# Joint Sensing Duty Cycle Scheduling for Heterogeneous Coverage Guarantee

Kin Sum Liu*, Tyler Mayer†, Hao Tsung Yang*, Esther Arkin†, Jie Gao*, Mayank Goswami‡, Matthew P. Johnson$^{\mathcal{S}}$, Nirman Kumar$^{\mathcal{P}}$, Shan Lin♭,

*Department of Computer Science, Stony Brook University. {kiliu,haotyang,jgao}@cs.stonybrook.edu
†Dept. of Applied Math. & Stats., Stony Brook University. tyler.mayer@stonybrook.edu,{estie,jsbm}@ams.stonybrook.edu
‡Department of Computer Science, Queens College, CUNY. mayank.goswami@qc.cuny.edu
$^{\mathcal{S}}$Department of Computer Science, Lehman College, CUNY. mpjohnson@gmail.com
$^{\mathcal{P}}$Department of Computer Science, University of Memphis, nkumar8@memphis.edu
♭Department of Electrical and Computer Engineering, Stony Brook University. shan.x.lin@stonybrook.edu

***Abstract*—In this paper we study the following problem: given a set of $m$ sensors that collectively cover a set of $n$ target points with heterogeneous coverage requirements (target $j$ needs to be covered every $f_j$ slots), how to schedule the sensor duty cycles such that all coverage requirements are satisfied and the maximum number of sensors turned on at any time slot is minimized. The problem models varied real-world applications in which sensing tasks exhibit high discrepancy in coverage requirements – critical locations often need to be covered much more frequently. We provide multiple algorithms with best approximation ratio of $O(\log n + \log m)$ for the maximum number of sensors to turn on, and bi-criteria algorithm with $(\alpha, \beta)$-approximation factors with high probability, where the number of sensors turned on is an $\alpha = O(\frac{\delta(\log(n)+\log(m))}{\beta})$-approximation of the optimal (satisfying all requirements) and the coverage requirement is a $\beta$-approximation; $\delta$ is the approximation ratio achievable in an appropriate instance of set multi-cover. When the sensor coverage exhibits extra geometric properties, the approximation ratios can be further improved. We also evaluated our algorithms via simulations and experiments on a camera testbed. The performance improvement (energy saving) is substantial compared to turning on all sensors all the time, or a random scheduling baseline.**

## I. Introduction

Sensor deployment and scheduling has been a fundamental problem for sensor network applications. The complexity of the problem comes from the variety of sensing modalities and sensor coverage ranges (ranging from omnidirectional disk coverage [22] to line-of-sight visibility coverage [23], [31]), the application coverage requirement (full/blanket coverage [5], [33], full-view coverage [38], $k$-coverage [42], barrier coverage [32], etc), and networking requirement (sensors remain connected by wireless radio [5], [23]). A substantial amount of work has been invested on the many variations of this problem in the past few years [16], [36].

In this work, we assume that the sensor deployment problem has been solved and a near optimal set of sensors have been placed to provide full coverage of the target locations. We instead look at the optimal duty cycle scheduling of the sensors. We may need a large number of sensors to provide full coverage — but it is possible that many sensors may simultaneously cover the same target, exhibiting a high level of redundancy in the system. Many sensors, such as cameras, may consume a large amount of energy to operate, or require a large amount of bandwidth to upload data. Thus efficient scheduling of the operations of such sensors can result in significant energy/communication bandwidth savings. Second, in many monitoring applications the target locations may require different levels of quality of service. In a building monitoring scenario for safety applications, certain locations (high activity areas) such as entrances to the building typically require continuous or highly frequent monitoring while other locations such as individual offices may only need to be checked once in a while as activity levels are low. This heterogeneity in service quality also provides an opportunity to schedule the sensors in a resource efficient way without hurting the overall application. This combined with the joint sensing/coverage features can provide opportunities for a smart scheduling to save substantially in energy/operational costs. See Figure 1 for an example.

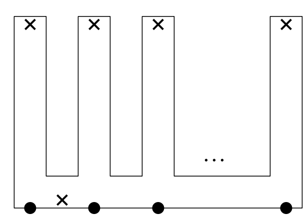



Fig. 1: An example of $n+1$ target points (shown as crosses) and $n$ sensor nodes (shown as solid disks) with line-of-sight coverage. $n$ sensors are needed to ensure full coverage of the targets. Suppose that the $n$ targets on top each need to be covered once in every $n$ slots (with $f = n$) while the bottom target needs to be covered every slot (with $f = 1$). Alternatively turning on each of the $n$ sensors would suffice, with savings of $n$ times the energy consumption compared to keeping all sensors on all the time.

We formulate our problem in the following manner. We are given a set of $m$ sensors and $n$ targets, where each sensor $i$ covers a subset of targets $S_i$. Further, we assume that time is slotted and in each time slot one may turn on only a subset of sensors. Each target location $j$ has a minimum coverage requirement – it must be covered at least once for every $f_j$

slots. These $f_j$'s can vary significantly. The problem asks to minimize the maximum number of sensors turned on at any time slot such that all the coverage requirements are satisfied.

In a closely related work by Liu et al. [27] a budget of $k$ sensors per slot is given, and the goal is to find a duty cycle schedule so that all target points are monitored sufficiently frequently. They considered two optimization objectives. The first is to minimize the maximum amount of time that any target point remains unseen (called the min-max version), and the second is to minimize the average time that any target point remains unseen (called the min-average version). They show for the min-max version, that if the sensor set is a minimal cover for the domain then the round robin schedule with each time slot having $\lceil m/k \rceil$ different sensors active is an optimal schedule, otherwise they can find an $O(\log n)$ approximation with a periodic schedule of an approximate set cover solution. For the min-average variant they show how to produce a schedule which serves as a $(2+\varepsilon)\alpha$-approximation where $\alpha = O(1)$ if $k \geq \log m / \log\log m$ and $\alpha = \log m / \log\log m$ otherwise.

The most significant difference in our problem formulation as compared to both the min-max and min-average version is that we strictly enforce the coverage quality requirement, which can be highly heterogeneous for different target locations. This is crucial for real-time applications with hard constraints on the coverage guarantee.

**Our Results.** We provide three algorithms for the joint duty cycle scheduling problem. First we show that restricting our solution to the family of periodic schedules can only suffer by a factor of two, and that finding the optimal periodic schedule is NP-hard. On the positive side, we have three different algorithms with different design ideas/approximation ratios.

- A combinatorial $O(\log n + \log m)$-approximation.
- An $O\left(\gamma \log(\frac{f_{\max}}{f_{\min}})\right)$-approximation in geometric instances where $\gamma$ is the best approximation ratio achievable in the corresponding geometric set cover problem and $f_{\max}$ ($f_{\min}$) is the largest (smallest) window size among all coverage requirements.
- An $O(\gamma \log m)$-approximation to the dual problem in geometric settings.
- An $(\alpha = O(\frac{\delta(\log n + \log m)}{\beta}), \beta)$ bi-criteria approximation algorithm where $\alpha$ is an approximation to the optimal number of sensors required to maintain full coverage, and $\beta$ is the approximation to the coverage requirement. In this case $\delta$ is the best approximation achievable for the set multi-cover problem.

