

Energy-Efficient Target Coverage in Wireless Sensor Networks

Mihaela Cardei
Dept. of Computer Science & Eng.
Florida Atlantic University
Boca Raton, FL 33431
mihaela@cse.fau.edu

My T. Thai Yingshu Li
Dept. of Computer Science & Eng.
University of Minnesota
Minneapolis, MN 55455
{mythai, yili}@cs.umn.edu

Weili Wu
Dept. of Computer Science
University of Texas at Dallas
Richardson, TX 75083
weiliwu@utdallas.edu

Abstract—A critical aspect of applications with wireless sensor networks is network lifetime. Power-constrained wireless sensor networks are usable as long as they can communicate sensed data to a processing node. Sensing and communications consume energy, therefore judicious power management and sensor scheduling can effectively extend network lifetime. To cover a set of targets with known locations when ground access in the remote area is prohibited, one solution is to deploy the sensors remotely, from an aircraft. The lack of precise sensor placement is compensated by a large sensor population deployed in the drop zone, that would improve the probability of target coverage. The data collected from the sensors is sent to a central node (e.g. cluster head) for processing.

In this paper we propose an efficient method to extend the sensor network life time by organizing the sensors into a maximal number of set covers that are activated successively. Only the sensors from the current active set are responsible for monitoring all targets and for transmitting the collected data, while all other nodes are in a low-energy sleep mode. By allowing sensors to participate in multiple sets, our problem formulation increases the network lifetime compared with related work [2], that has the additional requirements of sensor sets being disjoint and operating equal time intervals. In this paper we model the solution as the maximum set covers problem and design two heuristics that efficiently compute the sets, using linear programming and a greedy approach. Simulation results are presented to verify our approaches.

Keywords: wireless sensor network, energy efficiency, sensor scheduling, maximum set covers.

I. INTRODUCTION

Recent improvements in affordable and efficient integrated electronic devices have a considerable impact on advancing the state of wireless sensor networks, which constitute the platform of a broad range of applications including national security, surveillance, health care and environmental monitoring. Sensor nodes are small devices equipped with one or more sensors, one or more transceivers, processing, storage resources and possible actuators [1]. Sensor nodes organize in networks and collaborate to accomplish a larger sensing task. The characteristics of a sensor network include limited resources, large and dense networks (of hundreds or even thousands of sensor nodes) and a dynamic topology. A critical issue in wireless sensor networks is power scarcity, driven in part by battery size and weight limitations. As judicious

management of the available energy resources directly impacts the sensor network operation lifetime and the performance of the application, methods that optimize the sensor energy utilization have great importance.

A sensor node's radio can be in one of the following four states: transmit, receive, idle, or sleep. The idle state is when the transceiver is neither transmitting nor receiving, and the sleep mode is when the radio is turned off. As presented in [14], an analysis of the power usage for WINS Rockwell seismic sensor indicates power consumption for the transmit state between 0.38W and 0.7W, for the receive state 0.36W, for the idle state 0.34W and for the sleep state 0.03W. The receive and idle modes may require as much energy as transmitting, while the sleep mode requires the less energy. Another observation is the communication/computation power usage ratio, which can be higher than 1000 (e.g. for Rockwell WINS [14] is from 1500 to 2700), therefore local data processing, data fusion and data compression are highly desirable. Judiciously selecting the state of each sensor node's radio is accomplished through a *scheduling* mechanism.

Power saving techniques can generally be classified in the following categories:

- 1) schedule the wireless nodes to alternate between active and sleep mode
- 2) power control by adjusting the transmission range of wireless nodes
- 3) energy efficient routing, data gathering
- 4) reduce the amount of data transmitted and avoid useless activity.

In this paper we address the first method, that is, we design a mechanism that allows redundant nodes to enter the sleep mode. To design such a mechanism, one must answer the following questions [3]:

- 1) Which rule should each node follow to determine whether to enter sleep mode?
- 2) When should nodes make such a decision?
- 3) How long should a sensor remain in the sleep mode?

An important problem addressed in literature is the *sensor coverage problem*. This problem is centered around a fundamental question: “How well do the sensors observe the physical space?” As pointed out in [13], the coverage concept is a measure of the quality of service (QoS) of the sensing function and is subject to a wide range of interpretations due to a large variety of sensors and applications. The goal is to have each location in the physical space of interest within the sensing range of at least one sensor.

In this paper we address the *target coverage problem*, with the objective of maximizing the network lifetime of a power constrained wireless sensor network deployed for monitoring (coverage) of a set of targets with known locations. We consider that a large number of sensor nodes are dispersed randomly in close proximity of a set of objectives (targets) and send the sensed information to a central processing node. We define the sensor network lifetime as the time interval each target is covered by at least one sensor node.

In this paper we propose to extend the network lifetime by dividing the sensor nodes into a number of sets, such that each set completely covers all the targets. These sensor sets are activated successively, such that at any time instant only one set is active. The sensors from the active set are in the *active* state (e.g. transmit, receive or idle) and all other sensors are in the *sleep* state. If, while meeting the coverage requirements, sensor nodes alternate between the active and sleep mode, this will result in increasing the network and application lifetime compared with the case when all sensors are active continuously. Also, as a consequence, the spatial density of active nodes is lowered, thus reducing contention at the MAC layer. The contributions of this paper are the following:

- 1) introduce a new model of maximizing the network lifetime of the target coverage problem by organizing the sensor nodes in non-disjoint set covers; we define the maximum set covers (MSC) problem and prove that MSC is NP-complete
- 2) design two target coverage heuristics for efficiently solving the MSC problem using linear programming and greedy techniques, and
- 3) analyze the performance of our approach through simulation.

