

# Coverage by directional sensors in randomly deployed wireless sensor networks

Jing Ai · Alhussein A. Abouzeid

Accepted: 25 October 2005  
© Springer Science + Business Media, Inc. 2006

**Abstract** We study a novel “coverage by directional sensors” problem with tunable orientations on a set of discrete targets. We propose a Maximum Coverage with Minimum Sensors (MCMS) problem in which coverage in terms of the number of targets to be covered is maximized whereas the number of sensors to be activated is minimized. We present its exact Integer Linear Programming (ILP) formulation and an approximate (but computationally efficient) centralized greedy algorithm (CGA) solution. These centralized solutions are used as baselines for comparison. Then we provide a distributed greedy algorithm (DGA) solution. By incorporating a measure of the sensors residual energy into DGA, we further develop a Sensing Neighborhood Cooperative Sleeping (SNCS) protocol which performs adaptive scheduling on a larger time scale. Finally, we evaluate the properties of the proposed solutions and protocols in terms of providing coverage and maximizing network lifetime through extensive simulations. Moreover, for the case of circular coverage, we compare against the best known existing coverage algorithm.

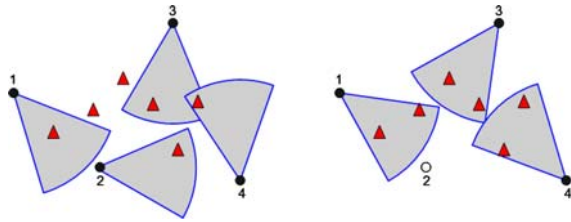
**Keywords** Directional sensors · Mathematical programming/optimization · Distributed algorithm · Scheduling · Network lifetime

## 1. Introduction

*Sensing coverage* is a fundamental problem in wireless sensor networks. It reflects how well the environment is monitored, and serves as a basis for applications such as physical phenomenon or target detection, classification and tracking. Due to the diversity of the sensor network applications, the concept of sensing coverage is subject to a wide range of interpretations. Nevertheless, only *isotropic sensors* have been studied in the literature. For example, in the most studied area or point coverage problems (e.g., (Slijepcevic and Potkonjak, 2001; Tian and Georganas, 2002; Wang et al., 2003; Zhang and Hou, 2005;

---

J. Ai (✉) · A. A. Abouzeid  
Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY  
12180, USA  
e-mail: aij@rpi.edu; abouzeid@ecse.rpi.edu



**Fig. 1** Two cases of four directional sensor nodes (black circles) deployed to cover six targets (red triangles) in a sensor field. [left] Case I: All four nodes are active while two targets are not covered (i.e. “uncovered”). [right] Case II: Three nodes are active with no targets uncovered; node 2 (white circle) is in sleep mode to conserve energy

Cardei and Du, 2005)), the sensing ability of sensors is abstracted as a circular region (or disk) and an event or target is detected in a binary sense depending on whether it is inside such a sensing disk or not.

To the best of our knowledge, no research work on sensor networks coverage for directional sensors has been done. Compared to isotropic sensors, directional sensors are obviously in that *the coverage region of a directional sensor is determined by both its location and orientation*. This can be best illustrated by the example shown in fig. 1.

In this paper, we study the problem of coverage by directional sensors with tunable orientations under the random deployment strategy. We develop solutions that maximize the number of targets to be covered while minimizing the number of sensors to be activated at any instant. We also present solutions that include sleep scheduling for nodes. Compared to deterministic deployment, random deployment is easy and less expensive for large wireless sensor networks, and may be the only feasible option in remote or inhospitable environments. Moreover, to compensate for the lack of exact positioning and improve the fault tolerance, nodes are typically deployed in excess, and thus redundant sensors usually arise. Furthermore, sensors are usually powered by batteries and it may not be possible to recharge or replace the batteries after deployment. In addition, target locations may change even after initial deployment, thus changing the optimal solution to the coverage problem.

The rest of the paper is organized as follows. In Section 2, we review related research. In Section 3, we propose the Maximum Coverage with Minimum Sensors (MCMS) problem. Section 4 presents its centralized exact and approximate solutions; the Integer Linear Programming (ILP) formulation and centralized greedy algorithm (CGA), respectively. Section 5 presents the distributed greedy algorithm (DGA) solution. In Section 6, we develop the Sensing Neighborhood Cooperative Sleeping (SNCS) protocol by incorporating the sensors’ residual energy into DGA to perform adaptive scheduling on a larger time scale. Detailed results of performance evaluations are presented in Section 7. Finally, we conclude the paper in Section 8.

## 2. Related work

There are two main threads of research, though dealing with the circular coverage by isotropic sensors, related to our research.

The first thread has in common the idea of turning off the redundant nodes online according to some off-duty eligibility rule derived based on geometric properties. Tian and Georganas (2002) define an off-duty eligibility rule from a concept called “sponsored area” to guarantee

the complete coverage. Wang et al. (2003) derived an off-duty eligibility rule based on the analysis of intersection points by sensing disks. Zhang and Hou (2005) proposed the OGDC algorithm where the off-duty eligibility rule for minimizing sensing overlap. Though the above schemes can be naturally implemented in a distributed way, most of them (except OGDC) did not present the optimal performance that can be achieved.

The second thread viewed the sensing coverage as a discrete problem in which nodes were usually organized in a power-aware fashion offline. Megrian and Potkonjak (2003) presented several ILP formulations and strategies to reduce the overall energy consumption while maintaining guaranteed 0/1 coverage. Slijepcevic and Potkonjak (2001) proposed the SET  $K$ -COVER problem to maximize the number  $K$  of disjoint set covers which can be activated successively along time, where a set cover is defined as a set of nodes that can completely cover the monitored area. Cardei and Du (2005) proposed the Maximum Disjoint Set Cover problem which shared the same notion as Slijepcevic and Potkonjak (2001) but in a different scenario where a set of targets with known locations need to be covered. By further relaxing the constraints of disjoint set covers, i.e., that one node can be in multiple set covers, Cardei et al. (2005) improved the network lifetime. Along a similar line, but with additional bandwidth constraint, Cheng et al. (2005) formulated the Minimum Breach problem in which the sizes of Set Covers are bounded. Compared with previous results without bandwidth considerations, they revealed that network lifetime can be enhanced further at the cost of coverage breach. Although it is valuable to derive the optimal scheduling by mathematical programming techniques and related approximate or heuristic algorithms, the centralized solutions are impractical to be realized in wireless sensor networks since they do not scale.

*Our work differs from prior work in several ways.* First, the desired configuration of directional sensors and orientations at any instant is formulated as the MCMS problem, which can be solved exactly by an ILP formulation in a small scale and approximately by CGA in a large scale. Second, we also provide the distributed solution (DGA) for the MCMS problem; moreover, the SNCS protocol can maintain sensing coverage and prolong the network lifetime simultaneously even on a large time scale with varying network conditions. Finally, our proposed framework to deal with coverage by directional sensors with tunable orientations can treat the coverage by isotropic sensors as a special case (we show comparisons against OGDC for this case in Section 7).

