

# Maximum Lifetime Effective-sensing Partial Target-coverage in Camera Sensor Networks

Yi Hong<sup>\*</sup>, Donghyun Kim<sup>†</sup>, Deying Li<sup>\*‡</sup>, Biaofei Xu<sup>\*</sup>, Wenping Chen<sup>\*</sup>, and Alade O. Tokuta<sup>†</sup>

<sup>\*</sup> School of Information, Renmin University of China, Beijing 100872, P.R. China.

E-mail: hongyi1003@hotmail.com, deyingli@ruc.edu.cn, xubiaofei@163.com, chenwenping@ruc.edu.cn.

<sup>†</sup> Department of Mathematics and Computer Science, North Carolina Central University, Durham, NC 27707, USA.

E-mail: donghyun.kim@ncsu.edu, atokuta@ncsu.edu.

<sup>‡</sup> Corresponding author.

**Abstract**—Due to the emerging multimedia applications, camera sensor network is getting more spotlight these days. The visual sensors adopted to implement camera sensor network are a kind of directional sensors, which are well-investigated in the context of directional sensor network in the literature. However, the design and implementation of camera sensor network pose additional challenges due to unique properties of visual sensors. In this paper, we consider an interesting sleep-wakeup scheduling problem in camera sensor network, namely the *maximum lifetime effective-sensing partial target-coverage (ML-EPT)* problem. Given a set of targets and a set of camera sensors with non-uniform remaining energy-level, the goal of ML-EPT is to compute a schedule of the camera sensors such that the continuous time duration, during which a partial coverage-level requirement over the targets is met, is maximized. We propose a new heuristic algorithm, namely the *MEASURE-and-SLICE (MaS)* algorithm, which are based on two sub procedures, MEASURE and SLICE. Our simulation result shows that MaS dominates its existing alternative on average, and is close to optimal in some restricted cases.

**Index Terms**—Camera sensor network, effective coverage, coverage problem, sensor network scheduling.

## I. INTRODUCTION

Recently, wireless sensor networks are being deployed for a wide variety of surveillance and monitoring applications. Many core applications of wireless sensor network such as battlefield surveillance need *full-coverage* over an area of interest by constantly monitoring the entire area in a real-time fashion [1], [2], [3], [4]. On the other hand, there are a number of applications of wireless sensor network which do not require full-coverage. In case of forest fire monitoring, the complete coverage of the forest is desirable only during dried seasons, and it is required to monitor at least 80 percent of the area during rainy seasons [5]. In the literature, the problem of covering only a certain required percentage of an area is referred as the *partial-coverage* problem [6].

Frequently, a wireless sensor network with directional sensors is referred as a *directional sensor network* [7]. In particular, a directional sensor network with embedded camera sensors is called as a *camera sensor network*. A directional sensor node is distinguished from a conventional sensor node by its sector-like sensing range (see Fig. 1). It is well-known that an object inside a video or a picture has a better chance to be recognized if it is taken from a specific angle [8]. For

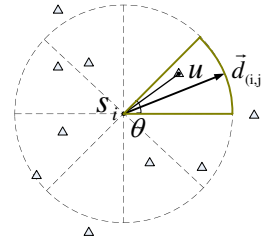


Fig. 1. This figure illustrates the basic concept of directional sensor node and our corresponding notations. The sector with a solid line is the area covered by a directional sensor  $s_i$ . Therefore,  $s_i$  is covering a target  $u$ . The beamwidth  $\theta$  is dependent on the sensor's physical characteristics.  $\vec{d}_{(i,j)}$  is a vector from  $s_i$  to the center of the surface of the  $j$ th sector.

example, a person's backhead in a picture taken by a camera is not useful to recognize his/her identity [9]. Due to the reason, the design and implementation of camera sensor networks pose additional challenges on the top of the known complications associated with directional sensors [10]. During the rest of this paper, we assume that a target is recognized or *effectively-covered* by a camera sensor if the facial view of the target is observed by a camera sensor (see Fig. 3).

One important issue of wireless sensor network is energy-efficiency, which is due to the fact that each node is battery-operated and in many application scenarios, it is very difficult or dangerous to replace or recharge the battery of the node once it is deployed. To deal with this issue, redundancy, one of the key features of wireless sensor network, is frequently exploited; in many applications, wireless sensor nodes are randomly deployed and a target of interest is likely covered by more than one node. Therefore, we can extend the time to monitor the target by having a *sleep-wakeup schedule* of the sensors covering the same target [3], [11].

This paper considers a partial-coverage problem of targets in camera sensor network. In detail, we assume each target has a weight associated with its importance and investigate a sleep-wakeup scheduling algorithm for a camera sensor network which needs to provide partial coverage over targets such that the total weight of the targets which are effectively-covered at any moment while the sensor network is active is at least a user-requested *coverage-level*. During the rest of this paper,

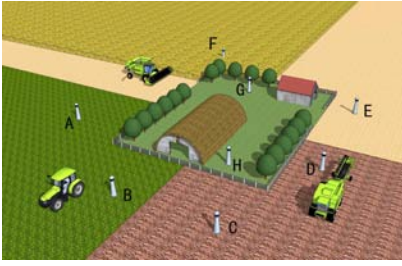


Fig. 2. In precision farming, it is generally enough to monitor sample crops and areas using camera sensors to make important decisions such as irrigating.

we will refer this problem as the *maximum lifetime effective-sensing partial target-coverage (ML-EPT)* problem. Precision agriculture is a good application of such a camera sensor network in which, it is enough to monitor sample crops and areas using camera sensors to make important decisions such as irrigating (see Fig. 2). We would like to emphasize that it is not trivial to solve ML-EPT with existing results since (a) existing partial-coverage algorithms cannot be applied to ML-EPT directly due to the effective-sensing model in camera sensor network, and (b) all of existing works in camera sensor network assume either full-coverage or barrier-coverage requirement.

The main contributions of this paper are three-fold. First, we propose a new partial target coverage problem in camera sensor network, ML-EPT, which considers both network lifetime maximization requirement and partial coverage requirement in camera sensor network at the same time. Second, we design a new heuristic algorithm for ML-EPT, namely the *MEASURE-and-SLICE (MaS)* algorithm. Third, we introduce a way to reuse an existing scheduling algorithm for directional sensor network for ML-EPT and compare its average performance with our MaS algorithm via simulation. Our simulation result shows our algorithm outperforms the existing alternative, and is close to optimal in some restricted cases.

