Selection and Orientation of Directional Sensors for Coverage Maximization

Giordano Fusco and Himanshu Gupta

Computer Science Department

Stony Brook University

Stony Brook, NY 11790

Email: {fusco,hgupta}@cs.sunysb.edu

Abstract—**Sensor nodes may be equipped with a "directional" sensing device (such as a camera) which senses a physical phenomenon in a certain direction depending on the chosen orientation. In this article, we address the problem of selection and orientation of such directional sensors with the objective of maximizing coverage area. Prior works on sensor coverage have largely focused on coverage with sensors that are associated with a *unique* sensing region. In contrast, directional sensors have multiple sensing regions associated with them, and the orientation of the sensor determines the actual sensing region. Thus, the coverage problems in the context of directional sensors entails selection as well as orientation of sensors needed to activate in order to maximize/ensure coverage.**

In this article, we address the problem of selecting a minimum number of sensors and assigning orientations such that the given area (or set of target points) is k-covered (i.e., each point is covered k times). The above problem is NP-complete, and even NP-hard to approximate. Thus, we design a simple greedy algorithm that delivers a solution that k-covers at least half of the target points using at most $M \log(k|C|)$ sensors, where $|C|$ is the maximum number of target points covered by a sensor and M is the minimum number of sensor required to k-cover all the given points. The above result holds for almost arbitrary sensing regions. We design a distributed implementation of the above algorithm, and study its performance through simulations. In addition to the above problem, we also look at other related coverage problems in the context of directional sensors, and design similar approximation algorithms for them.

I. Introduction

Coverage problems have been extensively studied in the context of sensor networks (see for example [1], [2], [3], [4]). The objective of sensor coverage problems is to minimize the number of active sensors, to conserve energy usage, while ensuring that the required region is sufficiently monitored by the active sensors. Most of the prior work has addressed the sensor coverage problem for single-orientation sensors, wherein each sensor is associated with a *unique* sensing region which is typically modeled as a uniform disk centered at the sensor's position. In this paper, we consider the sensor coverage problem in the context of "directional" sensors, such as cameras, whose sensing region depends upon the assigned orientation. Thus, the sensor coverage problem for directional sensors entails selection as well as orientation of active sensors to guarantee coverage.

In this article, we address the problem of selecting and orienting a minimum number of directional sensors to guarantee k-coverage of a given area or a set of target points (a point is k-covered if it is covered k times). The above problem is trivially NP-complete, and we show that it is actually NP-hard even to approximate it within any bounded factor. Thus, we design a simple greedy algorithm that has the following performance guarantee. If there is a solution of size M sensors that k-covers all the given target points, then our greedy algorithm delivers a solution that k-covers at least half of the target points (on average) using at most $M \log(k|C|)$ sensors, where $|C|$ is the maximum number of target points covered by a sensor. We also give a distributed implementation of the above greedy algorithm. In our experiments over dense random networks and target points, the greedy algorithm performs very well and selects only about 25% more sensors than the theoretical minimum.

In addition to the above problem of selection and orientation of directional sensors, we also present approximation algorithms for the following related problems on directional sensors: (i) Orient *all* the given sensors in order to *maximize* coverage, (ii) *Place* and orient a minimum number of sensors in order to cover the given area, (iii) Place and orient the *given* number of sensors to maximize the area covered.

Paper Organization. The rest of the paper is organized as follow. Section II contains detailed discussion about related work. The k-coverage problem is introduced in Section III. The greedy algorithm is presented in Section IV. We discuss related problems in Section V, and our experimental results in section VI.

II. **Related work**

In the recent years, there has been a lot of research done [2], [3], [5], [1] to address the coverage problem in sensor networks. In particular, Slijepcevic and Potkonjakthe [3] design a centralized heuristic to select mutually exclusive sensor covers that independently cover the network region. In [2], Charkrabarty et al. investigate linear programming techniques to optimally place a set of sensors on a sensor field for a complete coverage of the field. In [6], Shakkottai et al. consider an unreliable sensor network, and derive necessary and sufficient conditions for the coverage of the region and connectivity of the network with high probability. Recently, Hefeeda and Bagheri [7] extended the well-known ε-nets technique to solve the problem of k-covering the given sensor locations. In one of our prior works [1], we designed a greedy approximation algorithm that delivers a connected sensor-cover within a logarithmic factor of the optimal solution; this work was later generalized to k-coverage in [8].

Two closely related problems to the sensor-coverage problem are the art-gallery and hitting-set problems. The art-gallery problem (see [9] for a survey) is to place a minimum number of guards in a polygon so that each point in the polygon is visible from at least one of the guards. Guards may be looked upon as cameras with infinite range (angular and distance). The hitting-set problem is a "dual" of the set-cover problem. In both set-cover and hitting-set problems, we are given sets and elements. While in set-cover the goal is to select the minimum number of sets to cover all elements/points, in hitting-set the goal is to select a subset of elements/points such that each set is hit. The classical result for set cover [10] gives a $O(\log |C|)$ approximation, where $|C|$ is the size of the largest set. Brönnimann and Goodrich [11] were the first to use the ε-net technique [12] to solve the hitting-set problem and hence the set-cover with an $O(\log M)$ approximation, where M is the size of the optimal solution.

