each node in $V \setminus V'$ is adjacent to some node in V', which induces a connected subgraph. A dominating set is *weakly connected* if the graph induced by the stars of vertices of dominating set is connected. We notice that the problem of finding and maintaining of minimum size CDS in the corresponding unit-disk graph is equivalent to the (minimum energy) broadcast problem with the restriction that source is defined in advance. In this case we aim to minimize the number of transmitting nodes. Similarly, one can define a "partial" CDS of a given graph that corresponds to a multicast tree. It has been show that the problem of finding CDS (or weakly CDS) is NP-hard even for unit-disk graphs [12]. Guha and Khuller [20] gave an algorithm that works for general graphs and finds CDS that is a factor of $O(H(\Delta))$ far from the optimum CDS, where Δ is the maximal degree of the given graph and H is the harmonic function. Chen and Liestman [10] used the result in [20] in order to get $\ln \Delta$-approximation factor algorithm. Dubhashi et al. [14] obtained a $\log \Delta$ approximation algorithm for general graphs that finds weakly connected dominating set. Papers [2, 6, 32] all give a constant factor approximation for CDS in unit-disk graphs with the differences in algorithms complexity and approximation ratio. Cardei et al. [6] give an algorithm with factor 8 and message complexity $n * \Delta$, where Δ is the maximum degree of unit disk graph. Thus, in worst case they [6] have a quadratic message complexity. Wan et al. [32] present factor 8 algorithm with $O(n \log n)$ message complexity. However, they [32] used the broadcast model of transmission that counts only the broadcasted messages without received messages. In fact, the total amount of received messages can be as large as $O(n^2)$. An algorithm given by Alzoubi et al. [2] has linear time and message complexity, but the obtained approximation factor deteriorates to 192. Cheng et al. [11] gave a PTAS for finding CDS in unit disk graphs. Their algorithm has an approximation ratio of $(1 + 1/s)$ with running time $n^{O((s \log s)^2)}$. The hidden constant in the running time here is huge and thus the algorithm is impractical. We note that all these protocols do not ensure complete 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 sufficient accuracy. Thus, 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 a great deal of attention in recent time. Gupta et al. [22] design centralized and distributed approximation algorithms for the connected sensor cover problem that achieve a $O(\log n)$ approximation with respect to the size of an optimal sensor cover, where n is the network size. Zhang and Hou [36] prove that if the radio 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. Wang et al. [33] design a Coverage Configuration Protocol (CCP) that can provide different degrees of connected coverage and also present a geometric analysis of the relationship between coverage and connectivity.

Since a sensor network is usually deployed to perform surveillance and monitoring tasks, another definition of coverage is calculating a path with specific properties through a sensor network. The maximal support problem is to find a path that minimizes the maximal distance of a point on the path to the closest sensor and the maximal breach problem is to find a path which maximizes the minimal distance of a point on the path to the closest sensor. In [28] they derive centralized algorithms for finding a maximal breach path and a maximal support path in a sensor network using Voronoi diagram and Delaunay triangulation techniques. Distributed algorithms for both problems are given in [25]. In [17] they present constant-approximation algorithms for dynamic maintenance of the best-case and the worst-case coverage distances. They also improve the running time of the shortest maximal support path algorithms due to [28, 25].

When a sensor network already functions, the locations of the sensors are fixed. However, intelligent sensor placement algorithms can be applied prior to the deployment of the sensor network in order to optimize the underlying architecture. In [13] they present algorithms for finding efficient placement of sensors that guarantee probabilistic coverage of the grid points. Sensor placement for surveillance and target location is considered in [8]. They consider the problems of achieving the desired coverage while minimizing the cost (sensors may have different ranges and costs) and covering every grid point by a unique subset of the sensors.

Our results. We study the problem of providing coverage and connectivity in a unified framework. First we consider fixed sensor locations. We present the sector cover algorithm, which provides complete coverage and has an approximation factor of $\Theta(\log m)$, where m is the maximal number of neighbors of a single sensor in the corresponding unit disk graph. Then we derive the grid placement and the dominating cover algorithms, which achieve approximation factors of 6π and 32, respectively. However, these algorithms provide only partial coverage. We present an algorithm with sub-quadratic running time for the minimum connected dominating set problem in unit disk graphs that achieves an approximation factor better than 6. This improves upon the previous best known approximation factor of 8 due to [6, 32]. Thereafter we consider variable sensor locations. We develop the discrete grid algorithm that has a constant approximation guarantee and finds partial coverage. The accuracy of the coverage is a parameter, i.e., the higher the running time the better the accuracy. We also consider a model in which sensors must be placed only at the grid points. Finally, we propose algorithms for connecting a covering set.

