Joint Sensor Duty Cycle Scheduling with Coverage Guarantee

Kin Sum Liu∗, Jie Gao †, Shan Lin †, Hua Huang †, Brent Schiller †

ABSTRACT
Using optical sensors for indoor monitoring has been widely adopted in many smart building applications. An important design problem in this space is to explore the tradeoff between energy consumption and coverage quality. While it is important that the sensors achieve full coverage (i.e., every interesting target point can be monitored by at least one sensor), it is often a waste of energy to keep sensors on all the time as events are typically stochastic and rare and most of the time the sensors are on idle monitoring. In this paper we design efficient sensor duty cycles to ensure that any target point of interest is still covered sufficiently frequently while only a subset of sensors are kept on at any time slot. We denote by the maximum dark length for each target point \( p \) as the maximum duration in which \( p \) is covered at least once. We formulate two optimization problems: the min max dark length scheduling and the min average dark length scheduling. For both versions we provide efficient, practical algorithms with provable approximation guarantees. The two algorithms have been tested on two real testbed scenarios to evaluate its efficiency and coverage quality.

CCS Concepts
• Networks → Sensor networks; • Theory of computation → Scheduling algorithms;

Keywords
Sensor coverage, duty cycle scheduling, energy efficiency

1. INTRODUCTION
Optical networks have been widely deployed in modern buildings for various indoor monitoring applications, such as motion detection using passive infrared sensors [1], safety surveillance using camera networks [2], window system control using visible light sensors [3], and indoor localization using infrared systems [4]. By the nature of the directionality of signals used in these applications, it is often a nontrivial problem to deploy the sensors to obtain full coverage of the domain. Once the sensor locations are determined, it is often assumed that all sensors are kept on all the time. Keeping all sensors on may consume a significant amount of energy especially when the events happen in a stochastic manner. Most of sensing energy is wasted on idle monitoring, which is inefficient and sometimes unnecessary. There is a need to reduce energy consumption on idle sensing as long as the application requirements are met.

For most of the optical sensing applications such as surveillance and target localization, full coverage is important so that every location of the area can possibly be monitored. Fortunately, these applications may not require every location in the domain to be covered all the time, because 1) the physical events that appear in the domain usually last for a certain period of time. For example, an intruder usually passes through a camera’s view in a few seconds; the shadow cast by human movement lasts for hundreds of milliseconds; 2) these events can be collaboratively detected by a network of sensors. One location can be covered by multiple sensors. Although an individual sensor can miss an event, another sensor may still catch it. Therefore, our goal is to design efficient optical sensing schedules to turn sensors on and off, achieving a good tradeoff between detection delay and sensing energy consumption.

In this paper we formulate the sensor duty cycle scheduling problem in a general manner. We assume a domain \( D \) and a set of \( n \) sensors that collectively cover it. We do not make any special assumptions on sensing coverage range. Instead each node \( g \) is associated with a set \( P(g) \) representing the region of coverage. This allows the results in this paper to be applied in the most general setting, incorporating a variety of sensing models (omni-directional models or non-isotropic ones such as line-of-sight constraints). We assume that time is slotted and at each time slot, \( k \) out of \( n \) sensors are turned on. \( k \) is a parameter that can be used to tune the expected energy consumption level. The problem is to decide which set of sensors to turn on in each slot and in what order such that each target point is covered sufficiently frequently. The set of target points can be either discrete or continuous.

We discussed two different evaluation metrics on the quality of coverage. The first version, the min max dark length...
scheduling, minimizes the value $C$ such that in every con-
secutive $C$ slots every target is covered at least once. This
provides worst case guarantee on all target points. We show
that this problem in general is NP-hard, but if the set of
sensor nodes $V$ is a minimal set – meaning every node cov-
ers a unique target that is not covered by any other sensors
(i.e., every sensor is needed for full coverage), then round
robin on this set $V$ is optimal.

In the second evaluation metric, we try to minimize the
weighted average of the max dark length for each target in
the domain $D$. This has the freedom of being not tailored
by the worst case scenario but rather measures average cov-
erage quality. The weights can also be used to provide dif-
f erentiation on the importance of different targets. For this
problem we provide, again, an easy and intuitive algorithm
with an $O(1)$-approximation when $k = \Omega(\log n/\log \log n)$
and $O(\log n/\log \log n)$-approximation otherwise. The algo-
rithm is basically round robin on a random permutation in
which there are $\tau(g)$ copies of each sensor $g$. The parameter
$\tau(g)$ for each $g \in V$ can be solved by convex optimization.

In addition to providing theoretical guarantees of the algo-
rithms, we have also conducted both simulations and testbed
experiments to evaluate our algorithms. Under different de-
ployment topology and schedules, we can achieve a signif-
icant amount of energy saving on sensing with a modest
decay of coverage quality.

The contributions of this work are listed as follows:

- To the best of our knowledge, this is the first work to
investigate general sensor scheduling of optical sensor
networks with theoretical guarantees. The algorithms
are sufficiently simple and can be implemented in a
real system.