All of the above works are for single-orientation sensors, wherein each sensor is associated with a unique sensing region, typically a uniform disk. In contrast, in this article, 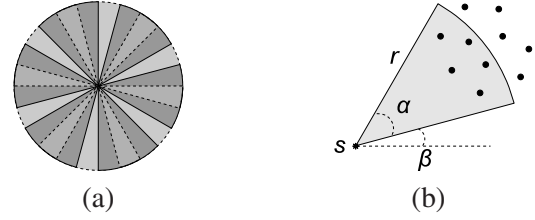we consider k-coverage of a region using directional sensors, wherein each sensor is associated with a set of sensing regions, among which one is chosen based on the selected orientation. To the best of our knowledge, there are only two works, [13] and [14], that have addressed coverage problems for directional sensors. In particular, for the specific case of cameras (with directional-cones as sensing regions), Ai and Abouzeid [13] propose a greedy heuristic (without any performance guarantee) to orient the given cameras in order to maximize 1-coverage (i.e., Problem 2 of Section V of this article). In another work, Hörster and Lienhart [14] address a number of camera-coverage problems, and design heuristics or exponential-time algorithms for each one of them. In this article, we address similar coverage problems for more general directional-sensors, and present approximation algorithms for them.

III. **Problem Formulation**

We start with formally defining a directional-sensor (d-sensor, in short) and the problem of selection and orientation of d-sensors to guarantee coverage of a given area. Informally, a d-sensor is a sensor associated with multiple sensing regions, out of which only one is active, depending on the orientation assigned to the d-sensor. The simplest example of a d-sensor is a camera – which senses a cone of a certain radius r and angle α, depending on the angle of orientation β. See Figure 1. *For the sake of simplicity and clarity*, we assume the d-sensors to be like cameras, i.e., the associated sensing regions to be uniform sized cones (as in [15] and [16]). *Our algorithms and their performance guarantees generalize to arbitrary sensing regions, as discussed later.*

Definition 1. (Directional Sensor (d-sensor); Orientation) We model a *d-sensor* as a sensor associated with multiple (actually infinite) sensing regions in the given 2D plane. Each associated sensing region is a 2D cone of uniform radius r and coverage angle α, centered at the d-sensor's position/location s. See Figure 1(a). The

Fig. 1. (a) Sensing regions associated with a d-sensor. (b) A selected selected sensing region corresponding to the orientation β, and given target points.
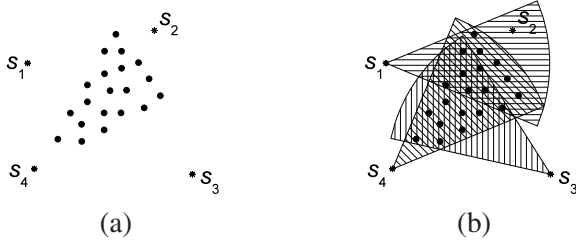
Fig. 2. Illustrating the SODkC problem. Suppose we are given 4 d-sensors of centers $s_1, \ldots, s_4$ and 20 points as in (a). Each d-sensor has associated many cone-shaped sensing regions as in Figure 1. The problem is to select the minimum number of d-sensor and orient them to k-cover all points. A possible solution for $k = 2$ is shown in (b), where 3 sensor suffice to 2-cover all points.

orientation of a d-sensor is an angle used to select **one** of its associated sensing region. For instance, in Figure 1(b), the orientation is β, and the shaded region is the selected sensing region. □

Definition 2. (Target Points) *Target points* are given points in the 2D plane that we wish to cover using the d-sensors. □

Problem 1. (Selecting and Orienting d-sensors for k-coverage (**SODkC**)) Given a set of d-sensors with fixed positions and a set of target points, select the minimum number of d-sensors and their assigned orientations, such that each target point is covered (is contained in the selected sensing region) by at least k of the selected d-sensors.

For simplicity, we have defined the above SODkC problem's objective as coverage of a set of given target *points*. However, as discussed later, our designed algorithms and techniques easily generalize to the problem of covering a given area.

Example 1. Suppose we are given 4 sensors and 20 points as in Figure 2(a), and we want to select the minimum number of d-sensor and orient them to 2-cover all points. In this particular example, 2 sensors are not enough to 2-cover all points. Instead, 3 sensors suffices, as shown in Figure 2(b). □

NP-Hardness of Approximating the SODkC Problem. We now show that it is NP-hard to approximate the above SODkC problem, by first showing the NP-completeness of the corresonding decision problem: Given a set of d-sensors with fixed locations, is there a way to orient the d-sensors so as to cover the given target points? This decision problem is NP-complete, as shown below using a reduction from 3-SAT.