The rest of the paper is organized as follows. Our model is described in Section 2. In Section 3 and Section 4 we consider fixed and variable sensor locations, respectively. Section 5 studies how to connect a covering set. We conclude with Section 6.

2 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 may or may not be given in advance. The sensor s_i can monitor objects that are within a distance R_s^i from s_i. This area is called the *sensing region* of s_i and is denoted by A_i. We define a *sector* to be a maximal region that is formed by intersection of a number of sensing regions such that all points within the sector are covered by the same set of sensors.

The *communication graph* G of the network is the undirected graph in which nodes are sensors and there is an edge between two nodes if they can communicate with 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 $S' = \{s_{j_1}, \ldots, s_{j_m}\}$ of sensors such that $P = A_{j_1} \cup \ldots \cup A_{j_m}$ and the communication subgraph induced by S' is connected. An example of a connected cover is presented in Figure 1.

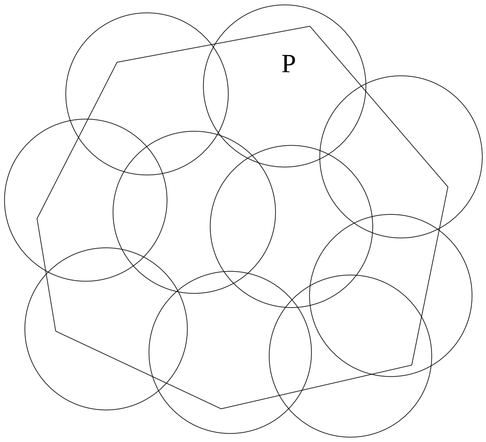

Figure 1: A connected cover example.

Definition 2.1 *Given a region of interest P, the* connected coverage problem *is to find a connected cover of P that uses a minimum number of sensors. We denote by OPT an optimal connected cover.*

Definition 2.2 *We say that an algorithm A has the approximation factor of c, if the size of the solution produced by A is at most $c \cdot |OPT|$ for any instance of the problem.*

We say that A finds a *partial coverage* if the set of the selected sensors does not cover P completely.

We make a few simplifying assumptions.

1. We assume that all sensors have the same sensing radius R_s.

2. We assume that two sensors can communicate with each other if the distance between them is at most R_c and $R_c \geq 2R_s$.

3. We assume that P is convex.

Theorem 2.1 ([33, 36]) *Under the above assumptions, complete coverage implies connectivity.*

In Section 5 we show how to replace assumption (3) by a more realistic assumption that $R_c = R_s$.

3 Fixed Sensor Locations

In this section we study the case in which the locations of the sensors are fixed. We derive the sector cover approximation algorithm, which provides complete coverage and has the approximation factor of $\ln n$ in the worst case. Then we present the grid placement and the dominating cover algorithms, which achieve constant approximation factors but provide only partial coverage. We also describe a new 6-approximation algorithm for the minimum connected dominating set problem in unit disk graphs.

3.1 Sector Cover Algorithm

In this section we develop the sector cover algorithm. We consider the sectors produced by the sensors as elements to be covered while each sensor represents a set. We apply the greedy set cover algorithm due to Johnson [18]. At each step, the algorithm selects the sensor that covers the maximal number of uncovered sectors.

In [3] they present an algorithm that determines in time $O(n^{1+\epsilon})$ whether a polygon is completely covered by a set of unit disks. We apply this algorithm to check whether a feasible solution exists, i.e., whether the region of interest P is completely covered by the available sensors.

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

Our algorithm is presented in Figure 3. The running time of the algorithm 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.

1. Apply the algorithm of [3] to check whether the sensors cover the region of interest P. If not, report failure.
2. Apply the greedy set cover algorithm to the sectors that intersect P. That is, until all sectors are covered, at each iteration select a sensor that covers the maximal number of uncovered sectors.