The rest of the paper is organized as follows. In section II we present energy efficient and coverage related works. Section III describes the target coverage problem. Next, in section IV, we introduce the maximum set covers (MSC) problem and prove that MSC problem is NP-complete. We propose a linear programming based heuristic in section V-B and a greedy solution in section V-C. Section VI presents the simulation results for our heuristics, and section VII concludes our paper.

II. RELATED WORK

Sensor nodes have size, weight and cost restrictions, with direct impact on resource availability. They have limited battery resources, processing and communication capabilities. As replacing the battery is not feasible in many applications, low power consumption is one of the most important requirements of a sensor network [8]. Various power efficient schemes have been proposed in literature [7], not only at the hardware and architectural design, but also when designing algorithms and protocols at all layers of the network architecture.

In sensor coverage problems, the goal is to have each location in the physical space of interest within the sensing range of at least one sensor. Cardei and Wu [3] survey recent sensor coverage problems proposed in literature and categorize them according to the following design criteria:

- 1) objective of the problem: maximize network lifetime or minimize the number of sensors deployed
- 2) sensor deployment method: deterministic versus random
- 3) relationship between sensing R_s and communication R_c ranges (e.g. $R_c = R_s$?; Do all sensors use the same R_c and the same R_s (homogeneous network)?)
- 4) additional critical requirements, such as energy-efficiency and connectivity
- 5) algorithms characteristics: centralized versus distributed and localized.

The coverage problems can be classified in the following types [3]:

- area coverage [4], [16], [17], [19], [15], where the main objective is to cover (monitor) an area,
- point (or target) coverage [2], [9], where the objective is to cover a set of points (targets) and
- coverage problems that have an objective to determine the maximal support/breach paths that traverse a sensor field [13], [10].

An important method for prolonging the network lifetime for the area coverage problem is to determine a localized and distributed protocol for selecting the set of active sensor nodes. To be distributed and localized are important properties of a node scheduling mechanism, as they better adapt to a scalable and dynamic network topology. The network activity can be organized in rounds, and the set of active sensor nodes is decided at the beginning of each round. Active node selection is determined based on the problem requirements (e.g. area monitoring, connectivity, power efficiency). Different techniques have been proposed in literature [4], [16], [17], [19] for efficiently deciding the eligibility rule, that is, to determine if a node will be in sleep mode for the next round. The basic idea is to design a mechanism such that each node can determine if its sensing disk is collectively covered by other nodes which

decided their active status. In this case, the node can go to sleep mode for the current round.

The coverage settings considered by Meguerdichian et al in [13] assume a given field instrumented with sensors and the initial and final locations of an agent that needs to move through the field. The problem asks to determine a maximal breach path (MBP) and the maximal support path (MSP) of the agent. The MBP (MSP) corresponds to the worst (best) case coverage and has the property that for any point on the path, the distance to the closest sensor is maximized (minimized). The model assumes homogeneous sensor nodes, known sensor locations (e.g. through GPS), with sensing effectiveness decreasing as the distance increases. The authors proposed a centralized solution, based on the observation that MBP lies on the Voronoi diagram lines and MSP lies on Delaunay triangulation lines.

The best coverage problem is further explored and formalized in [10], where Li et al proposed a distributed algorithm for MSP computation using the relative neighborhood graph. The authors also consider two extensions, namely MSP with least energy consumption and MSP with smallest path distance.

The works most relevant to our approach are [2] and [15]. Both of these papers propose energy efficient centralized mechanisms by dividing the sensor nodes into disjoint sets, such that every set can individually perform the coverage tasks. These sets are then activated successively, and while the current sensor set is active, all other nodes are in the sleep mode. The goal of this approach is to determine a maximum number of disjoint sets, as this has a direct impact on conserving sensor energy resources as well as on prolonging the network lifetime.

Cardei and Du [2] address the target coverage problem where disjoint sensor sets are modeled as disjoint set covers, such that every cover completely monitors all the target points. Disjoint set coverage problem is proved to be NP-complete, and a lower approximation bound of 2 for any polynomial-time approximation algorithm is indicated. The disjoint set cover problem [2] is reduced to a maximum flow problem, which is then modeled as a mixed integer programming.

Slijepcevic and Potkonjak [15] address the area coverage problem where the area is modeled as a collection of fields, where every field has the property that any enclosed point is covered by the same set of sensors. The most-constrained least-constraining algorithm [15] computes the disjoint covers successively, selecting sensors that cover the critical element (field covered by a minimal number of sensors), giving priority to sensors that: cover a high number of uncovered fields, cover sparsely covered fields and do not cover fields redundantly.

Our approach differs from these solutions by not requiring the sensor sets to be disjoint and by allowing the sets to operate for different time intervals. Because the solution space of the disjoint set cover problem is included in the solution space of the maximum set covers (MSC) problem, the optimal solution of the MSC problem produces better results in terms

of improving wireless sensor network lifetime.

III. TARGET COVERAGE PROBLEM

We consider a number of targets with known locations that need to be continuously observed (covered) and a large number of sensors randomly deployed closed to the targets. We also consider a central data collector node, which we will refer as the base station (BS). This BS can be the cluster head into a more general, cluster-based framework. Sensed data might be processed locally by the sensors or at the BS, from where it is aggregated and forwarded to the user. We also assume the sensors have location determination capabilities (e.g. GPS).

As we assume the number of sensors deployed in the field is greater than the optimum needed to perform the required task, an important energy-efficient method consists in scheduling the sensor nodes activity to alternate between *active* state and *sleep* state. We consider that a sensor node radio can go to the sleep mode when the node is not scheduled to perform the sensing task.