Algorithm 1: Greedy Algorithm (GA) — Selecting and orienting d-sensors for k-coverage
1 **while** there are targets not k-covered yet **and** there are d-sensors not yet oriented
2   Select a d-sensor (that has not been selected yet) and an orientation pair that covers the most number of target points not yet k-covered;

Reducing 3-SAT to the Decision Problem of SODkC. Given an instance of 3-SAT, i.e., a disjunction of conjunctive 3-literal clauses, we create a target point for each clause, a d-sensor for each variable, and two sensing regions (called positive and negative) for each sensor. The positive sensing region covers all the target points corresponding to clauses that contain the positive literal of the variable, and the negative sensing region covers all the target points corresponding to clauses that contain the negative literal of the variable. It is easy to see that the above reduction is a valid Karp-reduction from 3-SAT to our decision problem. Thus, the decision problem of SODkC is NP-complete.

Now, if it were possible to approximate (within any bounded factor) the SODkC problem in polynomial time, then we can solve the decision problem by just using the approximation algorithm. Note that the approximate algorithm returns a bounded solution iff there is some way (an optimal solution) to orient the d-sensors to cover the target points. Thus, it is NP-hard to approximate the SODkC problem.

IV. **GA: Greedy Algorithm**

In this section, we present our Greedy Algorithm, for the SODkC problem. We present the performance guarantee of the algorithm, and discuss its generalizations.

Greedy Algorithm (GA). Greedy Algorithm works in iterations. At each iteration, it considers all d-sensors that have not been yet selected in previous iterations. For each such d-sensor, it considers all its possible orientations, and picks the (d-sensor, orientation) pair that covers the largest number of target points that have not been yet k-covered (by previously selected d-sensors). The algorithm terminates when all the target points are k-covered (i.e., covered by at least k selected d-sensors). See Algorithm 1. Note that in the above algorithm, the selection of a d-sensor and the assignment of its orientation is permanent (i.e., not altered in later iterations). Morever, not all d-sensors are selected; the
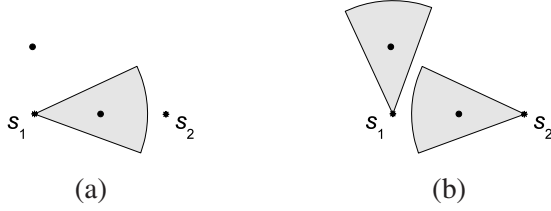
Fig. 3. Counterexample that GA may not find the optimal solution. GA might orient sensor s_1 towards the target point in the middle leaving the other point uncovered as in (a), while it is possible to cover both points as in (b).

d-sensors not selected by the algorithm are essentially kept inactive to save energy.

GA on the Running Example. Consider the example in Figure 2. GA selects and orients s_4 first, because it can cover the highest number of points (20 in this case). Then it selects and orients s_3 because it can cover 15 points for the 2nd time (note that s_2 could also cover 15 points, but ties are resolved arbitrarily). Finally, s_1 is selected to 2-cover the remaining 5 points.

Performance Guarantee of GA. Note that GA's solution may not k-cover all the given target points, even if there is some solution that does it. See Figure 3. The problem to determine whether there is a solution (i.e., a way to orient a set of given d-sensors) that k-covers the given target points is NP-complete as shown above. In the below theorem, we prove that if there is some solution that k-covers the set of given target points, then GA yields a solution which has a "benefit" of half the optimal and uses at most $O(\log |C|)$ times the optimal number of d-sensors.

We now define a concept of benefit of an arbitrary set of sensing regions, at a given stage of GA, which will be used in stating and proving the performance guarantee of GA.

Definition 3. (Benefit of a Set of Sensing Regions) Consider a given stage of GA, wherein GA has already selected and oriented some set of d-sensors. Informally, the *benefit of a sensing region* is the number of target points that it covers that are not yet k-covered by GA. Similarly, the *benefit of a set Q of sensing regions* is the "additional number of times" the set Q covers the uncovered (not yet k-covered by GA) target points. More formally, let G be the set of sensing regions already selected by the greedy algorithm. Also, let T' denote the set of yet uncovered (by GA) target points, i.e.,

$$T' = \{x \mid \text{number of sensing regions selected by GA} \text{ that cover } x \text{ is less than } k\}.$$

The benefit of Q is defined as

$$\sum_{x \in T'} \min \Big\{ \text{(number of sensing regions in } Q \text{ that cover } x),$$
$$k - \text{(number of sensing regions in } G \text{ that cover } x) \Big\}. \quad \square$$

Theorem 1. *Consider a given instance of SODkC problem. If there exists a solution that provides k-coverage to the given set of target points, then GA returns a solution that has a benefit of at least $k|T|/2$ and uses at most $O(\ln k|C|)$ times the number of d-sensors used by an optimal algorithm. Here, n is the total number of d-sensors, $|T|$ is the number of target points, and $|C|$ is the maximum number of target points covered by a sensing region. Also, GA can be implemented to run in $O(kn|T|^2 \log n|T|)$ time.*

Proof: Let n_g be the number of d-sensors selected (and oriented) by GA; let us number these d-sensors as $1, 2, \ldots, n_g$ in the order they were selected by GA. Also, let g_i be the benefit of the sensing region chosen in the ith step of GA.

