# $\varepsilon$-Net Approach to Sensor $k$-Coverage

Giordano Fusco and Himanshu Gupta

Computer Science Department
Stony Brook University
Stony Brook, NY 11790
{fusco,hgupta}@cs.sunysb.edu

**Abstract.** Wireless sensors rely on battery power, and in many application it is difficult or prohibitive to replace them. Hence, in order to prolongate the system's lifetime, some sensors can be kept inactive while others perform all the tasks. In this paper, we study the $k$-coverage problem of activating the minimum number of sensors to ensure that every point in the area is covered by at least $k$ sensors. This ensures higher fault tolerance, robustness, and improves many operations, among which position detection and intrusion detection.

The $k$-coverage problem is trivially NP-complete, and hence we can only provide approximation algorithms. In this paper, we present an algorithm based on an extension of the classical $\varepsilon$-net technique. This method gives a $O(\log M)$-approximation, where $M$ is the number of sensors in an optimal solution. We do not make any particular assumption on the shape of the areas covered by each sensor, besides that they must be closed, connected and without holes.

## 1 Introduction

Coverage problems have been extensively studied in the context of sensor networks (see for example [1–4]). 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. In an over-deployed network we can also seek $k$-coverage, in which every point in the area is covered by at least $k$ sensors. This ensures higher fault tolerance, robustness, and improves many operations, among which position detection and intrusion detection.

The $k$-coverage problem is trivially NP-complete, and hence we focus on designing approximation algorithms. In this paper, we extend the well-known $\varepsilon$-net technique to our problem, and present an $O(\log M)$-factor approximation algorithm, where $M$ is the size of the optimal solution. The classical greedy algorithm for set cover [5], when applied to $k$-coverage, delivers a $O(k \log n)$-approximation solution, where $n$ is the number of *target points* to be covered. Our approximation algorithm is an improvement over the greedy algorithm, since our approximation factor of $O(\log M)$ is independent of $k$ and of the number of target points.

Instead of solving the sensor's $k$-coverage problem directly, we consider a dual problem, the $k$-hitting set. In the $k$-hitting set problem, we are given sets and

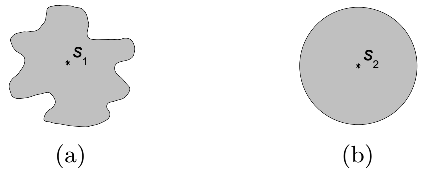


**Fig. 1.** Sensing regions associated with a sensor: (a) a general shape that is closed, connected, and without holes, and (b) a disk.

points, and we look for the minimum number of points that "hit" each set at least $k$ times (a set is hit by a point if it contains it). Brönnimann and Goodrich were the first [6] to solve the hitting set using the $\varepsilon$-net technique [7]. In this paper, we introduce a generalization of $\varepsilon$-nets, which we call $(k, \varepsilon)$-nets. Using $(k, \varepsilon)$-nets with the Brönnimann and Goodrich algorithm's [6], we can solve the $k$-hitting set, and hence the sensor's $k$-coverage problem. Our main contribution is a way of constructing $(k, \varepsilon)$-nets by random sampling. A recent Infocom paper [8] uses $\varepsilon$-nets to solve the $k$-coverage problem. However we believe that their result is fundamentally flawed (see Section 2.1 for more details). So, to the best of our knowledge, we are the first to give a correct extension of $\varepsilon$-nets for the $k$-coverage problem.

## 2 Problem Formulation and Related Work

We start by defining the sensing region and then we will define the $k$-coverage problem with sensors. In the literature, sensing regions have been often modeled as disks. In this paper, we consider sensing regions of general shape, because this reflects a more realistic scenario.

**Definition 1 (Sensing Region).** *The sensing region of a sensor is the area "covered" by a sensor. Sensing regions can have any shape that is closed, connected, and without holes, as in Fig. 1(a). Often, sensing regions are modeled as disks as in Fig. 1(b), but we consider more general shapes.*

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

**$k$-SC Problem.** Given a set of sensors with fixed positions and a set of target points, select the minimum number of sensors, such that each target point is covered (is contained in the selected sensing region) by at least $k$ of the selected sensors.