### 3. The MCMS problem statement

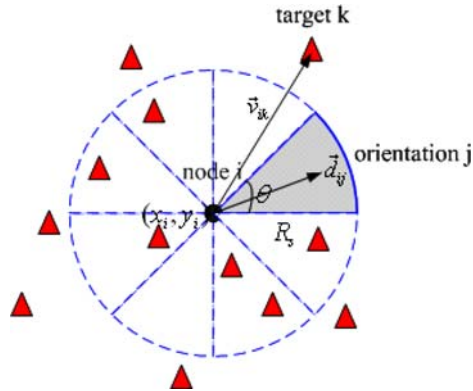
This section defined the sensing model of a directional sensor, formulates the MCMS problem and characterizes its complexity.

#### 3.1. The sensing model of a directional sensor

Unlike an isotropic sensor, a directional sensor has a finite angle of view and thus can not sense the whole circular area. Hence, by a simple geometric abstraction, its sensing region can be viewed as a sector in a two-dimensional plane as shown in fig. 2. Please note that, throughout the rest of the paper, unless otherwise mentioned, “sensor” or “node” refers to a directional sensor, as defined in the following section.

The following parameters completely characterize the sensing sector of a sensor node  $i$  (please refer to fig. 2).  $(x_i, y_i)$ : the Cartesian coordinates that denote the location of the sensor in a two-dimensional plane.  $\theta$ : the field of view (FOV), describing the maximum angle of sensing achieved by directional sensor.  $R_s$ : maximum sensing range of the sensor, beyond which a target will not be detected in a binary detection sense.  $\vec{d}_{ij}$ : a unit vector which cuts

**Fig. 2** A finite set of orientations for a directional sensor (black ball) covering targets (red triangles). The shadowed area is the current sensing sector of the directional sensor. A directional sensor can only choose one active sector at any time instant



the sensing sector into half. This parameter defines the orientation of the directional sensor (i.e., the direction it is looking).

### 3.2. Target in sector (TIS) test

To make the problem tractable, we assume that a directional sensor can only take a finite set of orientations. For instance, in the example shown in fig. 2, a directional sensor with  $\frac{\pi}{4}$  of FOV can pick eight orientations with mutually disjoint sensing sectors which can be combined to generate the full circular view of an isotropic sensor.

With each choice of orientation, a certain subset of targets is covered by the directional sensor. The relationship of a directional sensor, its orientation and a target can be determined by a Target in Sector (TIS) test.

The TIS test can be described as follows. Consider a target  $k$  located at  $\vec{t}_k$  and a directional sensor  $i$  located at  $\vec{l}_i$ . In order to determine whether the target  $k$  can be sensed by the directional sensor  $i$  with the  $j$ -th orientation, we follow the following steps:

1. calculate the distance vector  $\vec{v}_{ik}$  pointing from the directional sensor  $i$  to the target  $k$

$$\vec{v}_{ik} = \vec{t}_k - \vec{l}_i \tag{1}$$

2. check whether the resulting distance vector is within the FOV of the directional sensor  $i$  by performing the inner product operation

$$\vec{d}_{ij}^T \cdot \vec{v}_{ik} \geq \| \vec{v}_{ik} \|_2 \cos \left( \frac{\theta}{2} \right) \tag{2}$$

with equality when the target  $k$  is along the two edges of node  $i$  sensing sector.

3. verify whether target  $k$  is within the sensing range of the directional sensor  $i$  or not by checking

$$\| \vec{v}_{ik} \|_2 \leq R_s \tag{3}$$

with equality when the target  $k$  is on the arc of the sensing sector of the directional sensor  $i$ .

4. If both (2) and (3) hold, the result of the TIS test is true (i.e., node  $i$  covers the target  $k$  if it sets its orientation to  $j$ ); otherwise, it is false.

Let  $\Phi_{ij}$  denote the set of targets that are covered by sensor  $i$  when its orientation is  $j$ . Then we can determine all the sets  $\Phi_{ij} \forall i, j$ , by running the TIS test for every  $i, j$ .

### 3.3. Maximum coverage with minimum sensors (MCMS) problem

Under the random deployment strategy, not all targets are covered by the initial deployment. Further, all sensors are active. Our goal is to change the initial orientations in order to cover as many targets as possible while activating as few sensors as possible, at any time instant. We call this the MCMS problem. The MCMS problem can be stated as follows:

**Given:** A set of targets  $\mathcal{S} = \{s_1, s_2, \dots, s_m\}$  to be covered; a set of  $n$  homogenous directional sensors, each of which has  $p$  possible orientations, randomly deployed in a two-dimensional plane. Hence, a collection of subsets  $\mathcal{F} = \{\Phi_{ij}, 1 \leq i \leq n, 1 \leq j \leq p\}$  can be generated based on the TIS test, where each element  $\Phi_{ij}$  is a subset of  $\mathcal{S}$ .

**Problem:** Find a subcollection  $\mathcal{Z}$  of  $\mathcal{F}$ , with the constraint that at most one  $\Phi_{ij}$  can be chosen for the same  $i$ , to maximize the cardinality of the union of chosen  $\bigcup_{(i,j)} \Phi_{ij}$  (i.e., the number of covered targets) while minimizing the cardinality of  $\mathcal{Z} = \{\Phi_{ij}, (i, j) \text{ is chosen}\}$  (i.e., the number of activated directional sensors).

The following theorem shows the complexity of the MCMS problem.

**Theorem 3.1.** *The MCMS problem is  $\mathcal{NP}$ -complete.*

We present two proofs for this theorem.

**Proof 1:** We prove the  $\mathcal{NP}$ -completeness of the MCMS problem by showing its special case where sensors are isotropic (i.e.,  $p = 1$ ) is equivalent to another well-known  $\mathcal{NP}$ -complete problem. For any given number as the upper bound for the number of sensors to be activated, the problem of deriving the maximum number of targets to be covered can be treated as the classic Maximum Coverage Problem Hochbaum (1997) which is known to be  $\mathcal{NP}$ -complete. Hence, the result follows. □

**Proof 2:** First, we show that  $\text{MCMS} \in \mathcal{NP}$ . The decision version of the MCMS problem can be stated as follows.

**INSTANCE:** Given  $\mathcal{S}$  and  $\mathcal{F}$  as shown above.

**QUESTION:** Is there a subcollection of  $\mathcal{F}$  with  $u$  subsets covering at least  $v$  elements in  $\mathcal{S}$ ?

It is easy to see that any guessed solution can be verified in a polynomial time.

Second, we show that the MCMS problem is  $\mathcal{NP}$ -hard by proofing that  $\text{MAX\_COVER}_{\leq p}$  MCMS, where  $\text{MAX\_COVER}$  Hochbaum (1997) is a classic  $\mathcal{NP}$ -complete problem. The decision version of the  $\text{MAX\_COVER}$  problem can be stated as follows.

**INSTANCE:** Given  $\mathcal{S}$  and a collection  $\mathcal{G}$  of subsets.

**QUESTION:** Is there a subcollection of  $\mathcal{G}$  with  $u$  subsets covering at least  $v$  elements in  $\mathcal{S}$ ?