Definition 1: Target Coverage Problem (TCP): Given m targets with known location and an energy constrained wireless sensor network with n sensors randomly deployed in the targets' vicinity, schedule the sensor nodes activity such that all the targets are continuously observed and network lifetime is maximized.

The sensor scheduling mechanism can be accomplished as follows:

- 1) sensors send their location information to the BS
- 2) BS executes the sensor scheduling algorithm and broadcasts the schedule when each node is active
- 3) every sensor schedules itself for active/sleep intervals.

In this paper we are concerned with designing the node scheduling mechanism, and do not address the problem of selecting which protocol is used for data gathering or node synchronization. To efficiently transmit data from the sensors to the BS, a mechanism like LEACH[6] or PEGASIS[11] can be used. For node synchronization, one method is to have the BS periodically sending short beacons.

IV. MAXIMUM SET COVERS PROBLEM

In this section we define the maximum set covers problem (MSC) in section IV-A and prove its NP-completeness in section IV-B.

A. MSC Problem Definition

Let us assume that n sensors s_1, s_2, \dots, s_n are randomly deployed to cover m targets r_1, r_2, \dots, r_m . The base station (BS) has the coordinates of the sensor nodes and the targets, therefore it is able to compute for each sensor node which

targets it covers. One method is to assume that a sensor covers a target if the Euclidean distance between sensor and target is smaller or equal with a predefined sensing range.

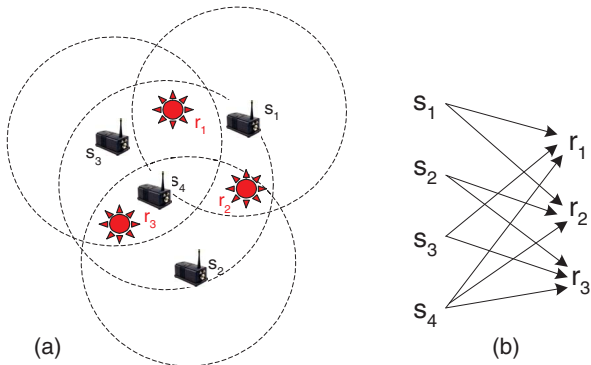


Fig. 1. Example with three targets $R = \{r_1, r_2, r_3\}$ and four sensors $C = \{s_1, s_2, s_3, s_4\}$

Figure 1 (a) shows an example with four sensor nodes s_1, s_2, s_3, s_4 and three targets r_1, r_2, r_3 . In this example we assume a node sensing area being the disk centered at the sensor, with radius equal to the sensing range. The coverage relationship between sensors and targets is also illustrated in the Figure 1 (b): $s_1 = \{r_1, r_2\}$, $s_2 = \{r_2, r_3\}$, $s_3 = \{r_3, r_1\}$ and $s_4 = \{r_1, r_2, r_3\}$. Note that a circular sensing area is not a requirement for our solution, we are just concerned with identifying which sensors cover each target.

We assume that all sensor nodes have the same remaining energy. In order to model the network lifetime, we assume that each sensor can be active for a unit time of 1. That is, if all sensors are active continuously, then the network lifetime is 1.

The work in [2], divides the sensors in disjoint sets, e.g. $S_1 = \{s_1, s_2\}$ and $S_2 = \{s_3, s_4\}$. This will result in a network lifetime of 2.

In this paper, we improve the scheduling scheme by allowing every sensor to be part of more than one set, and by allowing the sets to be operational for different time intervals. As illustrated in Figure 2, the sets in this case are: $S_1 = \{s_1, s_2\}$ for 0.5 time, $S_2 = \{s_2, s_3\}$ for 0.5 time, $S_3 = \{s_1, s_3\}$ for 0.5 time and $S_4 = \{s_4\}$ for 1 time. This organization results in a network lifetime of 2.5. This corresponds to 25% increase in network lifetime compared with the disjoint sets solution.

The problem of computing the set covers such that to maximize the network lifetime is formally defined next.

Definition 2: Maximum Set Covers (MSC) Problem: Given a collection C of subsets of a finite set R , find a family of set covers S_1, \dots, S_p with time weights t_1, \dots, t_p in $[0, 1]$ such

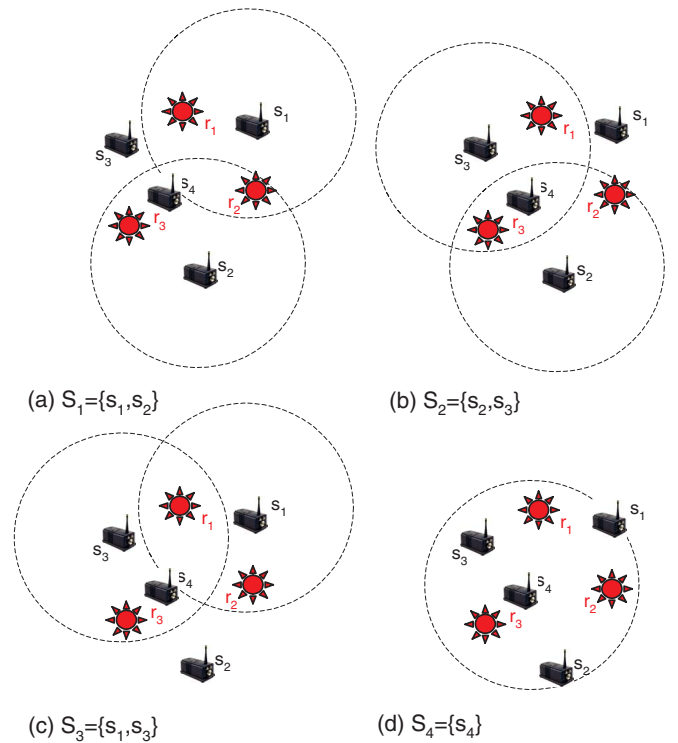


Fig. 2. Four cover sets: $S_1 = \{s_1, s_2\}$ for 0.5 time, $S_2 = \{s_2, s_3\}$ for 0.5 time, $S_3 = \{s_1, s_3\}$ for 0.5 time and $S_4 = \{s_4\}$ for 1 time

that to maximize $t_1 + \dots + t_p$ and for each subset s in C , s appears in S_1, \dots, S_p with a total weight of at most 1, where 1 is the life time of each sensor.

