Minimizing Service Delay in Directional Sensor Networks

Yi Wang and Guohong Cao
Department of Computer Science and Engineering
The Pennsylvania State University
E-mail: {yuw124, gcao}@cse.psu.edu

***Abstract*—In directional sensor networks, sensors can steer around to serve multiple target points. Most previous works assume there are always enough deployed sensors so that all target points can be served simultaneously. However, this assumption may not hold when the mission requirement changes or when more target points need to be served. Since it is not always practical to deploy new sensors, we propose to reconfigure the network by letting existing sensors steer and serve the targets periodically. As a result, targets may not be served continuously, and the service delay affects the quality of service. One important problem is how to choose the optimal set of targets to serve by each sensor such that the maximum service delay is minimized. We first show that this problem is NP-complete, and then we propose a centralized protocol whose performance is bounded by a logarithm factor of the optimal solution. We also propose a distributed protocol which achieves the same performance as the centralized protocol. Finally, we extend the optimization model and the protocols by considering the rotation delay, which is critical for some applications but ignored by previous work.**

I. Introduction

Directional sensor networks have been applied to many real applications such as camera networks for vision based sensing [2], [21] radar networks for weather monitoring [18], sonar network for underwater object detection [3], etc. Compared to a conventional omni-directional sensor, a directional sensor has limited sensing angle and the sensing range is often represented by a sector [1], [7], [10], such as the field of view of a camera sensor [2].

Most existing works in directional sensor networks focused on finding a static configuration of the sensing directions such that the coverage quality is maximized [1], [10], [17]. However, directional sensors can extend their sensing ability by dynamically switching among sectors. As an example, it has been shown in [7] that properly steering the sensors onto different sectors at different time can reduce the degree of coverage redundancy, and therefore increase the coverage efficiency and prolong the network lifetime. Similar to many previous works, they assume that the number of deployed sensors is sufficient and all the target points can be covered continuously during the lifetime of the network. In practice, this assumption may not hold as some deployed sensors may fail and some new target points may be added as the mission requirement changes [16].

As sensors may not cover (serve) every target point continuously, the delay to cover the target point affects the quality of service. For example, suppose a surveillance camera monitors two target points P_1 and P_2 on two different directions alternatively, i.e., at even time slots $t = 0, 2, 4, \ldots$, it monitors P_1; at odd time slots $t = 1, 3, 5, \ldots$, it monitors P_2. Then, there is one time slot delay between any two sequential services on each target. This delay, referred to as the *service delay*, determines how fast an interested event can be detected at a target point.

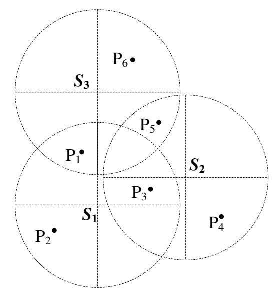

Fig. 1. A example of a directional sensor network. There are several different schedules: Schedule 1: $S_1 \to \{P_1, P_2, P_3\}, S_2 \to \{P_4, P_5\}, S_3 \to \{P_6\}$; Schedule 2: $S_1 \to \{P_2, P_3\}, S_2 \to \{P_4, P_5\}, S_3 \to \{P_1, P_6\}$; Schedule 3: $S_1 \to \{P_1, P_2\}, S_2 \to \{P_3, P_4\}, S_3 \to \{P_5, P_6\}$.

One important problem is how to serve all target points with minimum service delay given a fixed sensor deployment. A naive solution is to let each sensor steer and serve every target around it one by one. However, the service delay of this approach would be too long. Since there is some overlap between sensors' range, a target may be served by multiple sensors. As a result, a sensor does not have to serve all the targets around it, and hence it can reduce the service delay. However, it is a challenge to find the optimal set of targets to serve for each sensor such that all targets can be served and the longest service delay is minimized. Moreover, sensors should make the scheduling decision locally without global information.

Consider the example shown in Fig. 1, where the sensing area of each sensor is divided into 4 sectors. One way to schedule the sensors (Schedule 1) is to let sensor S_1 serve target points P_1, P_2 and P_3, S_2 serve P_4 and P_5, and S_3 serve P_6. Assume each sector has a service delay of 1. With this schedule, since S_1 has to serve three sectors covering P_1, P_2 and P_3, the maximum service delay is 2. However, in an optimal schedule (see Schedule 2 and 3 in Fig. 1), each sensor serves 2 sectors. Then, the maximum service delay is only 1. As can be seen, the service delay of a sensor is one less than

This work was supported in part by National Science Foundation under grant CNS-0916171.

the number of sectors it serves. Therefore, in order to minimize the service delay, we should minimize the number of sectors that a sensor should serve.

In the aforementioned schedule, we assume that the total service delay only depends on the number of sectors served by each sensor. This is true if the rotation delay is negligible compared with the total delay, which is a common assumption used in existing works (e.g. [7]). However, slowly moving devices such as radar always take non-negligible time to steer from one sector to another. Then, the total service delay also depends on the rotation delay from one sector to another. As a result, the number of served sectors and the distribution of these sectors both affect the service delay. As an illustration, we compare the two schedules (Schedule 2 and 3) shown in Fig. 1. In both cases, each sensor serves two sectors. Schedule 2 requires the sensors to move across 2 sectors (e.g., from the sector of P_2 to that of P_3), while Schedule 3 only requires them to cross 1 sector and thus it has shorter rotation delay. Since both schedules have the same number of selected sectors, Schedule 3 is preferred.

The above example shows that different schedules may result in different service delay. Although the optimal solution for this example can be easily found, in a complex network setting where many sensors are arbitrarily deployed, we need a systematic way to address the problem. Our contribution in this paper can be summarized as follows.

a) We rigorously define the service delay minimization problem in a steerable directional sensor network, which is shown to be NP-complete.

b) We propose a centralized protocol and prove that its performance is bounded by a logarithm factor of the optimal solution.

c) We design a distributed protocol and prove that it can achieve the same performance as the centralized protocol.

d) We extend the problem model by considering the rotation delay. We prove that the problem is NP-complete and then propose an efficient heuristic based solution.

To the best of our knowledge, this is the first work studying the service delay minimization problem in directional sensor networks. Although the problem of reducing *average* coverage delay has been studied before in disk (omni-directional) sensing model where each sensor has only two states (on and off) [8], the problem here is more challenging since each sensor now has many states (sectors) to consider. Also, we are able to provide a delay bound guarantee for the worst case, which is totally different from previous work.