In simulations we observe that the three algorithms have their unique strength strength under different scenarios. If the frequency requirements follow a uniform distribution the combinatorial algorithm needs only $60\%$ of sensors active compared to baseline random schedule. On the other hand, the geometric algorithm beats others and has only $36\%$ of sensors active when the frequency requirements follow an exponential distribution. Experiments on a camera testbed verify the energy saving of our scheduling algorithms.

## II. Related Work

Our version of the duty cycle problem arises as a combination of a set multi-cover and a scheduling problem in sensor networks. In the set multi-cover problem one is given a universe $\mathcal{U}$ of $n$ elements (with multiplicity) and a collection $\mathcal{S}$ of subsets of $\mathcal{U}$ and is asked to find a minimum cardinality sub-collection $\mathcal{C} \subseteq \mathcal{S}$ which covers $\mathcal{U}$ (with, at least, the appropriate multiplicity for each element). This problem is clearly a generalization of set cover which has been shown NP-Hard and can't be approximated to within a factor of $(1-\varepsilon)\ln n)$, $\forall \varepsilon > 0$, unless $NP \subseteq DTIME(n^{O(\log\log n)})$, by Feige [13]. Better approximations can be obtained in special cases. For example, [10] show a $O(\log |OPT|)$ approximation when the set system has a constant VC dimension. The same approximation can be obtained for the set multi-cover problem [12]. For covering points by half-spaces in 3D, an $O(1)$-approximation is possible [12]. Bansal et al. [7] showed a randomized $O(1)$-approximation algorithm for the weighted geometric set multi-cover problem when the covering objects have linear union complexity. The set multi-cover problem using disks as sets has a constant factor approximation [8], [1], [9], as well as a QPTAS [6]. Altinel et al. [4] show that, for targets with uniform coverage requirements and sensor locations chosen from a regular grid with some density $d$, an algorithm with a $(6\pi, f(d))$ bi-criteria approximation which uses at most $6\pi$ times as many sensors from the regular grid coordinates as the optimal solution and covers at least $f(d)$ percent of the total area where $f(d)$ is a function based on the density chosen for the grid.

Given a fixed placement of sensors, the problem of scheduling these sensors to collaboratively cover the domain has been studied in the line of sight model [28], [19]. In visual sensor networks Munishwar et al. [30] study the problem of finding an angle of view for each camera to maximize the number of target points covered. They give an exact centralized algorithm which is suitable to run on small sized networks, as well as a decentralized heuristic for larger networks with a $1/2$-approximation guarantee.

The problem of near-optimal placement of sensors given a duty cycle schedule in order to obtain certain coverage requirements has been studied in [25], [23], [41]. Most of these heuristics are validated via numerical experiments.

One related paper by Feinberg et al. [14] studies a scheduling problem called the generalized pinwheel problem for a single server and a set of recurring jobs. Each job has a duration and a maximum allowable time between two consecutive runs. Authors show necessary conditions for the feasibility of the problem and present an exact algorithm (with exponential running time) based on dynamic programming as well as several heuristic approximations as the problem is NP-hard.

There is a huge literature on duty cycle scheduling in sensor networks. See [37] for a survey. The duty cycle problem has been studied in communication networks and energy management systems [34], [39], [18], [21]. Yoo et al. [40] study the problem of scheduling sensors with energy harvesting capabilities based on the amount of remaining energy in the system, while Jaleel et al. [24] propose a probabilistic deployment algorithm based on a Poisson process and give upper bounds on the probability of coverage based on the energy decay function and spatial distribution of sensors over the network. These algorithms are largely non-collaborative, meaning that they don't take into account the overlapping coverage ranges.

Several approaches to the duty cycle scheduling problem using integer linear programming (ILP) have also been explored. Meguerdichian et al. [29] solve the natural set cover ILP to find a minimum 0/1 set cover of the targets. They also consider a sensing intensity model, where the intensity of coverage of a particular target is a function of the active sensor quality and its distance to the target. They show how to find a minimum cover which maintains an average sensing intensity for each target, and show how to find a maximum number of such coverings to get a redundant set which can be scheduled alternatively to minimize energy consumption. Chakrabarty et al. [11] show how to use ILP to find the minimum cover of sensors from a regular grid of positions for coverage as well as target localization in the network. Gu et al. [17] show how a column generation approach to solving the associated ILP can be used to find a minimum cover even for heterogenous coverage requirements of the targets. Although ILP methods can solve these problems exactly, they are often only feasible for modest problem sizes, as the class of problems are still NP-Complete.

## III. Joint Duty Cycle Scheduling

Given a set of $m$ sensors $\mathcal{S}$ and a set of $n$ targets $\mathcal{U}$ to monitor, sensor $i$ covers a subset of targets $S_i \subseteq \mathcal{U}$. For each target $j$, we are given the coverage requirement as the window size $f_j$ such that for any time window of length $f_j$ target $j$ is covered at least once. Time is slotted and at any time slot we wish to only turn on a subset of the sensors. The problem is to design a duty cycle schedule such that all targets meet their coverage frequency requirement and the maximum number of sensors turned on at any time slot (which is referred to by $k$) is minimized.

This problem is clearly NP-hard as set cover is a special case – when $f_i = 1$ for all $i$. Before we describe approximation algorithms for this problem. We first present a few important observations on periodicity of the optimal schedule.

It is not clear that an optimal schedule will be periodic, but here we show that if we restrict ourselves to periodic schedules with a sufficiently large period, such schedule is within a constant factor 2 of the optimal schedule. Moreover, we may force the period to be no greater than $m$, if we turn on only one extra sensor in every time slot on top of any periodic schedule.

We can model a schedule $\mathcal{F}$ as a set of computable functions $\mathcal{F} = \{F_1, \ldots, F_m\}$ where each $F_i : \mathbb{N} \to \{\mathsf{true}, \mathsf{false}\}$ specifies if sensor $i$ is to be turned on at time $t$, i.e., $F_i(t) = \mathsf{true}$ iff sensor $i$ is turned on at time $t$ for any $t \in \mathbb{N}$. Such a schedule is periodic with period $T$, if each $F_i$ is a periodic function of $t$ with period $T$, i.e., $F_i(t+T) = F_i(t)$ for each $i = 1, \ldots, m$ and any $t \in \mathbb{N}$.

Let $\mathcal{F}_{\mathsf{opt}}$ denote the optimal schedule and $\mathcal{F}_{\mathsf{opt}}^T$ denote the optimal schedule with period $T$. For any schedule $\mathcal{F}$ let $\mathsf{k}(\mathcal{F})$ denote the maximum number of sensors turned on at any time slot, called the cost of the schedule:

$$\mathsf{k}(\mathcal{F}) = \max_{t \in \mathbb{N}} |\{i : F_i(t) = \mathsf{true}\}|$$

Let $f_{\mathsf{max}} = \max_{j=1 \ldots n} f_j$ be the largest window size. We have the following lemmas. Omitted proofs of this section can be found in Appendix A.

**Lemma 3.1.** *For any $T \geq f_{\mathsf{max}}$, $\mathsf{k}(\mathcal{F}_{\mathsf{opt}}^T) \leq 2\mathsf{k}(\mathcal{F}_{\mathsf{opt}})$.*

From the above lemma, we can focus on finding a periodic schedule with period $T = f_{\mathsf{max}}$ such that we only lose an approximation factor of two in the cost. But $f_{\mathsf{max}}$ can still be fairly large, e.g., exponential in $m$ or $n$. The lemma below shows we can assume the period is $m$ (the number of sensors), at the cost of turning on only one more sensor in each slot, on top of any periodic schedule of period of $f_{\mathsf{max}}$.