Figure 2: The sector cover algorithm.

The next theorem shows that the sector cover algorithm finds a full connected coverage.

Theorem 3.1 *The covering set found by the sector cover algorithm is connected and P is completely covered.*

Proof: Step 1 of the algorithm ensures that the union of the sensing regions covers P and in Step 2 all sensing regions that intersect P are covered. Thus, P is completely covered. According to Theorem 2.1, this set is connected. ∎

The following theorem establishes the approximation factor of the sector cover algorithm.

Theorem 3.2 *The approximation factor of the sector cover algorithm is at most $\log m$, where m is the maximal number of sectors covered by a single sensor.*

We also present a matching lower bound. The proof of the following theorem is omitted due to the lack of space.

Theorem 3.3 *The approximation factor of the sector cover algorithm is at least $\Omega(\log m)$.*

Since in most real sensor networks m is typically a constant, in fact the sector cover algorithm achieves a constant factor approximation in the average case. Observe that this algorithm can be easily extended to the case of non-uniform sensing radii and the case of obstacles without affecting the approximation factor.

3.2 Grid Placement Algorithm

In this section we present the grid placement algorithm. A grid is defined as a packed tiling of regular rectangles called cells. We assume that the sides for each of the cells are parallel to the x and y axes of the plane. In a nutshell, we place a grid with cell size of $R_s/\sqrt{2} \times R_s/\sqrt{2}$ over the region of interest P. A specific instance of grid is defined by its position. Then we choose exactly one sensor in each cell to be in the cover. Finally, we add extra sensors to make the covering set connected.

Observation 2 *The selection of the cell size implies that each sensor covers its cell completely and sensors in neighboring cells are able 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. In [5] they give an algorithm for solving the grid placement problem that minimizes the number of grid cells not containing any point with running time of $O(n \log n)$ for a set of n points. We use the algorithm of [5] to optimize the grid placement.

Our algorithm is presented in Figure 3. The running time of the algorithm is $O(n^3)$: Step 2 takes $O(n \log n)$ time, Step $4(a)$ takes $O(n^3)$ time and Step $4(b)$ takes $O(n^2)$ time.

1. Define a grid with cell size of $R_s/\sqrt{2} \times R_s/\sqrt{2}$ covering P.
2. Apply the algorithm of [5] to find the grid placement that minimizes the number of empty cells.
3. Select the sensor closest to the center in each non-empty cell and add it to the covering set (the *basic* cover).
4. Add extra sensors to the covering set to make it connected:
   (a) Create a weighted graph GC in which nodes are the connected components in the communication subgraph induced by the nodes from the basic cover. There is an edge between two super-nodes u and v in GC if there exists a path between them 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.
   (b) Apply Prim's minimum spanning tree (MST) algorithm on the graph GC.
   (c) Add to the covering set the nodes that lie on the shortest paths corresponding to the edges of the MST (the *extended* cover).

Figure 3: The grid placement algorithm.

The following theorem states that the grid placement algorithm is correct.

Theorem 3.4 *The covering set found by the grid placement algorithm is connected. The uncovered area of P is bounded by the union of the empty cells.*

Proof: We argue that if there exists a path between two super-nodes u and v in G, then there also exists a path between them in GC. If GC contains edge (u, v), we are done. Otherwise, by our construction, u can reach v in GC through another super-node w. Therefore, the algorithm returns a connected covering set.

By Observation 2, all non-empty cells are covered. Hence, the uncovered area of P is bounded by the union of the empty cells. ∎

We note that the grid placement algorithm will find almost complete coverage for dense instances of the problem. Next we derive the approximation factor of the grid placement algorithm.

Definition 3.1 *We say that a super-node of GC* covers *a cell if it includes a node from the basic cover that is located in this cell.*

Theorem 3.5 *The approximation factor of the grid placement algorithm is at most 6π.*

Proof: 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 the grid.