The rest of the paper is organized as follows. In Section II, we introduce the service delay minimization problem and show that it is NP-complete. In Section III, a centralized protocol is presented and its performance is analyzed theoretically. A distributed protocol is then proposed in Section IV. In Section V, we generalize the problem model to consider rotation delay. Section VI evaluates the performance of these protocols. Related works are reviewed and compared in Section VII. Section VIII concludes the paper.

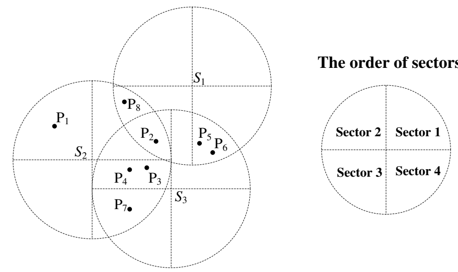

Fig. 2. An example of 3 sensors and 8 targets; each sensor has 4 sectors. In this example, $S_{1,1} = \emptyset$, $S_{1,2} = \emptyset$, $S_{1,3} = \{P_2, P_8\}$ and $S_{1,4} = \{P_5, P_6\}$; $S_{2,1} = \{P_2, P_8\}$, $S_{2,2} = \{P_1\}$, $S_{2,3} = \emptyset$ and $S_{2,4} = \{P_3, P_4, P_7\}$; $S_{3,1} = \{P_5, P_6\}$, $S_{3,2} = \{P_2, P_3, P_4\}$, $S_{3,3} = \{P_7\}$ and $S_{1,4} = \emptyset$

TABLE I
NOTATIONS

n	Total number of sensors
m	Total number of target points
a	The angle of a sensing sector
w	Number of sectors per sensor; $w = 360/a$
r	The radius of sensing sector
S_i	The i-th sensor
$S_{i,j}$	The j-th sector of i-th sensor
P_k	The k-th target point
d_s	Time duration to serve a sector (see Section V-A)
d_r	Time duration to cross a sector (see Section V-A)

II. Service Delay Minimization Problem

In this section, we formally define the service delay minimization problem (SDMP) and prove that it is NP-hard.

A. Problem Statement

Given m target points[1] $\{P_1, \dots, P_m\}$ and n directional sensors $\{S_1, \dots, S_n\}$, each sensor is associated with a sensing radius r and a sensing angle a. The sensing area (with radius r) around a sensor is partitioned into w sectors, where $w = 360/a$. We number the w sectors in counterclockwise order and let the first one start from the X(positive)-axis (Fig. 2). Let $S_{i,j}$ denote the j-th sector of the i-th sensor. Without ambiguity we also let $S_{i,j}$ denote the targets that can be served by sector $S_{i,j}$. For example in Fig. 2, S_1 has four sectors: $S_{1,1} = \emptyset$, $S_{1,2} = \emptyset$, $S_{1,3} = \{P_2, P_8\}$ and $S_{1,4} = \{P_5, P_6\}$. We assume that the time is divided into slots and a sensor can only serve one sector in one time slot. Table I summarizes the notations used throughout the paper. We formulate the service delay minimization problem as follows.

Definition 2.1 (SDMP) Suppose the rotation delay is 0. Given m targets and n sensors, each of which has w sectors, SDMP asks for a selection of at most W sectors for each sensor such that every target point is served by at least one selected sector and W is minimized.

B. NP-completeness of SDMP

We prove that SDMP is NP-hard via a reduction from the DCS (Directional Cover Set) problem, which has been proved NP-hard in [7].

Theorem 2.2 The service delay minimization problem defined in definition 2.1 is NP-complete.

[1] We use "targets" for short in this paper.

Proof: The DCS problem is a special case of SDMP when we ask for a selection in which each sensor can select at most 1 sector and all the target points can be served. On the other hand, it is not difficult to see that given a schedule, we can verify in polynomial time if the total number of selected sectors of each sensor is below a given threshold and if every target point is served by the selected sectors. Therefore, SDMP is NP-complete. ■

III. A Centralized Protocol for SDMP

Since SDMP is NP-complete, we can only propose heuristic based solution. In this section, we first present a centralized protocol, and then give its approximation bound.

A. Protocol Description

The protocol runs round by round and gradually selects the sectors for each sensor to serve. In each round, it selects at most one new sector for each sensor such that the number of new targets served by the selected sectors is maximized. The protocol stops when all targets are served by the selected sectors. The detailed description of the protocol is shown in Fig. 3.

Initially, as some targets can only be uniquely served by some specific sectors, they are selected first. Then the number of selected sectors for each sensor is counted and recorded. After that, the selection rounds begin. A sensor who has already selected w_i sectors in the initial step are not allowed to select any new sector in the first w_i rounds.

In each round those sensors that are eligible to choose can select at most one new sector for each of them. Ideally, we should make the selection such that the number of new targets being served in each round is maximized and thus the total number of selected sectors can be minimized. However, this is difficult due to the following observation.

Observation: Given an universal set $P = \{P_1, \ldots, P_m\}$ and n supersets $S_1, \ldots, S_n$, where S_i is a collection of some subsets $\{S_{i,1}, \ldots, S_{i,w}\}$ of P, the problem of selecting at most one $S_{i,j}$ from each S_i such that the total number of elements covered by them is maximized is NP-hard.

A proof can be found in [14]. Since it is NP-hard, we propose a greedy solution. Consider all sensors that can select in this round, we first find one that can serve the most number of new targets, and add the corresponding sectors into selection. Then this sensor is removed from our consideration and the remaining sensors are searched for the next best sector. This process continues until all sensors have been considered. If there are still some targets not served, a new round is started. The protocol stops when all targets are served.

Some optimization can be made to further improve the efficiency. We can group targets together as one unit if they can be served by the same set of sectors. For example in Fig. 2, targets P_3 and P_4 are grouped as one unit. Similarly, P_5 and P_6 are grouped as one. Then during the selection process, the contribution of each new sector is calculated based on how many new units it can serve. We expect that on average the number of units is less than the number of targets, which can help improve the performance (see next section for the reason). Another optimization is that when the protocol stops, we go through all the selected sectors and remove the sectors within which all targets can be served by other selected sectors.