The remainder of the paper is organized as follows. Section II reviews related work. We introduce some preliminaries and the formal definition of our ML-EPT problem in Section III. Our algorithm for ML-EPT, the MaS algorithm is presented in Section IV. The simulation result is in Section V, and we conclude this paper in Section VI.

## II. RELATED WORK

Frequently, the problem of computing a sleep-wakeup schedule of a wireless sensor network to maximize the continuous monitoring time satisfying a certain coverage requirement is referred as a *coverage problem*. In [12], Cardei and Du tackled the a full-coverage problem by (a) organizing the sensor nodes into a collection of disjoint sets, each of which completely covers the area of interest and (b) activating the nodes in each set one by one. Over years, various coverage problems have been intensively investigated to overcome the limitation of energy-restricted wireless sensor networks [13], [14], [15], [16]. Apart from the efforts on the coverage prob-

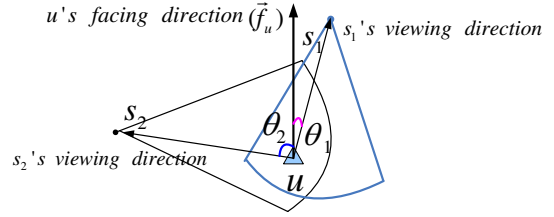


Fig. 3. In this figure, a target  $u$  is covered by two camera sensors  $s_1$  and  $s_2$ . The inner angle between the viewing orientation of  $s_1$  and  $u$ 's facing orientation,  $\theta_1$ , is smaller than that between the viewing orientation of  $s_2$  and  $u$ 's facing orientation,  $\theta_2$ . As a result,  $s_1$  can capture a more recognizable image of  $u$  than  $s_2$ .

lems in wireless sensor network with omni-directional sensors, significant amount of efforts have been made to deal with the coverage problems in directional sensor networks [17], [18], [19], [20], [21]. However, camera sensor network is a special kind of directional sensor network with several unique properties. Consequently, an algorithm working well for a directional sensor network may not fit to or can be inefficient for a camera sensor network.

In [10], Liu *et al.* defined the directional  $k$ -coverage problem in camera sensor networks, whose goal is to find the minimum number of camera sensors such that the area of interest is effectively- $k$ -covered, i.e. an object is effectively- $k$ -covered if this face direction is captured by  $k$  different cameras. In [22], the concept of the full-view coverage is introduced, in which a target is full-view covered by a camera sensor network only if the target's face is guaranteed to be captured independent from its face direction. In our previous work [23], we extend the result in [22] to deal with a full-view barrier coverage problem in camera sensor network. However, none of the existing work tackling the coverage problem in camera sensor network considered the partial-coverage requirement. In the literature, a sensor network is said to provide the *full-coverage* of an area if it completely covers the area constantly. In contrast, it provides the *partial-coverage* if the sensor network covers at least a certain percentage of the area while it operates. Over years, the partial-coverage problem has been investigated under various alias such as " $\alpha$ -lifetime" [2], [24], " $p$ -percentage coverage" [25], [26], [27], " $\theta$ -coverage" [28], or " $q$ -portion coverage" [29]. The common goal of those problems is to find the maximum lifetime sleep-wakeup schedule of a wireless sensor network such that at least " $p$  percent," " $\theta$  percent," " $\alpha$  portion," or " $q$  portion" of an area of interest is covered. However, the existing results on partial coverage problem is not directly applicable to ML-EPT since they are considering sensor nodes with omni-directional sensing area, and all of them focused on partial area coverage instead of partial coverage.

## III. PRELIMINARIES AND PROBLEM DEFINITIONS

In this section, we first review the effective-sensing model in camera sensor networks [10] and Identifiability Test in [30], which is a procedure to test if a target is effectively-sensed

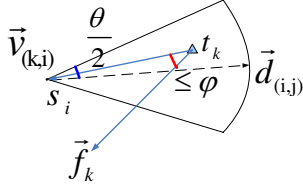


Fig. 4. In this figure, a target  $t_k$  is effectively-covered by a camera  $s_i$ .

by a camera sensor. Then, we finally introduce the formal definition of the *maximum lifetime effective-sensing partial target-coverage (ML-EPT)* problem.

#### A. Effective-sensing in Camera and Video Surveillance

As we mentioned earlier, this paper assumes that a target is recognized or *effectively-sensed (covered)* by a camera sensor if the facial view of the target is observed by a camera sensor. More formal definition of the effective-sensing model in camera sensor networks is as follow.

**Definition 1 (Effective-sensing Model).** Consider a target  $t_k$  located at  $(x_k, y_k)$  and a sensor  $s_i$  located at  $(x_i, y_i)$ .  $t_k$  is effectively-sensed (covered) by  $s_i$  if  $t_k$  is within the sensing area of  $s_i$  and the internal angle between two vectors  $\vec{f}_k$  and  $\vec{v}_{(k,i)}$  is no greater than  $\phi$ , i.e.  $\alpha(\vec{f}_k, \vec{v}_{(k,i)}) \leq \phi$  (see Fig. 4), where vector  $\vec{f}_k$  is the facing orientation of  $t_k$ ,  $\vec{v}_{(k,i)} = (x_k - x_i, y_k - y_i)$ , and  $\phi \in [0, \frac{\pi}{2}]$  is a predefined parameter called the maximum viewing angle.

#### B. Identifiability Test

This paper considers a camera sensor network of  $n$  camera sensors to monitor  $m$  targets. We assume each camera sensor has the same beamwidth  $\theta$  and therefore  $q = 2\pi/\theta$  available sectors. In this paper, we assume  $q$  is a non-zero integer. Each target  $t_k$  has a face direction represented by a vector  $\vec{f}_k$ . Under the effective-sensing model in Definition 1, we now provide the conditions that have to be satisfied if a target  $t_k$  is effectively-covered by a camera sensor  $s_i$  with its  $j$ th sector as below.

**Definition 2 (Identifiability Test).** A target  $t_k$  is effectively-covered by the  $j$ th sensing sector of a camera sensor  $s_i$  only if it passes all of the following three sub-tests.

