

Research Article

Desperate Coverage Problem in Mission-Driven Camera Sensor Networks

Yi Hong,¹ Junghyun Kim,² Donghyun Kim,² Deying Li,¹ and Alade O. Tokuta²

¹ Key Laboratory of Data Engineering and Knowledge Engineering, Renmin University of China, MOE and School of Information, Renmin University of China, Beijing 100872, China

² Department of Mathematics and Physics, North Carolina Central University, Durham, NC 27707, USA

Correspondence should be addressed to Deying Li; deyingli@ruc.edu.cn

Received 16 January 2014; Revised 25 March 2014; Accepted 31 March 2014; Published 24 April 2014

Academic Editor: KingShan Lui

Copyright © 2014 Yi Hong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, camera sensor network is attracting huge amount of attention due to the growing popularity of multimedia applications. This paper investigates a new scheduling problem in camera sensor network whose goal is to cover a set of targets as efficiently as possible during a given mission period. In particular, we consider a desperate situation in which we may not have enough camera sensors to cover all of the targets of interest during the mission. The goal of our problem of interest is to schedule the sensors such that (a) the number of the most important targets which are fully covered during the mission period is maximized, and (b) the overall target-temporal coverage which is defined as the gross sum of the weight of each target multiplied by the time period when the target is covered is maximized. We formally introduce the *desperate coverage problem in mission-driven camera sensor networks* (DCP-MCSN) and propose a new heuristic algorithm for this NP-hard problem. To evaluate the performance of the proposed algorithm, we compare it with a theoretical upper bound. Our simulation result shows that our algorithm performs very close to the upper bound and outperforms the existing alternative.

1. Introduction

Thanks to the growing popularity of multimedia applications of wireless sensor networks, *camera sensor networks*, which are wireless networks of camera sensors, are getting more attention very recently. A camera sensor node is a kind of directional sensor node whose sensing range resembles a sector of a disk. In Figure 1, 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 θ is dependent on the sensor's physical characteristics. $\mathbf{d}_{(i,j)}$ is a vector from s_i to the center of the surface of the j th sector. However, the design and implementation of camera sensor networks involve additional challenges on the top of the complexity of directional sensor network due to several key features of camera sensor networks. For instance, a picture of the occupant of a person is hardly useful to verify the identity of the person, which implies that a camera sensor node may not be able to cover an object within its sensing sector depending on the facing direction of the object. As shown in Figure 2,

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, θ_1 , is smaller than that between the viewing orientation of s_2 and u 's facing orientation, θ_2 . As a result, s_1 can capture a more recognizable image of u than s_2 .

Most wireless sensor nodes are battery-operated and it is very difficult or expensive to replace or recharge the battery. As a result, energy efficiency has been a major issue of wireless sensor networks. One popular approach to extend the lifetime of wireless sensor network is to exploit the redundancy of coverage in wireless sensor network. That is, in many application scenarios, sensor nodes are randomly deployed and thus it is highly likely that a target of interest can be monitored by more than one node at the same time. Then, we can make a sleep-wakeup schedule of the nodes which can cover the same target and activate them one by one while the rest of them fall asleep. In the literature, the problem of computing a sleep-wakeup schedule of sensor nodes such that the total time period satisfies a certain coverage quality requirement is referred as a "coverage problem." One popular

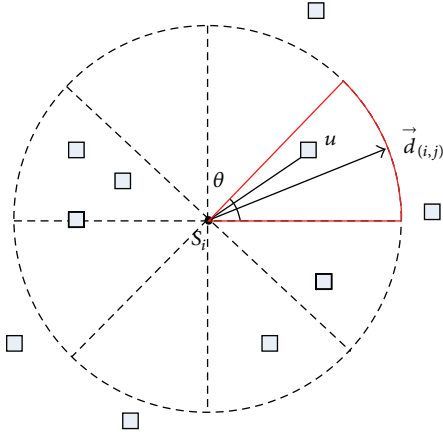


FIGURE 1: The sensing model of the directional sensor.

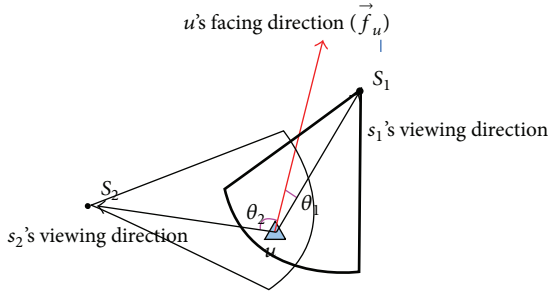


FIGURE 2: The sensing model of the camera sensor.

coverage problem is the full-coverage problem whose goal is to maximize the lifetime of a sensor network while seamlessly covering an area or targets of interest [1, 2].

Most coverage problems in wireless sensor networks have focused on the lifetime maximization issue. In those problems, the lifetime of the networks is an optimization goal, not a constraint. However, this is not always the case in reality. For instance, consider an application of wireless sensor networks in which we have to cover an area of interest or a set of targets during “a given mission period” (which is a constraint), but we do not have enough sensors to fully cover the whole area or the targets regardless of the scheduling algorithm we use. Certainly, the existing coverage algorithms that we discussed so far cannot deal with this situation. Recently, Liu and Cao introduced one way to handle this challenging issue. Given a required surveillance mission period, they suggested to maximize the spatial-temporal coverage of the area or targets of interest, which is defined as the gross sum of the total time that each distinct area is covered multiplied by the size of the area (in case of area coverage) [3] or the gross sum of the weight of each target covered multiplied by the total time duration that the target is covered (in case of target coverage) [4]. Very recently, Hong et al. extended this idea and investigated a spatial-temporal coverage problem in camera sensor networks [5].