A Centralized Scheduling Protocol for SDMP:

0. Let $S_{i,j} \subseteq P$ be the subset of targets that can be served
1. by the j-th sector of sensor S_i;
2. $C \leftarrow \emptyset, Selected \leftarrow \emptyset$;
3. $Delay_i \leftarrow 0, 1 \leq i \leq n$;
4. For $i = 1$ to n and $j = 1$ to w, do
5. If $S_{i,j}$ contains some targets that others do not, then
6. $Selected \leftarrow Selected \cup \{S_{i,j}\}$ and $C \leftarrow C \cup S_{i,j}$;
7. End for (i and j)
8. $Delay_i \leftarrow |\{S_{i,j} \in Selected, 1 \leq j \leq w\}|, 1 \leq i \leq n$;
9. For $W = 1$ to w, do
10. If $\{S_{i,j} : Delay_i < W, S_{i,j} \notin Selected, S_{i,j} \setminus C \neq \emptyset\} \neq \emptyset$;
11. Find S_{i_0,j_0} with maximum $|S_{i,j} \setminus C|$ in above set;
12. $Selected \leftarrow Selected \cup \{S_{i_0,j_0}\}$ and $C \leftarrow C \cup S_{i_0,j_0}$;
13. $Delay_{i_0} \leftarrow Delay_{i_0} + 1$;
14. If $C = P$, then stop and output $Selected$;
15. Else goto 10;
16. Else continue for next W;
17. End for (W).

Fig. 3. A Centralized Protocol for SDMP.

We use the example in Fig. 2 to show how the protocol works. Initially, since P_1 can only be served by sector $S_{2,2}$, $S_{2,2}$ is selected first. Then the first round begins. Because S_2 has already selected one sector, it can not participate in the first selection round. Among the others, since $S_{3,2}$ contains the highest number (3) of new targets (P_2, P_3, P_4), it is selected for S_3. Then we can consider S_1, which can select $S_{1,4}$ containing 2 new targets (P_5, P_6), and the first round ends. Since there are still targets not served, we start the second round. Now all sensors are eligible to select. We select the one that serves the most number of new targets: either $S_{2,4}$ or $S_{3,3}$, which can serve P_7. Then, we select $S_{1,3}$ for P_8. Now all targets are served. Since each sensor serves at most 2 sectors, the worst case service delay is 1.

B. Approximation Bound Analysis

In this section, we analyze the performance of the above protocol. We show that the maximum number of sectors selected for any sensor is upper bounded by a logarithm factor of the optimal value.

Theorem 3.1 (Approximation Bound) In the centralized protocol, the number of sectors each sensor serves is upper bounded by $\alpha \ln m \cdot W_{opt}$, where $\alpha \approx 2.31$, m is the number of targets, and W_{opt} is maximum number of sectors selected for a sensor in the optimal solution.

Proof: We can number the targets in the order of being first covered by some sector. Let $W(k)$ be the round number when target P_k is first covered and $C_{W(k)} = \{e_j : W(j) = W(k)\}$, i.e., the targets covered in round $W(k)$ but not covered before that. Let's define the price of target P_k as $price(P_k) = 1/|C_{W(k)}|$.

We assume W_{opt} is the optimal value achieved by the optimal solution. Then we claim that, for some constant α

$$price(P_k) = \frac{1}{|C_{W(k)}|} \leq \frac{\alpha}{m-k+1} \cdot W_{opt}. \quad (1)$$

If this is true, the total number of rounds is

$$\sum_{k=1}^{m} price(P_k) \leq \sum_{k=1}^{m} \frac{\alpha}{m-k+1} \cdot W_{opt} \leq \alpha \ln m W_{opt}.$$

which will demonstrate the correctness of the theorem.

Now we prove the claim is correct. Suppose the $(k-1)$-th round is just finished and the k-th round is about to start, which means that each sensor has selected at most $k-1$ sectors. Let MAX_k be the maximum number of new targets that can be covered in the k-th round, no matter what selection scheme is used. We first show that

$$MAX_k \geq \frac{m - \sum_{j=1}^{k-1} |C_j|}{W_{opt}}. \quad (2)$$

In fact, after the first $k-1$ rounds, there are still $m - \sum_{j=1}^{k-1} |C_j|$ leftover targets. Consider the optimal solution. Let $R_k^{OPT}(i)$ be the set of sectors of sensor S_i in the optimal solution which are not used by our algorithm before the k-th round. Obviously, $|R_k^{OPT}(i)| \leq W_{opt}$ and $\bigcup_{i=1}^{n} R_k^{OPT}(i)$ covers all the leftover targets.

To estimate a lower bound of MAX_k, consider the following process. First we select at most one sector from each $R_k^{OPT}(i)$, $1 \leq i \leq n$, such that the union of these sectors can cover the most number of new targets. Denote this number by n_1. Then we remove these targets from the leftover and also remove the selected sectors from each $R_k^{OPT}(i)$. We repeat the above process to select at most one sector from each $R_k^{OPT}(i)$, $1 \leq i \leq n$, such that the union of them covers the most number of new targets. Denote this number by n_2. Suppose this process has repeated t times until all leftover targets are covered.

In the above process, we can see $n_1 \leq n_2 \leq \ldots \leq n_t$, and $n_1 + n_2 + \ldots + n_t = m - \sum_{j=1}^{k-1} |C_j|$, and also $t \leq W_{opt}$. So

$$n_1 \geq \frac{m - \sum_{j=1}^{k-1} |C_j|}{t} \geq \frac{m - \sum_{j=1}^{k-1} |C_j|}{W_{opt}}.$$

Since we can choose among all unused sectors, including those in $R_k^{OPT}(i)$, it is not difficult to see that $MAX_k \geq n_1$. Thus (2) is proved.

Now we will show that the actual number of new targets served by our protocol in round k is no less than a constant portion of MAX_k. We can number the sectors selected in round k and the corresponding sensors in the order of being selected by our algorithm, i.e., the i-th selected sector is from S_i. Denote $benefit_i$ as the number of new targets covered by the i-th selected sector. To set up the comparison, let $OPT_k = \{S_{1,j_1}, S_{2,j_2}, \ldots, S_{n,j_n}\}$ be a selection of sectors that can yield MAX_k new targets to be served, where S_{i,j_i} is the j_i-th sector of sensor S_i. Let $benefit_i$ be the number of new targets served by S_{i,j_i}. Since we always choose the sector that can serve the most number of new targets, we have

$$benefit_1 \geq \frac{MAX_k}{n}. \quad (3)$$

We will show that for $i = 1, \ldots, n-1$,

$$benefit_{i+1} \geq \frac{MAX_k - 2\sum_{r=1}^{i} benefit_r}{n}. \quad (4)$$

To see this, consider the situation when we are about to select the $(i+1)$-th sector. There are $n-i$ sectors in OPT_k which can be used by us, namely $\{S_{i+1,j_{i+1}}, \ldots, S_{n,j_n}\}$. The total number of new targets served by them is at least