In MSC definition, C is the set of sensors and R is the set of targets, such that each sensor covers (monitor) a subset of targets. We want to determine a number of set covers S_1, \dots, S_p , where each set cover $S_i, i = 1..p$ completely covers all the targets, such that to maximize the network lifetime $t_1 + \dots + t_p$, where $t_j, j = 1, \dots, p$ is the time interval while the set cover S_j is active. Note that if a sensor belongs to more than one cover, then the sum of the time intervals of those covers cannot be greater than 1. This is because each sensor cannot be active more than 1.

B. MSC is NP-complete

In this section we first define the decision version of the MSC problem and then prove that MSC problem is NP-complete.

Definition 3: Decision Version of the MSC Problem: Given a collection C of subsets of a finite set R and a number k , does it exist a family of set covers S_1, \dots, S_p with time weights t_1, \dots, t_p such that $t_1 + \dots + t_p \geq k$ and for each subset s in C , s appears in S_1, \dots, S_p with total weight at most 1, where 1 is the life time of each sensor?

Theorem 1: MSC problem is NP-complete.

Proof: To show that $MSC \in NP$, consider that we are given a certificate and a number k . That is, we are given a family of set covers S_1, \dots, S_p with time weights t_1, \dots, t_p . Then, we can verify in polynomial time whether

- $t_1 + \dots + t_p \geq k$,
- each element in R is covered by at least one element in each $S_i, i = 1..p$
- for each subset s in C , s appears in S_1, \dots, S_p with total weight at most 1

To prove that the decision version of the MSC problem is NP-hard, we reduce the 3-SAT [5] problem to it in polynomial-time. A boolean formula is in conjunctive normal form (CNF) if it is expressed as an AND of clauses, each of which is the OR of one or more literals. A boolean formula is in 3-CNF if each clause has exactly three distinct literals. The 3-SAT problem is defined as follows: given a 3-CNF formula F , determine whether F has a satisfiable assignment.

Let F be a 3-CNF formula of a 3-SAT instance with n variables x_1, \dots, x_n and m clauses c_1, \dots, c_m . We define:

- $P_i = \{x_i, \bar{x}_i\} \cup \{c_j | c_j \text{ contains } x_i\}$,
- $Q_i = \{x_i, \bar{x}_i\} \cup \{c_j | c_j \text{ contains } \bar{x}_i\}$,
- $U = \{u\} \cup \{x_1, \dots, x_n\} \cup \{c_1, \dots, c_m\}$,
- $V = \{u\} \cup \{\bar{x}_1, \dots, \bar{x}_n\}$,

where u is a new element other than x_i, \bar{x}_i, c_j . These $2(n+1)$ sets form the collection of subsets of the set $R = \{u, x_1, \dots, x_n, \bar{x}_1, \dots, \bar{x}_n, c_1, \dots, c_m\}$. Let us set $k = 2$.

If the 3-SAT instance is satisfiable, then we can define two disjoint set covers

- $S_1 = V \cup \{P_i | x_i = 1\} \cup \{Q_i | \bar{x}_i = 1\}$
- $S_2 = U \cup \{P_i | x_i = 0\} \cup \{Q_i | \bar{x}_i = 0\}$

with time weights $t_1 = t_2 = 1$. In this case, we observe that the decision version receive the Yes-answer and we note that:

- 1) To cover u , every set-cover must contain either U or V . Since the total weight of the set-covers is at least 2, and each of U and V can appear in the set-covers with total weight at most 1, U and V cannot appear in the same set-cover.
- 2) Note that U and V cannot appear in the same set-cover. To cover both x_i and \bar{x}_i , every set-cover must contain either P_i or Q_i . It follows that P_i and Q_i cannot appear in the same set-cover. Otherwise, either P_i or Q_i would appear in the set-covers with total weight exceeding 1, which is a contradiction.

Let us now consider a set cover S containing V and exactly one of P_i and Q_i for $i = 1..n$. The set $x_i = 1$ if $P_i \in S$ and $x_i = 0$ if $Q_i \in S$. This is a satisfiable assignment for the 3-SAT instance.

Since MSC problem belongs to the class NP and is NP-hard, we can conclude that MSC is NP-complete. \square

V. OUR SOLUTIONS TO COMPUTE MAXIMUM SET COVERS

In this section we present two heuristics for the MSC problem. We first model the MSC problem as an Integer Programming in section V-A and then use the relaxation technique to design a Linear Programming based heuristic in section V-B. Then, we propose a greedy heuristic in section V-C, where set covers are formed individually, by covering first the most critical targets.

A. Integer Programming Formulation of the MSC Problem

Let us set a bound p for the number of set-covers. We formulate the MSC problem as follows:

Given:

- a set of n sensor nodes $C = \{s_1, s_2, \dots, s_n\}$
- a set of m targets $R = \{r_1, r_2, \dots, r_m\}$
- the relationship between sensor and targets, that is, for each sensor which is the set of targets it covers. This is modeled as having each element in C represented as a subset of the finite set R

Let us define $C_k = \{i | \text{sensor } s_i \text{ covers target } r_k\}$

Variables:

- x_{ij} , boolean variable, for $i = 1..n$ and $j = 1..p$; $x_{ij} = 1$ if sensor s_i is in the set cover S_j , otherwise $x_{ij} = 0$.
- $t_j \in \mathbb{R}$, $0 \leq t_j \leq 1$, for $j = 1..p$, represents the time allocated for the set cover S_j .