- With extensive simulations and system experiments
on an infrared localization system and a camera net-
work, our algorithms can reduce energy consumption
to about 1/3 with a maximum detection delay of 5
seconds.

In the rest of the paper we first survey related work on
sensor duty cycle scheduling. Then we present our theoret-
cal analysis and algorithm design under two optimization
metrics. We report the simulation results and testbed eval-
uation last.

2. RELATED WORK

There are three major lines of research relevant to this
work: scheduling algorithms in sensor networks, scheduling
algorithms in camera networks, and coverage algorithms.

Scheduling Algorithms in Sensor Networks. Schedul-
ing algorithms are studied in a wide range of computing
systems, such as wireless networks [5, 6, 7, 8, 9, 10, 11, 12,
13], sensor systems [14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24, 25, 26], and energy systems [27, 28]. Among these al-
gorithms, duty cycle based scheduling algorithms have been
widely adopted in communication [29, 30, 31] and energy
management [32, 33] in low power systems. Various duty
cycled sensing algorithms have also been developed with
different detection and energy saving objectives. In [34],
authors designed algorithms to adjust a node’s duty-cycle
according to the current amount of residual energy. In [35],
a probabilistic scheduling of the duty cycles is designed to
ensure event detection probabilities. These works usually
assume that a sensor is switched on-and-off once every cer-
tain period of time, which is easy to implement. What has
been missing in such work is the lack of coordination among
different sensors. In particular, when multiple sensors have
significantly overlapping sensing ranges it would be more ef-
icient to consider a joint optimization of multiple sensors at
the same time, to further reduce energy consumption while
maintaining the same level of coverage quality. This is also
the main aspect that our work differs. The challenge of our
problem is to consider the collaborative sensing on the tar-
get areas at a network scale with monitoring performance
guarantees and reduced energy consumption.

Scheduling Algorithms in Camera Networks. There
are a few papers that discuss specifically scheduling and con-
tral in optical sensor or camera networks. Authors of [36,
37] presented collaborative sensing algorithms using the field
of view sensing model. In [38, 39, 40, 41], authors designed
algorithms to determine the locations of cameras given dif-
f erent deployment requirements. In [42], authors formulate
an optimization problem that aims to maximize the dura-
tion of coverage for the most critical part of the monitored
region. Our work falls in this category as well, although we
do not limit our algorithm to be just for camera scheduling.
We allow a general definition of sensing ranges and in addi-
tion provide worst case guarantees for coverage quality. This
complements and enriches existing work in this domain.

Coverage Algorithms. There are a large body of re-
search literature on coverage algorithms in wireless sensor
networks. Some of these works consider the sensor selection
and scheduling problems to meet specific coverage goals [43].
In [44], an analytical framework for object detection in sen-
or networks is described, which allows mathematical anal-
ysis on average-case object detection quality in random and
synchronized sensing schedules. Authors of [45] investigated
the problem of sensor selection for minimizing error covari-
ance using mobile sensors. In [46], authors presented a cen-
tralized and a distributed algorithm to select the minimal
number of nodes to monitor $p$-percent of the monitored area.
These works usually assume circular sensing range, which is
not suitable for optical sensor network or camera networks
discussed in this paper.

3. SCHEDULING PROBLEM STATEMENT

We define the scheduling problem in a generic manner.
Given a domain $D$ to be monitored and a set of $n$ sensor
nodes $V$, we define by $P(g)$ the region of coverage by a
sensor node $g \in V$. Without loss of generality we can assume
that the union of coverage ranges by all nodes in $V$ is $D$. $D$
can be either a continuous domain (for example when the
entire room needs to be monitored) or a discrete set of tar-
get locations of interest. We would like to design an efficient
duty-cycle schedule of the sensor nodes such that $D$ is still
well covered and the total amount of time that the sensors
stay on/active is reduced. This problem explores the trade-
off between coverage quality and energy consumption in the
design space.

In our definition, we assume that time is slotted and at
any slot, only $k$ nodes are active. $k$ is a parameter that
d controls the amount of energy consumption expected in the
system and is assumed to be given. We will need to choose
which set of nodes to turn on and in what order to ensure
that the coverage quality remains high. Since not all sensors are on all the time, a point \( p \in D \) is not necessarily covered at all times. Thus we say \( p \) is **lighted up** if any of the active nodes at this slot covers \( p \); and **dark** otherwise. Naturally, we would like the dark period to be as short as possible. Rigorously, we define for each point \( p \) we would like the dark period to be as short as possible.

![Figure 1](image.jpg)

**Figure 1.** An example of four target points \( p_1, p_2, p_3, p_4 \) and three sensor nodes \( g_1, g_2, g_3 \). In the second problem formulation, we try to achieve every contiguous \( T(p) \) time slots. It is exactly one plus the maximum length of any contiguous dark period. An example of \( T(p) \) is shown in Figure 1.

\[
\begin{array}{cccccccc}
S & g_1 & g_2 & g_3 & g_2 & g_1 & g_2 & g_3 & g_2 & g_3 \\
p_1 & p_1 & p_2 & & p_1 & & p_2 & & \\
p_2 & p_2 & p_2 & p_2 & p_2 & & & & \\
p_3 & p_3 & p_3 & p_3 & & & & & \\
p_4 & & & & & p_4 & & &
\end{array}
\]

Now we formulate two research problems.

