Maximum Network Lifetime in Wireless Sensor Networks with Adjustable Sensing Ranges

Mihaela Cardei, Jie Wu, Mingming Lu, and Mohammad O. Pervaiz

***Abstract*— This paper addresses the target coverage problem in wireless sensor networks with adjustable sensing range. Communication and sensing consume energy, therefore efficient power management can extend network lifetime. In this paper we consider a large number of sensors with adjustable sensing range that are randomly deployed to monitor a number of targets. Since targets are redundantly covered by more sensors, in order to conserve energy resources, sensors can be organized in sets, activated successively. In this paper we address the Adjustable Range Set Covers (AR-SC) problem that has as its objective finding a maximum number of set covers and the ranges associated with each sensor, such that each sensor set covers all the targets. A sensor can participate in multiple sensor sets, but sum of the energy spent in each set is constrained by the initial energy resources. In this paper we mathematically model solutions to this problem and design heuristics that efficiently compute the sets. Simulation results are presented to verify our approaches.**

Keywords: wireless sensor networks, energy efficiency, sensor scheduling, linear programming, optimization.

I. Introduction

Wireless sensor networks (WSNs) constitute the foundation of a broad range of applications related to national security, surveillance, military, health care, and environmental monitoring. One important class of WSNs is wireless ad-hoc sensor networks, characterized by an *ad-hoc* or *random* sensor deployment method [9], where the sensor location is not known a priori. This feature is required when individual sensor placement is infeasible, such as battlefield or disaster areas. Generally, more sensors are deployed than required (compared with the optimal placement) to perform the proposed task; this compensates for the lack of exact positioning and improves fault tolerance. The characteristics of a sensor network [1] include limited resources, large and dense networks, and a dynamic topology.

An important issue in sensor networks is power scarcity, driven in part by battery size and weight limitations. Mechanisms that optimize sensor energy utilization have a great impact on prolonging the network lifetime. Power saving techniques can generally be classified in two categories: scheduling the sensor nodes to alternate between active and sleep mode, and adjusting the transmission or sensing range of the wireless nodes. In this paper we deal with both methods. We design a scheduling mechanism in which only some of the sensors are active, while all other sensors are in sleep mode. Also, for each sensor in the set, the goal is to have a minimum sensing range while meeting the application requirements.

All authors are with the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431. Email: {mihaela@cse., jie@cse., mlu2@, mpervaiz@}fau.edu.
Mihaela Cardei is the corresponding author (phone: 561-297-3459; fax: 561-297-2800).

In this paper we address the *target coverage* problem. The goal is to maximize the network lifetime of a power constrained wireless sensor network, deployed for monitoring a set of targets with known locations. We consider a large number of sensors, deployed randomly in close proximity of a set of targets, that send the sensed information to a central node for processing. The method used to extend the network's lifetime is to divide the sensors into a number of sets. Using the property that sensors have adjustable sensing ranges, the goal is to set up minimum sensing ranges for the active sensors, while satisfying the coverage requirements. Besides reducing the energy consumed, this method lowers the density of active nodes, thus reducing interference at the MAC layer.

The contributions of this paper are: (1) introduce the Adjustable Range Set Covers (AR-SC) problem and the mathematical model, (2) design efficient heuristics (both centralized and distributed) to solve the AR-SC problem, using linear programming and greedy techniques, and (3) analyze the performance of our approaches through simulations.

The rest of the paper is organized as follows. In section II we present related works on sensor coverage problems. Section III defines AR-SC problem and section IV presents our heuristic contributions. In section V we present the simulation results and section VI concludes our paper.

II. Related Work

In this paper we address the sensor coverage problem. As pointed out in [10], 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.

A survey on coverage problems in wireless sensor networks is presented in [4]. The coverage problems can be classified in the following types [4]: (1) area coverage [5], [12], [13], [14], [16], where the objective is to cover an area, (2) point coverage [3], [2], [6], where the objective is to cover a set of targets, and (3) coverage problems that have the objective to determine the maximal support/breach path that traverses a sensor field [10].