In this paper, we introduce another way to deal with the question of how to provide the best quality coverage over

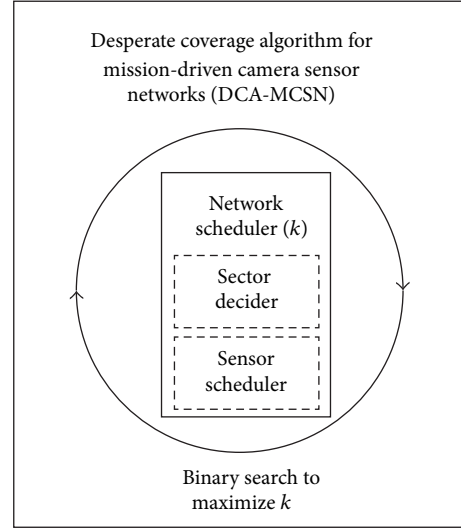


FIGURE 3: DCA-MCSN is a binary search algorithm to identify the maximum number of the most important targets which can be seamlessly covered by the camera sensors during the mission period and corresponding scheduling.

a set of targets of interest during a given mission period with insufficient number of camera sensor nodes. This study is motivated by our observation that when we solely aim to maximize the coverage quality (the total target-temporal coverage) like [4, 5], it is possible that some targets with the highest priority could be deprioritized by a number of insignificant targets, and as a result, relatively ignored in the course of the schedule building process. The problem of monitoring a warehouse with two entrances during a certain time period is a good example. Clearly, the entrances are the most important targets since the face of any trespasser can be easily detected by a camera observing the entrance. Meanwhile, a shelf inside the warehouse without any item is clearly a less-important target. To deal with this situation, we suggest providing full coverage to as many higher priority targets as possible during the mission period while the spatial-temporal coverage over the rest of targets can be maximized. We would like to emphasize that we are not trying to refute the significance of the target-temporal-maximization-only approach used in [4, 5], but we attempt to propose a new complement for those approaches to deal with this desperate situation. The main contributions of this paper can be summarized as follows.

- (a) We introduce a new scheduling problem in camera sensor networks, namely, *desperate coverage problem in mission-driven camera sensor networks (DCP-MCSN)*. More formal description of the problem is given in Section 4.
- (b) We propose a new heuristic algorithm for our new NP-hard problem, called *desperate coverage algorithm in mission-driven camera sensor networks (DCA-MCSN)*. In detail, given a set of camera sensors, a set of targets, and a mission period, DCA-MCSN performs a binary search (see Figure 3) to find a

constant k such that the most k important targets are fully covered during the mission period. During the search process, k is assumed to be a some constant (e.g., initially $k \leftarrow \lceil m/2 \rceil$) and NETWORK-SCHEDULER (Algorithm 3) is executed. NETWORK-SCHEDULER internally calls two subprocedures SECTOR-DECIDER (Algorithm 1) and SENSOR-SCHEDULER (Algorithm 2) to determine the sensing sector of each camera sensor and the activation schedule of each camera sensor. Dependent on NETWORK-SCHEDULER, the k value is increased or decreased and the whole process is repeated until we finalize k and obtain a schedule which allows the camera sensors to fully cover the most k important targets and the target-temporal coverage is maximized.

- (c) We compare the average performance of DCA-MCSN with the theoretical upper bound in simulation as well as with the only existing competitor in [5]. Our simulation result shows that our algorithm outperforms the target-temporal-maximization-only algorithm in camera sensor network in [5] in terms of the number of the most important targets which are fully covered during the whole mission period.

The rest of this paper is organized as follows. Section 2 presents related work. Some preliminaries are introduced in Section 3. We introduce the formal definition of DCP-MCSN and our algorithm for this problem in Section 4. Our simulation result is presented in Section 5. Finally, we conclude this paper in Section 6.

2. Related Work

Frequently, the problem of constructing a sleep-wakeup schedule of a wireless sensor network with the goal of maximizing the continuous monitoring time to meet a given coverage quality requirement is referred as a coverage problem [2–4, 6]. Most types of wireless sensor networks thoroughly investigated in the literature consider sensor nodes with omnidirectional sensing range. Consequently, the area covered by those sensor nodes resembles a disk with the node in the center of it. In contrast, the sensing area covered by a directional sensor node looks more like a sector of a disk [7]. In [8], Cai et al. have investigated the multiple directional cover set problem in directional sensor networks, whose objective is to compute a sleep-wakeup schedule of a given set of sensor nodes such that the time for the sensor network to fully cover a given set of targets is maximized. In the meantime, Ai and Abouzeid [9] studied a new coverage problem in directional sensor networks whose goal is given a set of sensor nodes with adjustable camera orientations, to determine the direction (orientation) of each sensor so that the number of covered targets is maximized (primary objective) while the number of sensor nodes used is minimized (secondary objective). In addition, the other aspects of directional sensor networks are also investigated like vulnerability [10], fault-tolerance [11], and so on [12, 13].

Briefly speaking camera sensor network is a kind of directional sensor network and thus they share several

properties. However, due to several unique features of camera sensors, especially the effective-sensing surveillance model in Definition 1, the scheduling algorithms for directional sensor networks are not directly applicable to camera sensor networks. Most importantly, a camera sensor node can cover a target only if the face of the target can be seen by the sensor. In [14], Liu et al. introduced the directional k -coverage problem in camera sensor networks whose objective is minimizing the number of camera sensors to cover an area of interest such that any spot in the area can be monitored by at least k different cameras. In [13], Wang and Cao assumed a camera sensor network version of the full coverage problem, called the full-view coverage problem. In this problem, a target is full-view covered by a camera sensor network if we can capture the face of a target which is within the sensing range of the camera sensor network independent of the face direction of the target. Previously, Ma et al. extended the result in [13] and investigated the full-view barrier coverage problem in camera sensor networks.

So far, most coverage problems in wireless sensor networks have concerned about the lifetime maximization issue. Therefore, they are not applicable to a desperate situation in which we are required to cover an area or a set of targets of interest during a given mission period. One possible solution of this challenging issue is introduced by Liu and Cao [3, 4], who suggested to maximize the spatial-temporal coverage of the area or targets of interest. In our previous work, we extended this idea and investigated a spatial-temporal coverage problem in camera sensor networks [5]. However, Liu and Cao's approach comes with a nonneglectable shortcoming—a target with the highest priority could be deprioritized by a number of insignificant targets, and as a result, is ignored by a scheduling algorithm. In this paper, we address this issue and attempt to propose an alternative solution of the problem as a complement of Liu and Cao's approach.