We will show that the number of sensors in the extended cover is at most $2(k-1)$. Clearly, the number of nodes in GC 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 GC is at most two. 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 from u to v must lie in a cell C not covered by the super-nodes u and v. Let w be the super-node covering C. We obtain that the node from C on the path between u and v is directly reachable from w and not directly reachable from u or v, which contradicts to our construction. [1] Therefore,

$$l \leq 2(k-1),$$

which establishes the theorem. ∎

3.3 Connected Dominating Set

In this section we describe the dominating cover algorithm. The algorithm uses the connected dominating set (CDS) algorithm as a subroutine. First we present a new algorithm for the CDS problem that achieves an approximation factor of 6. Let GD be a disk graph in which each sensor corresponds to a disk with radius $R_s/4$. We will compute a connected dominating set in GD. The following observation will be useful to demonstrate the coverage property.

Observation 3 *If a disk d in GD intersects another disk d', then the sensor located in the center of d completely covers d'.*

Our idea of building CDS is formed from two steps. At the first step, we construct a minimal size independent set of the unit disk (communication) graph and at the second step, we connect the chosen nodes by inserting additional disks. In order to accomplish the first step, we always select a node D that has a maximal number (at most 5) of the independent neighbors and remove all D's neighbors by taking D into the current independent set. Then we convert our independent set to a connected dominating set by increasing its size by at most a factor of two. The formal description of the algorithm is given in Figure 4.

Theorem 3.6 *The CDS algorithm finds a connected dominating set in a set of n unit radius disks that is of factor 6 from an optimal size using $O(n^{7/4} + k\log k)$ time, where k is the number of edges in the underlying graph.*

Proof: Steps $1-5$ find a maximal independent set IS. Let $OPTC$ be an optimal CDS solution. We claim that $|IS| < 3|OPTC|$. In order to see this, we classify the nodes of the given unit disk graph into two types: *complete* and *incomplete*. Complete node is a node that has been chosen during Steps $1-5$ to be in IS and at the time it was selected no one of its neighbors has been removed yet. All other nodes from

[1] Note that in the construction of GC we can restrict our attention to paths of length at most two.

1. Let $IS = \emptyset$.
2. Choose a disk D with a largest number of independent neighbors.
3. $IS = IS \cup \{D\}$.
4. Remove all the neighbors of D.
5. Repeat steps $2-4$ until $S = \emptyset$.
6. Mark each disk in IS as black; all other disks as grey. Each black node represents a different component.
7. Choose a grey node U (if exists) with at least two black neighbor nodes in different components.
8. Recolor U to black and union U and all its black neighbor nodes into one component.
9. Repeat steps $7-8$ until either there is no such grey node or we have only one black component. If we left with one black component we are done. Otherwise, continue with step 10.
10. Build a connectivity graph G', such that the nodes of G' correspond to the remaining black components and each such node is connected (if possible) to each other node by a path with only two grey nodes. In order to define such path we check for each grey node v (that is connected to black node w) all of its grey neighbor nodes that are connected to other than w black nodes. If such nodes exist, say $v_1, v_2, \ldots, v_k$ (v_1 connects to black w_1, v_2 connects to black w_2 and so on) we put additional nodes to G' that correspond to $v, v_1, v_2, \ldots, v_k$ and undirected edges $(w, v), (v, v_1), (v, v_2), \ldots (v, v_k), (v_1, w_1), (v_2, w_2), \ldots (v_k, w_k)$.
11. Connect all the black components nodes of G' using a BFS traversal.

Figure 4: The CDS algorithm.

IS are incomplete. The question we ask is: how many complete and incomplete nodes correspond (are adjacent) to the nodes in the optimal solution? The trivial case is when complete node v matches some node from the optimal solution. Thus, we assume that v doesn't match any node from $OPTC$. Therefore, some node o from $OPTC$ should be a neighbor of v. Clearly, to each complete node corresponds at least one node from $OPTC$ and there is no node from $OPTC$ that corresponds to two (or more) complete nodes. The problem is that the same node from $OPTC$ can correspond to some complete node and several incomplete nodes.

Definition 3.2 *We define the* degree *of a node to be the maximal number of its independent neighbors.*

We consider different cases when the degree of a complete node is $2, 3, 4, 5$. If the degree is 2, then the node from $OPTC$ corresponds to one complete and one incomplete node. If the degree is 3, then the node $OPTC$ corresponds to one complete and two incomplete nodes. If the degree is 4, then the worst case happens when 5 nodes from $OPTC$ correspond to one complete and 13 incomplete nodes. Finally, if the degree is 5, then the worst case occurs when 6 nodes from $OPTC$ correspond to one complete and 16 incomplete nodes (other situations are geometrically non-realizable due to the properties of unit disk graphs). Henceforth, $|IS| < 3|OPTC|$.