For simplicity, we have defined the above $k$-SC problem's objective as coverage of a set of given target *points*. However, as discussed later, our 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 Fig. 2(a), and we want to select the minimum number of sensors 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 Fig. 2(b).

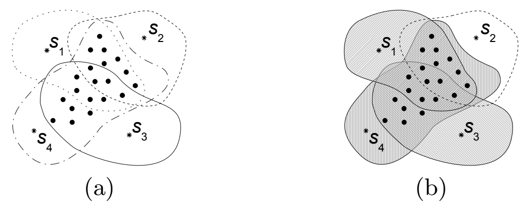


**Fig. 2.** Illustrating $k$-SC problem. Suppose we are given 4 sensors of centers $s_1, \ldots, s_4$ and 20 points as in (a). The problem is to select the minimum number of sensors to $k$-cover all points. A possible solution for $k = 2$ is shown in (b), where 3 sensor suffice to 2-cover all points.

### 2.1 Related Work

In recent years, there has been a lot of research done [2, 3, 9, 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 [10], 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. 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 [11].

Recently, Hefeeda and Bagheri [8] used the well-known $\varepsilon$-net technique to solve the problem of $k$-covering the sensor's *locations*. However, we strongly believe that their result is fundamentally flawed.[1] In this article we present a correct extension of the $\varepsilon$-net technique for $k$-coverage problem in sensor networks.

Two closely related problems to the sensor-coverage problem are set cover and hitting set problems. The area covered by a sensor can be thought as a set, which contains the points covered by that sensor. 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 [5] gives a $O(\log n)$-approximation algorithm, where $n$ is the number of *target points* to be covered. The same greedy algorithm also delivers a $O(k \log n)$-approximation solution for the $k$-SC problem. In contrast, the result in this article yields an $O(\log M)$-approximation algorithm for the $k$-SC problem, where $M$ is the optimal size (i.e., minimum number of *sensors* needed to provide

[1] Essentially, they select a set of *subsets* of size $k$ (called $k$-flowers) represented by the center of their locations. However, their result is based on the following incorrect claim that if the centers of a set of $k$-flowers 1-covers a set of points $N$, then the set of sensors associated with the $k$-flowers will $k$-cover $N$. In addition, in their analysis, they implicitly assume that an optimal solution can be represented as a disjoint union of $k$-flowers, which is incorrect.

k-coverage of the given target points). Note that our approximation factor is independent of $k$ and of the number of target points.

Brönnimann and Goodrich [6] were the first to use the $\varepsilon$-net technique [7] 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. In this article, we extend their $\varepsilon$-net technique to $k$-coverage. It is interesting to observe that our extension is independent of $k$ and it gives a $O(\log M)$-approximation also for $k$-coverage. For the particular case of 1-coverage with disks, it is possible to build "small" $\varepsilon$-nets using the method of Matoušek, Seidel and Welzl [12], and obtain a constant-factor approximation for the 1-hitting set problem. Their method [12] can be easily extended to $k$-hitting set, and this would give a constant-factor approximation for the $k$-SC problem when the sensing regions are disks. However, in this paper we focus on sensing regions of arbitrary shapes and sizes, as long as they are closed, connected, and without holes.

Another related problem is the art gallery problem (see [13] for a survey) which 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 sensors with infinite range. However, in this paper, we focus on *selecting* already deployed sensor.

## 3 The $\varepsilon$-Net Based Approach

In this section, we present an algorithm based on the classical $\varepsilon$-net technique, to solve the $k$-coverage problem. The classical $\varepsilon$-net technique is used to solve the hitting set problem, which is the dual of the set cover problem. The $k$-SC problem is essentially a generalization of the set cover problem – thus, we will extend the $\varepsilon$-net technique to solve the corresponding generalization of the hitting set problem.

### 3.1 Hitting Set Problem and the $\varepsilon$-Net Technique

We start by describing the use of classical $\varepsilon$-net technique to solve the traditional hitting set problem. We begin with a couple of formal definitions.