3. Preliminaries

In this section, we review two important preliminaries of our paper, the effective-sensing model in camera sensor networks [14] and the Identifiability Test [5], which is a procedure to check, given a set of targets and a set of camera sensors, if the target is effectively covered by the camera sensors. 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. The face direction of each target can be estimated by an existing head pose estimating strategy such as the one in [15]. A more formal definition of the effective-sensing model in camera sensor networks is as follows.

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 \mathbf{f}_k and $\mathbf{v}_{(k,i)}$ is not greater than ϕ ; that is, $\alpha(\mathbf{f}_k, \mathbf{v}_{(k,i)}) \leq \phi$ (see Figure 4), where vector \mathbf{f}_k is the facing orientation of t_k , $\mathbf{v}_{(k,i)} = (x_i - x_k, y_i - y_k)$, and $\phi \in [0, \pi/2)$ is a predefined parameter called the max viewing angle.

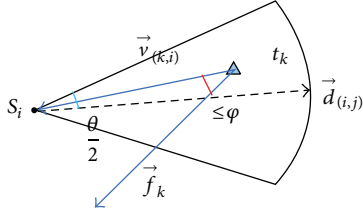


FIGURE 4: In this figure, a target t_k is effectively sensed by a camera s_i .

In this paper, we consider a camera sensor network of n camera sensors and m targets of interest. Each camera sensor is with uniform beamwidth θ and with $q \geq 1$ available sectors; that is, $q = \lceil 2\pi/\theta \rceil$. We denote the face direction of a target t_k by a vector \mathbf{f}_k . By the effective-sensing model in Definition 1, we can define the conditions to be satisfied for a target t_k to be 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 subtests.

Subtest 1. Check whether t_k is in the sensing range of s_i ; that is, check if $\|\mathbf{v}_{(k,i)}\| \leq R$ is true, where R is the maximum sensing range of s_i .

Subtest 2. Check whether t_k is within the j th sensing sector of s_i ; that is, check if $\mathbf{d}_{(i,j)}^T \cdot \mathbf{v}_{(k,i)} \geq \|\mathbf{v}_{(k,i)}\| \cdot \cos(\theta/2)$ is true, where $\mathbf{d}_{(i,j)}$ is a vector from s_i to the center of the surface of the j th sector.

Subtest 3. Check whether t_k is effectively covered by the j th sensing sector of s_i ; that is, check if $\mathbf{f}_k^T \cdot \mathbf{v}_{(k,i)} \geq \|\mathbf{v}_{(k,i)}\| \cdot \cos \phi$ is true.

From now on, $S_{(i,j)}$ will be used to represent (a) the set of targets effectively covered by the j th sensing sector of s_i and (b) the j th sensing sector of s_i , depending on the context. Note that in the first case, all $S_{(i,j)}$ s can be determined within a polynomial time.

4. Desperate Coverage Problem in Mission-Driven Camera Sensor Networks (DCP-MCSN) and Its Solution

Now, we introduce the formal definition of DCP-MCSN in Section 4.1 and our solution for this problem, namely, *desperate coverage algorithm in mission-driven camera sensor networks* (DCA-MCSN) in Section 4.2. Table 1 summarizes the terms, notations, and their semantics used in this paper.

4.1. Formal Definition of DCP-MCSN

Definition 3 (DCP-MCSN). DCP-MCSN is to find a subcollection (schedule) $\mathcal{X} \subset \mathcal{F}$ such that (a) for each sensor node s_i , at most one sector $S_{(i,j)}$ is used to form the subsets

in \mathcal{X} , (b) the number of the first k pivotal targets which are fully covered during B is maximized (primary objective), and (c) the overall target-temporal coverage is maximized (secondary objective).

Theorem 4. DCP-MCSN is NP-hard.

Proof. A simpler form of DCP-MCSN without the second requirement regarding maximizing the number of the first k pivotal targets is already proven to be NP-hard in [5]. Therefore, this problem is NP-hard. \square

4.2. Desperate Coverage Algorithm in Mission-Driven Camera Sensor Networks (DCA-MCSN). In this section, we introduce our algorithm for DCP-MCSN, namely, DCA-MCSN. Given a set of camera sensors, a set of targets, and a mission period, DCA-MCSN performs a binary search (see Figure 3). In detail, initially the number of the most important targets which can be seamlessly covered by the camera sensors is set to $k \leftarrow \lceil m/2 \rceil$ and executes NETWORK-SCHEDULER (Algorithm 3), which internally invokes two subprocedures SECTOR-DECIDER (Algorithm 1) and SENSOR-SCHEDULER (Algorithm 2). At the end, if NETWORK-SCHEDULER returns a schedule of camera sensors such that the most k important targets are fully covered during the mission period, we set $k \leftarrow \lceil (m + (m/2))/2 \rceil$ and execute NETWORK-SCHEDULER. Otherwise, we set $k \leftarrow \lceil ((m/2) - 1)/2 \rceil$ and execute NETWORK-SCHEDULER. At the end the search will be terminated, and we will find the maximum k that can be achieved by our algorithm as well as its corresponding schedule. We would like to emphasize that similar to our scheduling strategy in [5], the mission period B is divided into multiple cycles with a uniform length r , and NETWORK-SCHEDULER is invoked to construct a schedule for a short term with r unit time length. As a result, NETWORK-SCHEDULER is invoked for B/r times during the total mission period B . In the following, we introduce the description of SECTOR-DECIDER, SENSOR-SCHEDULER, and NETWORK-SCHEDULER.