**Lemma 3.2.** $\mathsf{k}(\mathcal{F}_{\mathsf{opt}}^m) \leq 2\mathsf{k}(\mathcal{F}_{\mathsf{opt}}) + 1$.

**Proof:** Separate the targets $\mathcal{U}$ based on their minimum window size into two sets $\mathcal{U}'$,$\mathcal{U}''$ where $\mathcal{U}' = \{j : f_j \leq m\}$ and $\mathcal{U}'' = \{j : f_j > m\}$. Let $\mathcal{F}_{\mathsf{opt}}$ be the optimal schedule for $\mathcal{U}$ and $\mathcal{F}'_{\mathsf{opt}}$ be the optimal schedule for $\mathcal{U}'$. Obviously, $\mathsf{k}(\mathcal{F}'_{\mathsf{opt}}) \leq \mathsf{k}(\mathcal{F}_{\mathsf{opt}})$. From Lemma 3.1, we know that $\mathsf{k}(\mathcal{F}'^m_{\mathsf{opt}}) \leq 2\mathsf{k}(\mathcal{F}'_{\mathsf{opt}})$.

To cover $\mathcal{U}''$, we can turn on one extra sensor in each time slot by simply doing round robin on the $m$ sensors. This suffices to cover all targets in $\mathcal{U}''$ with window size greater than $m$. In fact, these targets are covered more frequently.

Put together, the new schedule has period of $m$ and cost at most $2\mathsf{k}(\mathcal{F}_{\mathsf{opt}}) + 1$. □

Therefore, in the rest of the paper we will assume without loss of generality that $f_{\max} \leq m$ and will focus on periodic schedules with period at most $m$. This restriction only induces a constant factor.

## IV. Combinatorial Algorithm

In this section, we construct a periodic schedule $\mathcal{F}^m$ of period length $m$. The algorithm is for the purely combinatorial setting, i.e., the set of targets covered by a sensor can be arbitrary. We do not make any assumptions on sensing range and the result applies in the most general setting.

The main challenge lies in how to extend the concept of coverage to the temporal domain. For each target $j$, we partition one cycle of $m$ slots into $m/\lfloor f_j/2 \rfloor$ intervals of $\lfloor \frac{f_j}{2} \rfloor$ time slots each. For each interval $b$, introduce a new element $e_{j,b}$ which represents the target in the time frame $b$. The total number of elements is $\sum_{j=1}^{n} \frac{m}{\lfloor f_j/2 \rfloor} = O(nm)$. Let $\mathcal{E}$ be the universe of these new elements.

Sensors are place in the $m$ slots to create a periodic schedule $\mathcal{F}^m$. If a sensor $i$ is placed in slot of time frame $b$, and $i$ covers target $j$, we say the element $e_{j,b}$ is *covered*. Then, if our schedule covers all target $j$'s corresponding elements $\{e_{j,b} : b \in [1, \dots, \frac{m}{\lfloor f_j/2 \rfloor}]\}$, it is clear that $j$ is covered at least once in every window of size $f_j$ – since every such window will fully contain at least one interval.

Now we are ready to formulate a new set cover problem. We work with the family $\mathcal{V}$ of vectors of dimension $m$. Each vector $V \in \mathcal{V}$, $V =< i_1, i_2, \dots, i_m >$, corresponds to sensor $i_t$ being turned on at time slot $t$, for $1 \le t \le m$. We choose from this family $\mathcal{V}$ a vector and use it as one *layer* of the periodic schedule. Eventually we choose $k$ vectors of $V$, overlay them along the temporal domain, to create a schedule with cost of $k$. We want to minimize $k$ such that all elements $\{e_{j,b} : \forall j, \forall b\}$ are covered.

Formally, in this set cover $(\{S(V), V \in \mathcal{V}\}, \mathcal{E})$ problem, each set in the collection $\mathcal{V}$ is a subset of elements that are covered by the vector $V$, denoted by $S(V) \subseteq \mathcal{E}$, $V \in \mathcal{V}$. We wish to choose a minimum number of such vectors, such that the elements they collectively cover is $\mathcal{E}$.

**Lemma 4.1.** *$\mathcal{F}^m$ constructed from the optimal solution $\mathcal{C}^*$ of the set cover Problem $(\{S(V), V \in \mathcal{V}\}, \mathcal{E})$ is a 2-approximation to $\mathcal{F}^m_{\mathsf{opt}}$.*

**Proof:** $\mathcal{F}^m$ is the best $m$-period schedule to cover every element of $\mathcal{E}$. Clearly any schedule that covers all elements in $\mathcal{E}$ satisfies all the coverage window requirements. This establishes the correctness of the solution $\mathcal{F}^m$.

For the approximation factor of $\mathcal{F}^m$, note that the optimal solution $\mathcal{F}^m_{\mathsf{opt}}$ that satisfies all the coverage window requirements, may not cover all elements of $\mathcal{E}$. The latter is a stronger, sufficient but not necessary condition of the former.

To establish the approximation factor, we take $\mathcal{F}^m_{\mathsf{opt}}$ and make it a new schedule $\mathcal{F}^{m/2}$ by merging the sensors in the $(2t-1)$-th and $2t$-th time slot as the sensors turned on in the $t$-th time slot in $\mathcal{F}^{m/2}$, for each $1 \le t \le m/2$. We have two observations of $\mathcal{F}^{m/2}$:

- $\mathsf{k}(\mathcal{F}^{m/2}) = 2\mathsf{k}(\mathcal{F}^m_{\mathsf{opt}})$.
- The schedule $\mathcal{F}^{m/2}$ will have at least one sensor that covers target $j$ in each window of size $\lfloor f_j/2 \rfloor$, for each $j$. Therefore, $\mathcal{F}^{m/2}$ covers all elements in $\mathcal{E}$. By the optimality of $\mathcal{F}^m$, we know $\mathsf{k}(\mathcal{F}^m) \le \mathsf{k}(\mathcal{F}^{m/2})$.

Combining the two we have $\mathsf{k}(\mathcal{F}^m) \le 2\mathsf{k}(\mathcal{F}^m_{\mathsf{opt}})$. □

Running the classical greedy algorithm for set cover on this new instance $(\{S(V), V \in \mathcal{V}\}, \mathcal{E})$ immediately gives an $O(\log n + \log m)$-approximation. The issue is that there are exponentially many sets. $|\mathcal{V}| = m^m$, since in each time slot we may have $m$ choices of sensors. Therefore at each step, we cannot afford to enumerate over all remaining vectors in $\mathcal{V}$ to select the one that covers the maximum number of new elements of $\mathcal{E}$. However, it turns out that this greedy selection does not need to be done optimally. [20] shows that if we use a *pseudo*-greedy implementation that picks a $\beta$-approximate best set in each step (i.e., we choose a set that covers at least $\beta h$ new elements if the best set can cover $h$ new elements), such *pseudo*-greedy cover yields a $O(\frac{1}{\beta} \ln |\mathcal{E}|)$-approximation.

Below we show how to do this pseudo-greedy step. In particular, to find the next vector $V \in \mathcal{V}$, we run another greedy algorithm to gradually fill in the sensors for all the $m$ slots in vector $V$. The next sensor we turn on is the one that covers the maximum number of new elements in $\mathcal{E}$. We continue this iteration until all slots of $V$ are filled up. The entire algorithm is summarized in the pseudo code below.