Consider the stage when GA has selected (and oriented) $1, 2, \ldots, j$ d-sensors. In the next paragraph, we will prove the **claim** that at this stage (when GA has already selected and oriented 1 to j d-sensors), there *is* an orientation of some unselected d-sensor that covers at least $(k|T| - 2\sum_{i=1}^{j} g_i)/n_{opt}$ uncovered (not yet k-covered) target points, where n_{opt} is the number of d-sensors selected by an optimal algorithm. Since GA picks the best (d-sensor, orientation) pair in the following iteration, the above claim implies that

$$g_{j+1} \geq \frac{1}{n_{opt}} \left(k|T| - 2\sum_{i=1}^{j} g_i \right).$$

Now, we can show by induction that

$$\left(k|T| - 2\sum_{i=1}^{j} g_i \right) \leq k|T| \left(1 - \frac{2}{n_{opt}} \right)^{j}.$$

Thus, when $j = n_{opt}/2 \ln(k|T|/n_{opt})$, we have $(k|T| - 2\sum_{i=1}^{j} g_i) < n_{opt}$. Finally, observing that $n_{opt}|C| \geq k|T|$ and that GA will continue until $(k|T| - 2\sum_{i=1}^{j} g_i)$ is zero, we see that when GA terminates, it has a benefit of at least $k|T|/2$ and uses at most $O(\ln k|C|)$ d-sensors.

Proving the Claim. Consider the stage when GA has selected (and oriented) $1, 2, \ldots, j$ d-sensors. Let us consider the sensing regions selected by the optimal algorithm at this stage. Note that some of these sensing regions may belong to the 1 to j d-sensors already selected by GA, and may even be same as the one
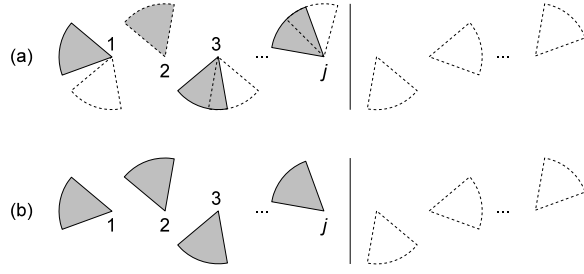
Fig. 4. The dotted cones are sensing regions picked by the optimal algorithm, and the shaded ones are those picked by GA. (a) The optimal sensing regions at the stage when GA has selected and oriented $1, 2, \ldots, j$ d-sensors. (b) The optimal sensing regions excluding the ones that belong to $1, 2, \ldots, j$ d-sensors.

selected by GA. See Figure 4(a). We make the following two observations about the benefit of sensing regions selected by the optimal algorithm.

- The benefit of the n_{opt} optimal sensing regions is at least $(k|T| - \sum_{i=1}^{j} g_i)$ at this stage of GA.
- The benefit of the optimal sensing regions excluding the ones that belong to the 1 to j d-sensors (which have been already selected by GA at this stage) is at least
$$(k|T| - \sum_{i=1}^{j} 2g_i).$$

Note that number of sensing regions constituting the above benefit is at most $(n\_opt - j)$. See Figure 4(b). The above observation is true because the benefit of an optimal sensing region o_i associated with an i d-sensor at this stage is at most the benefit of o_i at the stage when i d-sensor was selected by GA. The latter is at most g_i, the benefit of the sensing region chosen by GA in the ith iteration.

Finally, by the pigeon-hole principle, there must be at least one unselected d-sensor whose benefit at this stage is at least $(k|T| - 2\sum_{i=1}^{j} g_i)/(n_{opt} - j) > (k|T| - 2\sum_{i=1}^{j} g_i)/n_{opt}$.

Running Time. The algorithm can be implemented efficiently by keeping a priority queue of d-sensors' orientations based on the number of uncovered (not yet k-covered) targets that each d-sensor's orientation can cover. Note that the total number "relevant" orientations for a d-sensor is at most $|T|$, and thus, the size of the priority queue is at most $n|T|$. The loop is executed at most $k|T|$ times. In each iteration, extracting the best (d-sensor, orientation) pair from the queue requires $O(\log n|T|)$ time and updating the priorities of elements takes at most $O(n|T| \log n|T|)$ time (since there are at most $n|T|$ elements and each update takes at most $O(\log n|T|)$ time). ∎

Algorithm 2: Distributed GA (algorithm of each d-sensor) — Selecting and orienting d-sensors for k-coverage