4.2.1. Sector-Decoder. Algorithm 1 is the formal description of SECTOR-DECIDER. The input of this algorithm is the set of camera sensors S , the set of targets T , and the set of the most important k targets (those with the heaviest weights) in T . The goal of this greedy algorithm is to determine one active sector from all available ones for each camera sensor. In Lines 1–3, for each sector of each camera sensor, the set $S_{(i,j)}$ of targets which can be effectively covered by a sensor s_i with its j sector is computed.

In detail, in the preliminary phase (Lines 5–7), for each target $a_{k'}$ among the first k pivotal targets, the accumulated time recorder $C_{k'}$ that this target is effectively covered by some camera sensor is set to 0. Then, the loop spanning over Lines 8–24 is repeated to determine the set of sectors \mathcal{X} such that one sector (represented by those targets, e.g., $S_{(i,j)}$, which are covered by the sector) is selected from each sensor node. This loop largely consists of two parts.

- (a) The first part (Lines 9–15) is for the case that there exists at least one target among the most important

TABLE 1

| Terms, symbols | Semantics |
|---------------------------------|---|
| S | A set of homogenous camera sensor nodes $\{s_1, \dots, s_n\}$ |
| T | A set of targets $\{t_1, \dots, t_m\}$ |
| W | The set of the weights of the targets in T $\{w_1, \dots, w_m\}$ |
| θ | The uniform field-of-vision of the camera sensors |
| $R_s = 1$ | The normalized sensing range of the camera sensors |
| $R_t = 1$ | The transmission range of the camera sensors with omnidirectional transmission radios |
| L_i | The battery lifetime of s_i |
| \mathbf{f}_{t_k} | The face direction vector of t_k |
| B, r | The length of a given mission and the length of each cycle (we assume B/r is an integer for simplicity) |
| q | $2\pi/\theta$, the number of available orientations for each sensor |
| $\mathbf{d}_{(i,j)}, S_{(i,j)}$ | The vector of s_i 's j th available orientation and the set of targets covered by s_i using its j th sector |
| \mathcal{F} | A collection of the subsets of $\{S_{(i,j)} \mid 1 \leq i \leq n, 1 \leq j \leq q\}$ |

```

(1) for each  $(i, j)$  pair,  $1 \leq i \leq n, 1 \leq j \leq q$  do
(2)   Run Identifiability Test and compute  $S_{(i,j)}$ , the set of targets effectively covered by  $s_i$  with its  $j$ th sector.
(3) end for
(4) Set  $\mathcal{Z} \leftarrow \emptyset, UC_k \leftarrow \{a_{m_1}, \dots, a_{m_k}\}, UC' \leftarrow T \setminus UC_k$ , and  $UW \leftarrow S$ .
(5) for each target  $a_{k'} \in UC_k$  do
(6)    $C_{k'} \leftarrow 0$ .
(7) end for
(8) loop
(9)    $(i, j) \leftarrow \arg \max_{s_i \in UW, 1 \leq j \leq q} |S_{(i,j)} \cap UC_k|$ .
(10)  if  $S_{(i,j)} \cap UC_k \neq \emptyset$  then
(11)     $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{S_{(i,j)}\}, UW \leftarrow UW \setminus \{s_i\}, UC' \leftarrow UC' \setminus S_{(i,j)}$ .
(12)    for each  $a_{k'} \in S_{(i,j)} \cap UC_k$  do.
(13)       $C_{k'} \leftarrow C_{k'} + L_i = C_{k'} + L_i/(B/r)$ .
(14)    end for
(15)     $UC_k \leftarrow UC_k \setminus \{a_{k'} \mid C_{k'} \geq r, \text{ where } k' \in \{m_1, \dots, m_k\}\}$ .
(16)  else
(17)     $(i', j') \leftarrow \arg \max_{s_{i'} \in UW, 1 \leq j' \leq q} |S_{(i',j')} \cap UC'|$ .
(18)    if  $S_{(i',j')} \cap UC' = \emptyset$  then
(19)      break; /* quit the loop */
(20)    else
(21)       $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{S_{(i',j')}\}, UW \leftarrow UW \setminus \{s_{i'}\}, UC' \leftarrow UC' \setminus S_{(i',j')}$ .
(22)    end if
(23)  end if
(24) end loop
(25) Return  $\mathcal{Z}$ .

```

ALGORITHM 1: SECTOR-DECIDER $(S, T, \{a_{m_1}, \dots, a_{m_k}\})$.

k ones which may not be fully covered during the whole period r . And this part is used to determine an active sector for an undecided sensor node such that there are more uncovered targets among the most important k targets which will be covered. Once we find such a sector, for example, $S_{(i,j)}$, in Line 11, we add it to \mathcal{Z} . Line 13 is used to update the total time that the most important k targets can be covered by the sectors in \mathcal{Z} so far.

- (b) The second part (Lines 17–22) is for the case that all the most important k targets could be fully covered during the whole period r . This part is used to add

more sectors to \mathcal{Z} such that the total target-temporal coverage can be maximized.

4.2.2. Sensor-Scheduler. Algorithm 2 is the formal description of SENSOR-SCHEDULER. The inputs of this algorithm are as follows:

