Scheduling Sensor Activity for Point Information Coverage in Wireless Sensor Networks

Bang Wang, Kee Chaing Chua, Vikram Srinivasan and Wei Wang
Department of Electrical and Computer Engineering, National University of Singapore
(e-mail: {elewb, eleckc, elevs, g0402587}@nus.edu.sg)

***Abstract*— An important application of wireless sensor networks is to perform the monitoring missions, for example, to monitor some targets of interests at all times. Sensors are often equipped with non-rechargeable batteries with limited energy and energy saving is a critical aspect for wireless sensor networks. If a target is monitored simultaneously by serval sensors, some of them can be switched off to save energy without causing mission failure and by which their operational times as well as the network lifetime can be prolonged.**

In this paper, we study the problem of scheduling sensor activity to cover a set of targets with known locations such that all targets can be monitored all the time and the network can operate as long as possible. A solution to this scheduling problem is to partition all sensors into sensor covers such that each cover can monitor all targets and the covers are activated successively. In this paper, we propose to use the notion of information coverage which is based on the estimation theory to exploit the collaborative nature of wireless sensor networks, instead of using the conventional definition of coverage. Due to the use of information coverage, a target that is not within the sensing disk of any single sensor can still be considered to be monitored (information covered) by the cooperation of more than one sensor. This change of the problem settings complicates the solutions compared to that by using a disk coverage model and we propose a heuristic to approximately solve our problem. Simulation results show that our cover formation algorithm is better than an existing formation algorithm when only the conventional definition of coverage is used. Simulation results also illustrate that the network lifetime can be significantly improved by using the notion of information coverage compared with that by using the conventional definition of coverage.

I. Introduction

Recent advances in micro-electro-mechanical systems, digital electronics, and wireless communications have led to the emergence of *wireless sensor networks*, which consist of a great number of sensing devices each capable of sensing, processing and transmitting environmental information. Applications of wireless sensors networks include battlefield surveillance, environmental monitoring, biological detection, smart spaces, industrial diagnostics, and etc [1].

One of the fundamental issues in a wireless sensor network is the *coverage problem* [2], which addresses how well a target area is monitored. In some applications, the targets to be monitored are remote and dangerous and sensors have to be randomly deployed from, e.g., an aircraft. Due to the randomness of the deployment, it is not uncommon that targets might not be monitored if the sensor density is low. In literature, increasing sensor density is one way to provide adequate target coverage. This problem can also be solved by applying the notion of *information coverage* instead of the conventional *physical coverage* [3]. The information coverage will be briefly reviewed in the next section.

Another important issue in sensor networks is the *network lifetime*. The source of energy for a sensor is most often provided by its attached battery. Due to size and cost constraints, the energy available at each sensor is limited and in general non-rechargeable. The network lifetime can thus be measured as the time that the network cannot provide adequate coverage due to that some or all sensors deplete their energy. It has been shown in [4] that if a battery is used in short bursts with significant off-time other than in a continuous operation mode, its lifetime can be prolonged approximately twice times. Therefore, the lifetime of a sensor network can be prolonged if sensors are scheduled to alternate their active and inactive battery states.

Scheduling sensor activity to prolong network lifetime is not new in sensor networks. When the target area monitored by one sensor can also be monitored by other sensors, this sensor can temporarily switch off for saving energy and the network coverage can still be maintained. Therefore, we can partition the time line into contiguous intervals and in each interval only part of sensors are in the active state to monitor all targets and all other sensors are in the sleep state to save energy. A coverage-preserving node scheduling scheme has been proposed in [5], which is based on an off-duty eligibility rule to determine whether a sensor's sensing area is included in its neighbor's sensing area. In [6][7], the *disjoint set cover* has been used to solve the problem of preserving coverage and scheduling activity, which divides the available sensors into disjoint sets such that every set can completely cover all targets and are activated successively. However, these works are all based on the notion of physical coverage and cannot be applied in the domain of information coverage.

In this paper, we study the problem of scheduling sensor activity for sensor networks in which a set of targets with known locations should be monitored all the time. Sensors are randomly scattered in close proximity to a set of targets and send their sensed data to a processing center. The objective of the point information coverage problem is to ensure that every target should be information covered at all times and the network can operate as long as possible. Due to the use of information coverage, a target that is not within the sensing radius of any sensor can still be monitored by the cooperation of more than one sensor. Furthermore, even a target is within the sensing radius of only one sensor, it is not necessary for this sensor to be always active if some other sensors can cooperate to monitor the target. Therefore, scheduling sensor activity becomes more complicated in the domain of information coverage.

In this paper, we propose a heuristic to solve the scheduling problem for discrete points information coverage. The proposed heuristic includes three steps: The first step is to exhaustively search possible sensor sets for each target such that a target can be information covered by any set. The second step is to formulate sensor covers in a greedy way such that each cover can ensure that all targets are monitored. The last step is to compose a scheduling sequence by which sensors are activated successfully. The proposed heuristic is also applicable to generate disjoint physical covers in the domain of physical coverage by simply setting a parameter of the heuristic. Simulation results show that when only the conventional definition of coverage is used, the performance of our cover formation algorithm in terms of the number of generated covers is better than that of the cover formation algorithm proposed in [6]; and is comparable to that of another disjoint cover formation algorithm proposed in [7]. However, our formation algorithm is simpler than theirs. Simulation results also illustrate that the network lifetime can be significantly improved by using the notion of information coverage compared with that by using the conventional definition of coverage.

The rest of the paper is organized as follows. The definition of information coverage is briefly reviewed in Section II. Section III provides a detailed formulation for the problem of scheduling sensor activity for the point information coverage. A heuristic method is then proposed in Section IV which includes three main steps. Section V presents performance evaluation results and the paper is concluded in Section VI.