An important method for extending the network lifetime for the area coverage problem is to design a distributed and localized protocol that organizes the sensor nodes in sets. The network activity is organized in rounds, with sensors in the active set performing the area coverage, while all other sensors are in the sleep mode. Set formation is done based on the problem requirements, such as energy-efficiency, area monitoring, connectivity, etc. Different techniques have been proposed in literature [5], [12], [13], [14], [16] for determining the eligibility rule, that is, to select which sensors will be active in the next round. In [14], the authors addressed area coverage when sensors can adjust their sensing ranges.

For applications that require more stringent fault-tolerance or for positioning applications, k-coverage might be a requirement. In [8], the goal is to determine whether a given area satisfies the k-coverage requirement, when each point in the area of interest is covered by at least k sensors. Both uniform and non-uniform sensing ranges are considered, and the k-coverage property is reduced to the k perimeter coverage of each sensor in the network.

A different coverage formulation is given in [10]. A path has the worst (best) coverage if it has the property that for any point on the path, the distance to the closest sensor is maximized (minimized). Given the initial and final locations of an agent, and a field instrumented with sensors, authors [10] proposed centralized solutions to the worst (best) coverage based on the observation that worst coverage path lies on the Voronoi diagram lines and best coverage path lies on Delaunay triangulation lines.

The works most relevant to our approaches are [2] and [3]. Paper [2] introduces the target coverage problem, where disjoint sensor sets are modeled as disjoint set covers, such that every cover completely monitors all the target points. The disjoint set coverage problem is proved to be NP-complete, and a lower 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 mixed integer programming. This problem is further extended in [3], where sensors are not restricted to participation in only disjoint sets, that is, a sensor can be active in more than one set.

The coverage breach problem is introduced in [6], addressing the case when sensor networks have limited bandwidth. The objective of the problem is to organize the sensors in disjoint sets, such that each set has a given bounded number of sensors and the overall breach is minimized. The overall breach is measured as the number of targets uncovered by the sensor sets.

Our paper is an extension of the maximum set covers problem addressed in [3], for the case when sensor nodes can adjust their sensing range. Our goal is to reduce the sensing range of the active sensors, while maintaining the coverage requirements. This method has a double impact: first it reduces energy consumption, and second it reduces interference at the MAC layer. Sensors with adjustable sensing ranges are available commercially [11], [14].

Compared with [3], in this paper we are also concerned with designing a distributed and localized algorithm (see section IV-B.2) for the AR-SC problem. Distribution and localization are important properties of a node scheduling mechanism, as it adapts better to a scalable and dynamic topology.

III. Problem Definition

Let us assume that N sensors s_1, s_2,..., s_N are randomly deployed to cover M targets t_1, t_2,..., t_M. Each sensor has an initial energy E and has the capability to adjust its sensing range. Sensing range options are r_1, r_2,..., r_P, corresponding to energy consumptions of e_1, e_2,..., e_P.

We assume a base station (BS) located within the communication range of each sensor. One method to compute the sensor - target coverage relationship is to consider that a sensor covers a target if the Euclidean distance between the sensor and target is no greater than a predefined sensing range.

The formal problem definition is given below:

Definition 1: Target Coverage Problem [3]
Given M targets with known location and an energy constrained WSN with N sensors randomly deployed in the targets' vicinity, schedule the sensor nodes' activity such that all targets are continuously observed and network lifetime is maximized.

The approach we used in this paper is to organize the sensors in sets, such that only one set is responsible for monitoring the targets, and all other sensors are in sleep mode. Besides determining the set covers, we are also concerned with setting the sensing range of each active sensor. The goal is to use a minimum sensing range in order to minimize the energy consumption, while meeting the target coverage requirement.

Next we formally define the Adjustable Range Set Covers (AR-SC) problem, used to solve the target coverage problem.

Definition 2: AR-SC Problem
Given a set of targets and a set of sensors with adjustable sensing ranges, find a family of set covers $c_1, c_2, ..., c_K$ and determine the sensing range of each sensor in each set, such that (1) K is maximized, (2) each sensor set monitors all targets, and (3) each sensor appearing in the sets $c_1, c_2, ..., c_K$ consumes at most E energy.

In AR-SC definition, the requirement to maximize K is equivalent with maximizing the network lifetime. The sensing range of a sensor determines the energy consumed by the sensor when that set is activated. If a sensor participates in more than one set, then the sum of energy spent has to be at most E.