1 $status = \texttt{unselected}$;
2 **while** there are owned target points that are not k-covered
3   pick one such owned target point t;
4   send a `benefit-enquiry` message to neighboring `unselected` d-sensors that cover t, asking their maximum benefit if selected and oriented to cover t;
5   **if** there are no answers, **return** `failure`;
6   **else** select the d-sensor with maximum benefit and send it a `selection-request`;
7 **while** (*true*)   /* receive `benefit-enquiry` */
8   **if** ($status == \texttt{unselected}$)
9   **for each** `benefit-enquiry`
10   determine the best orientation to cover the given t, and inform the requesting d-sender of the benefit;
11 **while** (*true*)   /* receive `selection-requests` */
12   **if** ($status == \texttt{unselected}$) **and** (there is at least one `selection-request`)
13   satisfy the request that will result in maximum benefit;
14          $status = \texttt{selected}$;
15   Inform the near-by d-sensors (that cover a common target point) of the selection and orientation;

Generalizations to Arbitrary Sensing Regions, and Coverage of an Area. The Greedy Algorithm (GA) easily extends (along with its performance guarantee) to d-sensors with sensing regions of arbitrary shape, as long as the the sensing regions are compact.[1] For such d-sensors with arbitrary sensing regions, the concept of orientation may be defined as some way of identifying the different sensing regions. Note that the proof of the above Theorem 1 is independent of the shapes of the sensing regions.

Similarly, GA can also be used to k-cover a given area, rather than a given set of target points (as required by the formulation of SODkC problem). Essentially, coverage of an area requires dividing the given area into "subregions" as in our previous work [1]; a subregion is defined as a set of points in the plane that are covered by the *same* set of sensing regions. The number of such subregions can be shown to be polynomial in the total number of cone-shaped sensing regions in the system.

---

[1] A region is compact if it is closed and connected. In particular, this is true if each region covers a finite number of points.

Based on the above, we define the benefit of a sensing-region as the number of number of uncovered subregions contained in the sensing-region. The Greedy Algorithm can then be used without any other modification, and the performance guarantees still hold.

Distributed GA. In a distributed environment, *conceptually*, each yet-uncovered target point increases its coverage by selecting and orienting a d-sensor that covers it and has the highest benefit at that stage. The above process continues until all target points are k-covered. To facilitate the above, a target point (or a subregion) is "owned" by the highest-ID d-sensor that can cover it using one of its sensing regions. Thus, each d-sensor ensures coverage of the target points owned by it, through selection and orientation of near-by d-sensors. See Algorithm 2, for a pseudo-code of the distributed algorithm.

V. **Related problems**

In this section, we address problems similar to our SODkC problem that were presented in [14], in the context of coverage by placing and/or orienting cameras. We show that our GA can be modified to yield approximation algorithms for each of these related problems.

Problem 2. (Orienting all d-sensors for Maximizing Coverage) Given a set of d-sensors with fixed positions and a set of target points, orient all d-sensors so as to cover the maximum number of target points.

The above problem is also addressed in [13], wherein the authors present a greedy heuristic (different than ours) without a provable performance guarantee. In contrast, we present below a greedy approach with a constant-factor approximation.

Greedy Approach for Problem 2. The above problem can be solved using greedy approach similar to the GA, with the only difference that we continue to select and orient d-sensors until all the d-sensors have been oriented. At each stage, we pick a (d-sensor, orientation) pair that has the maximum benefit at that stage. See Algorithm 3. The running time of the algorithm is same as GA, i.e., $O(kn|T|^2 \log n|T|)$, but in this case, we get constant-factor approximation as shown in the following theorem.

Theorem 2. *The greedy approach (as described above) yields a 0.43-approximate solution to Problem 2.*

Proof: Let M be the *number of target points* covered by the optimal solution. This proof proceeds in a

---

Algorithm 3: Orienting all d-sensors for max-coverage

1 **while** there are d-sensors not oriented yet
2   Select a d-sensor (that has not been selected yet) and an orientation pair that covers the most number of target points not yet covered;

---

similar way as the proof of Theorem 1, and yields the below equation.

$$g_{j+1} \geq \frac{1}{n} \left(M - 2 \sum_{i=1}^{j} g_i \right)$$

Above, g_{j+1} is the benefit of the sensing-region chosen in the jth step of the greedy approach. By induction, we can show that

$$\left(M - 2 \sum_i g_{j+1} \right) \leq M \left(1 - \frac{2}{n} \right)^j,$$

which yields

$$\frac{\sum_{i=1}^{n} g_i}{M} \geq \frac{1}{2} \left(1 - \left(1 - \frac{2}{n} \right)^n \right) \geq \frac{1}{2} \left(1 - \frac{1}{e^2} \right) = 0.43.$$
∎

Problem 3. (Placing and Orienting d-sensors for Min-Cost Coverage) Given a polygon P and a set of d-sensors along with their *costs* and associated sensing regions. Cover a given percentage p of the area of the polygon P, by placing and orienting a minimum cost subset of given d-sensors.

Problem 4. (Placing and Orienting d-sensors for Max-Coverage) Given a polygon P and a set of d-sensors along with their *costs* and associated sensing regions. Cover a maximum percentage of P's area, by placing and orienting a subset of given d-sensors whose cost is less than the given budget B.