**Algorithm 1** Combinatorial Algorithm

**Inputs:** Universe $\mathcal{E}$
**Output:** Periodic Schedule $\mathcal{F}^m$

$\mathcal{C} = \emptyset$
**while** not all $e_{j,b} \in \mathcal{E}$ is covered **do**
  $k \leftarrow k + 1$
  **while** not all $m$ time slots are used **do**
    choose $(r, t)$ pair that covers the most remaining elements, where sensor $r$ is turned on at an empty time slot $t$
    $i_t \leftarrow r$
  **end while**
  $\mathcal{C} = \mathcal{C} \cup S(< i_1, ..., i_m >)$
**end while**
construct $\mathcal{F}^m$ from $\mathcal{C}$

**Lemma 4.2.** *The pseudo-greedy step gives a $\frac{1}{2}$-approximation, i.e., the number of elements covered is at least half of the maximum number of elements that can be covered by the optimal vector $V$.*

**Proof:** We observe that our problem of finding the best vector is an instance of Problem (1.5) in Fisher's analysis [15] with $P = 1$. Specifically, our objective can be re-stated as a matroid-constrained sub-modular maximization:

$$\max_{X \subseteq E}\{z(X) : \quad X \in I, \mathcal{M} = (E, I) \text{ a matroid, } z(X) \text{ sub-modular and non-decreasing,}\}$$

where

$$E = \{(i, t) : i \in \mathcal{S}, t \in \mathbb{N}_{\le m}\}$$
$$I = \{A : A \subseteq \{\{(i_1, 1), \dots, (i_m, m)\} : i_1, \dots, i_m \in \mathcal{S}\}\}$$
$$z(X) = z(\{(x_1, 1), \dots, (x_m, m)\}) = |S(< x_1, \dots, x_m >)|$$

Here $E$ is the set of all possible allocations of individual time slots to a sensor and $I$ is the collection of all possible (partial) allocations of $m$ time slots. $\mathcal{M}$ is a matroid because $z(\cdot)$ is a cardinality function which is clearly sub-modular and non-decreasing. For the hereditary property, if $X \subseteq Y$ and $Y \in I$, then $X \in I$ from our construction of the independent sets $I$. For the independent set exchange property, if $X, Y \in I$, and $|Y| > |X|$ (so $X$ must be a partial allocation), then we must be able to find a time slot $t_e$ allocated in $Y$ but not in $X$ ($(\cdot, t_e) \notin X$) and assigning that to $X$ is still feasible. So we have $\exists (i_e, t_e) \in Y \setminus X, X \cup (i_e, t_e) \in I$.

From this observation, we know that our algorithm is equivalent to using *greedy heuristic* for Problem (1.5) in [15] and we can directly apply Theorem (2.1) to get an approximation factor of $\frac{1}{2}$. □

**Theorem 4.3.** *The algorithm has an approximation factor* $O(\log n + \log m)$.

From Lemma 4.2, our algorithm picks a $\beta$-approximate best vector with $\beta = \frac{1}{2}$. Plugging in $\beta = \frac{1}{2}$ into Corollary 2 of [20], Algorithm 1 is a $O(2 \ln |\mathcal{E}|)$-approximation, i.e., $O(\log n + \log m)$-approximation.

## V. Geometric Algorithm

The algorithm we discussed so far does not make any assumptions on the coverage range of the sensors. In many scenarios the coverage ranges have some geometric properties which can be leveraged to obtain better approximation ratios.

For example, consider the following problem: among the sensors in $\mathcal{S}$, choose a minimum number of them to cover all targets in $\mathcal{U}$. As mentioned, this problem in general cannot be approximated better than $(1-\varepsilon)\log n$, $\forall \varepsilon > 0$. However, if the sensor coverage ranges have constant VC-dimension, the approximation ratio can be improved to $O(\log |OPT|)$ where $|OPT|$ is the number of sets in the optimal solution [10]. If the sensor coverage ranges are axis-parallel rectangles or disks (of possibly different radii), the set cover problem can be solved with approximation of $O(\log \log |OPT|)$ or $O(1)$ respectively [3]. If the sensor coverage ranges are unit disks, then a simple greedy algorithm gives a constant approximation [2] and a PTAS exists [26]. In the following we denote by $\gamma$ the best approximation ratio one can obtain for the particular instance of geometric set cover.

In this section, we will present a different algorithm that can benefit from results in geometric set cover problems.

**Uniform Window Size.** As an appetizer we look at the special case when all the targets have the same window size in coverage requirement, i.e., $f_j = f$, for all $j$. Define the optimal set cover solution as $\mathcal{C}^* \subseteq \mathcal{S}$ that covers all targets. We can find a geometric set cover solution $\mathcal{C}$ with size $|\mathcal{C}| \leq \gamma|\mathcal{C}^*|$ for the universe $\mathcal{U}$ with $n$ targets and $\gamma$ is the approximation ratio.

Now we have the following scheduling algorithm: Do round robin on $\mathcal{C}$ with $k = \lceil |\mathcal{C}|/f \rceil$ sensors. The period of this schedule is at most $f$, which satisfies the coverage frequency requirement.

We argue that the optimal schedule has to use $k^* \geq \lceil |\mathcal{C}^*|/f \rceil$. This is because all the targets must be covered at least once in one full period, thus the set of sensors that appear in one full period is a set cover solution, with size at least $|\mathcal{C}^*|$. Thus in this simple setting we find a sensor scheduling with approximation ratio of $\gamma$.

**General Setting.** When the targets have different coverage window sizes $f_j$, we take $f_{\min} = \min_j f_j$ and $f_{\max} = \max_j f_j$. Clearly, turning on $\lceil |\mathcal{C}|/f_{\min} \rceil$ at each time slot is a feasible solution and a lower bound is to turn on $\lceil |\mathcal{C}^*|/f_{\max} \rceil$ sensors.

Now we partition the targets into $\log R$ sets, where $R = f_{\max}/f_{\min}$, such that in each set $\mathcal{G}_i$ the frequencies of the targets do not differ by more than a factor of 2. Take $h_i = \min_{j \in \mathcal{G}_i} f_j$ and $H_i = \max_{j \in \mathcal{G}_i} f_j$. $H_i \leq 2h_i$. To cover the set of targets in $\mathcal{G}_i$, let $\mathcal{C}_i$ be the set cover solution and $\mathcal{C}_i^*$ be the optimal set cover solution. $|\mathcal{C}_i| \leq \gamma|\mathcal{C}_i^*|$, where $\gamma$ is the approximation factor in the geometric set cover problem.

For each set $\mathcal{G}_i$, we will use round robin and turn on $k_i = \lceil |\mathcal{C}_i|/h_i \rceil$ sensors at each time slot, which is sufficient to guarantee that all targets are covered frequently enough. The total number of sensors we need to turn on is $k = \sum_i k_i$.