When converting IS into connected dominating set, we observe that the number of the chosen grey nodes is less than or equal to the number of the original black nodes. The approximation factor of 6 follows.

Considering runtime of the algorithm, we assume that we are only given n disks (not the graph itself). In order to build a graph (which in naive fashion can take $O(n^2)$ time), we use the data structure for n disks given in Gupta et al. [21] that can be built in $O(n)$ time such that p disks that are intersected by a query disk can be reported in $O(n^{3/4} + p)$ time. We query this data structure n times for construction of the unit disk graph, spending $O(n^{3/4}+k)$ time. Next, in order to find a disk with the largest number of independent neighbors, we consider separately n circular arc graphs, where a disk plays a role of a circle and the disks intersecting it play a role of arcs. We use an algorithm with $O(m \log m)$ runtime [23, 24, 26], where m is the number of arcs, to compute a maximum independent set for each circular arc graph. To compute the total running time spent for these operations, we observe that each arc is involved in the computation only a constant number of times, since there is a constant number of independent 2-hop neighbors (at distance 2 in a unit disk graph) for each disk. Thus, it yields $O(k \log k)$ time. In order to construct a CDS, we start join black nodes by inserting additional grey nodes. This can be implemented as a disjoint-set data structure, where we use operations like make-set, find and union. The total time needed to accomplish Steps 6-8 is $O(n + k\alpha(k))$, where $\alpha(\cdot)$ is a very slow *inverse Ackermann's* function [30]. Finally, BFS takes $O(k)$ time in order to connect the remaining black components. ∎

The dominating cover algorithm is presented in Figure 5. The coverage property of the dominating cover algorithm follows by Observation 3.

1. Construct the disk graph GD in which all disks intersect P and have radius $R_s/4$.
2. Apply the CDS algorithm to calculate a connected dominating set in GD.

Figure 5: The dominating cover algorithm.

Theorem 3.7 *The covering set found by the dominating cover algorithm is connected. The uncovered area of P is bounded by the parts of P that is not covered by the same set of sensors having the sensing radius of $R_s/4$.*

The following theorem establishes the approximation factor of the dominating cover algorithm.

Theorem 3.8 *The approximation factor of the dominating cover algorithm algorithm is at most* 32.

Proof: The area covered by a single sensor from the IS computed by the CDS algorithm is $\pi R_s^2/16$. On the other hand, any sensor in OPT can cover the area of at most πR_s^2. The theorem follows since the size of the final CDS is at most twice that of IS. ∎

4 Variable Sensor Locations

In this section we consider the case in which we have to define the locations of the sensors. We develop the discrete grid algorithm that has a constant approximation factor and computes partial coverage. The running time of the discrete grid algorithm depends on the accuracy of the coverage. Then we study an extension of our model, where sensors must be placed only at the grid points.

4.1 Discrete Grid Algorithm

In this section we present the discrete grid algorithm, which is based on a discrete grid scheme. We divide the region of interest P into cells using a fine grid with cells of size $a \times a$, where $a \leq R_c/\sqrt{2}$. The vertices of the grid define a set of points. Then we cover these points with a minimal number of sensors. We use the PTAS of [19] for covering points with unit disks. Finally, we add extra sensors to connect the resulting set.

Observation 4 *The selection of the cell size implies that sensors covering adjacent vertices of the greed are able communicate with each other.*

The discrete grid algorithm appears in Figure 6. The running time of the algorithm is $O(k)$, where k is the number of cells. Next we analyze the performance of the discrete grid algorithm.

1. Define a grid with cell size of $a \times a$ covering P.
2. Apply the PTAS of [19] to find a coverage of the grid vertices (the *basic* cover).
3. Add extra sensors to the covering set to make it connected:
   (a) Create an unweighted graph GC in which nodes are the connected components in the communication subgraph induced by the nodes from the basic cover. There is an edge between every two super-nodes u and v in GC if they cover adjacent cells.
   (b) Calculate a spanning tree (ST) of the graph GC.
   (c) Add to the covering set a sensor that connects the endpoints of each ST edge (the *extended* cover).