II. Information Coverage based on Estimation

Consider a snapshot of the sensing field and a set of K geographically distributed sensors, each making a measurement on an unknown parameter θ at some location and time. We assume that each sensor knows its own coordinates. An example can be the sensing of an acoustic signal of amplitude θ. Let $d_k, k = 1, 2, ..., K$ denote the distance between a sensor k and a location with some parameter θ. The parameter θ is assumed to decay with distance, and at distance d it is θ/d^α, where $\alpha > 0$ is the decay exponent. The measurement of the parameter, x_k, at a sensor may also be corrupted by an additive noise, n_k. Thus

$$x_k = \frac{\theta}{d_k^\alpha} + n_k, k = 1, 2, ..., K. \qquad (1)$$

The objective of a parameter estimator is to estimate θ based on the corrupted measurements. Let $\hat{\theta}$ and $\tilde{\theta} = \hat{\theta} - \theta$ denote the estimate and the estimation error, respectively. When K measurements are available, a well-known *best linear unbiased estimator* (BLUE) [8] can be applied to estimate $\hat{\theta}_K$ and to achieve a minimum *mean squared error* (MSE).

In general, the estimate of a parameter at a point where an event happens (or a target is present) is different from the estimate of the same point without the occurrence of the event (or without the appearance of the target). This is because the signal energy of an event/target (e.g., seismic vibrations caused by a moving target) is larger than the background noise energy. Therefore, if an estimation error is small, not only the event/target can be claimed to be detected but also the event/target parameter can be obtained within a certain confidence level. Note that the estimation error $\tilde{\theta}_K$ is a random variable with zero mean (due to the zero mean uncorrelated noises) and variance $\tilde{\sigma}_K^2$. We can use the probability that the absolute value of the estimation error is less than a constant A, i.e., $\Pr\{|\tilde{\theta}_K| \le A\}$, to measure how well a point is monitored. The larger this quantity is, the more reliable the estimate. When it is equal to or larger than a predefined threshold ϵ, i.e., $\Pr\{|\tilde{\theta}_K| \le A\} \ge \epsilon$, we define the *information coverage* for K cooperative sensors as follows [3].

Definition 1: (K**-sensor** ϵ**-error information coverage**) *A point is said to be* (K, ϵ)*-covered if there exists* K *sensors to estimate any parameter at the point such that* $\Pr\{|\tilde{\theta}_K| \le A\} \ge \epsilon$*, where* $0 \le \epsilon \le 1$*. A region is said to be completely* (K, ϵ)*-covered if all the points of the region are* (K, ϵ)*-covered.*

A sensor is called *isotropic* if its sensing ability is the same in all directions. For such a sensor, its $(1, \epsilon)$ information coverage is a disk with radius D_s centered at the sensor. We can set D_s as the maximum distance between a sensor and a point such that the point can be $(1, \epsilon)$-covered. In this case, the $(1, \epsilon)$ information coverage is the same as the physical coverage: Both mean that a point is within D_s distance of at least one sensor. In this paper, we call the disk centered at a sensor with radius D_s as the *physical sensing disk* of the sensor and D_s as the *physical sensing range*. One of the properties of information coverage is that if a point is (k, ϵ) covered, it is also $(k + 1, \epsilon)$ covered. Therefore, when examining (K, ϵ) information coverage for a point, one can first check $(1, \epsilon)$ coverage for this point. If it is $(1, \epsilon)$-covered, then it is also (K, ϵ)-covered. If it is not $(1, \epsilon)$-covered, one more sensor is added to examine if it is $(2, \epsilon)$ covered and so on. We refer the reader to [3][9] for more details of information coverage.

Fig. 1 illustrates the concepts of physical and information coverage. The physically covered area by the six sensors is the union of their sensing disks. The point marked by the star in the figure is not physically covered. However, this point may be information covered. Let x_k denote the measurement of the parameter θ at the kth sensor. We assume that the measured parameter decays with distance, as illustrated by the dashed co-centric circles around the point. We can estimate θ by using more than one measurement to reduce estimation error. If the estimation error is small enough when using, e.g., all the six sensors for estimation, then the point is claimed to be information covered. From this example, we can see that information coverage can extend physical coverage in that a point not physically covered can still be information covered without sacrificing estimation reliability.

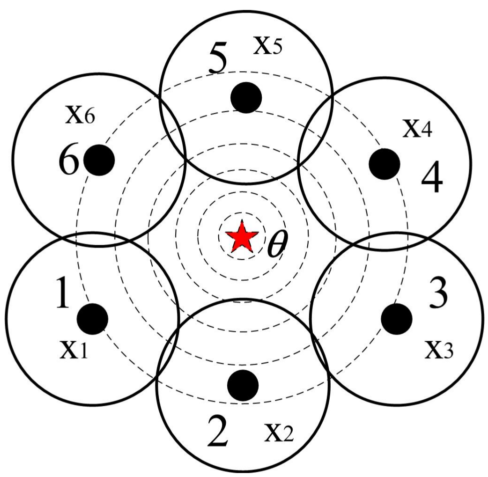

Fig. 1. Illustration of physical and information coverage. The point marked by the star is not physically covered, however, it may be $(6,\epsilon)$-covered if $\Pr\{\tilde{\theta}_6 \le A\} \ge \epsilon$, when 6 sensors are used for estimation.

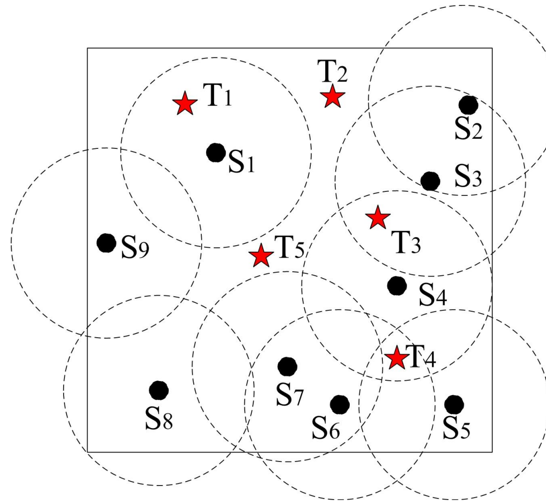

Fig. 2. Illustration of points coverage problem. There are 5 targets and 9 sensors in the field. Three targets (T_1, T_3, T_4) can be physically covered and two targets (T_2, T_5) should be information covered.