**Theorem 5.1.** *The algorithm has an approximation factor* $O(\gamma \log R)$, *where* $R = f_{\max}/f_{\min}$.

**Proof:** Just to cover the group of sensors $\mathcal{G}_i$, we have a lower bound $k^* \geq \lceil |\mathcal{C}_i^*|/H_i \rceil$. Thus

$$k_i = \lceil |\mathcal{C}_i|/h_i \rceil \leq \gamma|\mathcal{C}_i^*|/h_i \leq 2\gamma|\mathcal{C}_i^*|/H_i \leq 2k^*\gamma.$$

There are $\log R$ groups. This finishes the proof. □

**Discussion.** The approximation ratio of this algorithm depends on $R$, the ratio of the largest window size and the smallest window size. Recall that we can assume that $f_{\max}$ is at most $m$ with only a sacrifice of a factor of 2 in Lemma 4.1, thus we may assume that $\log R = O(\log m)$. In the case when the coverage ranges are disks this approximation ratio is no worse than that of the combinatorial algorithm. But when $R$ is much smaller, i.e., this algorithm can be better than the combinatorial algorithm. We tested a variety of geometric setting in our simulations and indeed it shows that in the geometric setting this algorithm starts to outperform the combinatorial algorithm.

## VI. Dual and Bicriteria Algorithms

In this section, we study the tradeoff between coverage requirement and resource constraints. In the original problem we fix the coverage frequency requirement and minimize $k$ (the maximum number of sensors to turn on in a slot). Here we would like to study two other variations:

- The *dual* problem: we fix $k$ (the number of sensors one can turn on in each slot) and minimize the factor $\beta$, by which we stretch the window size of the coverage requirement. That is, each target $j$ is covered at least once in every window of size $\beta f_j$, with $\beta \geq 1$. We denote this as the $\beta$-relaxed coverage requirement. We wish to minimize $\beta$.
- $(\alpha, \beta)$-approximation: we turn on at most $\alpha k^*$ sensors in each slot such that each target $j$ is covered at least once in every window of size $\beta f_j$ ($\beta \geq 1$), where $k^*$ is the optimal solution for the original problem (when the coverage window size is $f_j$ for target $j$).

### A. The Dual Problem

In the dual problem we look at the case when $k$ is given and we may not be able to meet the requirements for all targets. Thus we consider stretching each window size $f_j$ by a factor $\beta$ and we hope to minimize $\beta$.

We use the same partition of targets into groups $\mathcal{G}_i$ as in Section V. Now for a total quota of $k$ sensors to turn on at each time slot, we give a quota of $k/\log R$ to each group $\mathcal{G}_i$. The stretch $\beta_i$ beyond the requirement is bounded by

$$\beta_i \leq \frac{k_i}{k/\log R} = \frac{k_i \log R}{k}.$$

Let the stretch factor in the optimal solution be $\beta^*$. Consider only group $i$, neglecting other targets, and allocate the entire quota of $k$ sensors for monitoring targets in $\mathcal{G}_i$. This solution over $\mathcal{G}_i$ is no larger than the optimal solution. Thus a lower bound on the stretch factor over group $\mathcal{G}_i$ is a lower bound for $\beta^*$. By this argument we have

$$\beta^* \geq \lceil |\mathcal{C}_i^*|/h_i k \rceil.$$

Re-arrange, we have

$$\beta \leq \frac{k_i \log R}{k} \leq \frac{|\mathcal{C}_i^*| O(\gamma \log R)}{h_i k} \leq \beta^* O(\gamma \log R).$$

Thus, we have the following theorem.

**Theorem 6.1.** *The dual problem, can be approximated within a factor of $O(\gamma \log R) = O(\gamma \log m)$, where $\gamma$ is the approximation ratio of geometric set cover and $R = f_{\max}/f_{\min}$ is the aspect ratio of the window size.*

### *B. Bicriteria Approximation*

For an $(\alpha, \beta)$-bi-criteria approximation we relate the duty cycle scheduling problem to the *set multi-cover problem*. The set multi-cover problem is an extension of the set cover problem. Given a set of targets $\mathcal{U}$, we want to choose a multi-set $\mathcal{C}$ (i.e., a sensor can appear multiple times) with sensors from $\mathcal{S}$ with minimum number of sensors such that target $j$ must be covered at least $r_j$ times. $r_j$ is called the *coverage multiplicity*. In our case, we choose $r_j = f_{\max}/f_j$.

The optimal solution to this set multi-cover problem is denoted as $\mathcal{C}^*_{\mathsf{SMC}}$. We can run essentially the same greedy algorithm to find a solution to the set multi-cover problem denoted as $\mathcal{C}_{\mathsf{SMC}}$: choose the next sensor $i$ that maximizes the *utility*, defined as the number of targets covered by $i$ among those targets whose coverage multiplicity has not been met. By [35], we know that $|\mathcal{C}_{\mathsf{SMC}}|/|\mathcal{C}^*_{\mathsf{SMC}}| \leq \delta$, where $\delta$ is the best approximation ratio achievable for a particular instance of set multi-cover. $\delta = O(\log m)$ in general, and in the geometric case [12] show one can obtain a $O(\log |OPT|)$ for set systems with bounded VC-Dimension, and an $O(1)$-approximation when covering points in 3D with half-spaces, and an $O(\log \log \log |OPT|)$-approximation for covering points by triangles in the plane.

**Lemma 6.2 (Lower Bound on** $\mathsf{opt}$**).** *If we require each target to be covered at least once in every window of size $f_j$, in some time slot we need to turn on at least $\lceil |\mathcal{C}^*_{\mathsf{SMC}}|/f_{\max} \rceil$ sensors. That is, $k^* \geq \lceil |\mathcal{C}^*_{\mathsf{SMC}}|/f_{\max} \rceil$.*

**Proof:** Consider the optimal schedule that satisfies the coverage frequency requirement and consider all the sensors turned on in a contiguous window of $f_{\max}$ slots, denoted by a multi-set $Y^*$. We know that $Y^*$ will cover each target at least $r_j = f_{\max}/f_j$ times. Thus $Y^*$ is a solution to the set multi-cover problem. This says that $|Y^*| \geq |\mathcal{C}^*_{\mathsf{SMC}}|$. By pigeon hole principle, there is at least one slot in which $|Y^*|/f_{\max}$ sensors are turned on. This gives a lower bound on $k^*$: $k^* \geq |Y^*|/f_{\max} \geq \lceil |\mathcal{C}^*_{\mathsf{SMC}}|/f_{\max} \rceil$. $\square$

Now we are ready for the scheduling algorithm that makes use of the set multi-cover solution $\mathcal{C}_{\mathsf{SMC}}$. Define $z = |\mathcal{C}_{\mathsf{SMC}}|$. We take $q$ random permutations $\pi_i$ of $\mathcal{C}_{\mathsf{SMC}}$, for $i = 1, \cdots, q$. For each of the permutation $\pi_i$, we stretch it (or shrink it) to fit in a schedule of period $f_{\max}$:

- If $z \leq f_{\max}$, we spread the elements of $\pi_i$ uniformly on $f_{\max}$ slots. The first element in $\pi_i$ is turned on in each of the first $f_{\max}/z$ slots, the second element in $\pi_i$ is turned on in each of the second $f_{\max}/z$ slots and so on.
- If $z > f_{\max}$, we turn on multiple sensors in one slot. The first $z/f_{\max}$ elements in $\pi_i$ are turned on in the first slot, the second $z/f_{\max}$ elements in $\pi_i$ are turned on in the second slot, and so on.

The schedule created by $\pi_i$ is named $Z_i$. Then we overlay all $q$ schedules on top of each other to create a schedule $Z$, in which we have at most $qz/f_{\max}$ sensors in each time slot. Now we analyze the approximation factor of this schedule.