$$max\{0, MAX_k - 2\sum_{r=1}^{i} benefit_r\}. \quad (5)$$

The reason is as follows. Clearly the joint benefit of OPT_k is $MAX_k - \sum_{r=1}^{i} benefit_r$. Consider any sector S_{r,j_r} of the first i sectors in OPT_k. If it is also used by our algorithm, their benefit is 0 at present. If it is not selected, its current benefit does not exceed $benefit_r$, as if it does, it would be chosen in step r. So by subtracting the benefit of these sectors, we lower bound the joint benefit of the leftover sectors in OPT_k by (5). Finally, as we always choose the sector with the maximum benefit among the leftover, we have

$$benefit_{i+1} \geq \frac{max\{0, MAX_k - 2\sum_{r=1}^{i} benefit_r\}}{n-i}.$$

This is true for all $i = 1, \ldots, n-1$, which yields (4). Based on (3) and (4), we have for $i = 1, \ldots, n$,

$$\sum_{r=1}^{i} benefit_r \geq [1 - (1 - \frac{2}{n})^i] \cdot \frac{1}{2} MAX_k, \quad (6)$$

In fact, this can be proved by induction on i (omitted here). To lower bound the number of new targets served in round k, simply let $i = n$. Since $[1 - (1 - \frac{2}{n})^n]/2 \geq (1 - e^2)/2 \approx 0.43$ and let this number be $1/\alpha$, we have

$$|C_k| = \sum_{r=1}^{n} benefit_r \geq \frac{MAX_k}{\alpha},$$

with $\alpha \approx 2.31$. Combine this and (2), we have

$$price(P_k) \leq \frac{\alpha}{MAX_{w(k)}} \leq \frac{\alpha W_{opt}}{m - \sum_{j=1}^{w(k)-1} |C_j|} \leq \frac{\alpha W_{opt}}{m-k+1}$$

The last "$\leq$" is because $\sum_{j=1}^{w(k)-1} |C_j| \leq k$. Therefore (1) is proved and the whole proof is done. ■

IV. A Distributed Protocol for SDMP

In this section, we propose a distributed protocol for SDMP. We also prove that the distributed protocol can achieve the same selection result as the centralized protocol.

A. Protocol Description

The protocol runs at each sensor node, which is supposed to know the distribution of the targets within its sensing range. Two sensors can communicate with each other if there is a target that can be served by both of them. For each sensor, the protocol converges if all the targets around it are served. The detailed description of the protocol is shown in Fig. 4.

During initialization, each sensor finds targets that can only be covered by itself (e.g, by first asking all its neighbors for the target sets that they can cover and subtracting from its own set). Then the sectors needed to cover those targets are selected. After this, the sensor sends out message to its neighbors regarding how many sectors it has selected and how many new targets it can cover.

After initialization, each sensor begins to exchange information with its neighbors and selects sectors to serve. A sensor needs the following information to make a decision: (1) how many sectors has each neighbor selected (d_j); and (2) how many new targets can be served by each neighbor ($benefit_j$). Note that a sensor may have many unused sectors. The sector which can cover the most number of new targets is most likely to be selected next. The number of targets served by that sector is defined as the benefit of that sensor. This information is needed for a sensor to locally decide whether it can select a new sector. Specifically, a sensor can add a new sector if: (1) all of its neighbors have already selected at least the same number of sectors except for those who already stop selecting (sensors stop selecting if all the targets within its range are served); and (2) its benefit is greater than the benefit of those neighbors who have selected the same number of sectors. Ties are broken by giving preference to sensors with small sequence number. If a sensor decides to select a new sector, it immediately sends out a message to tell its neighbors about this decision. A sensor also sends out a message if its current benefit changes (possibly because some of the targets around it have been served by a newly selected sector of its neighbors).

We do not require any synchronization in the message exchanging process. Message from different sender may experience different delay. It is only required that messages sent from one sensor should be received in the original order.

A Distributed Protocol for SDMP - at S_i:

Initially

0. Find the set of targets that can only be covered by S_i;
1. Let $Selected_i$ be the set of sectors covering those targets;
2. $d_i \leftarrow |Selected_i|$;
3. $E_i \leftarrow \{P_j : P_j$ is covered by some sector in $Selected_i\}$;
3. $UC_i \leftarrow \{$ All targets within distance r from S_i $\} \setminus E_i$;
4. If $UC_i = \emptyset$, $d_i \leftarrow \infty$;
5. $N_i \leftarrow \{$ $benefit_j = \infty : S_j$ is S_i's neighbor $\}$;
6. Find the sector $S_{i,max}$ that covers the most number of
7. points, and $benefit_i \leftarrow |S_{i,max} \setminus E_i|$;
8. $New\_Covered_i \leftarrow \emptyset$;
9. Broadcast to all S_i's neighbors:
10. $M_i = \{New\_Covered_i, d_i, benefit_i\}$.

On receiving a message from neighbor S_j

11. If $New\_Covered_j = \emptyset$, go on to line 14; else
12. $UC_i \leftarrow UC_i \setminus New\_Covered_j$;
13. if $UC_i = \emptyset$, then $d_i \leftarrow \infty$;
14. Update d_j according to the message;
15. Calculates S_i's most beneficial sector $S_{i,max}$, and
16. $benefit_i \leftarrow |S_{i,max} \cap UC_i|$;
17. If for any neighbor S_j, one of the following is true:
18. $d_i < d_j$;
19. $d_i = d_j$, and either $benefit_i > benefit_j$
20. or $benefit_i = benefit_j$ but $i < j$
21. Then
22. $d_i \leftarrow d_i + 1$,
23. $New\_Covered_i \leftarrow S_{i,max} \setminus UC_i$,
24. $UC_i \leftarrow UC_i \setminus S_{i,max}$;
25. If $UC_i = 0$, $d_i \leftarrow \infty$;
26. $Selected_i \leftarrow Selected_i \cup \{S_{i,max}\}$;
27. Recalculate S_i's most beneficial sector $S_{i,max}$;
28. $benefit_i \leftarrow |S_{i,max} \cap UC_i|$;
29. Broadcast to all neighbors:
30. $M_i = \{New\_Covered_i, d_i, benefit_i\}$.

Fig. 4. A Distributed Protocol for SDMP.

B. Performance Analysis

In this section we show that the distributed protocol can achieve the same performance as the centralized protocol. To simplify the analysis, we modify the two protocols a little as follows without affecting their scheduling results.