III. The Point Information Coverage Problem

There are a set of points (targets) with known locations in a field and a number of sensors have been randomly scattered in the field. The objective of point information coverage problem is to ensure that every target should be information covered at all times and as long as possible. Let $\mathcal{T} = \{T_1, T_2, ...\}$ and $\mathcal{S} = \{S_1, S_2, ...\}$ denote the target set and the sensor set, respectively. Fig. 2 illustrates an example of a number of randomly deployed sensors to cover a set of targets, in which dashed lines are used to illustrate physical sensing disks. Since sensors are randomly scattered, it may happen that some sensors cannot physically cover even one target (e.g., sensors S_2, S_7, S_8 and S_9) and some targets are not physically covered by even one sensor (e.g., targets T_2 and T_5). However, due to the use of information coverage, those targets that are not physically covered can still be information covered. For example, target T_2 can be $(3,\epsilon)$-covered by using sensors S_1, S_2 and S_3; and target T_5 may be $(2,\epsilon)$ covered by using sensors S_7 and S_9. Furthermore, those targets that are physically covered can also use many far away sensors to achieve (K,ϵ) information coverage ($K > 1$) instead of using only one nearby sensor for $(1,\epsilon)$ coverage. For example, target T_3 can be $(3,\epsilon)$-covered by using sensors S_1, S_2 and S_6. This method may not be energy efficient for monitoring target T_3 since it uses 3 sensors. However, it may be energy efficient from viewpoint of the whole network. For example, if only sensors S_1, S_2 and S_6 are scheduled to be active, then T_1 can be $(1,\epsilon)$-covered by S_1; T_2 be $(2,\epsilon)$-covered by S_1 and S_2; T_4 be $(1,\epsilon)$-covered by S_6; T_5 be $(3,\epsilon)$-covered by S_1, S_2 and S_6.

As discussed above, all targets may be information covered by using only a fraction of all scattered sensors. Therefore, sensors can be scheduled to monitor all targets alternatively and the sensor network lifetime can be prolonged by scheduling sensor activity. Let C_i denote a subset of $\mathcal{S}$ such that **all** targets can be information covered by using **all** sensors in C_i and let $\mathcal{C}$ denote a collection of subsets C_i, $\mathcal{C} = \{C_1, C_2, ...\}$. It is the application requirement that all targets should be covered at all times. We note that different targets may be covered by different numbers of sensors. Furthermore, we emphasize that only part of sensors in C_i cannot fully cover all targets. For example, sensors S_1, S_2 and S_6 can form a subset $C_1 = \{S_1, S_2, S_6\}$ if and only if they can cover all targets (as discussed above) and all the three sensors have to be used for the coverage of all the targets. We note that C_i and C_j need not to exclusive of each other, i.e., $C_i \bigcap C_j \neq \emptyset$ is allowed for $i \neq j$. Suppose that the time axis is divided into contiguous intervals of equal duration. In each interval, a subset C_i is first chosen from $\mathcal{C}$; Then all sensors in C_i are scheduled to be in the *active state* and all the other sensors go into the energy efficient *sleep state*. A sensor k in the active state (or the sensor is active) means that it needs to generate its local measurement, i.e., x_k and to periodically send x_k to a processing center every some time (much less than the scheduling interval). Since sensors are assumed to be isotropic, it may happen that some active sensors can determine the occurrence of some event but cannot exactly tell where the event is. However, this problem is not unique to information coverage and also exists in the domain of physical coverage. For example, consider that two targets are within the sensing disk of a sensor. A sensor in the sleep state means that it needs not to produce and to send measurements; hence energy can be saved. Let $\overline{\mathbf{C}} \equiv (C_1, C_2, ...,)$ denote a sequence of subsets by which subsets are selected sequentially in each interval and only sensors in the selected subset are scheduled to be in the active state. We note that subsets may be repeated in the sequence, e.g., a subset C_i may appear twice in the sequence. However, $\overline{\mathbf{C}}$ is unique and not the repeat of its components. Now we can formulate *the point information coverage problem* as follows: Find a sequence $\overline{\mathbf{C}}$ such that the network lifetime is maximized if sensors are scheduled according to the sequence.

There are two main differences between the point information coverage problem defined above with the *set k-cover* problem defined in [6] and the *disjoint set covers* problem defined in [7]. The first is that we use the notion of information coverage instead of physical coverage used in [6][7]. The other difference is that the subsets which are called set covers are exclusive of each other, i.e., $C_i \bigcap C_j = \emptyset$, in [6][7]. It has been shown in [6][7] that both the set k-cover problem and the disjoint set covers problems are NP-complete problems. Due to the use of information coverage and the non-exclusive subsets, the point information coverage problem is more complex than the two problems. Since the disjoint set cover problem can be considered as a special case of our point information coverage by simply assuming that only $(1, \epsilon)$ coverage is considered in our problem, according to the *restriction method* [10], our point information coverage problem is also NP-complete. In the next section, we present a heuristic method to approximately solve the point information coverage problem.

IV. A Heuristic Method

The proposed heuristic is to compose an information cover sequence by which only sensors in an information cover are activated in an time interval and the covers are activated alternatively. The proposed heuristic is based on some motivations from the *greedy algorithm* for the classic *set covering problem* (e.g., [11] page 1033). The un-weighted version of the set covering problem can be formally stated as follows: Given a collection $\mathcal{C} = \{C_1, C_2, ...\}$ of subsets of a finite set $\mathcal{T}$, we say that $\mathcal{C}$ is a cover of $\mathcal{T}$ if every element in $\mathcal{T}$ belongs to at least one subset in $\mathcal{C}$. The minimum set covering problem is to find a cover $C^* \subseteq \mathcal{C}$ with the least size. The size of C^* is defined as the number of sets it contains, rather than the number of individual elements in these sets. The well-known greedy algorithm is to choose an unused set that covers the largest number of uncovered elements in each step. In our heuristic method, the greedy algorithm will be modified by including weights and be used for generating and selecting covers. But before the generation and selection for covers, we need to determine the possible information covers first, i.e., to determine C_i; and after that, we need to compose covers into the cover sequence, i.e., to form $\overline{\mathbf{C}}$.