**Theorem 6.3.** *The algorithm gives a $(q\delta, \beta)$-bicriteria approximation with high probability, where $q\beta = O(\log(n) + \log(m))$, $\delta$ is the approximation ratio for the set multi-cover problem.*

**Proof:** Let $B_j$ be the number of sensors in $\mathcal{C}_{\mathsf{SMC}}$ that cover target $j$. We know that $B_j \geq f_{\max}/f_j$ by the set multi-cover problem definition.

Now we consider a period of $f_{\max}$ time slots and partition it into windows of size $\beta f_i/2$. There are $2f_{\max}/(\beta f_i)$ windows. If in each window we have at least one sensor that covers target $j$, then target $j$ is covered in every window of size $\beta f_i$, which meets the $\beta$-relaxed coverage frequency requirement.

By the scheduling algorithm, in each schedule $Z_i$, there are $B_j$ sensors that are randomly placed in a total of $f_{\max}$ slots. Counting all $q$ schedules there are $qB_j$ sensors. Now we calculate the probability that a specific window $x$ of length $\beta f_j/2$ *does not* receive a sensor that covers $j$ – we say this window is not stabbed.

$$\text{Prob}\{\text{Window } x \text{ not stabbed for target } j\} \leq (1 - \frac{\beta f_j}{2f_{\max}})^{qB_j}.$$

Now, consider the probability that all windows are stabbed for all targets, that is, the $\beta$-relaxed coverage requirement is met. Let $\mu_j = 2f_{\max}/f_j$. $B_j \geq f_{\max}/f_j = \mu_j/2$.

$$\begin{array}{rl} & \text{Prob}\{\beta\text{-relaxed coverage requirement}\} \\ \geq & 1 - \sum_j \sum_x \text{Prob}\{\text{Window } x \text{ not stabbed for target } j\} \\ \geq & 1 - \sum_j \frac{2f_{\max}}{\beta f_j}(1 - \frac{\beta f_j}{2f_{\max}})^{qB_j} \\ \geq & 1 - \sum_j \frac{\mu_j}{\beta}(1 - \frac{\beta}{\mu_j})^{\mu_j q/2} \\ \geq & 1 - \sum_j \frac{\mu_j}{\beta}(1/e)^{q\beta/2} \\ \geq & 1 - \frac{2nf_{\max}}{\beta}(1/e)^{q\beta/2} \\ \geq & 1 - 1/n, \text{if } q\beta = O(\log n + \log f_{\max}) \end{array}$$

The total number of sensors we turn on is $qz/f_{\max} = q|\mathcal{C}_{\mathsf{SMC}}|/f_{\max} \leq q\gamma|\mathcal{C}^*_{\mathsf{SMC}}|/f_{\max} \leq q\gamma k^*$, in which the last inequality comes form the lower bound in Lemma 6.2. Therefore, this algorithm gives a $(q\gamma, \beta)$-bicriteria approximation with high probability, where $q\beta = O(\log(n) + \log f_{\max}) = O(\log n + \log m)$, $\gamma$ is the approximation ratio for the set multi-cover problem. □

## VII. Simulations & Experiments

### A. Simulations

We compare our algorithms to two baseline algorithms, Base_random, and Base_heuristic in geometric simulation. The baseline algorithms can be divided into two steps. First, for each target $j$, generate a periodic schedule with length $f_j$ that satisfies the target frequency requirement. The sensor chosen for each schedule is different: in Base_random, we randomly pick a sensor that covers this target, whereas Base_heuristic, selects the sensor which covers the most uncovered targets including this one. To avoid congestion in specific time slots, the schedule of each target randomly starts between 1 to $f_j$. Second, merge all schedules to create an overall schedule that satisfies frequency requirements of all targets.

We randomly generate 1000 targets and 64 sensors for the simulation. Each target is covered by at least one of the sensors. Each sensor is located at a point randomly generated in the plane. A sensor covers a target if the distance between them is at most one. Each target $j$ gets randomly assigned to a particular sensor $i$ and the location of the target is then generated randomly within the unit disk centered at sensor $i$. We introduce a parameter $\delta$ such that the frequency requirement of each target is chosen between 1 and $2^\delta$. The values of window sizes follow two different frequency distributions: uniform or exponential. In the uniform setting we sample the window size from 1 to $2^\delta$ with uniform probability; in the exponential setting the probability of choosing a larger window size is exponentially higher than others. That is, for a specific target $j$, $g \in \mathbb{N}_{\leq\delta}$ and

$$\mathrm{Prob}\{2^{g-1} \leq f_j \leq 2^g\} = 2^{g-1} \cdot 2^g / \sum_{1}^{\delta} (2^{g-1} \cdot 2^g).$$

Figure 2 shows a comparison between the algorithms. The $y$-axis represents the average of 50 simulations of the maximum number of sensors turned on in any time slot ($k$) in each setting. We run this experiment for $\delta = \{2, 3, ..., 6\}$. For fairness we set $\beta = 1$ and $q = 8$ to ensure coverage with high probability (where $q$ is the number of random permutations we overlay) in the bi-criteria algorithm.

Our results show that the combinatorial algorithm performs the best under a uniform frequency distribution in Figure 2a. However, if the frequency distribution is exponential, the geometric algorithm improves the most when the window size is more skewed to larger values. The reason is that the geometric algorithm has a smaller hidden constant factor than the combinatorial one and the effective grouping benefits (more targets with larger window size are grouped into the same cluster).

### B. Experiment

We have implemented a testbed to evaluate the algorithms. This testbed has four wireless camera nodes, and each of them is built based on the off-the-shelf BeagleBone low power development board. A 3.1 Megapixel Aptina CMOS digital image sensor MT9T111 and a USB Edimax EW-7811Un Wi-Fi adapter are plugged into the board. Events are emulated by an LED. The LED is controlled by a Raspberry Pi 2 development board.

For this experiment we set each time slot to be five seconds. Our schedule decides which cameras take a picture at each five seconds. Each LED $j$ is a target and is randomly turned on for $f_j$ slots multiple times. We check whether the cameras can capture every event. Our testbed is shown in Figure 3. Camera 1 and 3 cover three targets each, $\{a, b, c\}$, and $\{e, f, g\}$ respectively, and camera 2 covers $\{b, c, d, e, f\}$. Camera 4 only covers three targets $\{b, c, d\}$.

Generally, our three algorithms generate schedules with similar $k$. However, if we set targets with small window size ($f_g$ equals to 1 and others equal to 2) the bi-criteria algorithm obtains $k = 2$, while others obtain $k = 3$. The reason is that the geometric algorithm clusters targets with window size $= 1, 2$ in the same group. Also, the combinatorial algorithm requests each target $j$ to be covered within each $f_j/2$ time slots, which is also every slot. Therefore, they both cover all targets for each time slot which needs three sensors. On the other side, bi-criteria algorithm avoid this situation by taking directly the set multi-cover solution. For this simple setting, our scheduling algorithms reduce the number of active sensors in each time slot thus saving energy in the network.