For the  $\text{MAX\_COVER}$  problem, we first pick any  $u$  subsets,  $\Phi_1, \Phi_2, \dots, \Phi_u$ , from  $\mathcal{G}$ . Then, for each subset  $\Phi_i (1 \leq i \leq u)$ , we construct  $p$  copies of itself and rewrite them as  $\Phi_{i1}, \Phi_{i2}, \dots, \Phi_{ip}$ , similarly to that in the MCMS problem. Finally, we can use such an

“expanded” subcollection as the input into the MCMS problem. Obviously, the above reduction procedure is in a polynomial time and these two problems will obtain the same binary answers. This proves that the MCMS problem is  $\mathcal{NP}$ -complete.  $\square$

#### 4. Centralized solutions of the MCMS problem

In this section, we present an exact problem formulation of the MCMS problem as an Integer Linear Program (ILP). We also present an approximate but computationally efficient greedy solution of the problem.

##### 4.1. ILP formulation

The parameters used for the formulation can be summarized as follows.  $n$ : the number of directional sensors;  $m$  the number of targets,  $p$  the number of orientations available for each directional sensor.

The variables in the ILP are as follows.  $\psi_k$  a binary variable that has value one if target  $k$  is covered by any directional sensor, and zero otherwise;  $\chi_{ij}$  a binary variable that has value one if the directional sensor  $i$  uses the orientation  $j$ , and zero otherwise;  $\xi_k$  an integer variable that counts the number of directional sensors covering target  $k$ .

Under the random deployment, for each directional sensor  $i$ , there is an incidence matrix  $A^i_{(m \times p)}$  where each of its elements  $a^i_{kj}$  can be derived based on the TIS test:

$$a^i_{kj} = \begin{cases} 1 & k \in \Phi_{ij} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Therefore,  $\xi_k$  can be expressed as:

$$\xi_k = \sum_{i=1}^n \sum_{j=1}^p a^i_{kj} \chi_{ij} \tag{5}$$

Now the ILP for the MCMS problem can be stated as:

$$\max \sum_{k=1}^m \psi_k - \rho \left( \sum_{i=1}^n \sum_{j=1}^p \chi_{ij} \right) \tag{6}$$

subject to:

$$\frac{\xi_k}{n} \leq \psi_k \leq \xi_k \quad \forall k = 1 \dots m \tag{7}$$

$$\sum_{j=1}^p \chi_{ij} \leq 1 \quad \forall i = 1 \dots n \tag{8}$$

$$\psi_k = 0 \quad \text{or} \quad 1 \quad \forall k = 1 \dots m \tag{9}$$

$$\chi_{ij} = 0 \quad \text{or} \quad 1 \quad \forall i = 1 \dots n, \forall j = 1 \dots p \tag{10}$$

The objective function in (6) maximizes the number of targets to be covered and imposes a penalty by multiplying the number of sensors to be activated by a positive penalty coefficient  $\rho$  whose value must be small enough ( $\rho < 1$ ) to guarantee uniqueness of the solution. There

are  $(m + n \cdot p)$  variables and  $(2m + n + n \cdot p)$  constraints for the ILP. Eq. (7) represents a set of inequalities to indicate that whether any target  $k$  is covered or not: if none of the directional sensor covers target  $k$ , i.e.,  $\xi_k = 0$ , then  $\psi_k = 0$  to conform with the right inequality; if target  $k$  is covered by any direction sensor, i.e.,  $\xi_k > 0$ , since  $\xi_k$  is bounded by  $n$ ,  $\frac{\xi_k}{n}$  is a real number less than one, then  $\psi_k = 1$  to conform with the left inequality. Eq. (9) guarantees that one directional sensor has at most one orientation depending on whether it is activated or not.

#### 4.2. Centralized greedy algorithm (CGA)

Though the solution of the ILP formulation provides the optimal solution for the MCMS problem, it is not scalable for large problem instances. In this paper, rather than giving an LP-relaxation algorithm to this problem (similar to the approach in Cardei et al. (2005); Cheng et al. (2005)), we present here a greedy heuristic based polynomial-time algorithm for solving the MCMS problem approximately. We believe this is more useful since an LP-relaxation algorithm is not quite helpful for the design of a practical distributed algorithm.

The basic idea is based on the greedy principle and can be described as follows: we first construct  $\mathcal{F}$ , a collection of sets  $\{\Phi_{ij}, 1 \leq i \leq n, 1 \leq j \leq p\}$ , based on targets, directional sensors and all their possible orientations by the TIS tests as an instance of the problem. CGA runs in loops, where initially, all nodes are inactive (i.e., not selected). In each loop, for each sensor  $i$  that has not yet been activated, the number of additional targets that would change from uncovered to covered for each possible orientation is calculated. Then, the inactive sensor and its orientation that maximizes the number of newly covered targets is activated. Any ties are broken by selecting one of the choices at random. If there are no more targets to be covered or no more unselected directional sensors remaining, the algorithm terminates; otherwise, directional sensors are activated iteratively according to the above greedy rule. The pseudo-code of CGA is shown in Algorithm 1.

---

**Algorithm 1** CGA (Centralized Greedy Algorithm for the MCMS problem)

---

```

1:  $\mathcal{Z} \leftarrow \emptyset$  {a collection of (active node, orientation) pairs
   which, at the end of the algorithm, is the result of CGA}
2:  $\mathcal{V} \leftarrow \mathcal{S}$  {a set that, at the end of the algorithm, contains all
   uncovered targets}
3:  $\mathcal{Y} \leftarrow \{\text{set of inactive nodes}\}$ 
4: while 1 do
5:    $(i, j) \leftarrow \arg \max_{I \in \mathcal{Y}, 1 \leq J \leq p} \Phi_{IJ} \cap \mathcal{V}$  {i.e., maximizes
     the number of targets that become covered after this
     iteration}
6:   if  $|\Phi_{ij} \cap \mathcal{V}| == 0$  then
7:     break; {i.e., if none of the inactive nodes can cover
       additional targets if they are activated, exit the while
       loop}
8:   else
9:      $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{(i, j)\}$ 
10:     $\mathcal{V} \leftarrow \mathcal{V} \setminus \Phi_{ij}$ 
11:     $\mathcal{Y} \leftarrow \mathcal{Y} \setminus \{i\}$ 
12:   end if
13: end while
14: return  $\mathcal{Z}$ 

```

---

CGA, in the worst case, for each loop, to perform the TIS test for all nodes requires  $mnp$  steps, and then to choose a desired  $(i, j)$ , the running time is bounded by  $O(np)$ . Since there are at most  $n$  loops, the time complexity of CGA is  $O((m + 1)n^2 p)$  in the worst case.

## 5. Distributed solution for the MCMS problem

Without global information available in a centralized location, each directional sensor must make its decision independently based only on local information gathered from the “neighboring” nodes, as defined by the wireless communication range. Though intuitively the solution can not be expected to achieve as good a performance (in terms of coverage) as the centralized schemes, it is more computationally scalable and does not incur high communication overhead as required by a centralized solution.

### 5.1. Assumptions and definitions

To ensure individual sensors obtain perfect local information for their decision making, we make the following assumptions in our distributed scheme. Actually some of these assumptions can be relaxed, and we will discuss the extended results in Section 7.

- A1** Directional sensors are homogeneous. Specifically, all sensors have the same omnidirectional communication ranges  $R_c$ , shape of sensing sectors (i.e.,  $\theta$  and  $R_s$ ) and the available possible orientations are the same.
- A2** Every directional sensor knows its exact location information (e.g., He et al. (2003) provides a cost-effective localization service without equipping each sensor node with a GPS device) and every directional sensor knows and controls its orientation  $\vec{d}$ .
- A3** Every directional sensor is aware of the targets located within its sensing range  $R_s$ .
- A4** The initial deployed network is connected.
- A5** The sensors can not be located at the same coordinates in the two-dimensional plane.
- A6** The sensors utilize an ideal MAC for scheduling (no collisions) and the communication channel is error free.