Figure 6: The discrete grid algorithm.

Theorem 4.1 *The covering set found by the discrete grid algorithm is connected. The uncovered area of P is inversely proportional to the cell size.*

Proof: By Observation 4, the graph GC is connected. Thus, our algorithm returns a connected covering set. Note that all the grid vertices are covered. Thus, when the cell size tends to zero, we obtain complete coverage. Therefore, the uncovered area of P is inversely proportional to the cell size. ∎

Theorem 4.2 *The approximation factor of the discrete grid algorithm is at most $2(1+\epsilon)$.*

Proof: The number of nodes in the basic cover is at most $(1+\epsilon)|OPT|$ since an optimal solution has to cover all vertices of the grid. By our construction, the size of the extended cover is bounded by twice the size of the basic cover. ∎

4.2 Sensors Placement at Grid Points

We study the problem introduced in [13], but focus on deterministic (and not probabilistic) coverage. The model is as follows. We are given a $N \times N$ grid and the goal is to cover a subset of grid points using the minimum number of sensors. The restriction is that sensors must be placed only at the grid points. 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 $a \times a$. We present two constant-factor approximation algorithms for the model with and without obstacles. We improve upon the results of [13], where they give algorithms with running time $O(N^4)$ providing no worst-case performance guarantees.

Our first algorithm for the model *with obstacles* is based on greedy set cover. The greedy point cover algorithm is presented in Figure 7. The running time of the greedy point cover algorithm is $O(N^2 \log N)$.

Theorem 4.3 *The approximation factor of the greedy point cover algorithm is at most* $\log\left(4\pi R_s^2/a^2\right)$.

The theorem follows since each sensor covers at most $4\pi R_s^2/a^2$ points.

1. Consider N^2 sensors located at the grid points. Each sensor cover the points in its sensing region that are not obscured by the obstacles.
2. Apply the greedy set cover algorithm. That is, until all points of interest are covered, at each iteration select a sensor that covers the maximal number of uncovered points.

Figure 7: The greedy point cover algorithm.

The second algorithm for the model *without obstacles* appears in Figure 8. We use the PTAS of [19] to cover the points. Then we adjust the locations of the sensors to be at the grid points duplicating sensors if necessary. This algorithm has running time of $O(N^2)$.

Theorem 4.4 *The approximation factor of the PTAS point cover algorithm is at most* $4(1+\epsilon)$.

The theorem holds due to the fact that in Step 2 we replace each sensor by at most four sensors.

1. Apply the PTAS of [19] to find a coverage of the points of interest.
2. Replace each sensor that is located not at a grid point by a minimal set of sensors located at the neighboring grid points covering the same set of points.

Figure 8: The PTAS point cover algorithm.

We note that although the covering sets calculated by the greedy point cover and the PTAS point cover algorithms may be disconnected, one can connect them using the techniques developed in the next section.

5 Connectivity

In this section we show how to convert any *complete* covering set into a connected covering set under a realistic assumption that $R_c = R_s$. We present two algorithms based on MST and Steiner Tree techniques. Suppose that we are given a *basic* cover B calculated by an algorithm A.

Observation 5 *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 radius is R_s and if there is no such a sensor, the center of the disk is not covered by OPT. Observation 5 implies the following lemma.

Lemma 5.1 *Adding nodes in OPT makes the basic cover connected.*

The MST connection algorithm is presented in Figure 9. The next theorem derives the performance guarantee of the MST connection algorithm.

Theorem 5.2 *The size of the extended cover calculated by the MST connection algorithm is at most* $2(|B|-1)$.

Proof: According to Lemma 5.1, GC is connected. We argue that the weight of any edge in GC is at most two. Otherwise, A's coverage is incomplete because there exists a disk which is not directly reachable from any of the super-nodes and thus its center is not covered. The theorem follows since the number of edges in a spanning tree is $|B| - 1$. ∎

1. Create a weighted graph GC in which nodes are the connected components in the communication subgraph induced by the nodes in B. There is an edge between two super-nodes u and v in GC if there exists a path between them 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. Apply Prim's MST algorithm on the graph GC.

3. Add to the covering set the nodes that lie on the shortest paths corresponding to the edges of the MST (the *extended* cover).