**Set Cover (SC); Hitting Set (HS).** Given a set of points $X$ and a collection of sets $\mathcal{C}$, the *set cover* (SC) problem is to select the minimum number of sets from $\mathcal{C}$ whose union contains (covers) all points in $X$. The *hitting set* (HS) problem is to select the minimum number of points from $X$ such that all sets in $\mathcal{C}$ are "hit" (a set is considered hit, if one of its points has been selected).

Note that HS is a *dual* of SC, and hence solving HS is sufficient to solve SC.

We now define $\varepsilon$-nets. Intuitively, an $\varepsilon$-net is a set of points that hits all *large* sets (but may not hit the smaller ones). For the overall scheme, we will assign weights to points, and use a generalized concept of weighted $\varepsilon$-nets that must hit all large-weighted sets.

**Definition 3 ($\varepsilon$-Net; Weighted $\varepsilon$-Net).** *Given a set system $(X, \mathcal{C})$, where $X$ is a set of points and $\mathcal{C}$ is a collection of sets, a subset $H \in X$ is an* $\varepsilon$-net *if for every set $S$ in $\mathcal{C}$ s.t. $|S| \geq \varepsilon|X|$, we have that $H \cap S \neq \emptyset$.*

*Given a set system* $(X, \mathcal{C})$*, and a weight function* $w : X \to Z^+$*, define* $w(S) = \sum_{x \in S} w(x)$ *for* $S \subseteq X$*. A subset* $H \subseteq X$ *is a* weighted $\varepsilon$-net *for* $(X, \mathcal{C}, w)$ *if for every set* $S$ *in* $\mathcal{C}$ *s.t.* $w(S) \geq \varepsilon \cdot w(X)$*, we have that* $H \cap S \neq \emptyset$*.*

**Using $\varepsilon$-Nets to Solve the Hitting Set Problem.** The original algorithm for solving hitting set problem using $\varepsilon$-net was invented by Brönnimann and Goodrich [6]. Below, we give a high-level description of their overall approach (referred to as the BG algorithm), because it will help understand our own extension. We begin by showing how $\varepsilon$-nets are related to hitting sets, and then, show how to use $\varepsilon$-nets to actually compute hitting sets.

Let's assume that we have a black-box to compute weighted $\varepsilon$-nets, and that we know the optimal hitting set $H^*$ which is of size $M$. Now, define a weight function $w^*$ as $w^*(x) = 1$ if $x \in H^*$ and $w^*(x) = 0$ otherwise. Then, set $\varepsilon = 1/M$, and use the black-box to compute a weighted $\varepsilon$-net for $(X, \mathcal{C}, w^*)$. It is easy to see that this weighted $\varepsilon$-net is actually a hitting set for $(X, \mathcal{C})$, since $w^*(S) \geq \varepsilon w^*(X)$ for *all* sets $S \in \mathcal{C}$. There are known techniques [7] to compute weighted $\varepsilon$-nets of size $O((1/\varepsilon)\log(1/\varepsilon))$ for set systems with a constant VC-dimension (defined later); thus, the above gives us a $O(\log M)$-approximate solution. For the particular case of disks, it is possible to construct $\varepsilon$-nets of size $O(1/\varepsilon)$ [12], and hence obtain a constant-factor approximation.

However, in reality, we do not know the optimal hitting set. So, we iteratively guess its size $M$, starting with $M = 1$ and progressively doubling $M$ until we obtain a hitting set solution (using the above approach). Also, to "converge" close to the $w^*$ above, we use the following scheme. We start with all weights set to 1. If the computed weighted $\varepsilon$-net is not a hitting set, then we pick one set in $\mathcal{C}$ that is not hit by it and double the weights of all points that it contains. Then, we iterate with the new weights. It can be shown that if the estimate of $M$ is correct and using $\epsilon = 1/(2M)$, then we are *guaranteed* to find a hitting set using the above approach after a certain number of iterations. Thus, if we don't find a hitting set after enough iterations, we double the estimate of $M$ and try again. It can be shown [6] that the above approach finds an $O(\log M)$-approximate hitting set in polynomial time for set systems with constant VC-dimension (defined below), where $M$ is the size of the optimal hitting set.