Greedy Approaches for Problems 3 and 4. We use greedy approaches similar to GA to solve Problems 3 and 4. Essentially, at each stage, we pick the best (d-sensor, position, orientation) triplet that gives the highest ratio of *area-covered/cost*. See Algorithms 4 and 5. For a given d-sensor and orientation, we can find its best placement using the algorithm in [17]. The result in [17] is for placement of "360-degree guards," but their result can be easily extended for arbitrary closed and connected sensing-regions, provided that VC-dimension (defined below) remains bounded. The use of their approach allows us to finds a position for a given d-sensor and orientation that covers $(1 - \delta)$ times the optimal area possible in $O((n^2 l^2/\delta^4) \log^3(l/\delta))$ time, where l is the

Algorithm 4: Placing and orienting d-sensors for min-cost coverage
1 **while** P is not covered for at least p percent
2      $best\_d\_sensor = null$;
3   **for each** d-sensor type
4   **for each** discretized orientation
5   run the algorithm in [17] to find the best position for this d-sensor type and orientation;
6   **if** this combination gives a higher ratio of (area covered/d-sensor cost) than $best\_d\_sensor$
7   store this combination in $best\_d\_sensor$;
8   keep $best\_d\_sensor$;
9   remove the area covered by $best\_d\_sensor$ from the polygon to cover;

Algorithm 5: Placing and orienting d-sensors for max-coverage
1 **while** there is some budged available
2      $best\_d\_sensor = 0$;
3   **for each** d-sensor type whose cost is within the budged
4   **for each** discretized orientation
5   run the algorithm in [17] to find the best position for this d-sensor type and orientation;
6   **if** this combination gives a higher ratio of (area covered/d-sensor cost) than $best\_d\_sensor$
7   store this combination in $best\_d\_sensor$;
8   keep $best\_d\_sensor$;
9   reduce the budget by the cost of $best\_d\_sensor$;
10   remove the area covered by $best\_d\_sensor$ from the polygon to cover;

size of the given polygon. We start by stating the time complexity, and then we will prove the approximation factors of the above greedy approaches.

Theorem 3. *The above described greedy approaches for Problems 3 and 4 run in time $O(\tau n u R)$ time, where τ is the number of different d-sensors types, u is the number of discretized d-sensor's orientations, n is the total number of d-sensor, and R is the running time of the algorithm in [17]. In particular, $R = O((n^2 l^2/\delta^4) \log^3(l/\delta))$ where l is the number of edges of the polygon P and $\delta > 0$ is the parameter that affects the precision of the result.*

Before proving the approximation factors, we need to introduce the concept of Vapnik-Červonenkis (VC) dimension. The VC-dimension is defined in terms of set shattering, as follows.

Definition 4. (VC dimension) A set X is *shattered* by $\mathcal{C}$ if for each $Y \subseteq X$, there exists a set $S \in \mathcal{C}$ such that $X \cap S = Y$. The *VC dimension* is the cardinality of the largest set that can be shattered by $\mathcal{C}$. □

In our case, the VC dimension of the set of d-sensor's sensing regions is at most 23, as given by Valtr theorem [18]:

Theorem 4. *If $X \subset \mathbb{R}^2$ is compact and simply connected, then VC-dimension of the system $V(X) = \{V(x) \mid x \in X\}$ is at most 23.*

We now prove the approximation factors of Algorithms 4 and 5.

Theorem 5. *Consider an instance of Problem 3. If there exists a solution that covers p percentage of the polygon area, then our greedy approach for Problem 3 gives a solution that covers at least $p/2$ percentage of the given polygon using a cost of at most $O(1/(1-\delta) \cdot \ln \frac{p}{100}|C|)$ times the optimal cost. Here, $|C|$ is the maximum number of target points covered by a sensing regions and δ is the precision parameter.*

Proof: Let denote with $|P|$ the area of the input polygon P. This proof proceeds in a similar way as the proof of Theorem 1, and yields the following equation.

$$g_{j+1} \geq \frac{1-\delta}{n_{opt}} \left(\frac{p}{100}|P| - 2\sum_{i=1}^{j} g_i \right)$$

where g_{j+1} is the benefit of the sensing-region chosen in the jth step of the greedy approach, and n_{opt} is the number of d-sensors selected by an optimal algorithm.. Note that the $(1 - \delta)$ factor is given by the use of the algorithm in [17]. Also note that after placing each d-sensor, the area that it covers should be removed from the polygon. The way to handle this situation is discussed in paragraph 2.4 of [17], and it applies to our case because the VC dimension remains 23 at each iteration of the algorithm.