1. **Min Max Dark Length Scheduling.** Find a schedule \( S \) as an ordered list in which \( S_i \) is the set of nodes staying on in the \( i \)th time slot, \( |S_i| \leq k \) such that the maximum dark length for any point \( p \in D \) is minimized:

\[
\min_{p \in D} \max T(p).
\]

2. **Min Average Dark Length Scheduling.** Find a schedule \( S \) as an ordered list in which \( S_i \) is the set of nodes staying on in the \( i \)th time slot, \( |S_i| \leq k \) such that the weighted average of maximum dark length for all points \( p \in D \) is minimized:

\[
\min \sum_{p \in D} w(p) \cdot T(p).
\]

Here \( w(p) \) is a weight parameter and can be set higher for locations that are more important in applications.

In the first problem we minimize the maximum dark length for all points in the domain, achieving a worst case guarantee. In the second problem formulation, we try to achieve a good average and further we can use the weight \( w(\cdot) \) to allow differentiation of different target points in \( D \). In the following we will discuss the two versions separately.

### 4. MIN MAX DARK LENGTH SCHEDULING

The intuitive explanation for this objective function is that we want to cover every point regularly and we want to minimize the maximum time period that some point is uncovered, when only \( k \) nodes can be active during any time slot. Define \( C \) as the objective function \( \max_{p \in D} T(p) \). First we look at an intuitive schedule, by simply taking every \( k \) nodes in an arbitrary sequence of the \( n \) nodes in a Round Robin manner. That is, the first \( k \) nodes stay active in slot 1, the next \( k \) nodes stay active in slot 2 and so on. When \( n \) is not a multiple of \( k \), the last time slot is filled up by arbitrarily other nodes. This schedule with period of \( \lceil \frac{n}{k} \rceil \) is then repeated indefinitely.

**Theorem 4.1.** Given a domain \( D \) and a set of \( n \) nodes \( V \), at any time only \( k \) nodes are operating simultaneously, a round robin scheduling has max dark length at most \( \lceil \frac{n}{k} \rceil \).

**Proof.** For every \( \lceil \frac{n}{k} \rceil \) slots in an arbitrary sequence of the \( n \) nodes, each sensor node appears at least once in the schedule. Since all nodes together cover \( D \), every point of \( D \) is covered at least once in one period. Therefore, \( \forall p \in D, T(p) \leq \lceil \frac{n}{k} \rceil \). \( \square \)

Given a set of node \( V \), we say \( V \) is a **minimal cover**, if each node of \( V \) covers a unique target point (which is not covered by any other nodes of \( V \)). In other words, every node in \( V \) is crucial and cannot be omitted if we want \( D \) to be covered. In this case, the round robin scheduling is optimal. To show that we first state a useful property of an optimal schedule.

**Theorem 4.2.** If \( V \) is a minimal cover of \( D \), the Round Robin schedule is optimal.

**Proof.** We just need to show that the optimal scheduling achieves dark length no less than \( \lceil \frac{n}{k} \rceil \). Suppose \( S^* \) is the optimal schedule and suppose \( C^* \leq \lceil \frac{n}{k} \rceil \). Therefore, the total number of nodes that are active in the crucial sub-schedule is \( |\cup_{i=1}^{\lceil n/k \rceil} S_i| \leq kC^* < n \). Thus at least one sensor node (say node \( x \)) is not active in this period of time. The target node \( p \) that is uniquely covered by \( x \) is not covered. This means \( p \) must have dark length \( T(p) > C^* \), which is a contradiction that \( C^* \) is the optimal value. \( \square \)

When \( V \) is not a minimal cover, the challenge is to find a minimum number of nodes in \( V \) that provide full coverage of the domain \( D \), denoted by the **minimum cover**, which by itself is a classical set cover problem and thus is NP-hard.
But once this minimum set cover solution is given, doing a periodic round robin scheduling is the optimal.

**Theorem 4.3 (NP-hardness).** Given a domain $D$, a set of $n$ nodes $V$ among which $k$ nodes operate simultaneously, we have two statements:

1. Repeating a Round Robin schedule on a minimum cover is the optimal schedule that minimizes the max dark length of all points of $D$.

2. But finding this optimal schedule is NP-hard.

**Proof.** To prove the first statement, suppose $S^*$ is an optimal schedule. Recall that the nodes that are active in a crucial subschedule cover all points of $D$. This means that $C^*$ is no smaller than $\lceil n^*/k \rceil$, where $n^*$ is the size of the minimum cover. Since a round robin schedule using a minimum cover achieves max dark length of $\lceil n^*/k \rceil$, by Theorem 4.1, the optimality follows.

For the second claim, we use a reduction from the Set Cover problem. Given an instance of the Set Cover Problem $(U, S)$, $U$ is the universe and $S$ contains sets $\{G_i\}$, where $G_i \subset U$. We take $D$ as the universe, and create a set of nodes where node $i$ covers the set $G_i \in S$. We also set $k = 1$ in this reduction – if this special case of $k = 1$ is NP-hard it is obvious that the general problem is NP-hard as well. Our goal is to show that if we can solve the max dark length problem then we find the optimal set cover. For that, we will use two steps.