**VC Dimension.** We end the description of the BG algorithm, with the definition of Vapnik-Červonenkis (VC) dimension of set systems. Informally, the VC-dimension of a set system $(X, \mathcal{C})$ is a mathematical way of characterizing the "regularity" of the sets in $\mathcal{C}$ (with respect to the points $X$) in the system. A bounded VC-dimension allows the construction of an $\varepsilon$-net through random sampling of large enough size. The VC-dimension is formally defined in terms of set shattering, as follows.

**Definition 4 (VC-Dimension).** *A set* $S$ *is considered to be* shattered *by a collection of sets* $\mathcal{C}$ *if for each* $S' \subseteq S$*, there exists a set* $C \in \mathcal{C}$ *such that* $S \cap C = S'$*. The* VC-dimension *of a set system* $(X, \mathcal{C})$ *is the cardinality of the largest set of points in* $X$ *that can be shattered by* $\mathcal{C}$*.*

In our case, the VC dimension is at most 23 as given by the following theorem by Valtr [14].

**Theorem 1.** *If $X \subset \mathbb{R}^2$ is compact and simply connected, then VC-dimension of the set system $(X, \mathcal{C})$, where $X$ is a set of points and $\mathcal{C}$ is a collection of sets, is at most 23.*

Note that for a finite collection of sensors, whose covering regions are compact and simply connected, the dual is compact and simply connected too.

### 3.2 $k$-Hitting Set Problem and the $(k, \varepsilon)$-net Technique

We now formulate the $k$-hitting set ($k$-HS) problem, which is a generalization of the hitting set problem, viz., we want each set in the system to be hit by $k$ selected points.

**Definition 5 ($k$-Hitting Set ($k$-HS)).** *Given a set system $(X, \mathcal{C})$, the $k$-*hitting set ($k$-HS) *problem is to find the smallest subset of points $H \subseteq X$ with* at most one *point for each sibling-set such that $H$ hits every set in $\mathcal{C}$ at least $k$ times.*

<u>Connection Between $k$-HS and $k$-SC Problem.</u> Note that the above $k$-HS problem is the (generalized) dual of our sensor $k$-coverage problem ($k$-SC problem). Essentially, each point in the $k$-HS problem corresponds to a sensing region of a sensor, and each set in the $k$-HS problem corresponds to a target point. Below, we describe how to solve the $k$-HS problem, which essentially solves our $k$-SC problem. To solve the $k$-HS, we need to define and use a generalized notion of $\varepsilon$-net.

**Definition 6 (Weighted $(k, \varepsilon)$-Net).** *Suppose $(X, \mathcal{C})$ is a sibling-set system, and $w : X \to Z^+$ is a weight function. Define $w(S) = \sum_{x \in S} w(x)$ for $S \subseteq X$. A set $N \subseteq X$ is a* weighted $(k, \varepsilon)$-net *for $(X, \mathcal{C}, w)$ if $|N \cap S| \geq k$, whenever $S \in \mathcal{C}$ and $w(S) \geq \varepsilon \cdot w(X)$.*

**Using $(k, \varepsilon)$-Nets to Solve $k$-HS.** We can solve the $k$-HS problem using the BG algorithm [6], without much modification. However, we need an algorithm compute weighted $(k, \varepsilon)$-nets. The below theorem states that an appropriate random sampling of about $O(k/\varepsilon \log k/\varepsilon)$ points from $X$ gives a $(k, \varepsilon)$-net with high probability, if the set system $(X, \mathcal{C})$ has a bounded VC-dimension. For the sake of clarity, we defer the proof of the following theorem.