We also give the following definitions:

*Definition 5.1.* The set of **sensing neighbors** of a directional sensor  $i$ , denoted by  $\mathcal{N}_i$ , contains all the sensors for which any of their sensing sectors may intersect with node  $i$ 's sensing sector. Notice that all sensors located within  $2R_s$  distance of the sensor  $i$  are neighboring sensors.

*Definition 5.2.* Given a wireless sensor network consisting of a set of sensors, the **communication graph** for the sensor network is the undirected graph  $G$  with sensors as the vertices and an edge between any two sensors if they are located within  $R_c$  distance of each other.

*Definition 5.3.* The set of  **$k$ -hop communication neighbors** of a sensor  $i$ , denoted by  $\mathcal{C}_i^k$ , is all sensors within  $k$  hops of  $i$  in the **communication graph**  $G$ .



**Definition 5.4.** The set of *neighbor targets* of a sensor  $i$ , denoted by  $\mathcal{T}_i$ , contains all targets that may be covered by any of node  $i$ 's possible orientations. Thus, all targets located within  $R_s$  distance of  $i$  are neighbor targets.

It is worth noting that the above definitions are valid only when the assumption **A1** holds.

## 5.2. Distributed greedy algorithm (DGA)

In order for a node to make a globally optimal decision about its state (active or inactive) and its orientation, it needs to gather information about the current state and orientation of possibly all other nodes in the network. However, for the purpose of designing a low overhead distributed algorithm, each node should restrict the range of data gathering to a limited area localized around its own position.

Observe that for any two sensors, it is possible for these sensors to cover the same targets only if they are within a certain distance from each other, defined by the coverage range. Decisions made by these sensors are highly dependent. Therefore, in our distributed algorithm, under assumption **A2**, each sensor need to disseminate its information only within a  $2R_s$  radius, via, for example, a geocasting service.

In the DGA algorithm, every directional sensor has a unique variable, which we call “priority.” The priority needs to be unique only among its sensing neighbors. Each sensor is in one of two states; *active* or *inactive* state.

To simplify the description of the algorithm, we say a neighbor target is “acquired” by a sensor if the target was not covered by a higher priority neighboring sensor.

Initially, each directional sensor is in the *active* state, assigns itself a priority, and has a random initial orientation. Each directional sensor starts to collect its sensing neighbors information, i.e., priorities, locations and orientations of its sensing neighbors. Upon receiving this information, each sensor makes its decision independently as follows. It calculates, for each of its possible orientations, the number of acquired targets. There are two cases: (a) If the maximum number of acquired targets is positive (i.e., not zero), it will choose the orientation corresponding to the maximum number (random choice in case of a tie). If a directional sensor  $i$  has to switch to a new orientation, it sends out a new message to inform its sensing neighbors. (b) If the maximum number of acquired targets is zero, the sensor activates a transition timer, with duration  $T_w$ . The timer is canceled if new information from the sensing neighbors arrives and changes the maximum number of acquired targets to a non-zero value.

Note that the purposes of setting the transition timer  $T_w$  are 1) to prevent a sensor finalizing its decision before its sensing neighbors with higher priorities and 2) to transfer its state to *inactive* in time. As analyzed in Theorem 5.5, on the order of the time required to collect a sensor's sensing neighborhood information, at least one sensor finalizes its decision. Therefore, conservatively,  $T_w$  can be set to be a constant equal to the maximum value of that time. Currently, we estimate such a value by simulations and it is usually very small in practice.

Since the DGA is guaranteed to terminate (as shown in Theorem 5.5), the final set of sensors marked as *active* with their chosen orientations forms the solution of the MCMS problem. The pseudo-code of DGA, which is executed by each sensor  $i$ , is shown in Algorithm 2.

---

**Algorithm 2** DGA (Distributed Greedy Algorithm for the MCMS problem)

---

**I. Initialization Phase** (only performed once)

- 1:  $\text{orientation}(i) = j_0 \{1 \leq j \leq p\}$  {orientation of sensor  $i$  is randomly initialized}
- 2:  $\text{location}(i) = (x_i, y_i)$
- 3:  $\mathcal{N}_i \leftarrow \emptyset$  {the set of sensing neighbors of sensor  $i$ }
- 4:  $\text{priority}(i) = h(i)$  { $h(\cdot)$  is a hash function}
- 5:  $\text{state}(i)$  *active*  
 $\{a_{i,j}\} = \{0, \dots, 0\}$  {a  $p$ -array recording the number of acquired targets in each direction of sensor  $i$ }
- 6: send a coverage message containing orientation, location and priority information of sensor  $i$

**II. Decision Phase**

- 1: **while** coverage messages arrive from sensing neighbors **do**
  - 2: record information about corresponding orientations, locations and priorities if necessary and then update  $\mathcal{N}_i$  and  $\{a_{i,j}\}$  accordingly
  - 3:  $j = \arg \max_{1 \leq j \leq p} a_{i,j}$  {i.e., maximizes the number of acquired targets}
  - 4: **if**  $a_{i,j} > 0$  **then**
  - 5: send an updated message and adjust its orientation to  $j$  if necessary (i.e., the resultant orientation has to be different from the previous one)
  - 6: **if** transition timer of sensor  $i$  is *on* **then**
  - 7: turn the transition timer *off*
  - 8: **end if**
  - 9: **else**
  - 10: **if** transition timer of sensor  $i$  is *off* **then**
  - 11: turn the transition timer *on*
  - 12: **end if**
  - 13: **else**
  - 14: **if** transition timer of sensor  $i$  remains *on* for longer than  $T_w$  **then**
  - 15: the state of sensor  $i$  transitions from *active* to *inactive*
  - 16: Exit the while loop
  - 17: **end if**
  - 18: **end if**
  - 19: **end while**
- 

### 5.3. Algorithm properties

In this section, we analyze DGA properties in terms of its correctness and message complexity. In DGA, since sensors make their local decisions based on gathered sensing neighborhood information, two concerns may arise: (a) whether the algorithm terminates within a finite

time and (b) whether, when the algorithm terminates, there exists any target which is left uncovered because of a “misunderstanding,” where one sensor assumes other sensor has covered the target, while it actually has not. We call such a target a “hidden” target. Notice that since sensors tend to choose their directions to maximize the number of targets to be covered, it is possible that some targets may not be covered by any sensor. However, this type of targets uncovered is different from the “hidden” targets defined here. The following two theorems answer the above two questions.

**Theorem 5.5.** *DGA terminates in finite time.*

**Proof:** Let us first consider the node with highest priority and label it as node A. When A makes a decision on its orientation and sends a message to sensing neighborhood to update its status, we say that “A has reached a final decision.” Since A has the highest priority among its sensing neighbors, the decisions of its sensing neighbors can not affect its decision according to DGA. After node A reaching its final decision, the node with the second highest priority, labeled as node B, becomes the highest priority node among those that have not updated their decisions in the sensing neighborhood. Similarly, node B will also reach a final decision at most after one exchange of message among its sensing neighbors. By induction, the algorithm terminates in finite time.