1. Claim 1: Any optimal schedule $S^*$ can be turned into a periodic one.

2. Claim 2: For a periodic optimal schedule, the nodes that ever appear in the schedule is a minimum cover of $D$.

Therefore, any algorithm that solves the max dark length problem basically solves the Set Cover problem.

For Claim 1, using the same setting as before, we can simply repeat the crucial subschedule $S = S_iS_{i+1}...S_{j-1}$ indefinitely which achieves the same optimal value.

For Claim 2, recall that the union of the nodes that are active in the crucial subschedule covers all points of $D$. Since $k = 1$ this must be a minimum cover — otherwise we take round robin on a minimum cover and we get a contradiction on optimality.

To summarize the discussion on this metric, if $V$ is a minimal cover of $D$, we can find the optimal schedule immediately by repeating the Round Robin sub-schedule. Otherwise, we need to find a minimum cover of $D$ to obtain the optimal schedule which is NP-hard, but has an $O(\log |D|)$ approximation by a greedy algorithm.

### 5. MIN AVERAGE DARK LENGTH SCHEDULING

While the previous version provides the worst case guarantee of coverage quality for any point in $D$, it is possible that a small portion of $D$ controls the worst case value. For example, suppose we have three nodes $g_1, g_2, g_3$, $g_1$ covers only one target $p_1$; $g_2$ covers one target $p_2$; and $g_3$ covers four targets $p_3, p_4, p_5, p_6$. In the optimal min max schedule we will rotate between these three nodes and each target has a dark length of 3. But in the min average optimization we would favor $g_3$ (i.e., schedule it more frequently) as its coverage range is greater. To see that, just verify that the average dark length for repeating the schedule $g_1, g_3, g_2, g_3$ is $3/3 \leq 3$. The min average optimization is able to adjust to the differences in coverage range and in addition we can use the weight function $w(\cdot)$ to provide better coverage quality for targets of high importance. We assume that the weights are integer values.

Before we present our algorithm we first analyze properties of the optimal solution.

**Theorem 5.1 (Periodicity).** There is a schedule for min average dark length problem which is periodic and is a 2-approximation.

**Proof.** We define $A^* = \sum_{p \in T} w(p) \cdot T(p)$ as the optimal value. We also define $A(t)$ as the weighted average of the schedule in the first $t$ time slots, i.e., $A(t) = \sum_{p \in T} w(p) T_t(p)$, where $T_t(p)$ is the maximum dark length in the first $t$ slots for target point $p$. $A^*$ and $A(t)$ for any $t$ are integers. In fact the sequence $A(t)$ when $t$ goes to infinity will converge to $A^*$. For any target point $p$ we notice that $T_t(p)$ is a non-decreasing function when $t$ goes to infinity. Thus $A(t)$ is also non-decreasing. For a non-decreasing sequence to converge to a fixed value $A^*$, it means that for any $\varepsilon > 0$ there is a value $T$ such that for any $t \geq T$, $|A(t) - A^*| \leq \varepsilon$. If we take $\varepsilon = 1/2$ we know that $A(t) = A^*$ for any $t \geq T$, since both $A(t)$ and $A^*$ are integers. By the same argument we can say that $T_t(p) = T(p)$ for any $t \geq T$.

Now suppose the set of sensors active in the $t$th slot is denoted as $S_t$. We take the following sequence and repeat indefinitely:

$S_1, S_2, \cdots, S_{T-1}, S_T, S_{T-1}, \cdots, S_2.$

The new schedule $S'$ is a periodic schedule. Now we argue that for each target point $p \in D$, $T'(p) \leq 2T(p)$, where $T'(p)$ is the maximum dark length in the new schedule $S'$.

Recall that for each point $p$, $T(p)$ is realized in the first $T$ slots. We take $j \leq T$ to be the highest value such that $S_j$ covers $p$, and $i \geq 1$ to be the smallest value such that $S_i$ covers $p$. Clearly, $T - j \leq T(p)$; $i - 1 \leq T(p)$. Therefore, consider the dark gap formed by

$S_{j+1}, S_{j+2}, \cdots, S_{T-1}, S_T, S_{T-1}, \cdots, S_{j+1}.$

The dark length is no greater than $2T(p)$. Similarly, for the dark gap formed by

$S_{i-1}, S_{i-2}, \cdots, S_2, S_1, S_2, \cdots, S_{i-1}$

the dark length is no greater than $2T(p)$. This proves the claim that the new schedule $S'$ is a 2-approximation.

Motivated by the above theorem, we will only consider periodic schedules. The best periodic schedule is a 2-approximation of the optimal. In the following we first look at a special case when the schedule is formed by taking a permutation of the $n$ nodes and performing round-robin on the sequence. Then we generalize the results to a periodic schedule without the permutation requirement.