- (a) **Sub-test 1:** check whether  $t_k$  is in the sensing range of  $s_i$ , i.e. check if  $\|\vec{v}_{(k,i)}\| \leq R$  is true, where  $R$  is the maximum sensing range of  $s_i$ .
- (b) **Sub-test 2:** check whether  $t_k$  is within the  $j$ th sensing sector of  $s_i$ , i.e. check if  $\vec{d}_{(i,j)}^T \cdot \vec{v}_{(i,k)} \geq \|\vec{v}_{(k,i)}\| \cdot \cos \frac{\theta}{2}$  is true.
- (c) **Sub-test 3:** check whether  $t_k$  is effectively-covered by the  $j$ th sensing sector of  $s_i$ , i.e. check if  $\vec{f}_k^T \cdot \vec{v}_{(k,i)} \geq \|\vec{v}_{(k,i)}\| \cdot \cos \phi$  is true.

From now on, we will denote the set of targets effectively-covered by the  $j$ th sensing sector of  $s_i$  by  $S_{(i,j)}$ . Note that all  $S_{(i,j)}$ s can be determined within a polynomial time.

#### C. Formal Definition of ML-EPT

Now, we introduce the formal definition of the *maximum lifetime effective-sensing partial target-coverage (ML-EPT)* problem. Note that we assume that the remaining battery level of nodes is not uniform and each node is connected to the data collector, also known as the sink, directly.

**Definition 3 (ML-EPT).** Given

- (a) a set  $T$  of targets  $\{a_1, a_2, \dots, a_m\}$  to be covered and their corresponding weights  $W = \{w_1, w_2, \dots, w_m\}$ ,
- (b) a set  $S$  of homogenous camera sensors  $\{s_1, s_2, \dots, s_n\}$  randomly deployed in a 2-D plane, each of which has  $q$  available orientations, the same beamwidth  $\theta$ , and its own battery lifetime  $l_i$  ( $1 \leq i \leq n$ ) such that  $L = \{l_1, l_2, \dots, l_n\}$ ,
- (c) a collection  $\mathcal{F} = \{S_{(i,j)} | 1 \leq i \leq n, 1 \leq j \leq q\}$  subsets  $S_{(i,j)} \subseteq T$  computed by Identifiability Test, and
- (d) the required coverage-level  $CL$ , which is specific to a given mission, where

$$\max_{k \in \{1, \dots, m\}} w_k \leq CL \leq \sum_{k \in \{1, \dots, m\}} w_k,$$

ML-EPT is to schedule the active periods of each camera sensor such that the sum of weights of all targets which are effectively-covered is at least  $CL$  at any time and the network lifetime is maximized.

Largely, ML-EPT consists of the following two sub problems, determining the direction (active sector) of each node and assigning the sleep-wakeup schedule to it. After the schedule is determined, we will eventually obtain a collection of the disjoint subsets of camera sensors and use each subset one by one. During the rest of this paper, we will refer each of the subsets as a *covering-set*. Note that the possible active time of each covering-set is dominated by the node in the set with least amount of remaining energy. Based on this observation, we can restate the objective of ML-EPT as “to seek a collection of the disjoint subsets of  $S$  such that (a) the total weight of targets which are effectively-covered by each subset is at least  $CL$ , and (b) the total working time of these subsets is maximized.”

#### IV. A NEW HEURISTIC ALGORITHM FOR ML-EPT

In this section, we propose a new heuristic algorithm for ML-EPT, namely the *MEASURE-and-SLICE (MaS)* algorithm. The core strategy of our MaS is to maximize the total lifetime of disjoint subsets each of which meets the minimum coverage-level requirement  $CL$ . While this strategy is usual in many existing coverage problems in wireless sensor networks, ML-EPT poses unique challenges since (a) camera sensor network is differ in effective-sensing model, (b) ML-EPT is about partial-coverage, and (c) we assume the remaining energy-level of each node is not equal. As a result, it is important for us to build each covering-set in a way that the remaining energy level of the nodes in the covering-set is consumed as equal as possible (e.g. if there exists unbalanced energy consumption among the nodes, after using a covering-set for

---

**Algorithm 1 SLICE** ( $\mathcal{F}^*, CL, T, W, UC, L, L^R$ )

---

- 1: Sort  $\mathcal{F}^*$  in non-decreasing order based on  $\frac{l_i - l_i^R}{weight_{UC}(S_{(i,j)})}$ , where  $weight_{UC}(S_{(i,j)}) = \sum_{a_k \in S_{(i,j)} \cap UC} w_k$ .
- 2:  $C \leftarrow \emptyset, \mathbb{f}_i \leftarrow \emptyset$ , for  $1 \leq i \leq n$ .
- 3: **repeat**
- 4: Put the first sector  $S_{(i_0, j_0)}$  of  $\mathcal{F}^*$  into  $C$  ( $C \leftarrow C \cup \{S_{(i_0, j_0)}\}$ ). Update  $\mathbb{f}_{i_0} \leftarrow \{S_{(i,j)} | S_{(i,j)} \in \mathcal{F}^* \text{ and } i = i_0\}$  and  $\mathcal{F}^* \leftarrow \mathcal{F}^* \setminus \mathbb{f}_{i_0}$ .
- 5: **until** GETCOVERAGELLEVEL ( $C, T, W$ )  $\geq CL$
- 6: Return IMPROVECS ( $\mathcal{F}^*, CL, T, W, C, \{\mathbb{f}_i | \forall S_{(i,j)} \in C\}$ ).
- 7: **procedure** GETCOVERAGELLEVEL ( $C, T, W$ )
- 8:  $CL_C \leftarrow 0$  and  $flag_k \leftarrow 0$  for each  $a_k \in T$ .
- 9: **for** each  $S_{(i,j)} \in C$  **do**
- 10: For each  $a_k \in S_{(i,j)}$ , if  $flag_k$  is 0, then  $CL_C \leftarrow CL_C + w_k$  and  $flag_k \leftarrow 1$ .
- 11: **end for**
- 12: Return  $CL_C$ .
- 13: **end procedure**
- 14: **procedure** IMPROVECS ( $\mathcal{F}^*, CL, T, W, C, \{\mathbb{f}_i | \forall S_{(i,j)} \in C\}$ )
- 15:  $C' \leftarrow C$  and  $CL_{C'} \leftarrow 0$ .
- 16: **for**  $sn = 1$  to  $|C'|$  **do**
- 17:  $C' \leftarrow C' \setminus \{S'_{(i,j)}\}$ , where  $S'_{(i,j)}$  is the  $sn$ th element of  $C'$ .
- 18:  $CL_{C'} \leftarrow$  GETCOVERAGELLEVEL ( $C', T, W$ ).
- 19: If  $CL_{C'} \geq CL$ , then  $\mathcal{F}^* \leftarrow \mathcal{F}^* \cup \mathbb{f}_i$ . Otherwise, reverse Line 17.
- 20: **end for**
- 21: Return  $\langle C', \mathcal{F}^* \rangle$ .
- 22: **end procedure**