We first sketch the steps of our heuristic method, and then use a simple example to illustrate the main technical details. The proposed method is called *exhaustive-greedy-equalized heuristic* (EGEH) which consists of three steps. The pseudo-codes for the EGEH are shown in Tables I, II and III. The first step is to exhaustively search the possible coverage combinations for each target and to unite them as a pool for all targets. The second step is to greedily generate and select covers from the above mentioned pool. The third step is to compose the cover sequence in an equalized way such that the number of occurrences of each sensor is almost the same in the sequence. In the following, we use $|\cdot|$ to denote the operation that returns the number of elements of a set (or a tuple). For example, $|\mathcal{S}|$ returns the number of sensors.

TABLE I
EGEH Step 1: Exhaustive Search and Combination

(01)	**for** each $T_j \in \mathcal{T}$,
(02)	$K = 1$; $\mathcal{U}_j = \emptyset$;
(03)	**while** $K \le KMax$,
(04)	$\mathcal{W}$ = *combination*$(K, \mathcal{S})$;
(05)	**for** $a = 1 : \lvert\mathcal{W}\rvert$,
(06)	**if** *noninclusion*$(\mathcal{U}_j, w^a)$ **and** *coverage*(T_j, w^a)
(07)	$\mathcal{U}_j = \mathcal{U}_j \uplus w^a$;
(08)	**endif**
(09)	**endfor**
(10)	$K = K + 1$;
(11)	**endwhile**
(12)	**endfor**
(13)	$\mathcal{U} = \emptyset$;
(14)	**for** $j = 1 : \lvert\mathcal{T}\rvert$, **for** $a = 1 : \lvert\mathcal{U}_j\rvert$,
(15)	**if** $u_j^a \notin \mathcal{U}$,
(16)	$\mathcal{U} = \mathcal{U} \uplus u_j^a$; $\mathcal{T}(u_j^a) = \{j\}$;
(17)	**else** $\mathcal{T}(u_j^a) = \mathcal{T}(u_j^a) \bigcup \{j\}$; **endif**;
(18)	**endfor**; **endfor**;

For each target T_j, let $\mathcal{U}_j$ denote the set of all possible combinations of sensors such that T_j is information covered by any element of $\mathcal{U}_j$. The number of elements in the set $\mathcal{U}_j$ is $|\mathcal{U}_j|$. The elements of $\mathcal{U}_j$ are denoted as $u_j^1, u_j^2, ..., u_j^{|\mathcal{U}_j|}$ and an element itself is a subset of sensors, e.g., $u_j^1 = \{S_1\}, u_j^2 = \{S_2, S_3\}$. As discussed in Section II, if a point is (k, ϵ) covered, then it is also $(k+1, \epsilon)$ covered. Hence to avoid repetition when searching for (k, ϵ) coverage, we require that any element u_j^a is not a subset of another element u_j^b and vice versa, i.e., $u_j^a \not\subseteq u_j^b$ and $u_j^b \not\subseteq u_j^a$. This requirement is referred to as *non-inclusion condition*. We note that this condition does not mean the exclusiveness between u_j^a and u_j^b, i.e., $u_j^a \bigcap u_j^b \neq \emptyset$ is allowed. For example, $u_j^a = \{S_2, S_3\}$ and $u_j^b = \{S_2, S_4\}$. The pseudo-codes for the construction of $\mathcal{U}_j$ are lines 1 to 12 in Table I. The process is initialized by setting $\mathcal{U}_j = \emptyset$ and $K = 1$ (line 2). For each target, the exhaustive search of all possible combinations is terminated if the number of sensors to be selected is larger than a threshold $KMax$ $(1 \le KMax \le |S|)$, i.e., we are only interested in at most $(KMax, \epsilon)$ coverage. The function *combination* returns the set of all combinations of using K sensors (line 4). Note that an element w^a in $\mathcal{W}$ is a subset of $\mathcal{S}$. For example, when $K = 2$, it returns $\mathcal{W} = \{\{S_1, S_2\}, \{S_1, S_3\}, ..., \{S_2, S_3\}, ...\}$. We then check the (K, ϵ) coverage for each combination of K sensors (line 5). The function *noninclusion* checks whether any element in $\mathcal{U}_j$ is a subset of w^a and the function *coverage* checks whether the target T_j can be (K, ϵ) covered by using sensors in w^a. If a combination w^a satisfies the non-inclusion condition and can (K, ϵ) cover the target (line 6), then this combination is included into $\mathcal{U}_j$ (line 7). We use $\uplus$ to denote the operation such that w^a is included into $\mathcal{U}_j$ as a single element. After the exhaustive search of possible combinations for each target, we need to construct a pool $\mathcal{U}$ for all combinations and for all targets (lines 13 to 18). The set $\mathcal{T}(u_j^a)$ records all the targets that can be covered by using all sensors in u_j^a. Suppose that after the exhaustive