**Theorem 2.** *Let $(X, \mathcal{C}, w)$ be a weighted set system. For a given number $m$, let $N(m)$ be a subset of points of size $m$ picked randomly from $X$ with probability proportional to the total weight of the points in such subset. Then, for*

$$m \geq \max\left(\frac{2}{\varepsilon}\log_2\frac{2}{\delta},\ \frac{K}{\varepsilon}\log_2\frac{K}{\varepsilon}\right) \tag{1}$$

*the subset $N(m)$ is a weighted $(k, \varepsilon)$-sibling-net with probability at least $1 - \delta$, where $K = 4(d + 2k - 2)$, and $d$ is the VC-dimension of the set system.* $\square$

Now, based on the above theorem, we can use the BG algorithm with some modifications to solve the $k$-HS problem. Essentially, we estimate the size $M$ of an optimal $k$-HS (starting with 1 and iteratively doubling it), set $\varepsilon = k/(2M)$, and use Theorem 2 to compute[2] a $(k, \varepsilon)$-net $N$ of size $m$. If $N$ is indeed a $k$-hitting set, we stop; else, we pick a set in the system $\mathcal{C}$ that is not $k$-hit and double the weight of all the points it contains. With the new weights, we iterate the process. It can be shown[3] that within $(4/k)M \log_2(n/M)$ iterations of weight-doubling, we are guaranteed to get a $k$-HS solution if the optimal size of a $k$-HS is indeed $M$. Thus, after $(4/k)M \log_2(n/M)$ iterations, if we haven't found a $k$-HS, we can double our current estimate of $M$, and iterate. See Algorithm 1. The below theorem shows that the above algorithm gives an $O(\log M)$-approximate solution in polynomial time with high probability for general sets. The proof of the following theorem is again similar to that for the BG algorithm [6].

**Theorem 3.** *The algorithm described above (Algorithm 1) runs in time $O((|\mathcal{C}|+|X|)\,|X|\log|X|)$ and gives a $O(\log M)$-approximate solution for the $k$-HS problem for a general set systems $(X, \mathcal{C})$ of constant VC-dimension, where $M$ is the optimal size of a $k$-HS.*

*Proof.* The outer `for` loop, where $M$ is doubled each time, is run at most $O(\log|X|)$ times. The inner `for` loop, where the weights are doubled for a set, is executed at most $\frac{4}{k} M \log_2 \frac{|X|}{M} = O(|X|/k)$ times. Computing a $(k, \varepsilon)$-net using Theorem 2 takes at most $O(|X|)$ time, while the doubling-weight process may take up to $O(|\mathcal{C}|)$ time.

We now prove the approximation factor. An optimal algorithm would find a $k$-hitting set of size $M$. If the VC-dimension is a constant, the $k$-HS method of Theorem 2 finds a $(k, \varepsilon)$-net of size $O(\frac{k}{\varepsilon} \log \frac{k}{\varepsilon})$. So if $\varepsilon = \frac{k}{2M}$, the size of the $k$-hitting set is $O(M \log M)$, which is a $O(\log M)$-approximation. □

**Outline of Proof of Theorem 2.** There are two challenges in generalizing the random-sampling technique of [7], viz., (i) sampling with replacement cannot be used, and (ii) weights must be part of the sampling process.

Challenges in Extending the Technique of [7] to $k$-hitting set. The classical method [7] of constructing an $\varepsilon$-net consists of randomly picking a set $N$ of at least $m$ points, for a certain $m$, where each point is picked *independently* and randomly from the given set of points. This way of constructing an $\varepsilon$-net may result in duplicate points in $N$, but *the presence of duplicates does not cause a problem in the analysis.* Thus, we can also construct weighted $\varepsilon$-net easily by emulating weights using duplicated copies of the same point. The above described approach works well for 1-hitting set, partly because we do not count the number of times each set is hit. However, for the case of $k$-hitting set, when constructing a $(k, \varepsilon)$-net, we need to ensure that the number of *distinct* points that hit each set

[2] Theorem 2 gives a $(k, \varepsilon)$-net with high probability. It is possible to check efficiently if the obtained set is indeed a $(k, \varepsilon)$-net. If it is not, we can try again until we get one. On average, a small number of trials are sufficient to obtain a $(k, \varepsilon)$-net.