---

$t$  time units, some nodes might be exhausted, while some others still have lots of energy) while the subset (covering-set) satisfies the the minimum coverage-level requirement. Note that this also involves the problem of determining the active sector of each camera sensor in the covering-set, which makes the problem even more complicated. In the following, we give the detail of MaS, an iterative algorithm which consists of the following two sub procedures, (a) SLICE: computing a slice (subset) of  $S$  forming a minimum covering-set and (b) MEASURE: measuring the maximum coverage capacity (level) of each sensor (the maximum total weights of the targets effectively covered by each sensor), to tackle the daunting complexity of the ML-EPT.

#### A. SLICE: Greedy Computation of a Smaller Size Covering-set

In this subsection, we propose a greedy algorithm to select a smaller size covering-set from a given collection of  $\mathcal{F}^* \subseteq \mathcal{F}$ . The inputs of this algorithm are (a)  $\mathcal{F}^*$ , the sub collection of  $S_{(i,j)}$ s such that  $s_i$  is not included in any covering-set so far, (b)  $CL$ , a given mission critical coverage level, (c)  $T$ , the set of targets, (d)  $W$ , the weight of the targets, (e)  $UC$ , the set of uncovered targets so far, (f)  $L$ , the initial lifetime

of sensors, and (g)  $L^R$ , the remaining lifetime of sensors. SLICE is basically a greedy algorithm. Algorithm 1 is the formal description of the SLICE algorithm, which is to find a subcollection  $C \subseteq \mathcal{F}^*$  with the smaller size such that at most one sector can be selected for each available sensor and  $C$  satisfies the minimum coverage-level requirement  $CL$ .

Let us define the weight of each sector  $S_{(i,j)}$  in  $\mathcal{F}^*$  as  $weight(S_{(i,j)}) = \sum_{a_k \in S_{(i,j)}} w_k$ , which indicates the sector's coverage capability. SLICE first sorts the elements in  $\mathcal{F}^*$  in non-decreasing order of  $\frac{l_i - l_i^R}{weight_{UC}(S_{(i,j)})}$  for each  $S_{(i,j)} \in \mathcal{F}^*$ , where (a)  $l_i - l_i^R$  denotes the difference of the initial energy and the residual energy of sensor  $s_i$ , i.e. the consumed energy of  $s_i$ , (b)  $UC$  is the set of uncovered targets so far, and (c)  $weight_{UC}(S_{(i,j)}) = \sum_{a_k \in S_{(i,j)} \cap UC} w_k$ , which stands for the sector's ( $S_{(i,j)}$ 's) current coverage capacity, i.e. the sum of the weights of the uncovered targets but can be covered by  $S_{(i,j)}$  ( $\forall a_k \in S_{(i,j)} \cap UC$ ) at the current state. This step is included to give higher priority to those nodes with higher coverage capability per remaining energy level. Next, we greedily select one working sector of each available sensors with highest priority, add it to  $C$ , and update  $\mathcal{F}^*$ . This is repeated until  $C$  can meet the minimum coverage-level requirement. The current coverage-level of  $C$  can be computed by the procedure GETCOVERAGELLEVEL. Note that since we used a greedy strategy,  $C$  can be redundant. Therefore, finally, we improve this sub collection by the procedure IMPROVECS which removes the redundant sectors from  $C$ . Specifically, this is done by checking if the constructed covering-set still can meet the minimum coverage requirement without some nodes (Lines 16-20).

#### B. MEASURE: Evaluation of Coverage Capacity of Sensors

The goal of our next sub procedure, MEASURE, is to measure the maximal coverage capacity of each node and use this as a guideline to split the remaining energy-level of the node over multiple coverage-sets in which the node will be included later. For this purpose, we construct a maximal collection of disjoint covering-sets, each of which is composed of working sectors and can meet the required coverage-level. When we select a covering set to be added into the collection, we do not pick those sectors of sensors whose another sectors have considered already in the constructed covering-sets (so far). In this way, we can obtain the sample collection of covering-sets. Note that the collection of the covering-sets, constructed without the consideration of the energy-level of each node, are not our final output, but only will be used as a guideline in our final algorithm which will be introduced in the following subsection.

We construct a collection of available covering-sets by iteration: in each iteration, we construct a batch of covering-sets based on  $F$  and eliminate all sectors in the batch of covering-sets from  $F$ . We repeat the process until the current  $F$  cannot afford CL-coverage of the target set. It is important that the construction of a batch of covering-sets in each loop is according to the criterion mentioned above, which is