- (a) a single sector $c_i = S_{(i,j)}$ of a sensor node s_i for some j selected by SECTOR-DECIDER;
- (b) N_i which is the set of sectors, for example, $S_{(i,j)}$ s, selected by SECTOR-DECIDER such that there exist

```

(1)  $Sch_i \leftarrow \emptyset$ .
(2) if  $\exists a_{k'} \in c_i \cap T_k$  such that  $(t_{k'} - red_{k'}) < r$ , where
    (2.1)  $t_{k'} = \sum_{c_j \in A(k') \setminus \mathcal{Z}'} l_j$ ,
    (2.2)  $red_{k'} = \sum_{c_x, c_y \in A(k') \setminus \mathcal{Z}' \wedge x \neq y} \max\{\min(c_x \cdot f, c_y \cdot f) - \max(c_x \cdot b, c_y \cdot b), 0\}$ , and
    (2.3)  $A(k')$  is the set of selected sectors by SECTOR-DECIDER effectively covering  $a_{k'} \in T$ .
    then
(3)  $Sch_i \leftarrow Sch_i \cup \{\max_{c_{i'} \in N_i \cap A(k') \setminus \mathcal{Z}'} \{c_{i'} \cdot f\} \mid \forall a_{k'} \in c_i \cap T_k\}$ .
(4) else
(5)  $Sch_i \leftarrow Sch_i \cup \{c_{i'} \cdot b, c_{i'} \cdot f, c_{i'} \cdot b - l_i, c_{i'} \cdot f - l_i \mid \forall c_{i'} \in N_i \setminus \mathcal{Z}'\} \cup \{0, r\}$ .
(6) end if
(7) Find the  $c_i \cdot b_0$  satisfying  $\Delta TC_i = \min_{c_i \cdot b \in Sch_i} red_i$ , where
     $red_i = \sum_{c_j \in N_i \setminus \mathcal{Z}'} (\max\{\min(c_i \cdot f, c_j \cdot f) - \max(c_i \cdot b, c_j \cdot b), 0\} \cdot \sum_{a_k \in c_i \cap c_j} w_k)$ .
    Note that if  $c_i \cdot f$  is larger than the length of the round  $r$ , we should rewrite the original working
    period  $[c_i \cdot b, c_i \cdot f]$  as  $[c_i \cdot b, r]$  and  $[0, c_i \cdot f - r]$ .
(8) Return  $\langle c_i \cdot b_0, c_i \cdot f_0 \leftarrow c_i \cdot b_0 + L_i / (B/r) \rangle$ .

```

ALGORITHM 2: SENSOR-SCHEDULER $(c_i, N_i, \mathcal{Z}', \mathcal{Z}, T_k, Sch)$.

```

(1)  $\mathcal{Z} \leftarrow \text{SECTOR-DECIDER}(S, T, T_k)$ .  $TC \leftarrow 0$ .
(2) for each  $x \in \{i \mid S_{(i,j)} \in \mathcal{Z}\}$  do
(3)  $c_x \leftarrow \emptyset, \mathcal{Z}' \leftarrow \emptyset, T_k \leftarrow \{a_{m_1}, \dots, a_{m_k}\}$ , and  $N_i \leftarrow \emptyset$ .
(4) end for
(5) for each  $S_{(i,j)} \in \mathcal{Z}$  do
(6)  $c_i \leftarrow S_{(i,j)}, \mathcal{Z}' \leftarrow \mathcal{Z}' \cup \{c_i\}$ , and  $N_i \leftarrow \{c_x \mid c_x \in \mathcal{Z} \setminus c_i \text{ and } c_i \cap c_x \neq \emptyset\}$ .
(7) end for
(8) for each  $s_i \in S$  do
(9)  $Sch \leftarrow \emptyset$  and  $TC_i \leftarrow 0$ .
(10) end for
(11) while  $\mathcal{Z}' \neq \emptyset$  do
(12) Construct a maximal independent set  $M(\mathcal{Z}')$  of  $\mathcal{Z}'$ .
(13) for each  $c_i \in M(\mathcal{Z}')$  do
(14)  $\langle c_i \cdot b, c_i \cdot f \rangle \leftarrow \text{SENSOR-SCHEDULER}(c_i, N_i, \mathcal{Z}', \mathcal{Z}, T_k, Sch)$ .
(15) end for
(16)  $\mathcal{Z}' \leftarrow \mathcal{Z}' \setminus M(\mathcal{Z}'), Sch \leftarrow Sch \cup \{\langle c_i \cdot b, c_i \cdot f \rangle \mid \forall c_i \in M(\mathcal{Z}')\}$ .
(17) end while
(18) for each  $x \in \{m_1, \dots, m_k\}$  do
(19) if  $a_x$  is not fully covered during  $r$  time slots based on  $Sch$  then
(20) Return  $\langle \text{FALSE}, \emptyset, \emptyset, -1 \rangle$ .
(21) end if
(22) end for
(23)  $TC \leftarrow \sum_{k \in \{1, \dots, m\} \setminus \{m_1, \dots, m_k\}} w_k \cdot (t_k - red_k) + r \cdot \sum_{k \in \{m_1, \dots, m_k\}} w_k$ , where
    (23.1)  $t_k = \sum_{c_j \in A(k)} l_j$ , and
    (23.2)  $red_k = \sum_{c_i, c_j \in A(k) \wedge i \neq j} \max\{\min(c_i \cdot f, c_j \cdot f) - \max(c_i \cdot b, c_j \cdot b), 0\}$ .
(24) Return  $\langle \text{TRUE}, \mathcal{Z}, Sch, TC \rangle$ .

```

ALGORITHM 3: NETWORK-SCHEDULER $(B, S, T, W, T_k = \{a_{m_1}, \dots, a_{m_k}\})$.

some targets which can be covered by both the selected sector of s_i and another sector in N_i ;

- (c) a subcollection \mathcal{Z}' of \mathcal{Z} ;
- (d) \mathcal{Z} itself;
- (e) the set T_k of the most important k targets;
- (f) Sch , which is the schedule of cameras which are determined so far.

The goal of this algorithm is to determine the schedule of sector c_i within each short term with r unit time. The most important part is to make sure that the most important k targets are getting coverage during the whole period r . For this purpose, Line 2 checks if there exists a target in T_k which has not received sufficient coverage based on the schedule Sch (which is still under consideration) but still can be covered by c_i . If so, we add the latest finish time of sensors in Sch which covers such a target to Sch_i as the beginning time of c_i (Line 3).

If no such target exists, we just try to find a point to start to use c_i so that the target-temporal coverage is maximized. At the end, the activation period of c_i , which is represented by two moments $c_i \cdot b_0$ and $c_i \cdot f_0$, is returned. Note that we utilize a new metric, namely, the target-temporal coverage redundancy of a camera sensor node red_i in Line 7 of Algorithm 2. This metric is used to evaluate how much the node c_i is wasted by monitoring targets which are already monitored by the other nodes. By rescheduling the current node such that the overall target-temporal coverage redundancy can be decreased, we can improve the quality of the current schedule.