### 5.1 Scheduling with a Permutation

In this part we first look at a special case, when the schedule is determined by doing round robin on a permutation $\pi$ of the nodes. We can define the optimal schedule with this
additional restriction as the one that minimizes the objective function \( \sum_{p \in \mathcal{D}} w(p) \cdot T(p) \). We denote the permutation used in the optimal schedule as \( \pi^* \).

\[
G(p) \subseteq \{g_1, g_2, g_3, g_4, g_5, g_6\} \quad m = |V \setminus G(p)| = 10 \text{ balls}
\]

Assume \( G(p) \) appears in the permutation with this order \( g_1, g_3, g_4, g_6, g_5, g_2 \).

Figure 3. Repeat a random permutation of nodes in \( \mathcal{V} \).

Consider a point \( p \in \mathcal{D} \), define \( G(p) \) as the collection of nodes of \( \mathcal{V} \) that cover \( p \). Therefore the schedule with a permutation \( \pi \) can be represented as placing the nodes in the order of \( \pi \) on a circle. See Figure 3. The nodes of \( G(p) \) are spread out along the circle. Therefore the maximum dark length \( T(p) \) is precisely one plus the maximum gap between these nodes of \( G(p) \) on the circle. Suppose \( |G(p)| = h(p) \). Thus the nodes that are not in \( G(p) \) need to be placed in the gaps between the nodes of \( G(p) \) on the circle. This gives a trivial lower bound on \( T(p) \) for any permutation \( \pi \):

\[
T_\pi(p) \geq \left[ 1 + \frac{|V \setminus G(p)|}{k \cdot |G(p)|} \right] = 1 + \left[ \frac{n - h(p)}{k \cdot h(p)} \right] = B(p).
\]

Therefore,

\[
A_\pi = \sum_{p \in \mathcal{D}} w(p) T_\pi(p) \geq \sum_{p \in \mathcal{D}} w(p) \cdot \left[ 1 + \left[ \frac{n - h(p)}{k \cdot h(p)} \right] \right] = B.
\]

The lower bound on the right hand side of the above inequality is usually impossible to achieve. But we show that by using a random permutation \( \pi \) we can actually get a good approximation to the lower bound \( B \).

We now compute the expectation of the objective function with respect to running round robin on a random permutation using linearity of expectation:

\[
E[A_\pi] = E\left[ \sum_{p \in \mathcal{D}} w(p) T_\pi(p) \right] = \sum_{p \in \mathcal{D}} w(p) \cdot E[T_\pi(p)]
\]

To analyze \( E[T_\pi(p)] \) we will relate to the classical ‘Balls and Bins’ problem. For a particular target \( p \in \mathcal{D} \), let \( \ell = |G(p)| \) and \( m = |V \setminus G(p)| \). \( B(p) \geq 1 + m/(k\ell) \). Then \( T_\pi(p) \) under a random permutation is related to the maximum load of throwing \( m \) balls into \( \ell \) bins independently and randomly. Figure 3 illustrates this connection.

The maximum load in the balls and bins game has been studied a lot in the literature [47, 48]. We will basically state the results.

- If \( m = \Omega(\ell \log \ell) \), then \( E[T_\pi(p)] \leq 1 + O(1) \cdot m/(k\ell) = 1 + O(1) \cdot B(p) \) is the best possible (assuming balls are uniformly put in the bins).

- If \( m = o(\ell \log \ell) \), then \( E[T_\pi(p)] \leq (1 + O(1)) \cdot \frac{\ell}{k \log \ell} \leq O(1) \cdot \frac{n \log n}{\ell \log \log \ell} \).

If we take \( k \geq \log n / \log \log n \), then from the above results we see that \( E[T(p)] \leq O(1) \cdot B(p) \). Otherwise, \( E[T_\pi(p)] \leq O(\log n / \log \log n) \cdot B(p) \).

**Theorem 5.2 (Random Permutation).** Given \( n \) nodes \( V \) that cover a domain \( \mathcal{D} \), the solution of doing round robin of a random permutation \( \pi \) gives a \( \alpha \)-approximation for the min average dark length scheduling problem with additional restriction that only round robin of a permutation is used. Here \( \alpha = O(1) \) if \( k \geq \log n / \log \log n \), and \( \alpha = O(\log n / \log \log n) \) otherwise.

### 5.2 Approximation Algorithm

The observation from the previous section can be used to provide an approximation algorithm for the general case. We consider the optimal periodic schedule. By Theorem 5.1 if we have a \( \beta \) approximation of the best periodic schedule then it is a \( 2\beta \) approximation for the optimal solution (possibly non-periodic).

Suppose \( S \) is a periodic schedule. We consider the schedule of one period, denoted by \( S \). \( S \) is not necessarily a permutation of \( \mathcal{V} \). In fact, certain nodes in \( \mathcal{V} \) may not show up in \( S \) at all, while some other nodes of \( \mathcal{V} \) may show up multiple times in \( S \). Denote by \( \tau(g) \) the number of times that a node \( g \in \mathcal{V} \) appears in one full cycle. \( \tau(g) \) is always an integer. \( \tau(g) \geq 0 \). Denote by \( T \) the length of one full cycle in the schedule \( S \). Then we know \( \sum_{g \in \mathcal{V}} \tau(g) = T \cdot k \).