---

**Algorithm 2 MEASURE** ( $\mathcal{F}, CL, T, W$ )

---

```
1:  $p' \leftarrow 1, \mathcal{F}' \leftarrow \emptyset, \mathcal{A} \leftarrow \emptyset$ , and  $C_{p'} \leftarrow \emptyset$ .
2: while GETCOVERAGELEVEL ( $\mathcal{F}, T, W$ )  $\geq CL$  do
3:    $\mathcal{F}' \leftarrow \mathcal{F}$  and  $newRound \leftarrow true$ .
4:   while GETCOVERAGELEVEL ( $\mathcal{F}, T, W$ )  $\geq CL$  do
5:      $\langle C_{p'}, \mathcal{F}' \rangle \leftarrow \text{PREPARE-SLICE} (\mathcal{F}', CL, T, W)$ .
6:     If  $C_{p'} = \emptyset$  and  $newRound = true$ , then go
to Line 10. Otherwise, set  $newRound \leftarrow false$ ,  $\mathcal{A} \leftarrow$ 
 $\mathcal{A} \cup \{C_{p'}\}$ , and  $p' \leftarrow p' + 1$ .
7:   end while
8:    $\mathcal{F} \leftarrow \mathcal{F} \setminus \bigcup_{p_o=1}^{p'} C_{p_o}$ .
9: end while
10: For each sensor  $s_i \in S$ ,  $f_i \leftarrow \sum_{j=1}^q f(S_{(i,j)})$ , where
 $f(S_{(i,j)}) \leftarrow |\{C_{p'} | S_{(i,j)} \in C_{p'}, \forall C_{p'} \in \mathcal{A}\}|$ .
11: Return  $\{f_i | \forall s_i \in S\}$ .
12: procedure PREPARE-SLICE ( $\mathcal{F}^*, CL, T, W$ )
13:   Sort  $\mathcal{F}^*$  in non-increasing order based on the weight
of the sector, which is defined as (take sector  $S_{(i,j)}$  as an
example)  $weight(S_{(i,j)}) = \sum_{a_k \in S_{(i,j)}} w_k$ .
14:    $C \leftarrow \emptyset, \mathbb{f}_i \leftarrow \emptyset$ , for  $1 \leq i \leq n$ .
15:   while GETCOVERAGELEVEL ( $C, T, W$ )  $< CL$  and
 $\mathcal{F}^* \neq \emptyset$  do
16:     put the first sector  $S_{(i_0, j_0)}$  of  $\mathcal{F}^*$  into  $C$  ( $C \leftarrow$ 
 $C \cup \{S_{(i_0, j_0)}\}$ ), and  $\mathbb{f}_{i_0} \leftarrow \{S_{(i,j)} | S_{(i,j)} \in \mathcal{F}^*$  and  $i =$ 
 $i_0\}$ ,  $\mathcal{F}^* \leftarrow \mathcal{F}^* \setminus \mathbb{f}_{i_0}$ .
17:   end while
18:   if GETCOVERAGELEVEL ( $C, T, W$ )  $< CL$  then
19:     Return  $\emptyset$ .
20:   else
21:     Return IMPROVECS( $\mathcal{F}^*, CL, T, W, C, \{\mathbb{f}_i | \forall S_{(i,j)} \in$ 
 $C\}$ ).
22:   end if
23: end procedure
```

---

implemented by repeating Steps 3-8 of Algorithm 2 and based on the procedure PREPARE-SLICE ( $\mathcal{F}^*, CL, T, W$ ).

### C. MEASURE-and-SLICE (MaS): A New Heuristic Algorithm for ML-EPT

Now, we introduce a new heuristic algorithm for ML-EPT, namely MaS, based on Algorithms 1 (SLICE) and Algorithm 2 (MEASURE). MaS first prepares the guideline to assign an active period of each available sector in each covering-set according to the result of coverage capacity evaluation done by MEASURE. Using SLICE, MaS iteratively constructs a collection of covering-sets and assign the battery lifetime of each sector to be consumed in the scheduled covering-set. Algorithm 3 is the formal definition of MaS. Now, we analysis the running time of MaS.

**Theorem 1.** *The time complexity of Algorithm 3 for ML-EPT is  $O((nq)^3 \log(nq))$ , where  $n$  is the number of the camera sensors and  $q$  is the number of available orientations per sensor.*

---

**Algorithm 3 MEASURE-and-SLICE** ( $S, T, CL, L, W$ )

---

```
1:  $\mathcal{F} \leftarrow \emptyset, S_{(i,j)} \leftarrow \emptyset$ , for  $1 \leq i \leq n, 1 \leq j \leq q$ .
2: For each pair  $(i, j)$ , run the Identifiability Test on  $(i, j)$ 
and obtain  $S_{(i,j)}$ . Set  $\mathcal{F} \leftarrow \{S_{(i,j)} | 1 \leq i \leq n, 1 \leq j \leq q\}$ .
3: Compute  $\{f_i | 1 \leq i \leq n\} = \text{MEASURE} (\mathcal{F}, CL, T, W)$ .
4: For  $\forall s_i \in S$ ,  $l_i^R \leftarrow l_i$ . Let  $L^R \leftarrow \{l_i^R | 1 \leq i \leq n\}$ .
5:  $\mathcal{Z} \leftarrow \emptyset, \mathcal{F}' \leftarrow \emptyset, p \leftarrow 1, C_p \leftarrow \emptyset, L_p \leftarrow 0, LT \leftarrow 0$ .
6: while  $L^R \neq \emptyset$  and GETCOVERAGELEVEL ( $\mathcal{F}, T, W$ )  $\geq$ 
 $CL$  do
7:    $\mathcal{F}' \leftarrow \mathcal{F}, T_p \leftarrow \emptyset$ , and  $newRound \leftarrow true$ .
8:   while GETCOVERAGELEVEL ( $\mathcal{F}', T, W$ )  $\geq CL$  do
9:     For each  $s_i \in S$ , if  $l_i^R < \frac{l_i}{f_i}$ , then  $l_i^p \leftarrow l_i^R$ .
Otherwise,  $l_i^p \leftarrow \frac{l_i}{f_i}$ .
10:    if  $T \setminus T_p = \emptyset$  then
11:      go to Line 25. /* quit inner loop */
12:    end if
13:     $\langle C_p, \mathcal{F}' \rangle \leftarrow \text{SLICE} (\mathcal{F}', CL, T, W, T \setminus T_p, L, L^R)$ .
14:    if  $C_p = \emptyset$  then
15:      If  $newRound = true$ , then go to Line 27. /*
quit outer loop */
16:    else
17:       $L_p \leftarrow \min_{i \in \{i' | S_{i',j'} \in C_p\}} l_i^p, LT \leftarrow LT + L_p,$ 
 $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{C_p\}, T_p \leftarrow T_p \cup \{S_{ij} | S_{ij} \in C_p\}$ , and  $\mathcal{F}' \leftarrow$ 
 $\mathcal{F}' \setminus \bigcup_{S_{ij} \in C_p} \{S_{i',j'} | S_{i',j'} \in \mathcal{F}' \text{ and } i' = i\}$ .
18:      for each sector  $S_{ij} \in C_p$  do
19:         $l_i^R \leftarrow l_i^R - L_p$ .
20:        If  $l_i^R = 0$ , then  $L_R \leftarrow L_R \setminus \{l_i^R\}$ .
21:      end for
22:       $p \leftarrow p + 1$ .
23:    end if
24:  end while
25:   $\mathcal{F} \leftarrow \mathcal{F} \setminus \bigcup_{p_o=1}^p C_{p_o}$ .
26: end while
27: Return  $\langle \mathcal{Z}, LT \rangle$ .
```