The optimization problem can be written as:

$$\text{Maximize } t_1 + \dots + t_p$$

$$\text{subject to } \sum_{j=1}^p x_{ij} t_j \leq 1 \text{ for all } s_i \in C$$

$$\sum_{i \in C_k} x_{ij} \geq 1 \text{ for all } r_k \in R, j = 1, \dots, p$$

$$\text{where } x_{ij} = 0, 1 \text{ (} x_{ij} = 1 \text{ if and only if } s_i \in S_j \text{)}$$

Remarks:

- the first constraint, $\sum_{j=1}^p x_{ij} t_j \leq 1$ for all $s_i \in C$ guarantees that the time allocated for each sensor s_i , across all set covers, is not larger than 1, which is the life time of each sensor.
- the second constraint, $\sum_{i \in C_k} x_{ij} \geq 1$, for all $r_k \in R$, $j = 1..p$, and $C_k = \{i | s_i \text{ covers target } r_k\}$, guarantees that each target r_k is covered by at least one sensor s_i in each set cover S_j .

We observe that the term $x_{ij}t_j$ is not linear. Therefore we set $y_{ij} = x_{ij}t_j$, and reformulate the problem as:

$$\begin{aligned} & \text{Maximize} && t_1 + \dots + t_p \\ & \text{subject to} && \sum_{j=1}^p y_{ij} \leq 1 \text{ for all } s_i \in C \\ & && \sum_{i \in C_k} y_{ij} \geq t_j \text{ for all } r_k \in R, j = 1, \dots, p \\ & \text{where} && y_{ij} = 0 \text{ or } t_j \text{ and } y_{ij} \leq 1 \end{aligned}$$

B. LP-MSc Heuristic

To transform this formulation into a linear programming (LP), we further apply the relaxation technique:

$$\begin{aligned} & \text{Maximize} && t_1 + \dots + t_p \\ & \text{subject to} && \sum_{j=1}^p y_{ij} \leq 1 \text{ for all } s_i \in C \\ & && \sum_{i \in C_k} y_{ij} \geq t_j \text{ for all } r_k \in R, j = 1, \dots, p \\ & \text{where} && 0 \leq y_{ij} \leq t_j \leq 1 \end{aligned}$$

We are now ready to introduce our LP maximum set cover heuristic (LP-MSc):

LP-MSc Heuristic

Step 1 Solve the linear programming LP formulated above. Let (y_{ij}^*, t_j^*) , $i = 1..n$ and $j = 1..p$, be the optimal solution of the LP. Set the network lifetime $G = 0$.

Step 2 The first approximation solution can be obtained as follows:

```

for all  $j = 1$  to  $p$  do
  set  $y_{ij}^0 = 0$  for all sensors  $s_i \in C$ 
  set  $t_j^0 = \min_k \max_{i \in C_k} y_{ij}^*$ 
  for all  $k = 1$  to  $m$  do
    /* for each  $r_k \in R$  */
    choose an  $i \in C_k$  such that  $y_{ij}^* \geq t_j^0$  and set  $y_{ij}^0 = t_j^0$ 
  end for
end for
    
```

After the first approximation:

- each sensor s_i , $i = 1..n$, has a remaining life time $T_i = 1 - \sum_j y_{ij}^0$
- network lifetime $G = G + \sum_{j=1}^p t_j^0$

Step 3 We iteratively repeat step 1 and step 2 by solving the following linear program, in order to improve the current network lifetime G :

$$\begin{aligned} & \text{Maximize} && t_1 + \dots + t_p \\ & \text{subject to} && \sum_{j=1}^p y_{ij} \leq T_i \text{ for all } s_i \in C \\ & && \sum_{i \in C_k} y_{ij} \geq t_j \text{ for all } r_k \in R, j = 1, \dots, p \\ & \text{where} && 0 \leq y_{ij} \leq t_j \leq 1 \end{aligned}$$

The iteration is executed while each target is covered by at least one sensor having the remaining lifetime greater than 0.

Step 4 Return the network lifetime G .

End LP-MSc

After we run step 1 and step 2 of the LP-MSc Heuristic, we set up a solution (y_{ij}^0, t_j^0) , based on the optimal solution (y_{ij}^*, t_j^*) returned by the linear programming LP. In the step 3, we iteratively run the LP in order to improve the current solution. Let $(\bar{y}_{ij}^h, \bar{t}_j^h)$ be the optimal solution of the LP in the h th iteration. Let us define $t_j^h = \min_{r_k \in R} \max_{i \in C_k} \bar{y}_{ij}^h$ and let y_{ij}^h be set equal to 0 or t_j^h such that for every r_k , there exists an $i \in C_k$ such that $y_{ij}^h = t_j^h$.

Let us denote:

- $g_h = \sum_{j=1}^p t_j^h$
- $\bar{g}_h = \sum_{j=1}^p \bar{t}_j^h$
- $\rho = \max_{r_k \in R} |C_k|$

Theorem 2: $g_h \geq \frac{1}{\rho} \cdot \bar{g}_h$, for any iteration h .