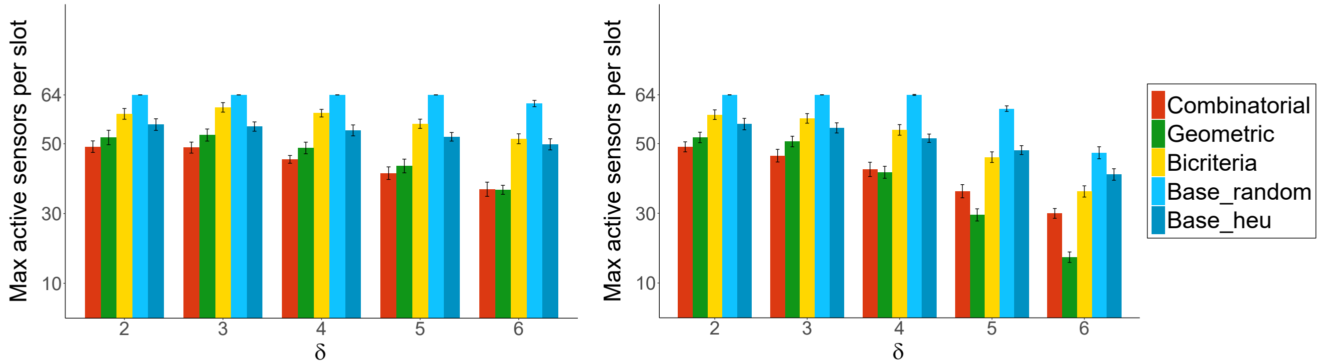
## VIII. Conclusion

In this paper we propose three algorithms for duty cycle scheduling of sensors with different design principles. They leverage properties and structures of the domain to construct efficient schedules with theoretical bounds while remain simple enough for implementation. The flexible formulation of coverage makes it appealing for many different applications in wireless sensor networks. In future work, we will explore the extension of this formulation to connectivity and load balancing.

## Appendix A
## Proofs for Section III

**Lemma 3.1.** *For any $T \geq f_{\max}$, $\mathsf{k}(\mathcal{F}^T_{\mathsf{opt}}) \leq 2\mathsf{k}(\mathcal{F}_{\mathsf{opt}})$.*

**Proof:** Let $\mathcal{F}_{\mathsf{opt}}$ denote the optimal schedule, and $F_{1,\mathsf{opt}}, \ldots, F_{n,\mathsf{opt}}$ denote the corresponding functions. We construct functions $F_1, \ldots, F_n$ each with period $T$ such that the cost of the corresponding schedule is at most 2 times that of the optimal schedule specified by $\mathcal{F}_{\mathsf{opt}}$. Our definition is as follows: For $t = 1, \ldots, T$ we define $F_i(t) = F_{i,\mathsf{opt}}(t) \vee F_{i,\mathsf{opt}}(t + T)$. Let $\mathcal{F}^T$ denote the




(a) Uniformly distributed window sizes

(b) Exponentially distributed window sizes

Fig. 2: Performance of algorithms in geometric domain against upper bound of maximum window size ($\delta$). Since every target must be covered once for every window, a larger window size allows it to be covered less frequently so the maximum number of active sensors in each time slot is reduced. All algorithms benefit when the targets are required to be covered less frequently.

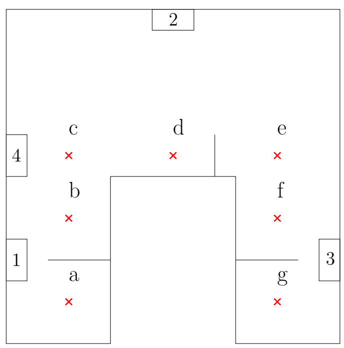



Fig. 3: Experiment testbed of covering targets using a camera network. There are 4 cameras (rectangles) with line-of-sight coverage and 7 point targets (represented by red crosses).

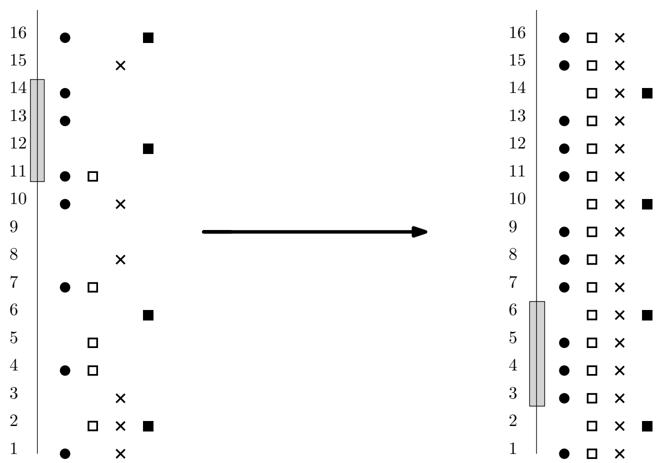



Fig. 4: Transforming a computable schedule to a periodic one. The example has $T = 4$. The disk, cross etc. represent different sensors, and their appearance represents the time slot they are scheduled. The interval $[11, 14]$ corresponds to a target covered by a sensor shown as a empty box. The corresponding interval $[3, 6]$ is also covered.

corresponding periodic schedule i.e., where we have extended these functions $F_i$ over the natural numbers periodically, i.e., for $t \in [\gamma T + 1, (\gamma + 1)T]$ we have $F_i(t) = F_i(t - \gamma T)$ for $\gamma \geq 1$. As we show later this schedule is valid (i.e., it satisfies the required coverage of the targets). For now, we show that its cost is at most 2 times the cost of optimal. See Figure 4 for an illustration of how to construct the periodic schedule from a given schedule.

For any $t$ in $\{1, \dots, T\}$, sensor $i$ is turned on iff it is turned on in $\mathcal{F}_{\mathsf{opt}}$ in either of times $t, t + T$. Thus the number of sensors turned on at any point in time is at most the sum of the number of sensors turned on at time $t, t + T$ which is at most $2\mathsf{k}(\mathcal{F}_{\mathsf{opt}})$.

Consider now any target $j$. Let $\mathcal{S}_j \subseteq \mathbb{N}_{\leq m}$ be the indices of the sensors covering target $j$. Consider an interval of length $f_j$. Since $T \geq f_{\mathsf{max}} \geq f_j$ such an interval can intersect at most two of the intervals $[1, T], [T + 1, 2T], \dots$. If such an interval lies inside $[\gamma T + 1, (\gamma + 1)T]$ then suppose it extends from $[\gamma T + x, \gamma T + x + f_j - 1]$ where $x \geq 1$. Now consider the corresponding interval $[x, x + f_j - 1]$. Since this interval is satisfied for target $j$ in the optimal schedule we have that at least one of the sensors with index in $\mathcal{S}_j$ is turned on during the times $[x, x + f_j - 1]$. By definition of $\mathcal{F}$ such a sensor will also be turned on once during the time interval $[\gamma T + x, \gamma T + x + f_j - 1]$. Now, consider an interval that intersects at least two periods, see Figure 4 for an example. Suppose it extends from $[\gamma T - f_j + x, \gamma T + x]$. In this case consider the interval $[T - f_j + x, T + x]$ in the optimal schedule. A sensor in $\mathcal{S}_j$ is turned on at least once during this interval. Suppose this was sensor $i$ turned on during some time in $[T - f_j + x, T]$ then as per the definition of $\mathcal{F}$ it is turned on during the corresponding interval in $\mathcal{F}$ as well. On the other hand, if it was turned on during $[T + 1, T + x]$ then, it will be turned on during $[1, x]$ in $\mathcal{F}$ (since $F_i(t) = F_i(t) \vee F_i(t + T)$ for $t$ in $1, \dots, T$), and by periodicity also during $[T + 1, T + x]$. So a sensor in $\mathcal{S}_j$ will be turned on at least once during $[T - f_j + x, T + x]$. By periodicity, that sensor will be turned on during $[\gamma T - f_j + x, \gamma T + x]$. This proves the validity of our periodic schedule. $\square$

**Acknowledgement:** E. Arkin would like to acknowledge NSF support through CCF-1526406. K. Liu, H. Yang and J. Gao would like to acknowledge the support through NSF DMS-1418255, CCF-1535900, CNS-1618391 and AFOSR FA9550-14-1-0193. K. Liu, H. Yang and S. Lin are grateful for support through DE-EE0007682, NSF CNS 1553273, CNS 1536086, CNS 1463722, and IIS 1460370.