TABLE II

EGEH Step 2: Greedy Cover Formulation

(19)	$\mathcal{C} = \emptyset$; $\mathcal{W} = \mathcal{T}$;
(20)	**while** $\mathcal{W} == \mathcal{T}$;
(21)	$c = \emptyset$; $\mathcal{V} = \mathcal{T}$;
(22)	**while** $\mathcal{V} \neq \emptyset$
(23)	select an $u \in \mathcal{U}$ that maximizes $\frac{\lvert\mathcal{T}(u) \bigcap \mathcal{V}\rvert}{\lvert u\rvert - \lvert c \bigcap u\rvert}$;
(24)	$\mathcal{V} = \mathcal{V} \backslash \mathcal{T}(u)$;
(25)	$\mathcal{U} = \mathcal{U} \backslash u$;
(26)	$c = c \bigcup u$;
(27)	**endwhile**
(28)	**if** *noninclusion*$(\mathcal{C}, c)$, $\mathcal{C} = \mathcal{C} \biguplus c$; **endif**
(29)	$\mathcal{W} = \emptyset$;
(30)	**for** $a = 1 : \lvert\mathcal{U}\rvert$, $\mathcal{W} = \mathcal{W} \bigcup \mathcal{T}(u^a)$; **endfor**
(31)	**endwhile**

TABLE III

EGEH Step 3: Equalized Sequence Composition

(32)	set $F(S_i)$ as the number of appearance of S_i in $\mathcal{C}$ for all $S_i \in \mathcal{S}$
(33)	set $F(C_j) = \max\{F(S_i) : \text{for all } S_i \in C_j\}$ for all $C_j \in \mathcal{C}$;
(34)	set $FLcm = leastcommonmultiple(F(C_j)$, for all $C_j \in \mathcal{C})$;
(35)	set $A(C_j) = FLcm/F(C_j)$, for all $C_j \in \mathcal{C}$;
(36)	set $CLen = \sum_{C_j \in \mathcal{C}} A(C_j)$; $\overline{\mathbf{C}} = nil$; $CPrev = \emptyset$;
(37)	**while** $CLen > 0$
(38)	$cmin =$MAXINT;
(39)	**for** $j = 1 : \lvert\mathcal{C}\rvert$,
(40)	**if** $A(C_j) > 0$
(41)	**if** $\lvert C_j \bigcap CPrev\rvert < cmin$,
(42)	$a = j$; $cmin = \lvert C_j \bigcap CPrev\rvert$;
(43)	**elseif** $\lvert C_j \bigcap CPrev\rvert == cmin$ **and** *rand*> 0.5,
(44)	$a = j$; $cmin = \lvert C_j \bigcap CPrev\rvert$;
(45)	**endif endif**
(46)	**endfor**
(47)	$A(C_a) = A(C_a) - 1$; $CLen = CLen - 1$;
(48)	$\overline{\mathbf{C}} = \overline{\mathbf{C}} \bigoplus C_a$; $CPrev = C_a$;
(49)	**endwhile**

search of all coverage combinations for T_1 and T_2, we have $\mathcal{U}_1 = \{\{S_1\}, \{S_2, S_3\}, \{S_2, S_4\}, \{S_3, S_4\}, \{S_2, S_5, S_6\}\}$ and $\mathcal{U}_2 = \{\{S_6\}, \{S_2, S_4\}, \{S_3, S_4\}, \{S_1, S_4, S_5\}\}$. Then the pool $\mathcal{U} = \{\{S_1\}, \{S_2, S_3\}, \{S_2, S_4\}, \{S_3, S_4\}, \{S_2, S_5, S_6\}, \{S_6\}, \{S_1, S_4, S_5\}\}$ and $\mathcal{T}(\{S_1\}) = \{1\}$, $\mathcal{T}(\{S_2, S_3\}) = \{1\}$, $\mathcal{T}(\{S_2, S_4\}) = \{1, 2\}$, and etc.

The next step is to produce a cover set $\mathcal{C}$ from $\mathcal{U}$ such that each element of $\mathcal{C}$ can cover all the targets (lines 19 to 31). Recall that for each $u \in \mathcal{U}$, $|u|$ is the number of sensors used to cover $|\mathcal{T}(u)|$ targets. Therefore, $|\mathcal{T}(u) \bigcap \mathcal{V}|$ is the number of targets that have not been covered by c but can be covered by selecting u and $|u| - |c \bigcap u|$ is the number of sensors that have to be added to c due to the selection of u. The selection function $\frac{|\mathcal{T}(u) \bigcap \mathcal{V}|}{|u| - |c \bigcap u|}$ can be considered as the *marginal profit* of selecting u. In the case that $|\mathcal{T}(u) \bigcap \mathcal{V}| > 0$ and $|u| - |c \bigcap u| = 0$, the marginal profit is infinite, i.e., some uncovered targets can be covered without using extra sensors. In the case that $|\mathcal{T}(u) \bigcap \mathcal{V} = 0|$ and $|u| - |c \bigcap u| = 0$, the marginal profit is set as 0. In each pass to select a cover (lines 22 to 27), the un-weighted greedy algorithm is modified to select a set that maximizes the marginal profit. Ties are broken arbitrarily. The meaning of 'greedy' here has two folds: One is that a cover is greedily selected (line 23); the other is that the pool $\mathcal{U}$ is greedily reduced (line 25). The greedy exclusion of selected u from $\mathcal{U}$ greatly simplifies the cover formulation; however, it may also reduce the possible available covers. The sensors in the selected u form a cover (line 26). However, it may happen that a newly formed cover contains all sensors in a previously formed cover. That is, only a fraction of sensors in the formed cover can cover all the targets. In this case, the newly formed cover is a redundant one and is not included into the cover set (line 28). The cover formulation is terminated if the remaining set $u \in \mathcal{U}$ cannot fully cover all targets (line 20, 29 and 30). Follow the example in the previous paragraph, the formed covers are $\mathcal{C} = \{\{S_1, S_6\}, \{S_2, S_4\}, \{S_3, S_4\}\}$.