Notice that the above convergence procedure also reveals the time complexity in the worst case where sensors reach final decisions in the order of their priorities along the time. Since notifying a decision of one node to its sensing neighbors takes at most  $O(n)$  time, the time complexity for all  $n$  nodes reaching their final decisions is  $O(n^2)$ . □

**Theorem 5.6.** *DGA guarantees no “hidden” targets.*

**Proof:** By gathering sensing neighborhood information, node  $i_v$  builds up an ordered list of three-tuple [node id, orientation, priority], e.g.,  $[i_1, j_{i_1}, p_{i_1}]$ ,  $[i_2, j_{i_2}, p_{i_2}]$ ,  $\dots$ ,  $[i_u, j_{i_u}, p_{i_u}]$ ,  $[i_v, j_{i_v}, p_{i_v}]$ ,  $\dots$ , where  $p_{i_1} > p_{i_2} > \dots > p_{i_u} > p_{i_v} > \dots$ . Also the coverage status of the node  $i_v$  is stored as a three-tuple entry in the ordered list of its sensing neighbors.

Take one of the sensing neighbors of node  $i_v$ , say node  $i_s$ , as an example. According to the description of DGA, node  $i_v$  marks itself as *inactive* only if it observes the number of acquired targets is zero and this situation lasts for duration  $T_w$ , which ensures that the decisions of sensing neighbors with higher priorities are finalized and that it has received the update messages from the higher priority neighbors. When node  $i_v$  is eligible to mark itself *inactive*, there could be only one of two cases:

- Case I*  $p_{i_v} < p_{i_s}$ : since node  $i_v$  has lower priority than that of node  $i_s$ , node  $i_s$  does not care about the decision of node  $i_v$  when it calculates its own decision. Hence, node  $i_v$ 's decision to turn its state to *inactive* can not cause hidden targets.
- Case II*  $p_{i_v} > p_{i_s}$ : when node  $i_s$  calculates its local decision, it must consider the orientation of node  $i_v$ . If node  $i_v$  marks itself *inactive*, this means that all its neighbor targets are already covered by higher priority sensing neighbors. Hence, for node  $i_s$ , it can still consider that these neighbor targets are covered with higher priority neighbors, and it does not need to explicitly know which higher priority nodes are covering these targets. Hence, there can be no confusion about making coverage decisions.

Extending the same analysis to all the neighbors of node  $i_v$ , the result follows. □

## 6. Sensing neighborhood cooperative sleeping protocol

Assuming static priorities of sensors, DGA runs once and terminates, providing a solution to the MCMS problem. Since the objective of the MCMS problem is minimizing the number of active sensors, DGA provides an energy-efficient configuration in the network. However, without dynamic energy balancing consideration among sensors, those *active* sensors set by DGA will ultimately deplete their batteries. Therefore, in this section, we will extend DGA so as to perform dynamic scheduling among sensors depending on the amount of residual energy. The new protocol is called the Sensing Neighborhood Cooperative Sleeping (SNCS) protocol.

The SNCS protocol works as follows. Each node continuously alternates between two phases; scheduling and sensing. In each scheduling phase, all sensors set their states to be *active* at the beginning and then perform DGA described above. At the end of the scheduling phase, as a result of running DGA, each sensor will be in one of two states; active or inactive. The *active* sensors will continue to be active in the followed sensing phase with its sensing and communication units turned on; whereas the *inactive* sensors will go to sleep immediately with its sensing and communication units turned off. In addition, these *inactive* sensors will reset themselves to be *active* state until the next scheduling phase.

In the scheduling phase of SNCS protocol, we assign the residual energy of a sensor as its priority in the DGA. Notice that the residual energy of sensors depends on their behaviors (i.e., transmit, receive, idle or sleep) and dynamically changes along the time, to maintain an unchanged order of priorities among sensors during one-time DGA execution to guarantee its termination, priorities are set to be the instantaneous value of residual energy of sensors only once in every scheduling phase. Moreover, to satisfy the uniqueness property of priorities in DGA, the residual energy of sensors is expected to be different. In case the equalities appear, though rare, by assumption **A5**, we can further compare two geographic coordinates of sensors accordingly until the tie is broken up.

Assigning the value of residual energy to the priority variables in DGA is essential for the SNCS protocol to achieve a trade-off between coverage and network lifetime. In each scheduling phase, residual energy of sensors acts as priorities in DGA to solve the MCMS problem. The sensors which have higher priorities (i.e., residual energy) are more likely to be selected to be *active* by the DGA to contribute to coverage, while the sensors with lower residual energy are more likely to go to sleep so as to conserve their energy. Since *active* sensors have larger energy dissipation rates than that of *inactive* sensors in the following sensing phase(s), those *active* sensors will, after a certain time, have less residual energy than that of *inactive* sensors. As a result, these *inactive* sensors may be turned to *active* by DGA when the residual energy of the neighboring active sensors depletes to a level that is lower than that of the inactive sensors. Thus, by using the residual energy of nodes as the priorities in the SNCS protocol, the SNCS protocol dynamically changes sensor's states (between *active* and *inactive*) so as to achieve energy balancing across the network while providing a solution to the MCMS problem.

The round duration  $\mathcal{R}$ , as an important protocol parameter, needs to be addressed: it must be chosen to be large enough compared to the time that DGA converges, so that the control overhead does not overwhelm the energy consumption; also it must be chosen to be short

enough compared to the average sensor lifetime such that a sensor can remain active for at least the duration of a round time.

## 7. Performance evaluation

First, the solutions of the MCMS problem by ILP, CGA and DGA are evaluated. Second we concentrate on the SNCS protocol, where we investigate the trade-off between coverage and network lifetime. Third, we examine the robustness of the SNCS protocol in scenarios that do *not* follow our simplified assumptions made in Section 5. Finally, we compare the performance of DGA with OGDC for the special case of circular coverage.

### 7.1. Effect of the various parameters on the solutions of the MCMS problem

The solutions for the MCMS problem in terms of the number of targets to be covered and the number of sensors to be activated are determined by  $(m, n, p, R_s)$ . In these experiments, we use  $m$  targets and  $n$  sensors uniformly distributed in a unit square area.

#### 7.1.1. Effect of the size of sensing sector on the ILP solution

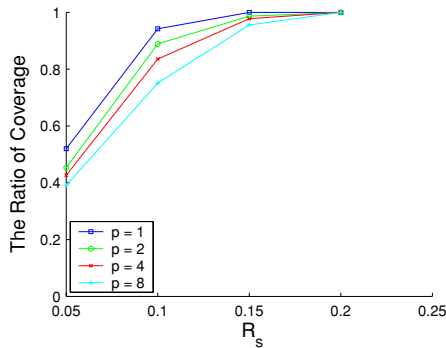
To identify the influence of the sensing sector of a sensor on the solutions of the MCMS problem, we fix  $m$  and  $n$  and obtain the results from ILP<sup>1</sup> solved by CPLEX 11 under different  $p$  and  $R_s$ , shown in fig. 3. Notice that smaller  $R_s$  means smaller sensing range and higher  $p$  means smaller angle of coverage. As expected, the coverage ratio (number of covered targets divided by the total number of targets) increases with the decrease of  $p$  and increases with the increase of  $R_s$ . Accordingly the number of sensors needed to solve the MCMS, i.e., the sensors labeled as *active* in the ILP solution, decreases with the decrease of  $p$  and/or the increase of  $R_s$ . Notice that this is true in most (but not all) cases, due to the integer nature of the solution.