[3] See [15] for the proof, which is similar to the one for BG in [6].

**Algorithm 1**: Solving $k$-HS Problem using $(k, \varepsilon)$-nets.
Since $k$-HS is the dual of $k$-SC, this algorithm also solves $k$-SC
(in $k$-SC, $X$ corresponds to the set of sensors, and $\mathcal{C}$ to the set of target points).

Given a set system $(X, \mathcal{C})$.
**for** $(M = 1;\ \ M <= |X|;\ \ M *= 2)$
  $\varepsilon = k/(2\,M)$;
  reset the weights of all points in $X$ to 1;
  **for** $(i = 0;\ \ i < (4/k)M \log_2(|X|/M);\ \ i++)$
    Compute a $(k, \varepsilon)$-net $N$ of size $m$ using Theorem 2;
    **if** each set in $\mathcal{C}$ is $k$-hit by $N$, **return** $N$;
    select a set in $\mathcal{C}$ that is not $k$-hit, and double the weight of all the points in the set;

is at least $k$. Thus, constructing a $(k, \varepsilon)$-net by picking points independently at random (with duplicates) does not lead to correct analysis. Instead, we suggest a novel method to construct a weighted $(k, \varepsilon)$-net $N$ by: (i) selecting a random subset of points (without duplicates) at once, and (ii) including the weights directly in the above sampling process. To the best of our knowledge, we are the first one to propose this extension.[4]

Proof Sketch of Theorem 2. Let $m$ be as given by equation (1), and $N$ be the subset of points randomly picked from $X$ as described in Theorem 2. After picking $N$, pick another set $T$ (for the purposes of the below analysis) in the same way as $N$. We now define two events

$$E_1 \;=\; \{\exists A \in \mathcal{C} \text{ s.t. } |A \cap N| < k,\ w(A) \geq \varepsilon\, w(X)\}$$
$$E_2 \;=\; \{\exists A \in \mathcal{C} \text{ s.t. } |A \cap N| < k,\ w(A) \geq \varepsilon\, w(X),\ |A \cap Z| \geq \varepsilon m\}$$

where $Z = N \cup T$ and $\varepsilon m \leq \mathrm{E}[\,|A \cap Z|\,]/2$. The proof consists of 3 major steps:

1. First, we show that $\Pr[E_1] \leq 2\Pr[E_2]$.
2. Then, it's easier to bound the probability of $E_2$
$$\Pr[E_2] \;\leq\; (2m)^{d+2k-2}\, 2^{-\varepsilon m}$$
3. Finally, we have that $m$ verifies $2(2m)^{d+2k-2}\, 2^{-\varepsilon m} \;\leq\; \delta$

The outline of each step follows:

1. From the definition of conditional probability
$$\Pr[E_2 \mid E_1] \;=\; \Pr[E_2 \cap E_1]/\Pr[E_1] \;=\; \Pr[E_2]/\Pr[E_1]$$
So we just need to show that $\Pr[E_2 \mid E_1] \geq 1/2$. Let $Z = \{y_1, \ldots, y_{2m}\}$ (where $y_i$'s *are* pairwise different). Define the random variable
$$Y_i \;=\; \begin{cases} 1 & \text{if } y_i \in A \\ 0 & \text{o.w.} \end{cases}$$

[4] As discussed before, [8] uses $\varepsilon$-nets to solve sensor's $k$-coverage, but their method is flawed (see Footnote 1 for more details).

Set $Y = \sum_{i=1}^{2m} Y_i$, and we have $Y = |A \cap Z|$. It is possible to show that $\mathrm{E}[Y] = \mu \geq 2\varepsilon m$, and $\mathrm{Var}[Y] \leq \mu$. Applying Chebyshev's inequality

$$\Pr[Y < \frac{\mu}{2}] \;\leq\; \Pr[|Y - \mathrm{E}[Y]| > \frac{\mu}{2}] \;\leq\; \frac{4}{\mu^2}\,\mathrm{Var}[Y] \;\leq\; \frac{1}{2}$$

and the result follows.