The final step is to compose a cover sequence $\overline{\mathbf{C}} = (C_1, C_2, ...,)$ by which sensors are scheduled to be in the active state alternatively. Each cover C_i in $\overline{\mathbf{C}}$ is selected from $\mathcal{C}$ and hence can cover all the targets. Furthermore, if any one sensor in a cover C_i fails, then not all targets can be covered. Since all sensors have the same power level at the time of sensor deployment, we propose to use an equalized sequence composition that tries to make each sensor appear equally and evenly in the sequence (lines 32 to 49). The rationale is that a cover with a sensor (or sensors) appeared many times in other covers should have less appearances in the sequence. Therefore, we first count the appearance of each sensor in the cover set (line 32) and use the worst sensor appearance of a cover as the cover weight (line 33). The number of appearance for each cover is reversely proportional to the cover weight (line 34 to 35). The length of the sequence in terms of the number of covers is then computed as the sum of the number of the appearances of all covers and some other initialization are performed before composing the sequence (line 36). In each pass of a cover selection, the process searches a cover exhaustively from the cover set such that this cover has not yet achieved its number of appearances in the sequence and has the least number of sensors covered by the previous cover in the sequence (lines 38 to 46). The function *rand* uniformly generates a number with value between 0 and 1. We use $\bigoplus$ to denote the operation that include a cover into the cover sequence. This sequence composition process will be terminated if the length of the sequence has achieved (line 37 and 49). Follow the previous example, we have $A(C_1) = 2$, $A(C_2) = 1$ and $A(C_3) = 1$ and $CLen = 4$. After the step 3, the composed sequence is $\overline{\mathbf{C}} = (C_1, C_2, C_1, C_3) = (\{S_1, S_6\}, \{S_2, S_4\}, \{S_1, S_6\}, \{S_3, S_4\})$.

We note that if only physical coverage is used, then only sensors S_1 and S_6 can cover the two targets and have to be activated all the times. Therefore, the network lifetime of our example by applying information coverage can be prolonged at least two times longer than that by applying the physical coverage.

The computation complexity of the heuristic is mainly from the exhaustive search part. For example, suppose we have

N sensors and M targets. We need to examine information coverage $M \times \sum_{K=1}^{KMax} \binom{N}{K}$ times for all targets. Furthermore, the computation time for checking $(K+1, \epsilon)$ coverage is in general larger than that for checking (K, ϵ) coverage for a point. Even with a simple assumption that computation cost is the same for checking (K, ϵ) coverage for all K, the computation complexity of the exhaustive search part increases exponentially with the network size N. For example, if $KMax = N$, the computation complexity is in the order of $M \times 2^N$. When only considering physical coverage (i.e., $KMax = 1$), the complexity of the exhaustive search part is in the order of $M \times N$, and is the same as that for checking physical coverage. In the case of $KMax = 1$, our heuristic is comparable to that proposed in [6] and less than that proposed in [7]. When information coverage is considered, i.e., $KMax > 1$, the complexity of EGEH becomes much larger than theirs. However, as pointed out in [7], this algorithm is executed by a central control center only once and hence trading off the running time in favor of more cover sets may be justified. Furthermore, according to our related work in [9], the system performance may not be significantly reduced by selecting only nearby sensors and using small values $KMax$ (e.g., $KMax = 6$) for each target with a given sensor density. Therefore, we may only need to consider to use sensors nearby a target to examine its information coverage instead of considering all sensors in the field and by which the computation complexity can be greatly reduced with a decreased network lifetime as a tradeoff.

V. Numerical Examples

The proposed EGEH also applies to generate disjoint set covers for physical coverage by simply setting $KMax = 1$. The covers that cover part or all targets under the physical coverage are called *physical covers* to distinguish from the information covers. The exhaustive search and combination algorithm first ensures that the physical covers each covering partial or all targets in the pool $\mathcal{U}$ are disjoint. This is because each target can only be $(1, \epsilon)$-covered by individual sensors and hence the physical covers in the pool $\mathcal{U}$ are individual sensors. Furthermore, the greedy cover formulation algorithm also ensures that the generated physical covers covering all targets are disjoint since it removes each selected physical cover from the pool.

We first compare the number of the disjoint physical covers generated by our algorithm with the number of physical covers produced by the *most constrained-minimally constraining heuristic* proposed by S. Slijepcevic and M. Potkonjak in [6] (called as the SP heuristic hereafter). The SP heuristic was designed for area monitoring. The area to be monitored is divided into a number of fields such that all points of a field are covered by the same set of sensors. By viewing every field as a target, we can directly apply the SP heuristic for generating disjoint covers each physically covering all targets. The SP heuristic builds each cover by successively adding the sensors that cover the sparsely covered parts of the area.

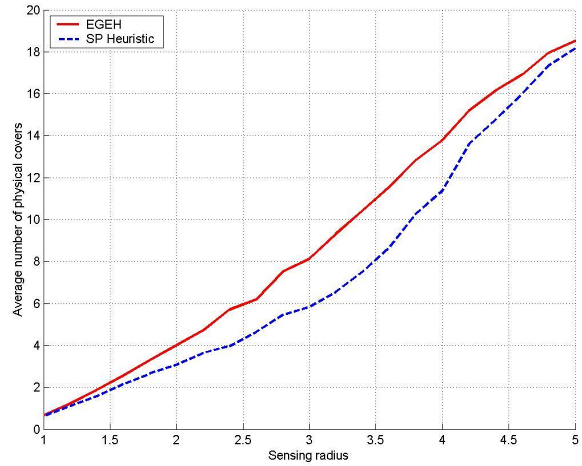

Fig. 3. The average number of physical covers vs the sensing radius (20 sensors and 4 targets in a square with side length of 5).

The selection priority is given to the sensors that (1) cover a high number of uncovered targets; (2) cover more sparsely covered targets; (3) do not cover targets redundantly and (4) redundantly cover the targets which are not sparsely covered.