#### 7.1.2. Effect of the number of sensors on ILP, CGA and DGA

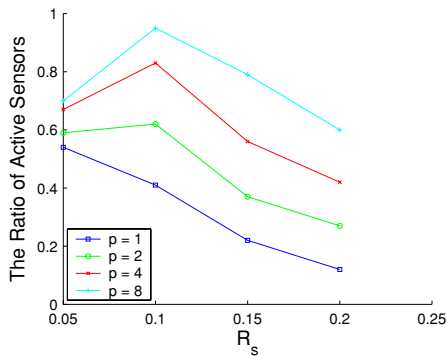
Next, we fix  $p = 8$  and  $R_s = 0.1$  and compare solutions for the MCMS problem of ILP, CGA and DGA by changing  $n$ . Figure 4 shows the scenario where  $m = 225$  and  $n$  ranges from 10 to 400. With the increase of sensors deployed, both the coverage ratio and active nodes for all three schemes increase linearly until  $n$  approaches 150; upon passing such a value, the number of active nodes increases slowly or even decreases whereas the coverage ratios continuously increase and then become saturated when  $n$  is above 350 or so. To state the differences: for the coverage ratio, ILP always behaves the best among the other two schemes and DGA tracks closely with that of CGA with all  $n$  values; for the number of sensors activated which is a bit complicated after  $n$  is more than 150, DGA activates the largest number of sensors in most of cases; while the curves depicting the number of active sensors by ILP and CGA may cross at some points but ILP stabilizes at a lower value than that of CGA.

<sup>1</sup> We set  $\rho = 0.001$  in the (6).

**Fig. 3** 225 targets and 100 sensors uniformly distributed in a unit square area:  $p$  picks 1, 2, 4, 8 and  $R_s$  picks 0.05, 0.01, 0.15, 0.20, respectively



(a) The ratio of coverage.



(b) The ratio of active sensors.

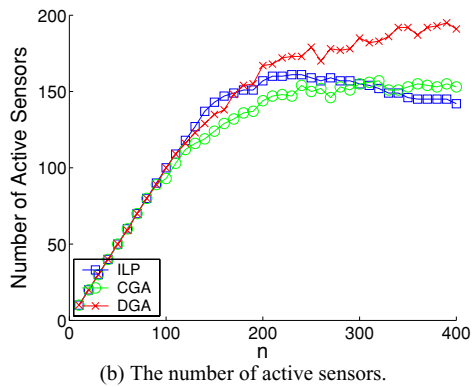
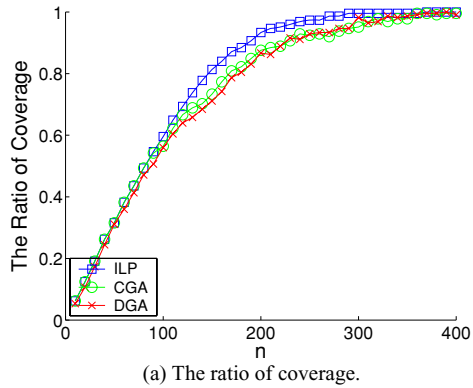
### 7.2. Performance of the SNCS protocol

In this experiment, we evaluate the performance of the SNCS protocol in terms of coverage and network lifetime. In the simulations, time is discretized into slots, where each sensor can transmit or receive a packet in a slot.

To characterize the network lifetime, we assume a simplified energy consumption model per slot of a sensor: transmission, reception, idle and sleep during a time slot consume 0.025, 0.022, 0.021 and 0.0002 units of energy, respectively. These energy consumption values are chosen to match the results in Feeney and Nilsson (2001), where they have noted that the receive and idle modes may require as much energy as transmitting while the sleep mode requires the least energy. Moreover, we assume that each sensor starts with equal initial energy of 500 units.

Other parameters for this experiment are:  $n = 200$ ,  $p = 8$ ,  $R_s = 0.1$ ,  $R_c = 0.2$ ,  $\mathcal{R} = 500$  slots and  $m$  ranges from 100 to 400 with an increment of 100. We examine the network lifetime, defined as the time until half of the sensors deplete their energy. We consider different densities of targets. figure 6 shows all aspects of the performance of the SNCS protocol.

**Fig. 4** Coverage performance comparison of ILP, CGA and DGA where  $p = 8$ ,  $R_s = 0.10$ ,  $m = 225$  and  $n$  ranges from 10 to 400



In fig. 5(a), for any  $m$ , initially the coverage ratio increases rapidly<sup>2</sup> and stabilizes at a higher level for a relative long period, then it drops sharply toward some lower level. Moreover, with the increase of  $m$ , both network lifetime and coverage ratio decrease.

In fig. 5(b), for any  $m$ , a similar “cutoff” property can also be observed in the curve for the active ratio of sensors. However, unlike the trend for the coverage ratio, the active ratio of sensors increases with the increase of  $m$ .

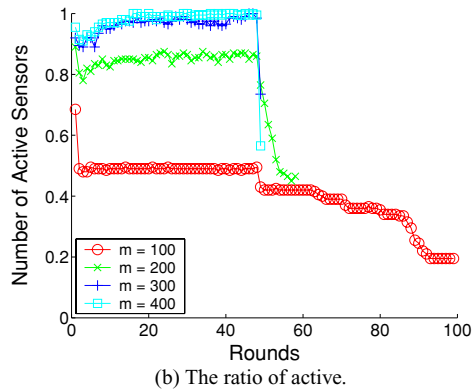
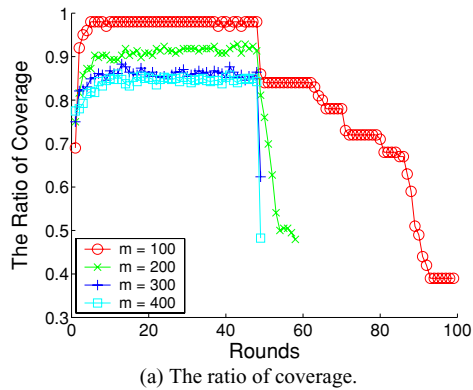
Notice that when the density of targets is above some threshold (say roughly 300 in our case), all aspects of performance of the SNCS protocol tend to be saturated due to the active ratio of sensors approaching 100%.

7.3. Robustness of the SNCS protocol

In these experiments, we examine the performance of the SNCS protocol by relaxing some of assumptions made before. Specifically, we introduce the following: (a) the sensors no longer know their exact location information (b) orientation errors are also introduced and (c) wireless communications may result in corrupted messages. To study the impacts of the above three factors, we evaluate the performance of the SNCS protocol under each of them independently and then compare the results with an ideal scenario (i.e., a scenario that follows

<sup>2</sup> Since we start the SNCS protocol with equal initial energy of sensors, the geographic locations of sensors are used to break up the tie in DGA initially. It also demonstrates that priorities are better to be random numbers rather than some values relating to the deployment parameters.