---

*Proof:* The analysis of the time complexity of Algorithm 3 partly depends on that of Algorithm 1. Thus let us consider Algorithm 1 first. The time complexity of Algorithm 1 is  $O(nq \log(nq))$ . For the *while* loop from Lines 8 to 24 in Algorithm 3, there are at most  $n^2 q \cdot \max_{s_i \in S} l_i$  iterations, where  $\max_{s_i \in S} l_i$  is the maximum battery lifetime among all the camera sensors. Thus, the *while* loop has time complexity with  $O(n^3 q^2 \log(nq) (\max_{s_i \in S} l_i))$ . Next, we consider the procedure of coverage capacity evaluation in Line 3. Since there are at most  $(nq)^2$  iterations in the nested *while* loop in Algorithm 2 and the time complexity of sub-procedure PREPARE-SLICE is  $O(nq \log(nq))$ , Algorithm 2's time complexity is  $O((nq)^3 \log(nq))$  where  $n$  is the number of the camera sensors and  $q$  is the number of available orientations per sensor. Based on the above analysis, the time complexity of Algorithm 3 is  $O((nq)^3 \log(nq))$ , where  $n$  is the number of the camera sensors and  $q$  is the number of available orientations per sensor. The proof is completed. ■

## V. SIMULATION RESULTS AND ANALYSIS

In [19], the authors proposed the *multiple directional cover sets (MDCS)* problem, whose goal is to organize a given set of directional sensors into a collection of non-disjoint cover sets to prolong the network lifetime, is formulated using a *mixed integer programming (MIP)* problem. Then, the authors introduced three heuristic algorithms for the linear programming relaxation of the MIP problem. Among the three algorithms, the *feedback algorithm (FB)* is shown to be the best on average via simulation. The feedback algorithm solves the linear programming relaxation by iteratively executing two internal processes, namely *conflicting direction elimination process* and *direction selection process*. In each iteration, one covering-set is determined from the linear programming relaxation, and the constraints which indicate this covering-set can satisfy the *coverage requirement* (full-coverage in MDCS, but partial-coverage in our problem) are added to the the linear programming relaxation for the next iteration. Then, the updated linear programming relaxation is solved again to obtain the next covering-set. Due to the reason, we can easily modify FB to solve ML-EPT by adopting our subprocedure GetCoverageLevel in Algorithm 1 to check if our partial coverage requirement is met. We would like to emphasize that *the FB algorithm modified in this way is the best existing alternative to solve ML-EPT, and thus we compare our algorithm against this modified FB*. Note that according to the time complexity analysis in [19], the running time of this variation is  $O((nq)^3 P^4)$ .

Now, we present our simulation results to evaluate the performance of the MaS algorithm with the variation of FB for ML-EPT in terms of network lifetime and the number of the total covering-sets. In this simulation, we randomly deploy  $n$  camera sensors and  $m$  targets in a  $10 \times 10$  2-D virtual space. Each camera sensor has the maximum sensing radius  $R$  and  $q$  available directions. The facing direction vector of each target is also randomly assigned. The maximum viewing angle  $\phi$  is set to be  $\frac{\pi}{4}$ . The battery lifetime of each sensor is a random real number between 1.0 and 5.0 time units and the weight of each target is a random integer between 0 and 10 units. We denote the coverage requirement as  $CL$ . For each parameter setting, we run 50 instances and compute their average for evaluation.

**MaS vs. FB (modified).** We first study how network lifetime is affected by the number of camera sensors  $n$ , the maximum sensing radius of each camera sensor  $R$ , the number of available directions per camera sensor  $q$ , and the number of targets  $m$  when the coverage requirement  $CL$  is 20 or 30. Fig. 5 shows the relationship between the network lifetime and the number of sensors. In Fig. 5(a), we set  $R = 5$ ,  $q = 4$ ,  $m = 10$ ,  $CL = 20$  and vary  $n$  from 20 to 40 with an increment of 5. In this figure, the network lifetime of both of MaS and FB increases smoothly when the number of sensors increases from 20 to 30. This trend becomes more prominent as the number of sensors increases from 30 to 40. When the number of sensors is 30, the average network lifetime of MaS is 2.970

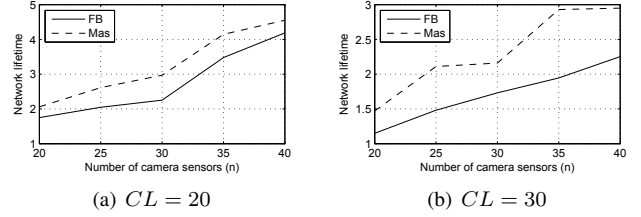


Fig. 5. Effect of  $n$  over network lifetime ( $m = 10$ ,  $R = 5$ ,  $q = 4$ ).

time units, while it is 2.251 time units for FB. In Fig 5(b), we set  $CL = 30$ , and repeat the same simulation. Now, the performance gap between MaS and FB is even greater. Based on the figures, we can conclude that MaS is getting better than FB as the size of network increases. The relationship

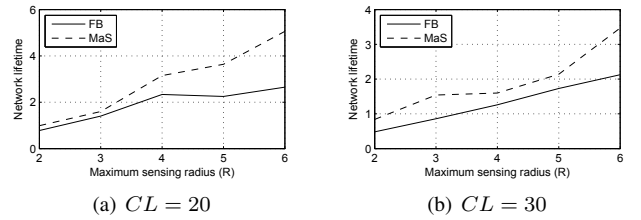


Fig. 6. Effect of  $R$  over network lifetime ( $m = 10$ ,  $n = 30$ ,  $q = 4$ ).

between network lifetime and the maximum sensing radius of each camera sensor is given in Fig. 6. Fig. 6(a) and Fig. 6(a) show as  $R$  increases, MaS is getting better than FB. From

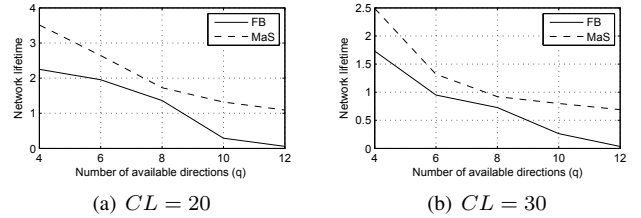


Fig. 7. Effect of  $q$  over network lifetime ( $m = 10$ ,  $R = 5$ ,  $n = 30$ ).