2. We use an alternate view. Instead of picking $N$ and then $T$, pick $Z \subseteq X$ of size $2m$, then pick $N \subseteq Z$ and set $T = Z \setminus N$. It can be shown that the two views are equivalent. Now, define

$$E_A \;=\; \{|A \cap N| < k,\, |A \cap Z| \geq \varepsilon m\}$$

Since $N$ and $T$ are disjoint $|A \cap Z| = |A \cap N| + |A \cap T|$ and then $|A \cap N| < k$ iff $|A \cap Z| < k + |A \cap T|$. So we have that $E_A$ happens only if $\varepsilon m \leq |A \cap Z| \leq m + k - 1$. By counting the number of ways of choosing $N$ s.t. $|A \cap N| < k$, we can bound $\Pr[E_A]$

$$\begin{aligned}\Pr[E_A] \;&=\; \Pr[|A \cap N| < k,\; |A \cap Z| \geq \varepsilon m] \\ &\leq\; \Pr[|A \cap N| < k \mid \varepsilon m \leq |A \cap Z| \leq m + k - 1] \\ &\leq\; (2m)^{2k-2} \tbinom{2m-\varepsilon m}{m} / \tbinom{2m}{m} \;\leq\; (2m)^{2k-2}\, 2^{-\varepsilon m}\end{aligned}$$

Since $E_A$ depends only on the intersection $A \cap Z$

$$\Pr[E_2] \;\leq\; \bigcup_{A \,|\, A \cap Z \text{ is unique}} \Pr[E_A] \;\leq\; |\mathcal{C}_{|Z|}|\,(2m)^{2k-2}\, 2^{-\varepsilon m}$$

3. Similar to [7]. □

Please, refer to [15] for the detailed proof, which is omitted here due to lack of space.

<u>Remark.</u> Note that the approximation factor of Theorem 3 could be improved, if we could design an algorithm to construct smaller $(k, \varepsilon)$-nets. For instance, if we could construct a $(k, \varepsilon)$-net of size $O(k/\varepsilon)$, then we would have a constant-factor approximation for the $k$-HS problem. For the particular case of disks, it is easy to extend[5] the method in [12] to build a $(k, \varepsilon)$-net of size $O(k/\varepsilon)$ (see [15] for more details).

### 3.3 Distributed $\varepsilon$-Net Approach

Distributed implementation of the $\varepsilon$-net algorithm requires addressing the following main challenges.

1. We need to construct a $(k, \varepsilon)$-net, through some sort of distributed randomized selection.

[5] Essentially, it is enough to replace $\delta = \varepsilon/6$ with $\delta = \varepsilon/(6k)$ and the proof follows through. Also note that the dual of disks and points is also composed by disks and points.

2. For each constructed $(k, \varepsilon)$-net $N$, we need to verify in a distributed manner whether $N$ is indeed a $k$-coverage set ($k$-hitting set in the dual).
3. If $N$ is not a $k$-coverage set, then we need to select *one* target point (a set in the dual) that is not $k$-covered by $N$ and double the weights of all the sensing regions covering it.

We address the above challenges in the following manner. First, we execute the distributed algorithm in *rounds*, where a round corresponds to one execution of the inner `for` loop of Algorithm 1 (i.e., execution of the sampling algorithm for a particular set of weights and a particular estimate of $M$). We implement rounds in a weakly synchronized manner using internal clocks. Now, for each of the above challenges, we use the following solutions.

1. Each sensor keeps an estimate of the total weight of the system, and computes $m$ independently. To select $m$ sensors, each sensor decides to select itself independently with a probability $p = m * own\_weight/total\_weight$, resulting in selection of $m$ sensors (in expectation). Each selected sensor picks an orientation (sensing-region) of highest weight.
2. Locally, verify $k$-coverage of the owned target points, by exchanging messages with near-by (that cover a common target point) sensors. If a target point owned by a sensor $D$ and its near-by sensors are all $k$-covered for a certain number of rounds (for example 10), then $D$ exits the algorithm.
3. Each sensor decides to select one of the owned target points with a probability of $q = 1/((1 - \varepsilon)\, n)$, which ensures that the expected number of selected target point is 1.