4.2.3. Network-Scheduler. Based on the above two subprocedures SECTOR-DECIDER and SENSOR-SCHEDULER, NETWORK-SCHEDULER's formal description is shown in Algorithm 3. The inputs of this algorithm are the mission period B , the set of camera sensors S , the set of targets T , the weight of the targets W , and the set of the most important k targets, $\{a_{m_1}, \dots, a_{m_k}\}$. The outcome of this algorithm consists of four tuples. The first one is either true or false. It is true if the schedule generated by this algorithm is providing full coverage of the most important k targets during the mission period B . Otherwise, it returns false. The second item returned is \mathcal{Z} which includes one sector for each camera sensor to achieve the target-temporal coverage TC , which is the fourth item of the output of this algorithm. The third one is Sch , which is the schedule of each camera sensor.

This algorithm is based on the assumption that all camera sensor nodes are already scheduled; that is, the activation period of each camera sensor node is already determined randomly, and we try to refine the schedule of each node (one by one) based on the current schedule of the other nodes such that the overall target-temporal coverage can be maximized. Note that the preassignment of the schedule of all nodes can be done via assigning each node's beginning time with 0. Based on the preassignment of the schedule, we can improve the quality of it by iteratively identifying a node which can make the overall target-temporal coverage increase and rescheduling this node.

In detail, in Line 1, SECTOR-DECIDER is called to determine the active sector for each sensor node. Lines 2–10 are used to initialize the variables. Lines 11–17 are used to determine the schedule of camera sensors. In particular, Line 12 utilizes maximal independent set of \mathcal{Z}' so that the camera sensors whose activation period is determined do not overlap. In this way, we can further maximize the target-temporal coverage. Finally, Lines 18–24 are checking if the determined schedule is satisfiable.

4.2.4. A Big Picture. Now, we give a simple example to illustrate how the overall algorithm works. The camera sensor network is composed of 5 camera sensors, each of which has 8 available orientations, and 5 targets, each of which has its own facing direction (as shown by the shorter arrow of each target in Figure 5). Among these 5 targets, there are 2 pivotal targets a_1 and a_2 ; that is, $T_k = \{a_1, a_2\}$ and $w_1 = 4, w_2 = 3, w_3 = 2, w_4 = 1, w_5 = 1$. In the scheduling of each sensor,

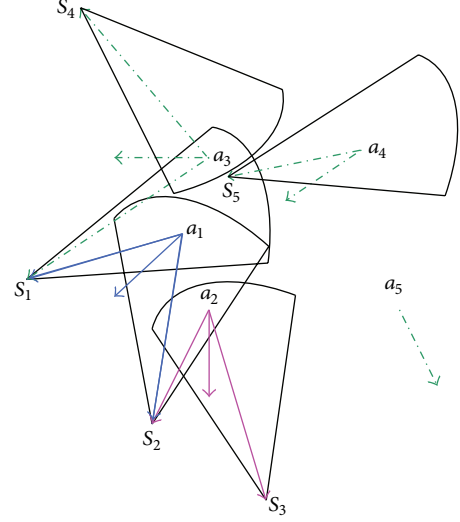


FIGURE 5: An instance of the overall algorithm.

the length of each round $r = 10$ and the sensors' battery lifetime per round l_1, l_2, l_3, l_4, l_5 are 5, 6, 4, 4, 6, respectively.

In Figure 5, each sensor has been assigned the working direction; that is, Lines 1–7 of Algorithm 3 have been accomplished for this network, and we obtain each sensor's neighbor set: $N_1 = \{S_2, S_4\}$, $N_2 = \{S_1, S_3\}$, $N_3 = \{S_2\}$, $N_4 = \{S_1\}$, $N_5 = \{\emptyset\}$. Then we begin to generate a sleep-wakeup schedule for each sensor (Lines 8–17 of Algorithm 3). We first go into the while loop (Lines 11–17 of Algorithm 3).

- (a) In the first loop, $\mathcal{Z}' = \{S_1, S_2, S_3, S_4, S_5\}$ and an MIS of \mathcal{Z}' is $\{S_1, S_3, S_5\}$, which is obtained based on each sensor's neighbor set. For each sensor in $\{S_1, S_3, S_5\}$, we run Algorithm 2 to get its starting time point.
 - (i) For S_1 , $S_1 \cap T_k = \{a_1\}$ and $t_1 - red_1 = 0 < r$, so $Sch_1 = \{S_2 \cdot f\} = \{\emptyset\}$. It is easy to find that $S_1 \cdot b = 0$ satisfying $\Delta TC_1 = 0$; thus, we can return $S_1 \cdot b = 0$.
 - (ii) The output of Algorithm 2 on S_3 is similar to that of S_1 and $S_3 \cdot b = 0$.
 - (iii) For S_5 , since $S_5 \cap T_k = \emptyset$ and $N_5 = \{\emptyset\}$, $Sch_5 = \{0, 10\}$.

We can easily get that $S_5 \cdot b = 0$ satisfying $\Delta TC_5 = 0$; thus, we can return $S_5 \cdot b = 0$.

- (b) In the second loop, \mathcal{Z}' is updated into $\{S_2, S_4\}$ and the MIS of the current \mathcal{Z}' is $\{S_2, S_4\}$. For each sensor in $\{S_2, S_4\}$, we run Algorithm 2 to get its starting time point.
 - (i) For S_2 , $S_2 \cap T_k = \{a_1, a_2\}$ and $t_1 - red_1 = 5 < r$, $t_2 - red_2 = 0 < r$, so $Sch_2 = \{S_1 \cdot f, S_3 \cdot f\} = \{5, 4\}$. It is easy to find that $S_2 \cdot b = 4$ satisfying $\Delta TC_2 = 4 (= \min\{7, 4\})$; thus, we can return $S_2 \cdot b = 4$.
 - (ii) For S_4 , since $S_4 \cap T_k = \emptyset$, $Sch_4 = \{S_1 \cdot b, S_1 \cdot f, 0, 10\} = \{0, 5, 10\}$.