Figure 9: The MST connection algorithm.

The Steiner Tree connection algorithm appears in Figure 10. We use the algorithm of [16] for the node-weighted steiner tree problem. The following theorem establishes the performance guarantee of the Steiner Tree connection algorithm.

Theorem 5.3 *The size of the extended cover calculated by the Steiner Tree connection algorithm is at most* $(\ln |B|) \cdot |OPT|$.

Proof: By Lemma 5.1, the cost of an optimal steiner tree is at most $|OPT|$. The approximation factor follows by Theorem 5.1 and Theorem 5.2 [16]. ∎

1. Create an unweighted graph GC in which nodes are the connected components in the communication subgraph induced by the nodes in B and the other nodes of G not included into B.

2. Apply the algorithm of [16] for the node-weighted steiner tree with unit weights on the graph GC, where nodes in B represent the terminal nodes.

3. Add to the covering set the steiner nodes (the *extended* cover).

Figure 10: The Steiner Tree connection algorithm.

Remark 1 *We note that even if A provides only partial coverage, the algorithms will still return a connected set as long as OPT provides complete coverage.*

6 Concluding Remarks

In this paper we investigate the problem of maintaining coverage and connectivity by keeping a minimal number of sensor nodes in active mode in wireless sensor networks. We present centralized algorithms with provable worst-case constant approximation factors for the fixed as well as the variable sensor locations. We also propose an improved approximation algorithm for the minimum connected dominating set problem in unit disk graphs. In addition, we derive algorithms for connecting a covering set. An interesting future research direction is to design distributed versions of our algorithms.

References

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

[2] K. Alzoubi, P.-J. Wan and O. Frieder, “Message-optimal connected dominating sets in mobile ad hoc networks”, *Proc. of ACM Sympos. on Mobile ad hoc networking and computing*, 2002.

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

[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] P. Bose, A. Maheshwari, P Morin, and J. Morrison, ”The grid placement problem,” *Proceedings of the 7th Annual Workshop on Algorithms and Data Structures*, volume 2125 of Lecture Notes in Computer Science, pp. 180–191, 2001.

[6] M. Cardei, Xiayoan Cheng, Xiuzhen Cheng and D. Du, “Connected domination in multihop ad hoc wireless networks”, manuscript.

[7] A. Cerpa and D. Estrin, ”ASCENT: Adaptive Self-Configuring Sensor Networks Topologies,” *International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002)*, New York, NY, USA, June 2002.

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

[9] 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.

[10] Y. Chen and A. Liestman, “Approximating minimum size weakly connected dominating sets for clustering mobile ad hoc networks”, *MOBIHOC 2002*.

[11] X. Cheng, X. Huang, D. Li and D.-Z. Du, Polynomial-time approximation scheme for minimum connected dominating set in ad hoc wireless networks, *Networks*, to appear.

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

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

[14] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan and A. Srinivasan, “Fast Distributed Algorithms for (weakly) Connected Dominating Sets and Linear-Size Skeletons”, *ACM-SIAM Sympos. on Discrete Algorithms*, 2003.

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

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

[17] 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, to appear.

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

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

[20] S. Guha and S. Khuller, “Approximation algorithms for connected dominating sets”, *Algorithmica*, 20(4), pp. 374–387, 1998.

[21] P. Gupta, R. Janardan and M. Smid. “Algorithms for some intersection searching problems involving circular objects”, *International Journal of Mathematical Algorithms*, (1), pp. 35–52, 1999.

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

[23] W.-L. Hsu and K.-H. Tsai, “Linear time algorithms on circular-arc graphs”, *Information Processing Letters* 40 (1991), pp. 123–129.

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

[25] X. Li, P. Wan, and O. Frieder, “Coverage in wireless ad-hoc sensor networks,” *In ICC 2002*, New York City, 2002.

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

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

[28] S. Meguerdichian, F. Koushanfar, M. Potkonjak and M. B. Srivastava, "Coverage Problems in Wireless Ad-hoc Sensor Networks," *Proc. IEEE Infocom 2001*.

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

[30] R.E. Tarjan and J. van Leuween, "Worst-case analysis of set union algorithms", *Journal of ACM*, 31(2), pp. 245–281, 1984.

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

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

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

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

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

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