Fig. 7(a) and Fig. 7(b), we can observe that as the number of available directions per camera sensor increases, network lifetime decreases. This result is natural since as we have larger  $q$ , each sector has less chance to effectively cover more targets. As a result, to meet a given  $CL$  requirement, we need more nodes in a covering-set, which will results in faster energy-depletion of the sensor network. Overall, this result shows MaS is better than FB. Fig. 8 shows that network lifetime increases as the number of targets increases. This is easy to understand since as we have more nodes, we may need to operate less number of nodes to meet a given  $CL$  requirement. Both of Fig. 8(a) and Fig. 8(b) also well support our claim that MaS outperforms FB. Fig. 9 shows the relationship between network lifetime and  $CL$ . In general, this result shows as  $CL$  increases, we need to operate more number of camera sensors to form a valid covering-set. As a result, network lifetime is shortened. This result also shows that MaS outperforms FB.

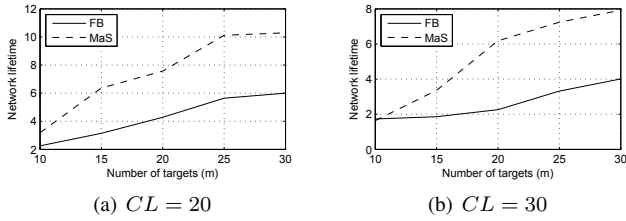


Fig. 8. Effect of  $m$  over network lifetime ( $n = 30, R = 5, q = 4$ ).

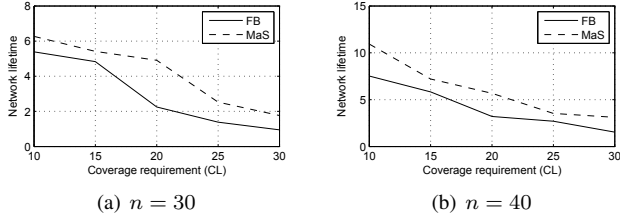


Fig. 9. Effect of  $CL$  over network lifetime ( $m = 10, R = 5, q = 4$ ).

**MaS vs. Optimum.** Now, we compare the performance of MaS algorithm for ML-EPT against an optimal solution. Note that ML-EPT is NP-hard and thus it is extremely difficult and time-consuming to obtain optimal solutions in general cases. Therefore, we only focus on restricted cases in which exhaustive search for optimal solutions is possible. In detail, we prepare a 7 by 7 grid space with the side length of one unit and randomly deploy 6 sensors  $s_1, s_2, s_3, s_4, s_5, s_6$ , and 5 targets  $t_1, t_2, t_3, t_4, t_5$  on some grid points in the space as shown in Fig. 10. Each sensor has the maximum sensing radius of 1 and four available directions, whose vectors are  $(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}), (-\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}), (-\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}),$  and  $(\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2})$ . And each target's facing direction vectors will be chose in the range of  $\{(1, 0), (0, 1), (-1, 0), (0, -1)\}$ . The maximum viewing angle  $\phi$  is set to be  $\frac{\pi}{4}$ . The battery lifetime of each sensor  $b_i$  is  $i$  time units (for  $i \in \{1, 2, 3, 4, 5, 6\}$ ) and the weight of each target  $w_j$  is  $j$  units (for  $j \in \{1, 2, 3, 4, 5\}$ ). We set  $CL$  to either 6 or 8. Then, we exhaustively search the optimal solution, and compare them with MaS. Note that the working time of each sensor can only be an integer. After 50 random combinations of targets' facing directions (e.g. it is a combination that the facing directions of  $t_1, t_2, t_3, t_4, t_5$  are  $(-1, 0), (0, 1), (1, 0), (1, 0), (0, -1)$  respectively), we compute the average and Fig. 11 shows the results of this simulation. In this figure, the first group shows the result when  $CL = 6$  and the second group indicates the case of  $CL = 8$ . We can find the average network lifetime of optimal solution is 2.5 and the average network lifetime by MaS is 2.083 when  $CL = 6$ , and the two values are 3 and 2.833 respectively when  $CL = 8$ . Our simulation result in Fig. 11 indicates that the average (experimental) performance ratio of MaS is nearly 1.

## VI. CONCLUSION

This paper studies a partial target-coverage problem in camera sensor network. While many efforts are made to study various coverage problems in both wireless sensor network

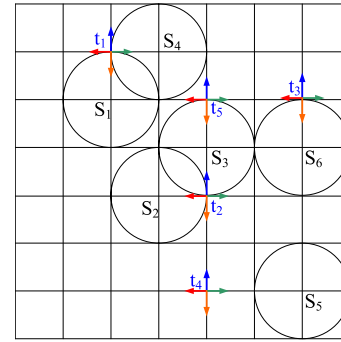


Fig. 10. A grid network instance.

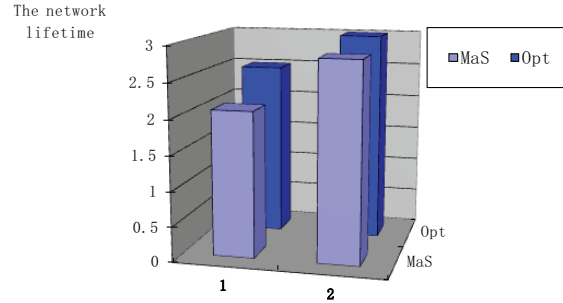


Fig. 11. The performance comparison of MaS against optimal solutions in a grid network instance.

and directional sensor network, our problem is still very challenging due to the unique properties of camera sensors. We formally define our problem as the *maximum lifetime effective-sensing partial target-coverage (ML-EPT)* problem, and propose a heuristic algorithm for ML-EPT, namely the *MEASURE-and-SLICE (MaS)* algorithm. Our simulation results show that MaS outperforms an existing alternative solution for ML-EPT as well as performs very closely to optimal solutions at least in some restricted circumstances.

## ACKNOWLEDGMENT

This research was jointly supported in part by National Natural Science Foundation of China under grants 61070191 and 91124001, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China under grant 12XNH179. And the National Research Foundation for the Doctoral Program of Higher Education of China under grant 20100004110001. This work was also supported in part by US National Science Foundation (NSF) CREST No. HRD-0833184 and by US Army Research Office (ARO) No. W911NF-0810510.