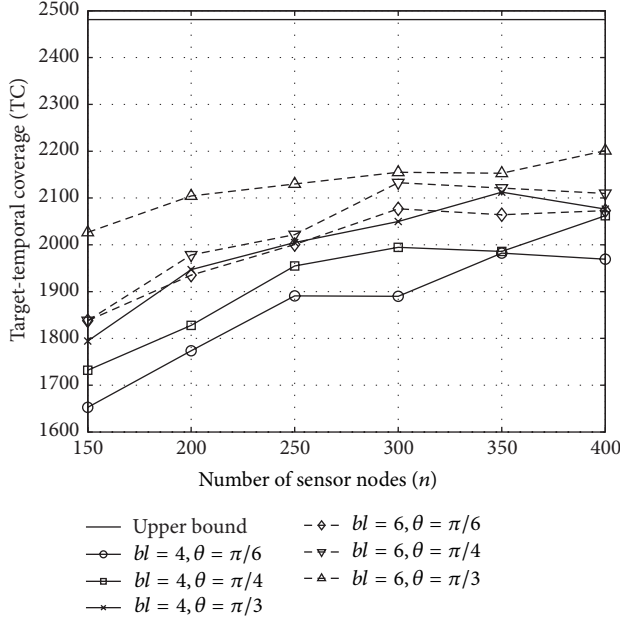


FIGURE 6: Target-temporal coverage achieved by DCA-MCSN versus the number of sensors.

We can easily get that $S_4 \cdot b = 5$ satisfying $\Delta TC_4 = 0$; thus, we can return $S_4 \cdot b = 5$.

After the second loop, \mathcal{E}' is updated into \emptyset , that is, the while loop can be terminated. Finally we obtain that the scheduling is $S_1 \cdot b = 0$, $S_2 \cdot b = 4$, $S_3 \cdot b = 0$, $S_4 \cdot b = 5$, $S_5 \cdot b = 0$ and $TC = 94 (= 4 * (5 + 6 - 1) + 3 * (4 + 6) + 2 * (5 + 4) + 1 * 6 + 1 * 0)$.

5. Simulation Results and Analysis

In this section, we present our results and discuss the performance of our algorithm for DCA-MCSN. Specifically, we randomly deploy n camera sensor nodes and m targets within a 100×100 2-D virtual space, where n is set to 150, 200, ..., 400. It is expected that the number n of available camera sensor nodes, whose battery lifetime is significantly lower than the mission lifetime L , is greater than the number of m of targets, and thus we set m as 20, 30, ..., 70. We set maximum sensing radius R of each camera sensor to be 50. We assume the weight of each target is a random number within the range of (0, 10). We assume that the mission lifetime per a unit r is 10 unit time. Lastly, we randomly assign the face direction vector of each target, and the maximum viewing angle is $\pi/4$. In this problem, each camera sensor node has q sensing sectors and we will use one of them; that is, the beamwidth θ of each sector is $2\pi/q$.

In the simulation, we study the average characteristic behavior of the proposed algorithm and evaluate its average performance with regard to the target-temporal coverage TC and the number of the most important targets fully covered by sensors given. Especially, we will check how this algorithm works with two important parameters: the uniform battery lifetime, bl , of each sensor and the beamwidth θ of each

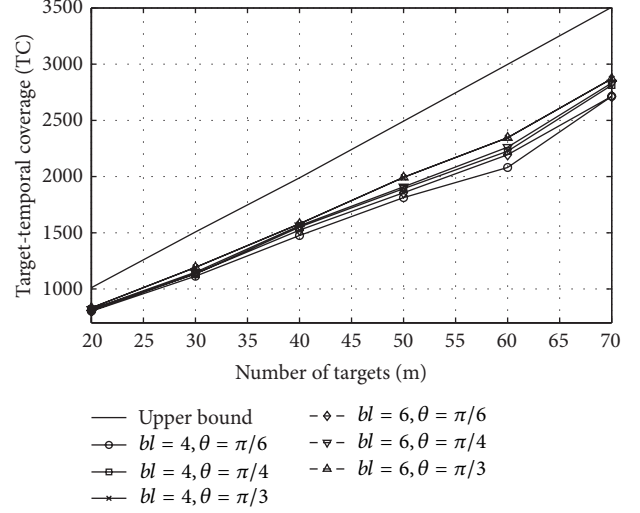


FIGURE 7: Target-temporal coverage achieved by DCA-MCSN versus the number of targets.

camera sensor. In particular, we will check the following six cases. Case (a) $bl = 4$, $\theta = \pi/6$, Case (b) $bl = 4$, $\theta = \pi/4$, Case (c) $bl = 4$, $\theta = \pi/3$, Case (d) $bl = 6$, $\theta = \pi/6$, Case (e) $bl = 6$, $\theta = \pi/4$, Case (f) $bl = 6$, $\theta = \pi/3$. We also use the sum of the weights of all targets, $\sum_{1 \leq k \leq m} w_k$, multiplied by the mission time duration per a unit r as the theoretical upper bound, which is notated by UB . Clearly UB can be larger than actual achievable optimum.

5.1. Simulation Results for DCA-MCSN. In Figure 6, we compare the performance of DCA-MCSN against UB under the 6 parameter settings (Cases (a)–(f)) while the number of targets to be covered is fixed to 50. By comparing the results with $bl = 4$ (Cases (a), (b), and (c)) against $bl = 6$ (Cases (d), (e), and (f)), we can learn that if the beamwidth θ of each sensor's sector is fixed, the performance of the algorithm becomes close to UB with larger, longer battery lifetime bl . This is because with larger bl value, we have more number of sensor nodes to fully cover more number of targets during the mission period. Next, we study how the total TC is affected by the beamwidth θ of each sensor's sector while we fix bl to 4 and 6, respectively. By comparing Cases (a), (b), and (c) in Figure 6 as well as Cases (d), (e), and (f) in Figure 6, we can learn that TC increases as θ grows. This is because the number of targets covered by each sensor increases as θ is getting larger. Overall, from Figure 6, we can observe that the initial battery lifetime bl of each sensor has more significant influence than the beamwidth θ on the performance of the algorithm.