By induction we can show that

$$\left(\frac{p}{100}|P| - 2\sum_{i=1}^{j} g_i \right) \leq \frac{p}{100}|P| \left(1 - 2\frac{1-\delta}{n_{opt}} \right)^j.$$
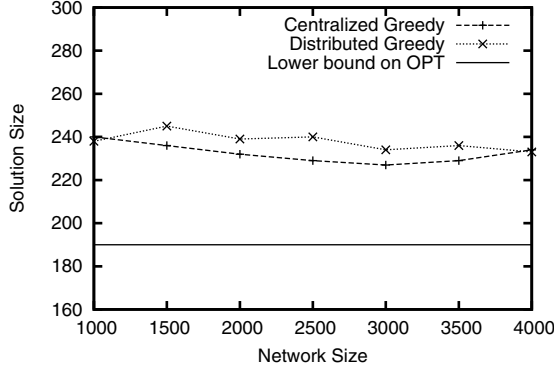
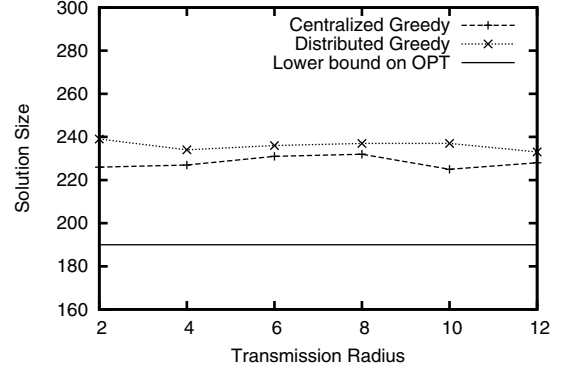Fig. 5. Solution size of 4-cover varying network size. Communication radius = 4 units.

Fig. 6. Solution size of 4-cover varying communication radius. Sensor network size = 3000 sensors.

Thus, when $j = n_{opt}/(2(1 - \delta)) \ln(\frac{p}{100}|P|/n_{opt})$, we have $(\frac{p}{100}|P| - 2(1-\delta) \sum_{i=1}^{j} g_i) < n_{opt}$. Finally, observing that $n_{opt}|C| \geq |P|$ and that GA will continue until $(\frac{p}{100}|P| - 2(1-\delta) \sum_{i=1}^{j} g_i)$ is zero, we see that when GA terminates, it has a benefit of at least $\frac{p}{100}|P|/2$ and uses at most $O(1/(1-\delta) \cdot \ln \frac{p}{100}|C|)$ d-sensors. ∎

Theorem 6. *Our greedy approach for the Problem 4 gives a solution which covers at least $(1 - e^{-2(1-\delta)})/2$ times the area covered by an optimal solution, for a given parameter $\delta > 0$.*

Proof: Let M be the *number of target points* covered by the optimal solution. This proof proceeds in a similar way as the proof of Theorem 1, and yields the following equation.

$$g_{j+1} \geq \frac{1 - \delta}{n}\left(M - 2\sum_{i=1}^{j} g_i\right)$$

where, g_{j+1} is the benefit of the sensing-region chosen in the jth step of the greedy approach. Note that the $(1 - \delta)$ factor comes form the use of the approximation method of [17]. By induction, we can show that

$$g_j \geq \frac{1 - \delta}{n} \cdot M \cdot \left(1 - 2\frac{1 - \delta}{n}\right)^{j-1}$$

Summing all equations we get

$$\sum_{j=1}^{n} g_j \geq \frac{1 - \delta}{n} \cdot M \cdot \sum_{j=1}^{n}\left(1 - 2\frac{1 - \delta}{n}\right)^{j-1}$$

which gives

$$\frac{G}{M} \geq \frac{1}{2}\left(1 - \left(1 - 2\frac{1 - \delta}{n}\right)^{n}\right) \geq \frac{1}{2}\left(1 - \frac{1}{e^{2(1-\delta)}}\right)$$

where $G = \sum_{j=1}^{n} g_j$ is the total benefit of the greedy algorithm. ∎

VI. Experimental results

In this section we present experimental results about the execution of GA and Distributed GA. The setup is similar to the one of [8]. Sensors are placed randomly in a square of 40×40 units. The visibility radius is set to 8, the d-sensor's cone spans an angle of 60 degrees, and we discretize the orientations by shifts of 30 degrees. In all our experiments, k is set to 4. Note that each d-sensor's cone has an area of $8^2\pi/6$, so to 4-cover the square we need at least $4 \cdot 40^2 \cdot 6/(8^2\pi) \geq 190$ d-sensors.

Communication in Distributed Algorithms. In distributed algorithms, d-sensors need to communicate with near-by d-sensors (those that cover a common target point). To reach a near-by d-sensor, a sensor broadcasts a message up to a distance ℓ. A safe choice for ℓ is the *link radius*, which is the maximum communication distance between any two sensors whose sensing regions intersect. We use the same methodology of [1] and [8] to compute the link radius ℓ of a sensor network. In particular, we define dense networks as networks with more than r/t sensors within a distance $2r$, where r and t are the *sensing* and *transmission* radii respectively. For a 40×40 area, a dense network should have at least $(80/t)^2$ sensors, and this is the case of all our experiments. For dense networks the link radius is estimated to be $\ell = (2r/t + 1)$.