Suppose we are able to guess the correct value of \( \tau(g) \) and \( T \), then we can simply reduce the problem to the special case studied in Theorem 5.2 and get an \( O(\alpha) \) approximation. In the following we show how to obtain good values for \( \tau(g) \) and \( T \) to achieve that.

If we define \( f(g) = \frac{\tau(g)}{kT} \) as the frequency of occurrence of \( g \) in the schedule \( S \), we have

\[
\sum_{g \in \mathcal{V}} f(g) = \sum_{g \in \mathcal{V}} \frac{\tau(g)}{kT} = 1.
\]

For each point \( p \in \mathcal{D} \), define the number of times that \( p \) is covered in one cycle by \( X(p) := \sum_{g \in G(p)} \tau(g) \). Thus, \( T(p) \), the largest dark length in the schedule, is lower bounded by the average separation if all occurrences of nodes that cover \( p \) are uniformly spread out:

\[
T(p) \geq \frac{T}{X(p)} = \frac{1}{\sum_{g \in G(p)} \tau(g) / T} = \frac{1}{k \cdot \sum_{g \in G(p)} f(g)}
\]

The objective function for \( S \) is lower bounded in the following inequality:

\[
A = \sum_{p \in \mathcal{D}} w(p) T(p) \geq \sum_{p \in \mathcal{D}} \frac{w(p)}{k \cdot \sum_{g \in G(p)} f(g)} = B
\]

Notice that in the above discussion we didn’t make any assumption on what \( S \) is. Take the best periodic schedule \( S^* \) the inequality holds:

\[
A^* = \sum_{p \in \mathcal{D}} w(p) T^*(p) \geq \sum_{p \in \mathcal{D}} \frac{w(p)}{k \cdot \sum_{g \in G(p)} f^*(g)} = B(f^*(g), \forall g \in \mathcal{V})
\]

The right hand side (i.e., \( B \)) is a lower bound of the optimal value, in which the only parameter is the frequency \( f(g) \) of each node \( g \). In the following we use two steps to get a good approximation.
1. We show that we can minimize the value $B$ in polynomial time by choosing proper values of $f(g)$ for all $g \in \mathcal{V}$.

2. We show an algorithm that achieves a $\beta$ approximation of the minimum possible $B$, which immediately gives us a $\beta$ approximation to the best periodic schedule, or a $2\beta$ approximation to the optimal schedule.

**Lemma 5.3.** The following optimization problem is convex and thus can be solved in polynomial time:

$$\min \sum_{p \in D} \sum_{g \in \mathcal{C}(p)} \frac{w(p)}{f(g)}$$

subject to

$$\sum_{g \in \mathcal{C}} f(g) = 1, f(g) \geq 0$$

**Proof.** In this minimization problem the only parameters are the values $f(g)$ for $g \in \mathcal{V}$. We only need to show that the objective function is convex. Since the sum of convex functions is still convex, we only need to verify that $1/\sum_{g \in \mathcal{C}(p)} f(g)$ is convex. In other words,

$$\frac{1}{\sum_{g \in \mathcal{C}(p)} [f(g) + f'(g)]} \leq \frac{1}{\sum_{g \in \mathcal{C}(p)} f(g)} + \frac{1}{\sum_{g \in \mathcal{C}(p)} f'(g)}$$

This can be verified easily. $\square$

Remark that in the optimization problem it is easiest to understand when the domain $D$ has a discrete, finite number of target points. But the same can be applied when the domain $D$ is continuous. In this case, we can find the arrangement defined by the sensor coverage ranges (i.e., the decomposition of the domain $D$ into pieces $D_i$ such that all points in each piece $D_i$ are covered by the same subset of sensors). The number of pieces in the arrangement is bounded, typically in polynomial of $n$. For example, if line of sight constraints is the only constraint in sensor coverage, with $n$ sensors inside a polygon of $h$ vertices, the number of pieces in the arrangement is bounded by $O(n^2h^2)$. Then the summation over all points of $p$ boils down to a weighted sum of the finite number of pieces in the arrangement, which can be solved in the same manner.

Now we solve this optimization problem for the best frequency $f'(g)$ for all $g \in \mathcal{V}$, which are real numbers. Since $f'(g)$ minimizes the function $B$, we have

$$B(f'(g), \forall g \in \mathcal{V}) \leq B(f^*(g), \forall g \in \mathcal{V}) \leq A^*$$

We can approximate these real number $f'(g)$ by rational numbers $\hat{f}(g)$ while losing only a tiny factor $(1 + \epsilon)$ for an $\epsilon > 0$ that can be made arbitrarily small:

$$B(\hat{f}(g), \forall g \in \mathcal{V}) \leq B(f'(g), \forall g \in \mathcal{V})(1 + \epsilon)$$

For rational numbers $\hat{f}(g)$ we can find integer values $\hat{T}(g)$ for all $g \in \mathcal{V}$ and integer value $\hat{T}$ such that $\hat{f}(g) = \hat{T}(g)/(k\hat{T})$. The scale of $\hat{T}$ depends on the resolution of $\hat{f}(g)$ which depends on the error $\epsilon$. Thus for a fixed error $\epsilon$, $\hat{T}$ is bounded.