Next, we study Figure 7 in which we fix the number of sensor nodes deployed to 250 and vary the number of targets to be covered from 20 to 70. This result clearly shows that the performance of our algorithm is close to the theoretical upper bound independent of the other parameters including bl , θ ,

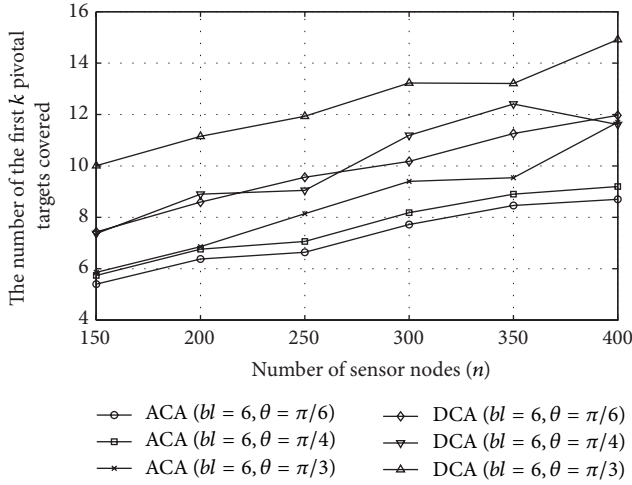


FIGURE 8: The number of the first k important targets covered by sensors by DCA-MCSN versus ACA.

and the number of targets m once we obtain enough coverage over the area.

5.2. Simulation Results for DCA-MCSN versus ACA. In this subsection, we compare the performance of DCA-MCSN against ACA (TEC-NC-Adjuster) in [5] (which showed outstanding performance for solving the target-temporal effective-sensing coverage with adjustable cameras problem, TEC-AC, in the simulations) under the 3 parameter settings (Cases (d)–(f)) while the number of targets to be covered is fixed to 50. We study how many important targets are fully covered by the two strategies (see Figure 8). From this result, we can observe that for both of the strategies, the number k of the most important targets which are fully covered during the mission period is affected by θ ; that is, k is steadily increasing along with the growth of θ . Furthermore, DCA-MCSN works better than ACA under all parameter settings, which implies that DCA-MCSN is more efficient than ACA for the mission-driven application considering the coverage quality of the targets with higher priority.

6. Conclusion

In this paper, we investigate a new coverage problem in camera sensor networks. In this problem, we study how to schedule a given set of camera sensor nodes to cover a set of given targets with weight related to their importance and fixed face directions such that the number of the first k important targets effectively covered during a given mission period can be maximized as well as the overall target-temporal coverage is maximized (as the secondary optimization goal). The main contribution of this paper is the possibility of using insufficient number of camera sensors to best support a coverage mission in a way that was never considered before. Since we do not always have enough number of sensor nodes to complete a coverage mission, our paper is of great applicability in many real application scenarios. As a future

work, we plan to study the applicability of this model in the hybrid sensor network of both a number of cheap static sensor nodes and a few expensive mobile sensor nodes, which we believe also have a wide range of applications in many real-life scenarios.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was jointly supported by National Natural Science Foundation of China under Grant 91124001, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China 10XNJ032.

References

- [1] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proceedings of the 24th IEEE International Conference on Computer Communications (INFOCOM '05)*, pp. 1976–1984, March 2005.
- [2] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Computer Communications*, vol. 29, no. 4, pp. 413–420, 2006.
- [3] C. Liu and G. Cao, "Spatial-temporal coverage optimization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 465–478, 2011.
- [4] C. Liu and G. H. Cao, "Distributed critical location coverage in wireless sensor networks with lifetime constraint," in *Proceedings of the 31th IEEE International Conference on Computer Communications (INFOCOM '13)*, 2012.
- [5] Y. Hong, D. Kim, D. Li, W. Chen, A. O. Tokuta, and Z. Ding, "Target-temporal effective-sensing coverage in mission-driven camera sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN '13)*, 2013.
- [6] M. Cardei and D.-Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333–340, 2005.
- [7] H. Ma and Y. Liu, "On coverage problems of directional sensor networks," in *Proceedings of the 1st International Conference on Mobile Ad-Hoc and Sensor Networks (MSN '05)*, 2005.
- [8] Y. Cai, W. Lou, M. Li, and X.-Y. Li, "Target-oriented scheduling in directional sensor networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 1550–1558, May 2007.
- [9] J. Ai and A. A. Abouzeid, "Coverage by directional sensors in randomly deployed wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 1, pp. 21–41, 2006.
- [10] J. Adriaens, S. Megerian, and M. Potkonjak, "Optimal worst-case coverage of directional field-of-view sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (SECON '06)*, pp. 336–345, September 2006.
- [11] G. Fusco and H. Gupta, "Selection and orientation of directional sensors for coverage maximization," in *Proceedings of the 6th*

Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09), June 2009.

- [12] W. Cheng, S. Li, X. Liao, S. Changxiang, and H. Chen, "Maximal coverage scheduling in randomly deployed directional sensor networks," in *Proceedings of the International Conference on Parallel Processing Workshops (ICPPW '07)*, September 2007.
- [13] Y. Wang and G. Cao, "On full-view coverage in camera sensor networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 1781–1789, April 2011.
- [14] L. Liu, H. Ma, and X. Zhang, "On directional K-coverage analysis of randomly deployed camera sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '08)*, pp. 2707–2711, May 2008.
- [15] B. Raytchev, I. Yoda, and K. Sakaue, "Head pose estimation by nonlinear manifold learning," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 462–466, August 2004.