**Fig. 5** Performance of the SNCS protocol where  $n = 200$ ,  $p = 8$ ,  $R_s = 0.1$ ,  $R_c = 0.2$ ,  $\mathcal{R} = 500$  slots and  $m$  ranges from 100 to 400 with an increment of 100



all the assumptions mentioned in Section 5). The parameters used for these experiments are  $m = 225$ ,  $n = 200$ ,  $p = 8$ ,  $R_s = 0.1$ ,  $R_c = 0.2$  and  $\mathcal{R} = 1000$  slots.

7.3.1. Effect of localization errors

We introduce location errors as follows. For each sensor, its actual location is randomly selected within a circle with the error bound  $E_l$  as the radius centered around its perceived location. We experiment with high values of  $E_l$ , even comparable to  $R_s$ .

Figure 6 shows the performance comparison of the SNCS protocol where the error bound  $E_l$  is 10, 5 and 1% of  $R_s$ . The coverage ratios continuously decrease down to 12% whereas the network lifetime remains the same when error bounds increase from 0.001 to 0.01. On the other hand, the ratios of active sensors are insensitive to  $E_l$  (which is a good feature).

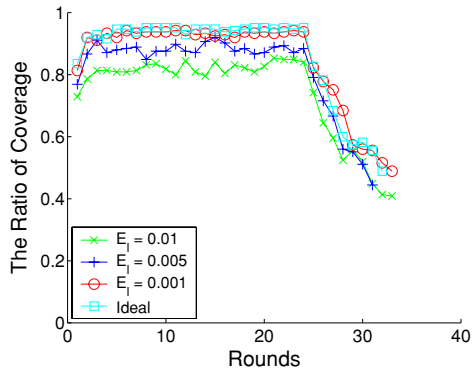
7.3.2. Effect of orientation errors

We introduce orientation errors as follows. For each sensor, we add a deviation to the orientation. The deviation is randomly selected within the error bound  $E_o$ , which is comparable to its field of view  $\theta$ .

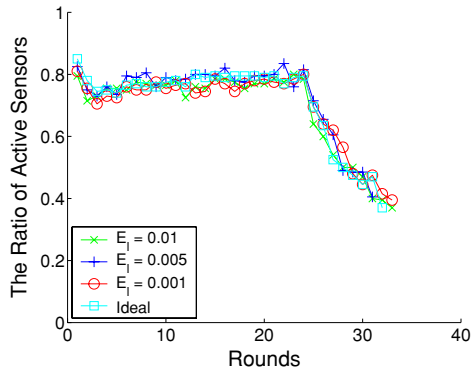
Figure 7 shows the performance comparison of the SNCS protocol with and without orientation errors, where the error bound  $E_o$  is 20, 10 and 5% of  $\theta$ . The coverage ratio only slightly decreases, but still remains within 6% of the ideal case. The network lifetime remains



**Fig. 6** Performance comparison of the SNCS protocol where  $m = 225, n = 200, p = 8, R_s = 0.10, R_c = 0.20, \mathcal{R} = 1000$  slots with localization errors



(a) The ratio of coverage.



(b) The ratio of active sensors.

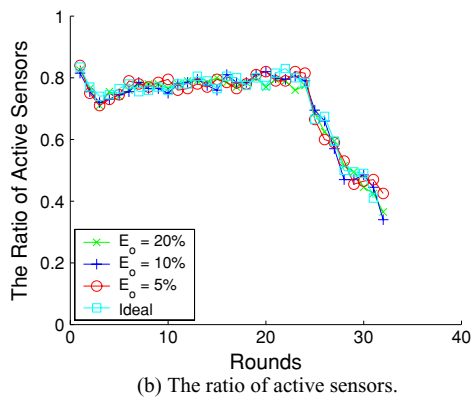
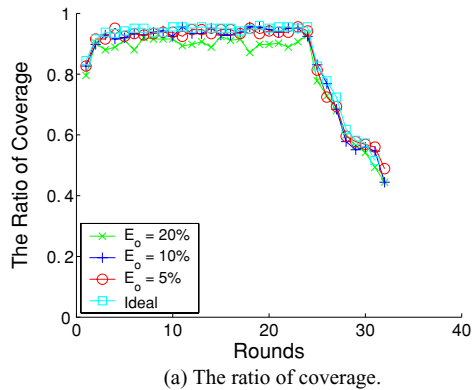
the same when orientation error bounds increase from 5 to 20%. The ratio of active sensors is nearly unchanged under various  $E_o$ . These results illustrate the robustness of the SNCS algorithm to orientation errors.

### 7.3.3. Effect of packet losses

To model communication errors, we assume that messages are corrupted, and hence lost, with a probability  $E_c$ . Figure 8(a) shows the performance comparison of the SNCS protocol where  $E_c$  is 10, 5 and 1%. We observe that the coverage ratios remain the same under different  $E_c$ . Also, the lifetime is the same (around 31 rounds) for all the simulated values of  $E_c$  except when  $E_c = 10\%$  where the lifetime is approximately 28 rounds, since the algorithm consumes more messages during the scheduling phases of the rounds—the reason is that some of the packets are lost and hence it takes more time to ultimately converge on the MCMS solution. Figure 8(b) shows that the ratio of active sensors increases proportionally with the increase in  $E_c$ .

Notice that, comparing  $E_c = 10\%$  with the ideal case, the increase in the average active ratio is only about 10%, which indicates the robustness of the SNCS protocol to packet errors.

**Fig. 7** Performance comparison of the SNCS protocol with and without orientation errors.  $m = 225$ ,  $n = 200$ ,  $p = 8$ ,  $R_s = 0.10$ ,  $R_c = 0.20$ ,  $\mathcal{R} = 1000$  slots



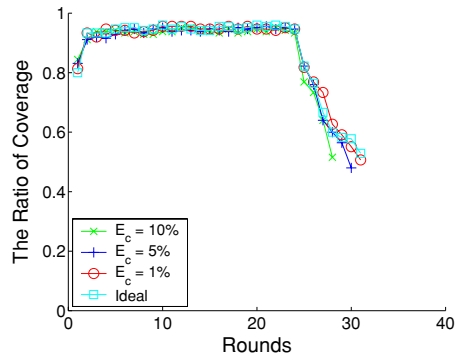
#### 7.4. Performance comparison of DGA and OGDC

By setting  $p = 1$  in DGA, coverage by directional sensors degenerates to the circular (isotropic) coverage, the most common case studied in the literature so far, which allows us to compare the performance of our proposed scheme with previous ones. Therefore, in this experiment, we focus on two distributed schemes, DGA and OGDC, which are regarded as one-time deployment schemes in this context. The performance metrics of interests are the ratio of coverage and the number of active sensors when finishing a configuration of the randomly deployed network.

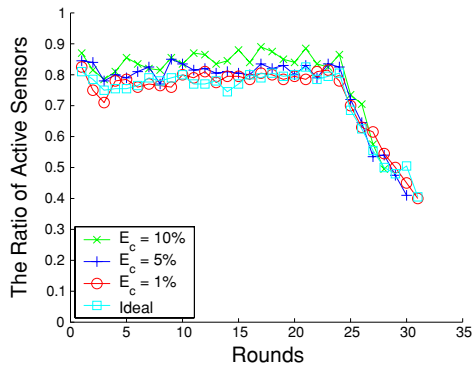
Although OGDC provides area coverage rather than target coverage, there are two reasons for choosing OGDC for comparison. First, as mentioned in Yan et al. (2003), by configuring the set of targets as a set of regular grid points with a certain density, target coverage can be approximated as area coverage. Second, as stated in (Zhang and Hou, 2005), OGDC outperforms other existing distributed algorithms (Ye et al., 2003; Tian and Georganas, 2002; Wang et al., 2003) in terms of the above interested performance metrics.