## REFERENCES

- [1] C. Huang and Y. Tseng, "The Coverage Problem in a Wireless Sensor Network," in *Proc. of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'03)*, 2003.
- [2] H. Zhang and J. Hou, "On Deriving the Upper Bound of  $\alpha$ -Lifetime for Large Sensor Networks," *Proc. of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'04)*, May 2004.

- [3] M. Cardei, M. Thai, Y. Li and W. Wu, "Energy-Efficient Target Coverage in Wireless Sensor Networks," *IEEE Annual International Conference on Computer Communications (INFOCOM)*, Mar. 2005, Miami, USA.
- [4] M.T. Thai, Y. Li, F. Wang, and D.-Z. Du, "O(log  $n$ )-Localized Algorithms on the Coverage Problem in Heterogeneous Sensor Networks," in *Proc. of the 26th IEEE International Performance Computing and Communications Conference (IPCCC)*, 2007.
- [5] L. Yu, N. Wang, and X. Meng, "Real-Time Forest Fire Detection with Wireless Sensor Networks," *Wireless Communications, Networking and Mobile Computing (WiCOM)*, vol. 2, nos. 23-26, pp. 1214-1217, Sept. 2005
- [6] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao, "Transforming Complete Coverage Algorithms to Partial Coverage Algorithms for Wireless Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 22, issue 4, pp. 695-703, 2011.
- [7] H. Ma and Y. Liu, "On Coverage Problems of Directional Sensor Networks," *Proc. of the First International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 721-731, 2005.
- [8] V. Blanz, P. Grother, P.J. Phillips, and T. Vetter, "Face Recognition based on Frontal Views Generated from Non-frontal Images," *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [9] C. Sanderson, T. Shang, and B.C. Lovell, "Towards Pose-invariant 2d Face Classification for Surveillance," *Proc. of International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, 2007.
- [10] L. Liu, H.-D. Ma, and X. Zhang, "On Directional  $k$ -coverage Analysis of Randomly Deployed Camera Sensor Networks," *Proc. of IEEE International Conference on Communications (ICC)*, 2008.
- [11] M. Cardei and J. Wu, "Energy-efficient Coverage Problems in Wireless Ad-hoc Sensor Networks," *Computer Communications*, vol. 29, issue 4, pp. 413-420, 2006.
- [12] M. Cardei and D.-Z. Du. Improving wireless sensor network lifetime through power aware organization. *Wireless Networks*, 11(3): 333-340, 2005.
- [13] R. Cohen and B. Kapchits. An optimal algorithm for minimizing energy consumption while limiting maximum delay in a mesh sensor network. In: *INFOCOM 2007*.
- [14] H. Liu, X.-W. Chu, Y.-W. Leung, X.-H. Jia, and P.-J. Wan. Maximizing lifetime of sensor-target surveillance in wireless sensor networks. In: *GLOBECOM 2009*.
- [15] C. Liu and G.-H. Cao. Spatial-temporal coverage optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 10(5): 465-478, 2011.
- [16] Y. Wang and G.-H. Cao. Minimizing service delay in directional sensor networks. In: *INFOCOM 2011*.
- [17] J. Ai and A. A. Abouzeid, "Coverage by Directional Sensors in Randomly Deployed Wireless Sensor Networks," *Journal of Combinatorial Optimization (JOCO)*, vol. 11, issue 1, pp. 21-41, 2006.
- [18] J. Adriaens, S. Megerian, and M. Potkonjak. Optimal worst-case coverage of directional field-of-view sensor networks. In: *SECON 2006*.
- [19] Y. Cai, W. Lou, M. Li, and X.-Y. Li. Target-oriented scheduling in directional sensor networks. In: *INFOCOM 2007*.
- [20] W.-F. Cheng, S.-S. Li, X.-K. Liao, S.-C. Xiang, and H.-T. Chen. Maximal Coverage Scheduling in Randomly Deployed Directional Sensor Networks. In: *ICPPW 2007*.
- [21] G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In: *SECON 2009*.
- [22] Y. Wang and G.-H. Cao, "Minimizing Service Delay in Directional Sensor Networks," *Proc. of the 30th IEEE International Conference on Computer Communications (INFOCOM)*, 2011.
- [23] H. Ma, M. Yang, D. Li, Y. Hong and W. Chen, "Minimum Camera Barrier Coverage in Wireless Camera Sensor Networks," *Proc. of the 31th IEEE International Conference on Computer Communications (INFOCOM)*, 2012.
- [24] AH. Zhang and J. Hou, Maximizing  $\alpha$ -Lifetime for Wireless Sensor Networks, *Proc. of the 3rd International Workshop Measurement, Modeling, and Performance Analysis of Wireless Sensor Networks (SenMetrics)*, July 2005.
- [25] S. Gao, X. Wang, and Y. Li, " $p$ -Percent Coverage Schedule in Wireless Sensor Networks," *Proc. of the 17th International Conference on Computer Communication and Networks (ICCCN)*, Aug. 2008.
- [26] H. Bai, X. Chen, Y. Ho, and X. Guan, "Percentage Coverage Configuration in Wireless Sensor Networks," *Proc. of the 3rd International Conference on Parallel and Distributed Processing and Applications (ISPA)*, Nov. 2005.
- [27] Y. Wu, C. Ai, S. Gao, and Y. Li, " $p$ -Percent Coverage in Wireless Sensor Networks," *Proc. of the 3rd International Conference on Wireless Algorithms, Systems and Applications (WASA)*, Oct. 2008.
- [28] Y. Liu and W. Liang, "Approximate Coverage in Wireless Sensor Networks," *Proc. of The IEEE Conference on Local Computer Networks, 30th Anniversary (LCN)*, Nov. 2005.
- [29] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient Energy Management in Sensor Networks," *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, Y. Xiao and Y. Pan, eds., vol. 2, Nova Science Publishers, 2005.
- [30] Y. Hong, D. Kim, D. Li, W. Chen, A.O. Tokuta, and Z. Ding, "Target-temporal Coverage in Mission-driven Camera Sensor Networks," submitted to *the IEEE International Conference on Computer Communications and Networks (ICCCN 2013)*, November 21, 2012.