Proof: Note that for any $j = 1..p$,

$$\bar{t}_j^h \leq \sum_{i \in C_k} \bar{y}_{ij}^h \leq \rho \max_{i \in C_k} \bar{y}_{ij}^h, \text{ for all } r_k \in R,$$

therefore,

$$\bar{t}_j^h \leq \rho \min_{r_k \in R} \max_{i \in C_k} \bar{y}_{ij}^h = \rho \cdot t_j^h$$

Hence $\sum_{j=1}^p \bar{t}_j^h \leq \rho \sum_{j=1}^p t_j^h$, resulting in $\bar{g}_h \leq \rho \cdot g_h$. \square

Let us now analyze the runtime complexity of the LP-MSc heuristic based on the following variables: n -number of sensors, m -number of targets, and p -upperbound for the number of set covers. The runtime of the LP is $O(N^3)$ if Ye's algorithm [18] is used, where N is the number of variables. We can therefore consider for the step 1 runtime $O(N^3)$, where $N = p(1+n)$. The complexity of the step 2 is $O(nmp)$. Since $m \ll n$, the step 2 runtime complexity is dominated by the step 1 runtime complexity. If, for the step 3, we assume that steps 1 and 2 are repeated for a constant number of times, we get a total runtime complexity for the heuristic of $O(p^3 n^3)$.

C. Greedy-MSc Heuristic

In this section we propose a greedy approach for the MSc problem. Our heuristic takes as the input parameters C -the

set of sensors, R -the set of targets, and w -sensor lifetime granularity, $w \in (0, 1]$. The heuristic returns i -the number of set covers and the set covers C_1, C_2, \dots, C_i .

Greedy-MSc Heuristic (C, R, w)

```

1: set lifetime of each sensor to 1
2: SENSORS = C
3: i=0
4: while each target is covered by at least one sensor in
   SENSORS do
5:   /* a new set cover  $C_i$  will be formed */
6:   i = i + 1
7:    $C_i = \emptyset$ 
8:   TARGETS = R
9:   while TARGETS  $\neq \emptyset$  do
10:    /* more targets have to be covered */
11:    find a critical target  $r_{critical} \in TARGETS$ 
12:    select a sensor  $s_u \in SENSORS$  with greatest
       contribution, that covers  $r_{critical}$ 
13:     $C_i = C_i \cup s_u$ 
14:    for all targets  $r_k \in TARGETS$  do
15:      if  $r_k$  is covered by  $s_u$  then
16:        TARGETS = TARGETS -  $r_k$ 
17:      end if
18:    end for
19:  end while
20:  for all sensors  $s_j \in C_i$  do
21:    lifetime_ $s_j$  = lifetime_ $s_j$  - w
22:    if lifetime_ $s_j$  == 0 then
23:      SENSORS = SENSORS -  $s_j$ 
24:    end if
25:  end for
26: end while
27: return  $i$ -number of set covers and
    the set covers  $C_1, C_2, \dots, C_i$ 

```

The heuristic recursively builds set covers, in lines 5 to 19. The set $SENSORS$ maintain the list of sensors that have the residual energy greater than zero, thus these sensors can participate in additional set covers. The set $TARGETS$ contains the targets that still have to be covered by the current set cover C_i .

At each step, a critical target is selected, in line 11, to be covered. This can be for example the target most sparsely covered, both in terms of number of sensors as well as with regard to the residual energy of those sensors. Once the critical target has been selected, the heuristic selects the sensor with the greatest contribution that covers the critical target. Various sensor contribution functions can be defined. For example we can consider a sensor to have greater contribution if it covers a larger number of uncover targets and if it has more residual energy available. Once a sensor has been selected, it is added to the current set cover in line 13, and all additionally covered targets are removed from the $TARGETS$ set. When all targets are covered, the new set cover was formed.

The condition in line 4, that each target is covered by at

least one sensor in the set $SENSORS$, guarantees that a new set cover will be formed in lines 5 to 19. A target is either covered by the sensors already selected in the set cover, or it becomes a critical target, at which point the sensor with the greatest contribution, that covers the critical target, is selected. Based on the condition in line 4, at least one such sensor exists.

After a set cover C_i has been formed, the lifetime of each sensor in C_i is updated in line 21. We use w to represent the time that each set cover is active. For example, for $w = 0.2$, each sensor can be part of at most five set covers. If $w = 1$, this corresponds to the disjoint set covers, when each sensor can be part of only one set cover. Once a sensor finishes its time, it is removed from the set of available sensors $SENSORS$.

The network lifetime is computed as $w * i$, with i -number of set covers and w -time each cover is active.

The complexity of the Greedy-MSc Heuristic is $O(im^2n)$, where i is the number of set covers. The variable i is upperbounded by d/w , where d is the number of sensors that covers the most sparsely covered target. Usually w is a constant and $d < n$. Thus the heuristic runtime is $O(dm^2n)$.

VI. SIMULATION RESULTS

In this section we evaluate the performance of the LP-MSc (see section V-B) and Greedy-MSc (see section V-C) heuristics, designed to compute a maximum number of set covers. We simulate a stationary network with sensor nodes and target points randomly located in a $500m \times 500m$ area. We assume the sensing range is equal for all the sensors in the network. In the simulation we consider the following tunable parameters:

- n , the number of sensor nodes. We vary the number of randomly deployed sensor nodes between 25 and 750 to study the effect of node density on the performance.
- m , the number of targets to be covered. We vary the number of targets between 5 and 15.
- r , the sensing range. We vary the sensing range between 100m and 300m.

To solve the linear programming, we used the optimization toolbox in Matlab [12].

In the first experiment, we consider 5 and 15 target points randomly distributed, and we vary the number of sensors between 25 and 75 with an increment of 5, when the sensing range is 250m. We take the starting value p in our heuristic equal to the number of sensor nodes ($p = n$).