Now we have a set of nodes $\hat{\mathcal{V}}$ and for each node $g \in \mathcal{V}$ we repeat it $\hat{T}(g)$ times so that we have a new set of nodes $\hat{\mathcal{V}}$. By the discussion in the previous section, we simply choose a random permutation of $\hat{\mathcal{V}}$ and run round robin on it. This generates a schedule whose average dark length value is at most $\alpha$ times the value $B(\hat{f}(g), \forall g \in \mathcal{V})$, according to Theorem 5.2. Therefore the random permutation using repetition $\hat{T}(g)$ for $g \in \mathcal{V}$ will generate average dark length value at most

$$\alpha \cdot B(\hat{f}(g), \forall g \in \mathcal{V}) \leq \alpha(1 + \epsilon) \cdot A^*$$

This finishes the proof for the following theorem.

**Theorem 5.4.** In polynomial time we can generate a schedule which is round robin on a random permutation in which node $g$ is repeated $\hat{T}(g)$ times, $\hat{T}(g)$ is produced by the convex optimization procedure. This schedule achieves an $(2 + \epsilon)\alpha$ approximation to the optimal schedule. Here $\alpha = O(1)$ if $k \geq \log n/\log \log n$, and $\alpha = O(\log n/\log \log n)$ otherwise.

At the end, we would like to summarize the algorithm for the sake of clarity:

1. Solve the convex optimization problem for $f(g)$ for each node $g \in \mathcal{V}$:

$$\min \sum_{p \in D} \sum_{g \in \mathcal{C}(p)} \frac{w(p)}{f(g)}$$

subject to

$$\sum_{g \in \mathcal{V}} f(g) = 1, f(g) \geq 0$$

2. Suppose the optimal values for the above problem is $f'(g)$ for $g \in \mathcal{V}$, we take rational numbers $\hat{f}(g)$ to approximate $f'(g)$ such that the objective function is only losing at most a factor $(1 + \epsilon)$ for $\epsilon > 0$.

3. With $\hat{f}(g)$ being rational numbers, we find integers $\hat{T}(g)$ and $\hat{T}$ such that $\hat{f}(g) = \hat{T}(g)/(k\hat{T})$.

4. Given the nodes $\mathcal{V}$, we generate $\hat{\mathcal{V}}$ which is a random permutation in which node $g \in \mathcal{V}$ is repeated $\hat{T}(g)$ times. We run round robin on $\hat{\mathcal{V}}$.

We remark that although the analysis of the theoretical guarantee of the above algorithm is non-trivial, the algorithm itself is essential taking a random permutation which is sufficiently simple and practically appealing.

6. SIMULATION

We ran simulations to evaluate the performance of the scheduling algorithm. We consider the scenario of an indoor localization application, in which the topology of a room is represented by a polygon and the possible target points (the domain set $\mathcal{D}$) monitored are selected by a grid sampling of the interior of the polygon. Then we run a Greedy Set Cover to find the set of vertex guards (the node set $\mathcal{V}$) that can fully cover these target points since the general Art Gallery Problem is NP-hard. To maintain a full coverage of the room, all guards in the greedy set need to be turned on at all times. This is the baseline energy consumption.

Assume $k = 1$ and each time slot is 1 second in the localization system. Now we can only turn on one node per second instead of $|\mathcal{V}|$ nodes. So the targets are not covered in every time slot. We want to know the dark length of the target points to understand the detection delay in such an application.
Figure 4 illustrates an example with $|D| = 113$ and $|V| = 3$. There are three nodes where $g_1$ covers 109 targets, $g_2$ covers 108 targets and $g_3$ covers 101 targets. If we want to minimize the average dark length, we run the approximation algorithm in Section 5.2. After the optimization, the weight of $g_1$, $g_2$ and $g_3$ are 0.4173, 0.3883 and 0.1944 respectively. In a periodic schedule with the length of period $= 100$ time slots, $g_1$, $g_2$ and $g_3$ should appear 42, 39 and 19 times respectively. Random permutation according to the weights gives out an average dark length of 1.74 time slots for each target. The weight of $g_3$ is lowest because it covers the least number of target but is necessary as a target point (represented by red cross) is uniquely covered by $g_3$. The maximum dark length of this periodic schedule is very high as $g_3$ is not frequently scheduled so it is dominated by the dark length of this unique target point. This illustrates the difference between min average dark length and min max dark length.

We simulated with 10 different room topologies (A few figures are provided after the conclusion). If we want to minimize max dark length, we need to find the minimum cover. The Greedy Set Cover solution is a $O(\log n)$-approximation to the optimal cover. For min average dark length, we run the algorithm to get the periodic schedule. The result of the approximation scheduling algorithm is reported in Table 1. Note that this algorithm tries to minimize the average dark length but not the maximum dark length so there is a large difference between average and max dark length of the schedule.

For such a localization application and topology, if we can only turn on one sensor in each time slot, the mean detection delay is 3.588 time slots which could be less than 2 seconds in our experience of infrared localization system.