In the distributed protocol, the sensor will continue to select and communicate even when all the targets around it have been served. It will select a virtual sector if the two conditions mentioned above are met. Then it sends out a message containing a "void" decision and zero benefit value. Its neighbors will treat this as a regular decision and update the corresponding information accordingly. Similarly, in the centralized protocol, if a sensor can not find a new sector with non-zero benefit, it selects a virtual sector.

It is not difficult to see that this modification will not affect the selection result in both protocols. Now we can prove the following result.

Theorem 4.1 Given the same sensor deployment and target distribution, the distributed protocol yields the same selection result as the centralized protocol, i.e., for any sensor S_i, a sector $S_{i,j}$ is selected by the distributed protocol if and only if it is selected by the centralized protocol.

Proof: We use DP to stand for the distributed protocol and CP for the centralized protocol. The benefit of S_i (denoted as $benefit_i$) is defined as the maximum number of new targets that can be served by a new sector of S_i. For $i = 1, \dots, n$, N_i denotes the neighbors of S_i. In DP, we say a sensor S_i "has selected a new sector" only if it has sent out the decision message and the message has been received by at least one of S_i's neighbor. Without loss of generality, we assume S_i updates d_i and $benefit_i$ right after it has selected a new sector. A sensor is said to be in W level if it has selected exactly W sectors. Due to space limitation, we only give the sketch of the proof.

For each S_i, we re-order its sectors in the sequence as they are selected in DP; i.e., S_i selects $S_{i,1}$ first, and then $S_{i,2}$ and so on. Denote $Sector_W = \{S_{i,W} : i = 1, \dots, n\}$, i.e., the set of W-th sectors that are selected in DP. Then the theorem is equivalent to the following claim.

Claim: $Sector_W$ is exactly the set of sectors that are selected by CP in round W.

We prove it by induction on W. Suppose it is true for $Sector_1, \dots, Sector_W$. Let us consider $Sector_{W+1}$. Suppose $S_{i_1,j_1} \in Sector_{W+1}$ and S_{i_1} is the first to select the $(W+1)$-th sector in DP. This means at that time all of S_{i_1}'s neighbors are in W-level and all the other sensors are in W or lower level. So for $S_i \in N_{i_1}$:

$$benefit_{i_1} \geq benefit_i, \text{“} = \text{” only if } i_1 < i, \qquad (7)$$

where $benefit_{i_1}$ and $benefit_i$ are, respectively, the benefit of S_{i_1} and S_i when S_{i_1} selects the $(W+1)$-th sector in DP.

Now we can show that S_{i_1} will select a sector earlier than its neighbors in the $(W+1)$-th round of CP. If this is not true, we can always find some $S_{i_0} \in N_{i_1}$ which makes selection the earliest among all S_{i_1}'s neighbors in round $W+1$ in CP. Then this implies

$$benefit'_{i_0} \geq benefit'_{i_1}, \text{“} = \text{”} \text{ only if } i_0 < i_1. \quad (8)$$

where $benefit'_{i_0}$ and $benefit'_{i_1}$ are, respectively, the benefit of S_{i_0} and S_{i_1} when S_{i_0} selects its $(W+1)$-th sector in CP.

Then based on the induction assumption and the fact that the benefit of a sensor can not increase as the selection proceeds, we can show

$$benefit_{i_0} \geq benefit'_{i_0}, \text{ and } benefit_{i_1} = benefit'_{i_1}. \quad (9)$$

Thus we have a contradiction to (7) by combining (8) and (9) together. Then we know that when S_{i_1} selects a new sector in CP, its benefit is just $benefit_{i_1}$, and the most beneficial sector is just S_{i_1,j_1}.

We can further prove by induction within round $W+1$ that all the other sectors in $Sector_{W+1}$ are also selected by CP. To see this, consider the sectors in $Sector_{W+1}$ in the order as they are selected in DP: $S_{i_1,j_1}, \ldots, S_{i_n,j_n}$. We assume $S_{i_1,j_1}, \ldots, S_{i_k,j_k} (k < n)$ are known to be selected by CP in round $W+1$, and for sensors $S_{i_1}, \ldots, S_{i_k}$, any neighbors of them can not select the $(W+1)$-th sector before they select in round $W+1$ of CP.

Then consider $S_{i_{k+1}}$, which is the $(k+1)$-th sensor that has selected its $(W+1)$-th sector in DP. This implies $\forall S_{i_0} \in N_{i_{k+1}}$ and $S_{i_0} \notin \{S_{i_1}, \ldots, S_{i_k}\}$, $S_{i_{k+1}}$ selects its $(W+1)$-th sector before S_{i_0} does in DP. So,

$$benefit_{i_{k+1}} \geq benefit_{i_0}, \text{“} = \text{”} \text{ only if } i_{k+1} < i_0, \quad (10)$$

where $benefit_{i_{k+1}}$ and $benefit_{i_0}$ are, respectively, the benefit of $S_{i_{k+1}}$ and S_{i_0} when $S_{i_{k+1}}$ selects its $(W+1)$-th sector in DP.

Now suppose S_{i_0} is the earliest to select among all $S_{i_{k+1}}$'s neighbors that are not in $\{S_{i_1}, \ldots, S_{i_k}\}$ and it does so earlier than $S_{i_{k+1}}$ in round $(W+1)$ in CP. This implies:

$$benefit'_{i_0} \geq benefit'_{i_{k+1}}, \text{“} = \text{”} \text{ only if } i_0 < i_{k+1}, \quad (11)$$

where $benefit_{i_0}$' and $benefit'_{i_{k+1}}$ are, respectively, the benefit of S_{i_0} and $S_{i_{k+1}}$ when S_{i_0} is about to select its $(W+1)$-th sector in CP.

Then similarly we can show that

$$benefit_{i_0} \geq benefit'_{i_0}, \text{ and } benefit'_{i_{k+1}} \geq benefit_{i_{k+1}}. \quad (12)$$

Thus we have a contradiction to (10) by combining (11) and (12) together. Then we know that none of $S_{i_{k+1}}$'s neighbors that are not in $\{S_{i_1}, \ldots, S_{i_k}\}$ can select a new sector before $S_{i_{k+1}}$ in the $(W+1)$-th round of CP. Since those neighbors in $\{S_{i_1}, \ldots, S_{i_k}\}$ are known to select earlier than $S_{i_{k+1}}$ in CP, which is the induction assumption, $benefit_{i_{k+1}}$ is just the benefit when $S_{i_{k+1}}$ selects in CP, and the most beneficial sector is just $S_{i_{k+1},j_{k+1}}$.