In Figure 3, we present the lifetime computed by LP-MSc and Greedy-MSc heuristics, depending on the number of sensors and number of targets deployed. Network lifetime results returned by the two heuristics are close and they increase with sensor density. When more sensors are deployed, each target is covered by more sensors, thus more set covers can be formed. Also, considering the same number of sensors, for a smaller number of targets, the lifetime (and the heuristic

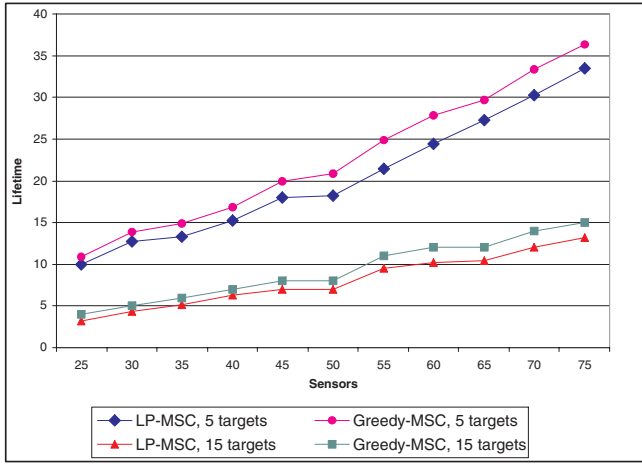


Fig. 3. Network lifetime with number of sensors when range $r=250m$

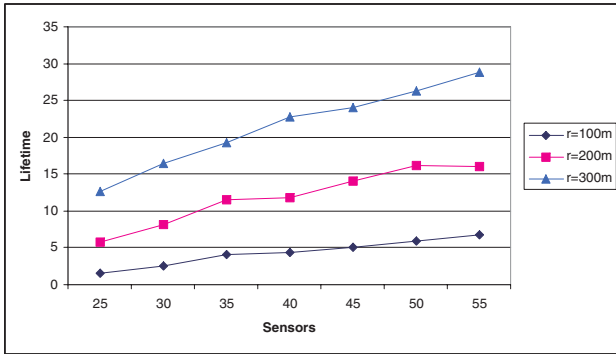


Fig. 4. LP-MSC heuristic, network lifetime with number of sensors for 10 targets

runtime) increases. This is because more cover combinations between the sensor nodes are feasible.

In Figure 4, we measure the network lifetime when the number of sensors vary between 25 and 55 and the sensing range is 100m, 200m, and 300m. We consider 10 targets randomly deployed. The network lifetime increases with the number of sensors and with the sensing range, as each node can now participate in more covers.

Figure 5 shows the convergence of the network lifetime value with the number of iterations for 0.1 and 0.01 tolerance. For example, for 45 sensors we obtained the lifetime 15.859 after 41 iterations, for a tolerance of 0.1 and the lifetime 17.016 after 61 iterations, for a tolerance of 0.01.

In Figure 6 we use Greedy-MSC to compute the network lifetime for large sensor networks, since it has lower complexity and running time. We vary the number of sensors between 250 and 750. We compare the results returned by Greedy-MSC with an upperbound computed as the number of sensors that covers the most sparsely covered sensor. Since each sensor has a lifetime of 1 and each target has to be observed continuously,

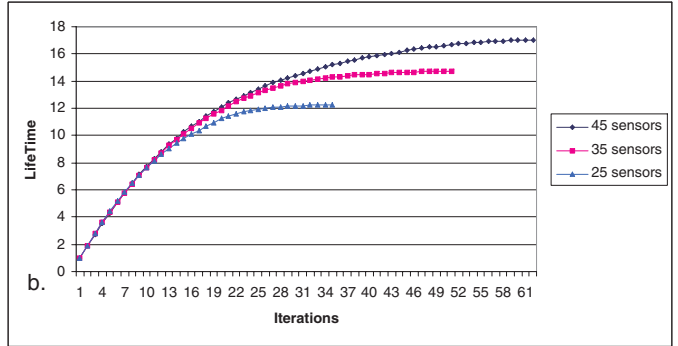
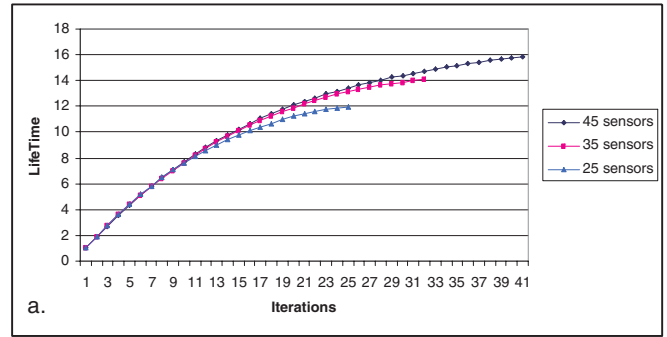


Fig. 5. LP-MSC heuristic, lifetime and number of iterations for tolerance 0.1 (a.) and tolerance 0.01 (b.)

this number is an upperbound of the network lifetime.

Sensors	LP-MSC		Greedy-MSC	
	Lifetime	Runtime (s)	Lifetime	Runtime (s)
25	10.004	12.428	10.900	0.100
30	12.715	24.235	13.900	0.150
35	13.320	32.237	14.900	0.150
40	15.293	52.886	16.900	0.290
45	17.957	127.843	19.900	0.331
50	18.236	220.738	20.900	0.450
55	21.405	334.361	24.900	0.620
60	24.456	511.095	27.800	0.631
65	27.318	3262.181	29.700	0.851
70	30.260	11789.452	33.400	0.871
75	33.410	2976.460	36.300	1.202

TABLE I

Runtime of LP-MSC and Greedy-MSC heuristics

In Table I, we present the runtime for LP-MSC and Greedy-MSC heuristics, when we vary the number of sensors deployed to monitor 5 targets. We set the range $r = 250m$. As the theoretical complexity analysis indicates, the Greedy-MSC runtime is much smaller than LP-MSC runtime. This is due to the recursive calls to LP-solver in LP-MSC heuristic.