### 7. TESTBED EVALUATIONS

We had implemented the approximation scheduling algorithm on a camera network. The objective of this network is to detect the random occurrence of an event in a room. We had deployed networked camera nodes to monitor the testing environment. One camera node is shown in Figure 5. A full description of a similar network can be found in [39]. Several valuable points on the ground are marked (the weight $w(p)$ is non-zero). The coverage of those points with respect to each camera is recorded ($P(g)$ is known). The resource constraint on the system is designed as follows: only one camera node can be turned on in each 5 seconds time slot ($k = 1$) and the camera takes a picture in the beginning of the slot. With the information about the weight, coverage and resource constraint, we can run the scheduling algorithm to determine an efficient ordering for the nodes to be turned on in each time slot.

An event is represented by a blinking LED light bulb controlled by a Raspberry-Pi in Figure 6. This is analogous to a fire event or smoking behavior. Each blinking event lasts for 6 seconds. Therefore, if a camera that can cover the event point is turned on during the period, the event is detected and reported.

A case study of 4 target points and 3 camera nodes in a room is illustrated in Figure 7. A target point is represented by a red cross. Therefore, $P(g_1) = \{p_1, p_3\}$, $P(g_2) = \{p_1, p_2\}$, $P(g_3) = \{p_2, p_4\}$. If the targeta are equally important $w(p_1) = w(p_2) = w(p_3) = w(p_4)$, then $f(g_2)$ is set to be 0 in our approximation algorithm. Therefore, we will alternate between camera $g_1$ and $g_3$. Instead of turning on 3 cameras all the time, we will cut the energy consumption to $\frac{1}{3}$ while increasing the detection delay by one time slot (5 seconds). Our experiment confirms the delay but we cannot achieve exactly the theoretical energy saving because of system overhead. Keeping the cameras connected and ready causes energy even when they are not taking photos. The energy saving is around 60% for the experiments.

The weights of the target points also play an important role in the optimization process. We simulated with 10 different room topologies (A few figures are provided after the conclusion). If we want to minimize the max dark length, we need to find the minimum cover. The Greedy Set Cover solution is a $O(\log n)$-approximation to the optimal cover. For min average dark length, we run the algorithm to get the periodic schedule. The result of the approximation scheduling algorithm is reported in Table 1. Note that this algorithm tries to minimize the average dark length but not the maximum dark length so there is a large difference between average and max dark length of the schedule.

For such a localization application and topology, if we can only turn on one sensor in each time slot, the mean detection delay is 3.588 time slots which could be less than 2 seconds in our experience of infrared localization system.

![Figure 4](image-url) A topology with 113 targets in the domain and 3 nodes to cover the domain. The target point represented by a red cross is uniquely covered by node $g_3$.

![Figure 5](image-url) A camera node. Each node is able to communicate over WiFi, take pictures and store pictures in the USB drive.

| Topology | $|D|$ | $|V|$ | Avg Dark Len. | Max Dark Len. |
|----------|------|------|--------------|--------------|
| 1        | 114  | 3    | 3.34         | 9            |
| 2        | 113  | 3    | 1.74         | 26           |
| 3        | 135  | 4    | 4.15         | 26           |
| 4        | 181  | 4    | 6.13         | 20           |
| 5        | 175  | 3    | 3.45         | 13           |
| 6        | 165  | 2    | 1.96         | 6            |
| 7        | 146  | 4    | 3.97         | 22           |
| 8        | 157  | 3    | 4.11         | 13           |
| 9        | 118  | 2    | 1.90         | 9            |
| 10       | 243  | 3    | 5.13         | 13           |

Table 1. Approximation algorithm on 10 topologies.
Figure 6. The LED bulb is connected and controlled by the program on Raspberry-Pi. The timing of the random blinking event is recorded.

Figure 7. Physical layout of experiments on camera network in a room. 3 cameras (black dot) is deployed to cover 4 target points (red cross).

role. Suppose that the event occurs more likely at \( p_1 \) and \( p_2 \). Then the weight of \( p_1 \) and \( p_2 \) is increased. For example, we can change the control program on Raspberry-Pi so the probability of blinking event is higher at \( p_1 \) and \( p_2 \). Then the frequency of \( g_2 \) will be non-zero and higher in the schedule.

8. CONCLUSION

In this paper we proposed the first set of solutions for joint sensor duty cycle scheduling for indoor applications when the sensors collaboratively cover the domain with quality guarantee while significantly reducing energy consumption. We remark that the efficiency of the algorithm makes it appealing to be applied in practical applications. There are a couple of open questions that we wish to answer in future work. First we would like to know if it is NP-hard to find the optimal schedule in the second optimization metric (min max dark length). Second we wish to establish hard coverage frequency requirement for each target and develop efficient scheduling algorithms.
Acknowledgment

We would like to acknowledge support through AFOSR FA9550-14-1-0193, NSF DMS-1418255, NSF CCF-1535900, NSF CNS-1536086, NSF CNS-1463722, NSF IIS-1460370.

9. REFERENCES


on, 2005.