Then we know that all sectors in Sector$(W+1)$ is also selected by CP in round $W+1$. By the fact that one sensor can select exactly one sector in each round, we know $Sector_{W+1}$ is exactly the subset of sectors selected in round $W+1$ in CP. Therefore, our claim is correct and the proof is complete. ■

V. SDMP WITH ROTATION DELAY

In the SDMP problem, the total service delay only depends on the number of sectors served by each sensor. This is true if the rotation delay is negligible compared with the service delay. However, some slow moving devices such as radar may take non-negligible time to steer from one sector to another. In this section, we extend our previous optimization model by considering rotation delay.

A. Problem Statement - SDMP-R

Let d_s denote the time needed for the sensor to serve a sector and d_r denote the time needed to rotate across a sector. For example if a sensor S_1 currently stays in sector $S_{1,1}$ and wants to serve $S_{1,3}$ which is 2 sectors away (Fig. 6(a)), it will first take $2d_r$ time to reach that sector and then d_s time to finish the service.

For a specific target, the service delay is the length of the time period between any two sequential services by a sensor, including all the rotation time and service time. The longest service delay among all the targets is defined as the worst case service delay of the network. Now we can define the problem to be considered.

Definition 5.1 (SDMP-R) Given the above assumptions, the service delay minimization problem with rotation delay asks for a selection of sectors for each sensor such that each target point can be served by at least one selected sectors and the longest time for any sensor to traverse and serve all its selected sectors and then return back to a start position is minimized.

Based on the NP-completeness of SDMP, we can show that:

Theorem 5.2 The service delay minimization problem with rotation delay defined in definition 5.1 is NP-complete.

Proof: In fact, SDMP is a special case of SDMP-R if we let $d_r = 0$ and $d_s = 1$. From Theorem 2.2, SDMP-R is NP-hard. On the other hand, given a schedule we can verify if every target is served within the given delay bound in polynomial time. Therefore SDMP-R is NP-complete. ■

B. A Centralized Protocol for SDMP-R

The protocol has similar idea as the centralized protocol for SDMP, and it works round by round. Each round is associated with a parameter called global delay bound (W). In each round, every sensor can select new sectors as long as its total delay stays below W. The sectors with greatest benefit are selected first. The current round ends when no more sector can be selected given the current W. If there are still targets not served, a new round will be started with a new W. The detailed description of the protocol is shown in Fig. 5.

Initially, as some targets can only be uniquely served by some specific sectors, those sectors are selected first. Given these selected sectors, the total delay of each sensor to serve them ($Delay_i$) is calculated. There are two cases to consider (e.g., in Fig. 6 the greyed sectors are selected). First, if all

A Centralized Protocol for SDMP-R (d_s, d_r fixed):

0. Let subset $S_{i,j} \subseteq P$ be the targets served by j-th sector of sensor S_i;
1. $C \leftarrow \emptyset, Selected_i \leftarrow \emptyset, 1 \leq i \leq n$;
2. $Delay_i \leftarrow 0, 1 \leq i \leq n$;
3. For $i = 1$ to n and $j = 1$ to w, do
4.   If $S_{i,j}$ contains targets that no other can serve, then
5. $Selected_i \leftarrow Selected_i \cup \{S_{i,j}\}$ and $C \leftarrow C \cup S_{i,j}$;
6. End for (i and j)
7. $Delay_i \leftarrow DelayCompute(Selected_i), 1 \leq i \leq n$;
8. $W_{int} \leftarrow max\{Delay_i : 1 \leq i \leq n\}$;
8. For $W = W_{int}$ to $w \cdot (d_s + d_r)$, do
9.   If $\{S_{i,j} : DelayCompute(Selected_i \cup \{S_{i,j}\}) \leq W$, $S_{i,j} \notin Selected_i, S_{i,j} \setminus C \neq \emptyset\} \neq \emptyset$, then
10.   Find $S_{i,j}$ with maximum $|S_{i,j} \setminus C|$ in above set;
11. $Selected_i \leftarrow Selected_i \cup \{S_{i,j}\}$ and $C \leftarrow C \cup S_{i,j}$;
12. $Delay_i \leftarrow DelayCompute(Selected_i)$;
13   If $C = P$, then stop and output $\bigcup_{i=1}^{n} Selected_i$;
14   Else goto 9;
15.   Else increase W (specified in section V-B);
16. End for (W)

/*Subroutine to calculate the delay */
$DelayCompute(Selected_i)$
17. If $|Selected_i| \leq 1$, then return $|Selected_i| \cdot d_s$;
18. Find the maximum distance $dmax_i$ between any two sequential sectors in $Selected_i$;
19. If $dmax_i < w/2$, then Return $|Selected_i| \cdot d_s + w - d_s$;
20. Else return $|Selected_i| \cdot d_s + (w - dmax_i) \cdot 2 - ds$.

Fig. 5. A Centralized Protocol for SDMP-R.

the selected sectors lie within a "half circle", then the best way to serve them is to move forth and back between the two end sectors ($S_{1,1}$ and $S_{1,5}$ in Fig. 6(a)). In this case, the total rotation time equals to the total number of sectors between the two end positions multiplied by $2d_r$. Second, if the "distance" between any two adjacent selected sectors is smaller than half circle, the best way to serve them is just rotating around the full circle (Fig. 6(b)). In this case, the total rotation time equals to the total number of sectors (w) multiplied by d_r. The other part of the total delay is the time spent on serving the sectors, which equals to the number of selected sectors (3 in Fig. 6) multiplied by d_s. Then we combine these two parts together and get the total delay. Note that the total delay for each sensor will be dynamically changed if new sectors are selected during the following selection.

After initialization, if all targets are served, the protocol will stop. Otherwise, the selection rounds begin with W equal to the maximum total delay of all sensors.

At the beginning of each round, all sensors are considered eligible to select. We greedily select new sectors for eligible sensors; i.e., we find a sector such that: (1) it contains the most number of new targets; (2) selecting this sector will not cause the total delay (calculated as above) of that sensor to be greater than W. This process continues until no more sectors can be selected without violating the second rule. Then if there are still targets not served, we increase W and start a new round.

A large increasing step of W may result in more sectors being selected in the next round, which is beneficial for fast convergence of the protocol. However, we may also select more unnecessary sectors. Increasing W by 1 is the most conservative way, but this will cause the total number of rounds to be $\Theta(d_r + d_s)$, which means the running time of the protocol is not polynomial of the input length. In fact, we will ensure that at least one new sector can be selected in each round. Thus our method is to let each sensor propose a new value of W such that the sensor is able to select a new sector. Then the proposed values are compared and the minimum one is used as the new value for W.