In the simulations, 20 sensors and 4 targets are randomly scattered within a square sensing field with side length of 5. In a simulation run, if the scattered sensors cannot cover all the scattered targets, then no covers are generated. The average number of generated physical covers are then averaged over 100 simulation runs. Fig. 3 compares the average number of physical covers by EGEH and the SP heuristic against the sensing radius. Not unexpectedly, the average number of physical covers increases with the increase of the sensing radius. When the sensing radius increases, a sensor may cover more targets. In other words, the network redundant sensors grows and hence the more disjoint physical covers. It is observed that our algorithm consistently generates more physical covers than the SP heuristic. Furthermore, we notice that if we only compare the greedy cover formation in EGEH and the cover formation in the SP heuristic, the greedy cover formation is easier to be implemented and requires less computations. In [7], M. Cardei and D. Du have proposed to solve the disjoint set physical covering problem by converting it to a *maximum flow problem* based on a *mixed integer programming*. It has also been reported that algorithm proposed by Cardei and Du consistently performs slightly better than the SP heuristic. We note that the proposed EGEH has comparable performance as theirs, however, the complexity of EGEH is much less than theirs.

Before we compare the network lifetime for information coverage and physical coverage, we briefly introduce how to determine (k, ϵ) coverage for a point and refer the reader to [3] for details. Consider a special case that all noises are Gaussian and independent. The sum of these noises is still Gaussian with zero mean and variance $\tilde{\sigma}_K^2 = \sum_{k=1}^{K} a_k^2 \sigma_k^2$, where $a_k =$

$B_K/d_k^{\alpha}\sigma_k^2$ and $B_K = \left(\sum_{k=1}^{K} 1/d_k^{2\alpha}\sigma_k^2\right)^{-1}$. We further assume that all noises have the same variance, i.e., $\sigma_k^2 = \sigma^2$ for all $k = 1, 2....$ Hence we have

$$\Pr\{|\tilde{\theta}_K| \leq A\} = 1 - 2Q\left(\frac{A}{\sigma\sqrt{D_K}}\right), \qquad (2)$$

where $D_K = (\sum_{k=1}^{K} d_k^{-2\alpha})^{-1}$ and $Q(x) = \frac{1}{\sqrt{2\pi}}\int_x^{\infty} \exp(-\frac{t^2}{2})dt$. Define the sensing range of a single sensor, D_s , as the distance where the estimation error performance equals ϵ. Therefore, D_s satisfies $Q(\frac{A}{D_s^{\alpha}\sigma}) = \frac{1-\epsilon}{2}$. For simplicity, A can be set as $\beta\sigma$, $\beta > 0$. Here, we set $A = \sigma$ and choose a certain ϵ, and compute D_s as the sensing range used to relate the physical coverage and information coverage in a single sensor case, i.e., $(1, \epsilon)$ coverage is the same as the physical coverage when the radius of any sensing disk is D_s. Another way is to set $A = \sigma$ and set D_s as the unit for distance, and compute ϵ accordingly. Here, we set $D_s = 1$, and hence $\epsilon = 0.683$.

We assume that all sensors have the same energy level at the time of the network setup. The time axis is divided into contiguous intervals with equal duration. For simplification, we assume that a fixed amount of energy is consumed in each interval for an active sensor and no energy consumption for a sleep sensor. A sensor is dead if its energy level is zero. A cover is called *invalid* if any sensor of the cover is dead. Recall that our cover formulation requires all targets can be covered if and only if all sensors of the cover are active. Therefore, an invalid cover leads to the mission failure (not all targets are covered), although it may cover some targets. An invalid cover is then no longer used and removed from cover sequence. The *network lifetime* hence is defined as the duration in terms of the number of intervals from the time when the network is setup to the time when no valid cover is in the cover sequence. To avoid trivialities, we require that all the randomly scattered targets can be physically covered (i.e., $(1, \epsilon)$-covered) by some or all scattered sensors in each simulation run; If not, we re-scatter sensors and targets. The average network lifetime is then obtained from 100 simulation runs. To compare with the physical coverage, the average lifetime gain is defined as the average lifetime of using at most $KMax$ sensors for information coverage divided by the average lifetime of using only 1 sensor for the physical coverage.

Fig. 4 plots the average lifetime gain against $KMax$ for different decay exponent. The lifetime gain is significant when using the notion of the information coverage. Furthermore, the higher values of $KMax$, the higher lifetime gain; and the lower values of decay exponent; the higher lifetime gain. When the information coverage is used, some sensors that cannot be used for monitoring targets due to their distances to targets larger than the sensing radius can now be used for monitoring targets also, which increases the network lifetime. Furthermore, the number of sensors for the information coverage of all targets can even be less than the number of sensors for the physical coverage of all targets in some cases. For example, a target is close to two sensors (but not within the sensing radius of the two sensors) each physically covering a target; Then this target may also be $(2, \epsilon)$-covered by the two sensors without incurring the activation of another sensor. Therefor, the network lifetime can be prolonged when the information coverage is used.

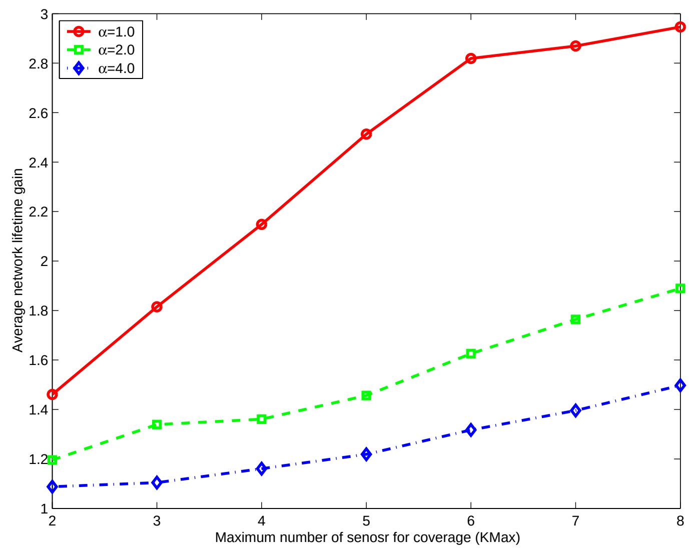

Fig. 4. Average network lifetime gain vs the maximum allowable number of sensors used for monitoring each target $(KMax)$ (20 sensors and 4 targets in a square with side length of 5).