Experiments. In all our experiment, both GA and distributed GA were able to attain k-coverage (even if theoretically it is not guaranteed).

Figure 5 shows the solution size for densities that vary from 1000 to 4000, and Figure 6 shows the solution size for different communication radii. Note that 1000 sensors are barely enough to obtain 4-coverage. As the plot shows, the solution size of the greedy algorithm
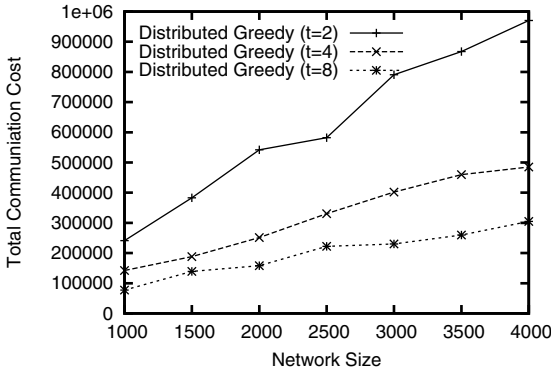
Fig. 7.   Communication cost of 4-cover varying the network size.

does not increase as the network size increases. This implies that the solution obtained for $n = 1000$ is already quite close to the optimal one. Also, note that the greedy solution is only 25% higher than the theoretical lower bound on the optimal solution Moreover, the solution of Distributed GA is very close to the one of Centralized GA. This means that the method used to distribute the algorithm does not compromise much the accuracy of the solution.

Figure 7 shows how the communication cost varies for different network sizes.

VII.   **Conclusions**

In this paper, we studied the problem of selecting and orienting a minimum number of direction sensor for k-coverage problem. The problem is NP-hard to approximate; however, we designed a greedy algorithm with certain performance guarantees. We designed a distributed version, and through experiments showed that the distributed implementation does not compromise much on the performance of the solution. We also addressed three other related problems that arise in the context of coverage using direction sensors, and analyze the performance of appropriate greedy approximation schemes for them.

References

[1] H. Gupta, Z. Zhou, S. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," *ACM/IEEE Transactions on Networking (TON)*, vol. 14, no. 1, 2006.

[2] K. Charkrabarty, S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transaction on Computers*, 2002.

[3] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor ad-hoc networks," in *Proceedings of the International Conveference of Communication (ICC)*, 2001.

[4] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *Proceedings of the International Conference on Mobile Computing and Networks (MobiCom)*, 2001.

[5] F. Ye, G. Zhong, S. Lu, and L. Zhang, "Peas: A robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the International Conference on Distributed Computing Systems*, 2003.

[6] S. Shakkottai, R. Srikant, and N. Shroff, "Unreliable sensor grids: Coverage, connectivity and diameter," in *Proceedings of the IEEE INFOCOM*, 2003.

[7] M. Hefeeda and M. Bagheri, "Randomized k-coverage algorithms for dense sensor networks," in *26th IEEE International Conference on Computer Communications (INFOCOM)*, 2007, pp. 2376–2380.

[8] Z. Zhou, S. R. Das, and H. Gupta, "Connected k-coverage problem in sensor networks," in *Proceedings of the 13th International Conference on Computer Communications and Networks (ICCCN)*, 2004, pp. 373–378.

[9] J. O'Rourke, *Art Gallery Theorems and Algorithms*, ser. International Series of Monographs on Computer Science.   New York, NY: Oxford University Press, 1987, vol. 3.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*, 2nd ed.   Cambridge Boston: MIT Press McGraw-Hill, 2001.

[11] H. Brönnimann and M. T. Goodrich, "Almost optimal set covers in finite vc-dimension," *Discrete & Computational Geometry*, vol. 14, no. 4, pp. 463–479, 1995.

[12] D. Haussler and E. Welzl, "Epsilon-nets and simplex range queries," *Discrete & Computational Geometry*, vol. 2, pp. 127–151, 1987.

[13] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, pp. 21–41, 2006.

[14] E. Hörster and R. Lienhart, "On the optimal placement of multiple visual sensors," in *Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks (VSSN)*, 2006, pp. 111–120.

[15] H. Ma and Y. Liu, "On coverage problems of directional sensor networks," in *Proceedings of the 1st International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, ser. LNCS, vol. 3794, 2005, pp. 721–731.

[16] J. Adriaens, S. Megerian, and M. Potkonjak, "Optimal worst-case coverage of directional field-of-view sensor networks," in *Proceedings of 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 2006, pp. 336–345.

[17] O. Cheong, A. Efrat, and S. Har-Peled, "On finding a guard that sees most and a shop that sells most," *Discrete and Computational Geometry*, vol. 37, no. 4, pp. 545–563, 2007.

[18] P. Valtr, "Guarding galleries where no point sees a small area," *Israel Journal of Mathematics*, vol. 104, pp. 1–16, 1998.