### 3.4 Generalizations to *k*-coverage of an Area

The $\varepsilon$-net approach can also be used to $k$-cover a given area, rather than a given set of target points (as required by the formulation of $k$-SC 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 sensing regions in the system. The algorithm described here can then be used without any other modification, and the performance guarantees still hold.

## 4 Conclusions

In this paper, we studied the $k$-coverage problem with sensors, which is to select the minimum number of sensors so that each target point is covered by at least $k$ of them. We provided a $O(\log M)$-approximation, where $M$ is the number of sensors in an optimal solution. We introduced a generalization of the classical $\varepsilon$-net technique, which we called $(k, \varepsilon)$-net. We gave a method to build $(k, \varepsilon)$-nets based on random sampling. We showed how to solve the sensor's $k$-coverage problem with the Brönnimann and Goodrich algorithm [6] together with our $(k, \varepsilon)$-nets. We believe to be the first one to propose this extension.

As a future work, we would like to extend this technique to *directional sensors*. A directional sensor is a sensor that has associated multiple sensing regions, and its *orientation* determines its actual sensing region. The $k$-coverage problem with directional sensors is NP-complete and in [16] we proposed a greedy approximation algorithm. We believe that using $(k, \varepsilon)$-nets can give better approximation factor for this problem.

## References


1. Gupta, H., Zhou, Z., Das, S., Gu, Q.: Connected sensor cover: Self-organization of sensor networks for efficient query execution. TON **14**(1) (2006)
2. Charkrabarty, K., Iyengar, S., Qi, H., Cho, E.: Grid coverage for surveillance and target location in distributed sensor networks. Transaction on Computers (2002)
3. Slijepcevic, S., Potkonjak, M.: Power efficient organization of wireless sensor ad-hoc networks. In: ICC. (2001)
4. Meguerdichian, S., Koushanfar, F., Qu, G., Potkonjak, M.: Exposure in wireless ad-hoc sensor networks. In: MobiCom. (2001)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. 2nd edn. MIT Press McGraw-Hill, Cambridge Boston (2001)
6. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite vc-dimension. Discrete & Computational Geometry **14**(4) (1995) 463–479
7. Haussler, D., Welzl, E.: Epsilon-nets and simplex range queries. Discrete & Computational Geometry **2** (1987) 127–151
8. Hefeeda, M., Bagheri, M.: Randomized k-coverage algorithms for dense sensor networks. In: INFOCOM. (2007) 2376–2380
9. Ye, F., Zhong, G., Lu, S., Zhang, L.: Peas: A robust energy conserving protocol for long-lived sensor networks. In: ICDCS. (2003)
10. Shakkottai, S., Srikant, R., Shroff, N.: Unreliable sensor grids: Coverage, connectivity and diameter. In: INFOCOM. (2003)
11. Zhou, Z., Das, S.R., Gupta, H.: Connected k-coverage problem in sensor networks. In: ICCCN. (2004) 373–378
12. Matoušek, J., Seidel, R., Welzl, E.: How to net a lot with little: Small $\varepsilon$-nets for disks and halfspaces. In: SCG. (1990) 16–22 Corrected version available at http://kam.mff.cuni.cz/∼matousek/oldpaps.html.
13. O'Rourke, J.: Art Gallery Theorems and Algorithms. Volume 3. Oxford University Press, New York, NY (1987)
14. Valtr, P.: Guarding galleries where no point sees a small area. Israel Journal of Mathematics **104** (1998) 1–16
15. Fusco, G., Gupta, H.: $\varepsilon$-net approach to sensor $k$-coverage. Technical report, Stony Brook University (2009) http://www.cs.sunysb.edu/∼fusco/publications.html.
16. Fusco, G., Gupta, H.: Selection and orientation of directional sensors for coverage maximization. In: SECON. (2009)