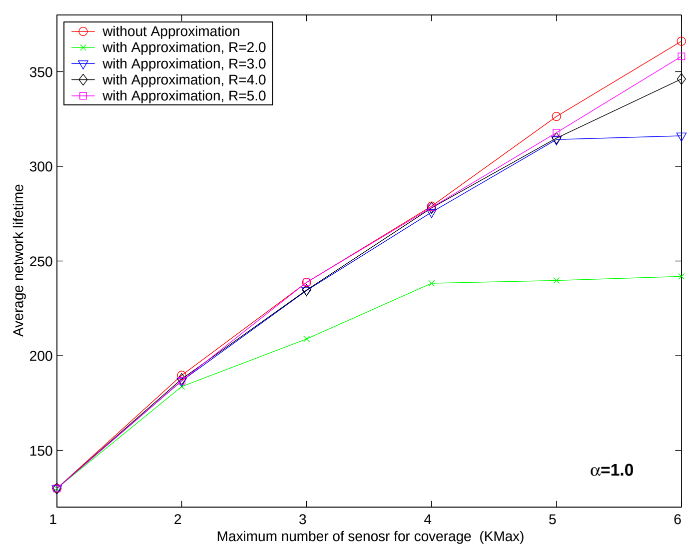

Fig. 5. Average network lifetime of the approximation scheme vs the maximum allowable number of sensors used for monitoring each target $(KMax)$ (20 sensors and 4 targets in a square with side length of 5).

As discussed in the previous section, the exhaustive search part is very time consuming and its computation complexity increases exponentially with the number of deployed sensors. We next simulate an approximation scheme which only considers sensors nearby a target for its information coverage instead of considering all sensors in the sensor field. For each target, only sensors in the disk centered at this target with a radius

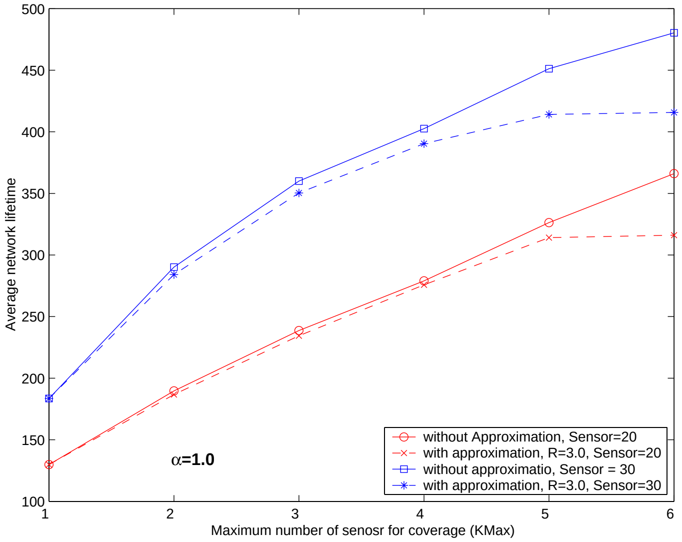

Fig. 6. Average network lifetime of the approximation scheme ($R = 3.0$) vs the maximum allowable number of sensors used for monitoring each target ($KMax$) (20 and 30 sensors and 4 targets in a square with side length of 5).

of R are used to information cover this target. Fig. 5 plots the average network lifetime for different values of R. It is not unexpected to observe that the network lifetime of using smaller values R is smaller than that using larger values of R. This is because that the chance for information covering a target is reduced due to less available sensors nearby a target when a smaller value of R is used. Fig. 6 compares the average network lifetime for the approximation scheme when different numbers of sensors are scattered into the field. It is seen that the approximation scheme performs poorer when more sensors are scattered. This indicates that the radius of R should be increased for better approximation when the sensor density increases.

VI. Concluding Remarks

Scheduling sensor activity is an effective way to prolong network lifetime. This paper has studied the problem of scheduling sensor activity, however, under the definition of information coverage instead of the conventional definition of physical coverage. We have proposed an exhaustive-greedy-equalized heuristic to approximately solve the problem. Simulation results have shown that in the case of physical coverage, the proposed heuristic performs better than an existing algorithm in terms of the number of generated disjoint covers. When information coverage is used instead of physical coverage, the network lifetime can be significantly improved.

References

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks, Elsevier Publishers*, vol. 39, no. 4, pp. 393–422, 2002.

[2] M. Cardei and J. Wu, *Coverage Problems in Wireless Ad Hoc Sensor Networks (Mohammad Ilyas and Imad Mahgoub eds.), Handbook of Sensor Networks*, chapter 19, CRC Press, 2004.

[3] B. Wang, W. Wang, V. Srinivasan and K. C. Chua, “Information coverage for wireless sensor networks,” *IEEE Communications Letters*, vol. 9, no. 11, pp. 967–969, November 2005.

[4] L. Benini, G. Castelli, A. Macii, E. Macii, M. Poncino and R. Scarsi, “A discrete-time battery model for high-level power estimation,” *Design, Automation and Test in Europe Conference*, pp. 35–39, 2000.

[5] D. Tian and N. D. Georganas, “A coverage-preserving node scheduling scheme for large wireless sensor networks,” *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pp. 32–41, 2002.

[6] S. Slijepcevic and M. Potkonjak, “Power efficient organization of wireless sensor networks,” *IEEE International Conference on Communications (ICC)*, vol. 2, pp. 472–476, June 2001.

[7] M. Cardei and D.-Z. Du, “Improving wireless sensor network lifetime through power aware organization,” *Wireless Networks, Kluwer Academic Publishers*, vol. 11, no. 3, pp. 333–340, May 2005.

[8] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications and Control*, Prentice Hall, Inc, 1995.

[9] B. Wang, K. C. Chua, V. Srinivasan and W. Wang, “Sensor density for complete information coverage in wireless sensor networks,” *accepted by European Workshop on Wireless Sensor Networks (EWSN)*, 2005.

[10] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, The MIT Press; 2nd Edition, 2001.