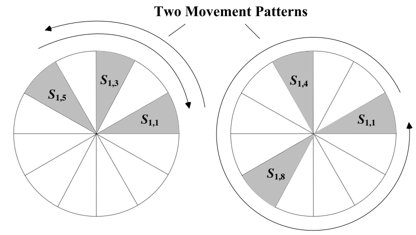

Fig. 6. Two cases on calculation of service delay (greyed sectors are selected).

Finally, the protocol stops when all targets are served.

C. Discussion on Distributed Implementation

We sketch the idea of a distributed implementation of the protocol for SDMP-R. The protocol is similar to the distributed protocol for SDMP presented in Section IV. Sensors exchange information about the current delay and the benefit with the neighbors. Based on this, a sensor can decide to select a new sector if its current delay is smaller than its neighbors' and selecting a new sector would not cause the current delay to be larger than the maximum of its neighbors'. Also, the sensor needs to check if its benefit is greater than those neighbors who have the same current delay. Some subtle issues should be further investigated. For example, how does the sensor locally increase the current delay bound (W) when no progress can be made? This is obvious if we have a global coordinator. But in distributed protocol, a sensor only knows the information from its neighbors. Some coordination policy must be specified.

VI. EVALUATIONS

In this section, we evaluate the performance of the proposed protocols in various scenarios.

A. Simulation Setup

In our simulation, there are 1000 targets ($m = 1000$) and the number of sensors (n) varies from 50 to 300. All the targets and sensors are uniformly distributed in a 400 $\times$ 400 area. The sensing radius of each sensor is 50 and each sensor's sensing area is partitioned into 16 sectors. For the evaluation of SDMP-R, per sector service delay (d_s) and rotation delay (d_r) are fixed to be 1. Each result shown here is a statistical average of 50 experiments.

Since no existing protocols are designed for minimizing the service delay in directional sensor networks, we compare our protocols (SDMP and SDMP-R) to the following three protocols.

- *Static*: In the static protocol, the sensor does not steer around; i.e., it chooses one sector and serves that sector continuously. To maximize the number of served targets, each sensor selects one sector containing the maximum number of targets.
- *Random:* In the random selection scheme, each target is assigned to a sector randomly among all sectors that can serve it.

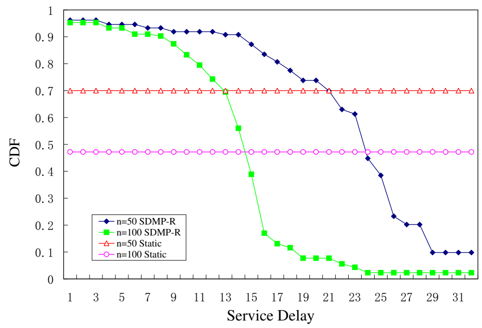

Fig. 7. Comparison with a static configuration.

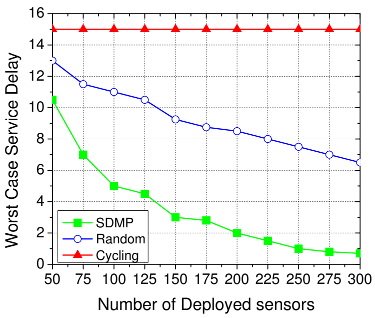

Fig. 8. Worst case delay comparison for SDMP.

- *Cycling:* In this scheme, each sensor simply rotates and serves all the sectors around it.

B. Comparison with the static scheme

In this section, we compare the performance of SDMP-R to the static scheme. The static scheme is designed for scenarios where there are enough sensors to cover the targets. When the number of sensors is not enough to cover all the targets, it may not perform well, and we need our SDMP-R to steer the sensors dynamically.

The results are shown in Fig. 7. We calculate the percentage of targets that have a total service delay above each value (X-axle). Results for $n = 50$ and $n = 100$ are included. From this figure, we can see that the static configuration only serves part of the targets with zero service delay (about 30% when $n = 50$ and 50% when $n = 100$). All the other targets are left with no service at all (with infinite delay). On the other hand, our protocol is able to serve much more targets with graceful performance degradation. When $n = 50$, about 80% of the targets are served with delay smaller than 26. When $n = 100$, about 90% of the targets are served with delay smaller than 18. The delay drops as the number of sensors increases. The scheduling protocol fills the gap between continuous service and no service at all in the static configuration.

C. The Performance of SDMP

We compare the worst case service delay of our protocol and the random selection for SDMP. The performance of the Cycling is also included as a comparison baseline.

The results are shown in Fig. 8. As can be seen, carefully selecting the sectors for each sensor can greatly reduce the worst delay in every density configuration. The delay in Random is twice as much as our protocol when n is 100, and it is more than four times when n is larger than 200. Also, as the number of sensors increases, the worst case service delay of both Random and our protocol drops. This is because as the density of sensor increases, the expected work load for each sensor drops. However, the delay drops much faster in our protocol, which further demonstrates the effectiveness of the scheduling.

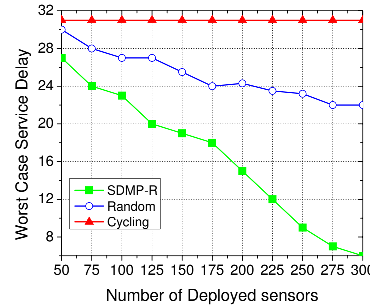

Fig. 9. Worst case delay comparison for SDMP-R.

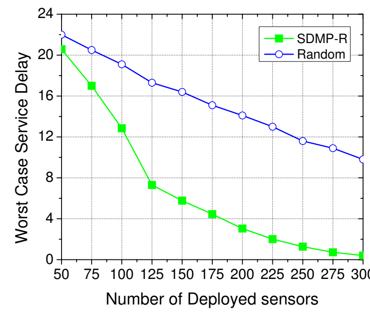

Fig. 10. Average delay for SDMP-R and Random.

D. The Performance of SDMP-R

Finally, we show the performance of our protocol for SDMP-R. The worst case delay comparison is shown Fig. 9, and we also include average delay comparison results in Fig. 10.

In both figures, our protocol has a lower delay than Random. As the number of sensor increases, the worst case delay and average delay drop much faster in our protocol. The worst case delay in random selection is 20% higher than ours when n is 100, and it is almost twice as much as ours when n is 200. This difference is even larger for the average delay. The average delay in Random is 50% higher when n is 100, and it is almost five times of ours when n is 200. From these results, we see that the proposed protocol for SDMP-R can effectively reduce both worst case and average case service delay.