In the simulations, to run DGA, we configure the set of targets as follows. We divide the unit square area into  $l \times l$  square grids and targets are at the centers of these grids. To evaluate the effect of target density on the quality of deployment, we set  $l$  to 10, 15

**Fig. 8** Performance comparison of the SNCS protocol where  $m = 225, n = 200, p = 8, R_s = 0.10, R_c = 0.20, \mathcal{R} = 1000$  slots with communication errors



(a) The ratio of coverage.



(b) The ratio of active sensors.

and 20. The rest of the parameters for both DGA and OGDC<sup>3</sup> are:  $R_s = 0.1, R_c = 0.2$  and  $n$  ranges from 75 to 400 with an increment of 25. As for the performance results, the ratio of coverage is measured as follows. To compare DGA with OGDC, we need to provide some estimate of area coverage from DGA, since DGA provides target coverage. So, we configure a set of grid points by setting  $l = 100$  and then define the ratio of coverage as the ratio of the number of grid points that are covered by at least one sensor to the total number of grid points. Figure 9 shows the performance comparison of DGA and OGDC.

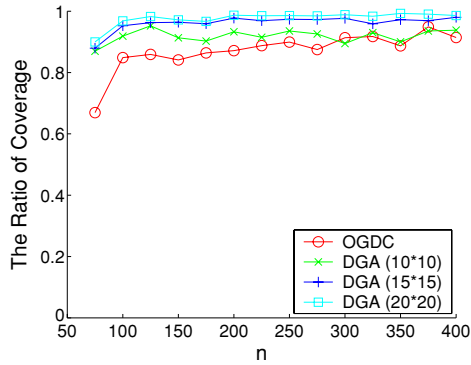
In fig. 9(a), the ratio of coverage by DGA increases with the increase of  $l$  for all  $n$  and tends to be saturated when  $l \geq 15$ . On the other hand, the ratio of coverage by OGDC is less than the  $l = 10$  case by DGA for most of  $n$  and gets close to it with the increase of  $n$ .

In fig. 9(b), the number of active sensors by DGA increases with the increase of  $l$  for all  $n$ . On the other hand, the curve depicting the number of active sensors of OGDC crosses the curve of the  $l = 10$  case by DGA when  $n \leq 175$  and then exceeds it.

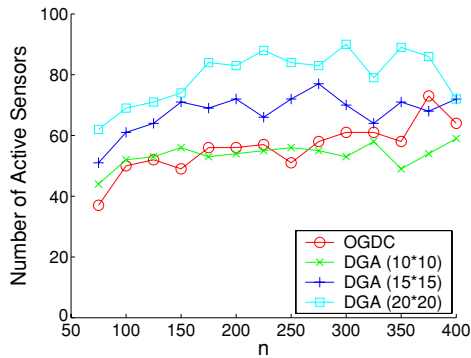
The above results show that, by appropriately configuring the density of targets as a set of regular grid points (i.e., the  $l = 10$  case by DGA where the interval of grid points is the same as the sensing range  $R_s$  of sensors), the performance of DGA is comparable to that of OGDC. The reason is that to maximize the number of acquired targets with a certain granularity in DGA is in the same spirit of minimizing the overlaps among sensing disks in OGDC.

<sup>3</sup> Other parameters specific to OGDC are chosen from Table I reported in Zhang and Hou (2005).

**Fig. 9** Performance comparison of OGDC and DGA, where  $p = 1, R_s = 0.1, R_c = 0.2$  and  $n$  ranges from 75 to 400



(a) The ratio of coverage.



(b) The number of active sensors.

### 8. Conclusions

In this paper, we study the problem of coverage by directional sensors in randomly deployed wireless sensor networks. To characterize the desired node and orientation configuration at any instant, we first propose the MCMS problem which is proved to be  $\mathcal{NP}$ -complete. Then we present its exact solution by an ILP formulation and approximate solution by CGA in a centralized fashion, respectively. Followed we provide the distributed solution of the MCMS problem by DGA and show its properties. Furthermore, to maximize the network lifetime in a larger time scale, we develop the SNCS protocol based on DGA with residual energy consideration of sensors. Finally, we systematically evaluate the performance of proposed solutions and protocol through extensive simulations.

To be noted, some components in our model may not be practical, such as the sensing region of a directional sensor with a binary detection model. However, our proposed framework can be easily generalized to accommodate other practical sensing and detection model of a sensor as long as we can measure the coverage and establish the relationship between sensing regions of sensors and the objects (e.g., target, region or volume) to be covered.

**Acknowledgments** The authors would like to thank Professor John Mitchell for discussing the proof of Theorem 3.1. The authors would also thank the anonymous reviewers for their valuable feedback.

## References

- Slijepcevic S, Potkonjak M (2001) Power efficient organization of wireless sensor networks, in IEEE International Conference on Communications (ICC), Helsinki, Finland, Vol. 2, pp. 472–476
- Tian D, Georganas ND (2002) A coverage-preserving node scheduling scheme for large wireless sensor networks, in Proceedings of the ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), Atlanta, GA, United States, pp. 32–41
- Wang X, Xing G, Zhang Y, Lu C, Pless R, and Gill C (2003) Integrated coverage and connectivity configuration in wireless sensor networks, in Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, United States, pp. 28–39
- Zhang H, Hou JC (2005) Maintaining sensing coverage and connectivity in large sensor networks, *The Wireless Ad Hoc and Sensor Networks: An International Journal*
- Cardei M, Du DZ (2005) Improving wireless sensor network lifetime through power aware organization, *ACM Wireless Networks* 11:333–340
- Megriani S, Potkonjak M (2003) Low power 0/1 coverage and scheduling techniques in sensor networks, University of California, Los Angeles, Department of Computer Science, Tech. Rep. 030001
- Cardei M, Thai MT, Li Y, Wu W (2005) Energy-efficient target coverage in wireless sensor networks, in Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3, Miami, FL, United States, March, 1976–1984
- Cheng MX, Ruan L, Wu W, (2005) Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks, in Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3: Miami, FL, United States, pp. 2638–2645
- Hochbaum DS (1997) Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. *Approximation Algorithms for NP-hard Problems*. PWS
- He T, Huang C, Blum BM, Stankovic JA, Abdelzaher T (2003) Range-free localization schemes in large-scale sensor networks, in Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM), San Diego, CA, United States, pp. 81–95
- “Ilog cplex” version 9.0. “[Online]. Available: <http://www.ilog.com/products/cplex/>”
- Feeney LM, Nilsson M (2001) Investigating the energy consumption of a wireless network interface in an ad hoc networking environment, in Proceedings of 20th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Vol. 3: Anchorage, AK, United States, pp. 1548–1557
- Yan T, He T, Stankovic J (2003) Differentiated surveillance for sensor networks, in Proceedings of the First International Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, United States, pp. 51–62
- Ye F, Zhong G, Cheng J, Lu S, Zhang L (2003) Peas: A robust energy conserving protocol for long-lived sensor networks, in Proceedings of 23rd International Conference on Distributed Computing Systems (ICDCS), Providence, RI, United States, pp. 28–37