The simulation results can be summarized as follows:

- for a specific number of targets, the network lifetime output by our heuristics increases with the number of sensors and the sensing range

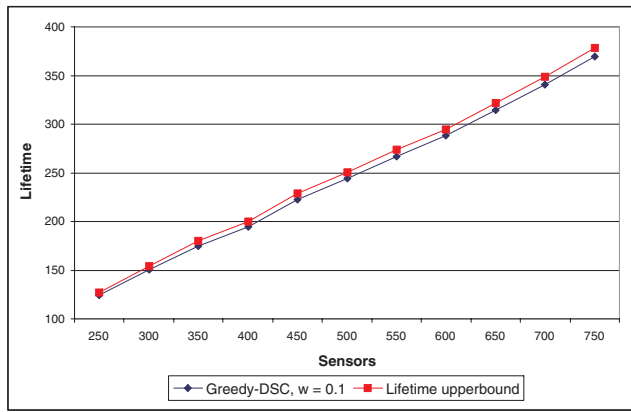


Fig. 6. Greedy-MSC, network lifetime for 5 targets and range $r=250m$

- for a specific number of sensors and sensing range, the network lifetime increases as the number of targets to be monitored decreases
- for smaller tolerance values, the lifetime value increases over time as result of additional execution of steps 1 and 2 of the LP-MSC heuristic (see section V-B). There is a trade-off between the higher lifetime value and the increase in the runtime, triggered by additional LP-solver calls.
- Greedy-MSC has a lower running time, thus it is more scalable to large sensor networks

VII. CONCLUSIONS

Wireless sensor networks are battery powered, therefore prolonging the network lifetime through a power aware node organization is highly desirable. An efficient method for energy saving is to schedule the sensor node activity such that every sensor alternates between sleep and active state. One solution is to organize the sensor nodes in set covers, such that every cover completely monitors all the targets. These covers are activated in turn, such that at a specific time only one sensor set is responsible for sensing the targets, while all other sensors are in the sleep state. This problem is modeled as maximum set covers problem. We proved that this problem is NP-complete and proposed two efficient heuristics, LP-MSC and Greedy-MSC heuristics, using a linear programming formulation and greedy approach, respectively. Simulation results are presented to verify our approaches.

As part of our future work, we will investigate the impact of k -coverage and $p\%$ -coverage ($p \leq 100$) on the network lifetime. k -coverage problem requires a robust coverage, when each target has to be covered by at least k sensors. $p\%$ -coverage (e.g. $p = 90$) problem requires that each target be covered at least $p\%$ of the time. Another direction is to design a distributed and localized algorithm for energy-efficient target coverage.

ACKNOWLEDGMENT

Weili Wu is supported in part by the National Science Foundation under grant ACI-0305567.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian and E. Cayirci, A Survey on Sensor Networks, *IEEE Communications Magazine*, (Aug. 2002), pp 102-114.
- [2] M. Cardei, D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, accepted to appear in *ACM Wireless Networks*.
- [3] M. Cardei, J. Wu, Energy-Efficient Coverage Problems in Wireless Ad Hoc Sensor Networks, accepted to appear in *Computer Communications*, special issue on Sensor Networks.
- [4] J. Carle and D. Simplot, Energy Efficient Area Monitoring by Sensor Networks, *IEEE Computer*, Vol 37, No 2 (2004), pp 40-46.
- [5] M. R. Garey, D. S. Johnson, *Computers and Intractability. A guide to the Theory of NP-Completeness*, Freeman, New York, 1979.
- [6] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *HICSS 2000*.
- [7] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, A survey of energy efficient network protocol(I), *Wireless Networks*, 7 (2001), pp 343-358.
- [8] J. Kahn, R. H. Katz and K. Pister, Next Century Challenges: Mobile Networking for Smart Dust, *ACM MOBICOM Conference* (Aug. 1999).
- [9] K. Kar and S. Banerjee, Node Placement for Connected Coverage in Sensor Networks, *Proc. of WiOpt 2003: Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks* (2003).
- [10] X.-Y. Li, P.-J. Wan, and O. Frieder, Coverage in Wireless Ad-hoc Sensor Networks, *IEEE Transactions on Computers*, Vol 52 (2002), pp 753-763.
- [11] S. Lindsey, C. S. Raghavendra, PEGASIS: Power Efficient GATHERing in Sensor Information Systems, *IEEE Aerospace Conference* (Mar. 2002).
- [12] Matlab, <http://www.mathworks.com/>
- [13] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, Coverage Problems in Wireless Ad-Hoc Sensor Networks, *IEEE Infocom* (2001), pp 1380-1387.
- [14] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, Energy-Aware Wireless Microsensor Networks, *IEEE Signal Processing Magazine*, 19 (2002), pp 40-50.
- [15] S. Slijepcevic, M. Potkonjak, Power Efficient Organization of Wireless sensor Networks, *IEEE International Conference on Communications*, (Jun. 2001).
- [16] D. Tian and N. D. Georganas, A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks, *Proc. of the 1st ACM Workshop on Wireless Sensor Networks and Applications* (2002).
- [17] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. D. Gill, Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks, *First ACM Conference on Embedded Networked Sensor Systems* (2003).
- [18] Y. Ye, An $o(n^3l)$ potential reduction algorithm for linear programming, *Mathematical Programming*, Vol 50 (1991), pp 239-258.
- [19] H. Zhang and J. C. Hou, Maintaining Sensing Coverage and Connectivity in Large Sensor Networks, *NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks* (Feb. 2004).