## REFERENCES


[1] A. K. Abu-Affash, P. Carmi, M. J. Katz, and G. Morgenstern. Multi cover of a polygon minimizing the sum of areas. *International Journal of Computational Geometry & Applications*, 21(06):685–698, 2011.

[2] R. Acharyya, G. K. Das, et al. Unit disk cover problem. *arXiv preprint arXiv:1209.2951*, 2012.

[3] P. K. Agarwal and J. Pan. Near-linear algorithms for geometric hitting sets and set covers. In *Proceedings of the thirtieth annual symposium on Computational geometry*, page 271. ACM, 2014.

[4] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy. Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks. *Computer Networks*, 52(12):2419–2431, 2008.

[5] X. Bai, Z. Yun, D. Xuan, T. H. Lai, and W. Jia. Deploying four-connectivity and full-coverage wireless sensor networks. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, April 2008.

[6] S. Bandyapadhyay and K. Varadarajan. Approximate clustering via metric partitioning. *arXiv preprint arXiv:1507.02222*, 2015.

[7] N. Bansal and K. Pruhs. Weighted geometric set multi-cover via quasi-uniform sampling. In *European Symposium on Algorithms*, pages 145–156. Springer, 2012.

[8] R. Bar-Yehuda and D. Rawitz. A note on multicovering with disks. *Computational Geometry*, 46(3):394–399, 2013.

[9] S. Bhowmick, K. Varadarajan, and S.-K. Xue. A constant-factor approximation for multi-covering with disks. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 243–248. ACM, 2013.

[10] H. Brönnimann and M. T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

[11] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers*, 51(12):1448–1453, 2002.

[12] C. Chekuri, K. L. Clarkson, and S. Har-Peled. On the set multicover problem in geometric settings. *ACM Transactions on Algorithms (TALG)*, 9(1):9, 2012.

[13] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[14] E. A. Feinberg and M. T. Curry. Generalized pinwheel problem. *Mathematical Methods of Operations Research*, 62(1):99–122, 2005.

[15] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions?ii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

[16] A. Ghosh and S. K. Das. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, 4(3):303 – 334, 2008.

[17] Y. Gu, Y. Ji, J. Li, and B. Zhao. Qos-aware target coverage in wireless sensor networks. *Wireless Communications and Mobile Computing*, 9(12):1645–1659, 2009.

[18] T. He, S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 270–283. ACM, 2004.

[19] T. A. Hobson and I. V. L. Clarkson. Collaborative sensor scheduling for space situational awareness. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 640–643. IEEE, 2012.

[20] D. S. Hochbaum and A. Pathria. Analysis of the greedy approach in problems of maximum k-coverage. *Naval Research Logistics*, 45(6):615–627, 1998.

[21] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan. Adaptive duty cycling for energy harvesting systems. In *Proceedings of the 2006 international symposium on Low power electronics and design*, pages 180–185. ACM, 2006.

[22] C.-F. Huang and Y.-C. Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2Nd ACM International Conference on Wireless Sensor Networks and Applications*, WSNA '03, pages 115–121, New York, NY, USA, 2003. ACM.

[23] H. Huang, C.-C. Ni, X. Ban, J. Gao, A. T. Schneider, and S. Lin. Connected wireless camera network deployment with visibility coverage. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1204–1212. IEEE, 2014.

[24] H. Jaleel, A. Rahmani, and M. Egerstedt. Duty cycle scheduling in dynamic sensor networks for controlling event detection probabilities. In *Proceedings of the 2011 American Control Conference*, pages 3233–3238. IEEE, 2011.

[25] P. Kulkarni, P. Shenoy, and D. Ganesan. Approximate initialization of camera sensor networks. In *European Conference on Wireless Sensor Networks*, pages 67–82. Springer, 2007.

[26] J. Li and Y. Jin. A ptas for the weighted unit disk cover problem. In *International Colloquium on Automata, Languages, and Programming*, pages 898–909. Springer, 2015.

[27] K. S. Liu, J. Gao, S. Lin, H. Huang, and B. Schiller. Joint sensor duty cycle scheduling with coverage guarantee. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 11–20. ACM, 2016.

[28] A. Makhoul and C. Pham. Dynamic scheduling of cover-sets in randomly deployed wireless video sensor networks for surveillance applications. In *2009 2nd IFIP Wireless Days (WD)*, pages 1–6. IEEE, 2009.

[29] S. Meguerdichian and M. Potkonjak. Low power 0/1 coverage and scheduling techniques in sensor networks. Technical report, Citeseer, 2003.

[30] V. P. Munishwar and N. B. Abu-Ghazaleh. Coverage algorithms for visual sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 9(4):45, 2013.

[31] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, Inc., New York, NY, USA, 1987.

[32] A. Saipulla, C. Westphal, B. Liu, and J. Wang. Barrier coverage of line-based deployed wireless sensor networks. In *INFOCOM 2009, IEEE*, pages 127–135, April 2009.

[33] A. V. Savkin, F. Javed, and A. S. Matveev. Optimal distributed blanket coverage self-deployment of mobile wireless sensor networks. *IEEE Communications Letters*, 16(6):949–951, June 2012.

[34] L. Shi, M. Epstein, B. Sinopoli, and R. M. Murray. Effective sensor scheduling schemes in a sensor network by employing feedback in the communication loop. In *2007 IEEE International Conference on Control Applications*, pages 1006–1011. IEEE, 2007.

[35] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[36] B. Wang. Coverage problems in sensor networks: A survey. *ACM Comput. Surv.*, 43(4):32:1–32:53, Oct. 2011.

[37] L. Wang and Y. Xiao. A survey of energy-efficient scheduling mechanisms in sensor networks. *Mobile Networks and Applications*, 11(5):723–740, 2006.

[38] Y. Wang and G. Cao. On full-view coverage in camera sensor networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1781–1789, April 2011.

[39] L. Xu, J. Cao, S. Lin, H. Dai, X. Wu, and G. Chen. Energy-efficient broadcast scheduling with minimum latency for low-duty-cycle wireless sensor networks. In *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*, pages 163–167. IEEE, 2013.

[40] H. Yoo, M. Shim, and D. Kim. Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks. *IEEE communications letters*, 16(2):202–204, 2012.

[41] C. Yu and G. Sharma. Camera scheduling and energy allocation for lifetime maximization in user-centric visual sensor networks. *IEEE Transactions on Image Processing*, 19(8):2042–2055, 2010.

[42] Z. Zhou, S. Das, and H. Gupta. Connected k-coverage problem in sensor networks. In *Computer Communications and Networks, 2004. ICCCN 2004. Proceedings. 13th International Conference on*, pages 373–378, Oct 2004.