VII. Related Works

The coverage problem in sensor network has been the focus of research during the past few years [9], [20]. Recently, some new problems and models have been identified [5], [19], [17], [22], [15], [4], etc. Among these new problems, the directional sensor network model is the most relevant to ours. In [1], the problem of how to maximize the coverage with minimum sensors is studied. This problem is proved to be NP-complete and a greedy heuristic is proposed. An extension work which requires each target to be k-covered can be found in [10]. They considered the dual optimization model of the problem, and they proposed greedy heuristics and derived performance bounds. A similar problem in $3D$ case was considered in [17]. They use a potential field based approach to decide the orientation of the sensors such that the area covered is as large as possible.

All the above works study how to achieve the best static configuration for a directional sensor network. They assume enough sensors are deployed such that the target points can be covered simultaneously. They do not consider the possibility of steering sensors to cover more target points when the original deployment is not enough. Although the authors of [7] considered the steering capability of the sensors, their focus was how to steer the sensors to different directions at different time to maximize network lifetime. They did not address the issue of how to accommodate new targets points with less number sensors. Also their work was based on heuristics and no performance bound was given.

Our work is also inspired by the study on the tradeoff between the coverage delay and the network lifetime in the traditional sensor model. Some of these works are inspired by an earlier work on "sweep coverage", which was first introduced in [11] for multi-robot system. In [13], a method called "virtual patrol" was proposed. The idea is to only activate a small number of sensors covering a stripe zone of the surveillance area each time. As sensors switch between on and off, the covered stripe zone can virtually move from one place to another. Thus each point of the area can be monitored periodically. In their work, the time between two sequential "on" statues determines the coverage delay for the area covered by the sensor. The key problem is how to schedule the on and off statuses of the sensors such that each target point is covered within a bounded time and the sensors do not drain their energy too fast. Some similar problems have been studied in [12], [6], [8]. Among them, [8] has proposed a distributed protocol to reduce the average coverage delay under given "on" time percentage, which is very interesting work, although no worst case performance bound is given.

The above works focus on the omni-direction (disk) sensing model, where each sensor has only two statuses: on and off. The coverage delay in these works is caused by turning off some sensors for power conservation. Compared with these works, the service delay in our study is inherent due to the insufficient number of sensors. Minimizing the service delay is much more complicated because each sensor has much more statuses (sectors) to consider. Moreover, the worst case performance of our protocols is proved to be bounded.

VIII. Conclusions

In this paper, we studied the service delay minimization problem in directional sensor networks. When the number of sensors is not enough to serve all the target points simultaneously, the sensors are scheduled to steer and serve the target points periodically. As targets may not be served continuously, the service delay affects the quality of service. One important problem is how to choose the optimal set of targets to serve for each sensor such that the maximum service delay is minimized. We first show that this problem is NP-complete, and then we propose a centralized protocol whose performance is bounded by a logarithm factor of the optimal solution. We also design a distributed protocol which can achieve the same performance bound as the centralized protocol. Finally, we extend the optimization model by considering the rotation delay, and propose scheduling protocols for this new model.

References

[1] J. Ai and A. A. Abouzeid. Coverage by directional sensors in randomly deployed wireless sensor networks. *J. Comb. Optim.*, 11(1):21–41, 2006.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, 2007.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia. State-of-the-art in protocol research for underwater acoustic sensor networks. In *Underwater Networks*, pages 7–16, 2006.

[4] X. Bai, Z. Yun, D. Xuan, T.-H. Lai, and W. Jia. Optimal patterns for four-connectivity and full coverage in wireless sensor networks. *IEEE Trans. Mob. Comput.*, 9(3):435–448, 2010.

[5] P. Balister, Z. Zheng, S. Kumar, and P. Sinha. Trap coverage: Allowing coverage holes of bounded diameter in wireless sensor networks. In *INFOCOM*, pages 136–144, 2009.

[6] Y. M. Baryshnikov, E. G. C. Jr., and K. J. Kwak. High performance sleep-wake sensor systems based on cyclic cellular automata. In *IPSN*, pages 517–526, 2008.

[7] Y. Cai, W. Lou, M. Li, and X.-Y. Li. Target-oriented scheduling in directional sensor networks. In *INFOCOM*, pages 1550–1558, 2007.

[8] Q. Cao, T. F. Abdelzaher, T. He, and J. A. Stankovic. Towards optimal sleep scheduling in sensor networks for rare-event detection. In *IPSN*, pages 20–27, 2005.

[9] M. Cardei and J. Wu. Energy-efficient coverage problems in wireless ad-hoc sensor networks. *Comput. Commun.*, 29(4):413–420, 2006.

[10] G. Fusco and H. Gupta. Selection and orientation of directional sensors for coverage maximization. In *IEEE SECON*, pages 556–564, 2009.

[11] D. W. Gage. Command control for many-robot systems. *Unmanned Systems Magazine*, 10(4):28–34, 1992.

[12] Y. Gu, J. Hwang, T. He, and D. H.-C. Du. usense: A unified asymmetric sensing coverage architecture for wireless sensor networks. In *ICDCS*, page 8, 2007.

[13] C. Gui and P. Mohapatra. Virtual patrol: a new power conservation design for surveillance using sensor networks. In *IPSN*, pages 246–253, 2005.

[14] D. S. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. International Thomson Publishing, 1996.

[15] G. S. Kasbekar, Y. Bejerano, and S. Sarkar. Lifetime and coverage guarantees through distributed coordinate-free sensor activation. In *MOBICOM*, pages 169–180, 2009.

[16] C. Liu and G. Cao. Spatial-temporal coverage optimization in wireless sensor networks. *to appear in IEEE Trans. Mob. Comput.*, 2010.

[17] H. Ma, X. Zhang, and A. Ming. A coverage-enhancing method for 3d directional sensor networks. In *INFOCOM*, pages 2791–2795, 2009.

[18] V. Manfredi and J. Kurose. Scan strategies for meteorological radars. In *NIPS*, 2007.

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

[20] P.-J. Wan and C.-W. Yi. Coverage by randomly deployed wireless sensor networks. *IEEE/ACM Trans. Netw.*, 14(SI):2658–2669, 2006.

[21] Y. Wang and G. Cao. On full-view coverage in camera sensor networks. In *INFOCOM 2011*.

[22] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C.-W. Yi. Data fusion improves the coverage of wireless sensor networks. In *MOBICOM*, pages 157–168, 2009.