AR-SC problem is NP-complete, by restriction method [7]. Maximum Set Covers [3] is a special case of AR-SC problem when the number of sensing ranges $P = 1$ and when the time a sensor is active is considered to be the energy consumed.

Figure 1 (a) shows an example with four sensors s_1, s_2, s_3, s_4 and three targets t_1, t_2, t_3. Each sensor has two sensing range r_1, r_2 with $r_1 < r_2$. In this example we assume a node's sensing area is the disk centered at the sensor, with

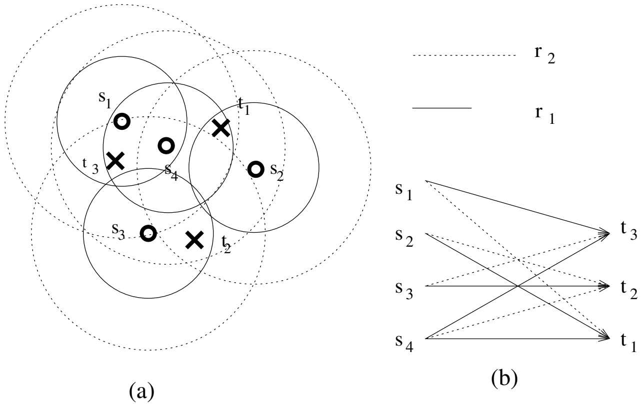


Fig. 1. Example with three targets $T = \{t_1, t_2, t_3\}$ and four sensors $S = \{s_1, s_2, s_3, s_4\}$

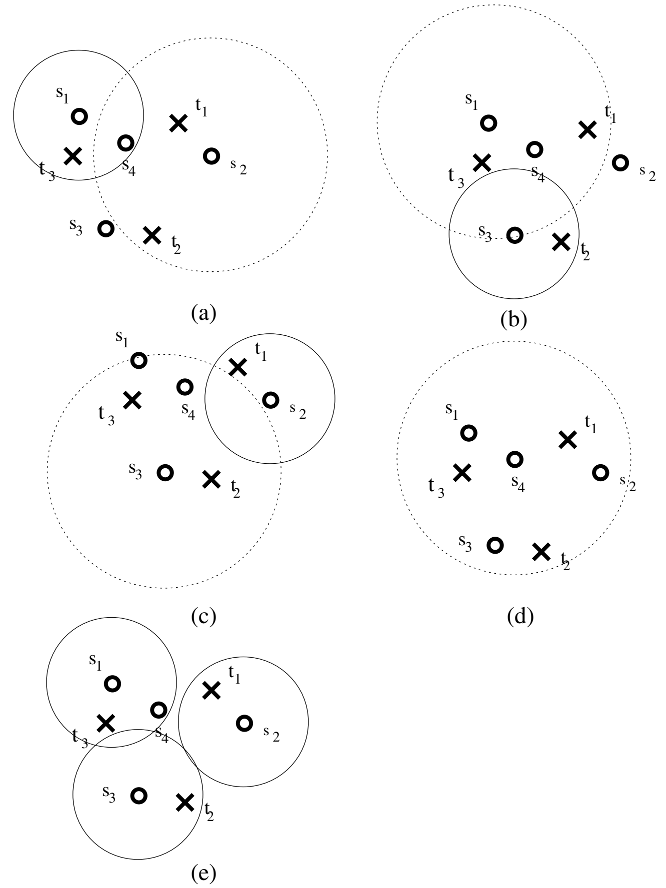


Fig. 2. Five set covers: $C_1 = \{(s_1, r_1), (s_2, r_2)\}$, $C_2 = \{(s_1, r_2), (s_3, r_1)\}$, $C_3 = \{(s_2, r_1), (s_3, r_2)\}$, $C_4 = \{(s_4, r_2)$, and $C_5 = \{(s_1, r_1), (s_2, r_1), (s_3, r_1)\}$

a radius equal to the sensing range. We use a solid line to denote range r_1 and a dotted line for range r_2. The coverage relationships between sensors and targets are also illustrated in Figure 1 (b): $(s_1, r_1) = \{t_3\}$, $(s_1, r_2) = \{t_1, t_3\}$, $(s_2, r_1) = \{t_2\}$, $(s_2, r_2) = \{t_1, t_2\}$, $(s_3, r_1) = \{t_2\}$, $(s_3, r_2) = \{t_2, t_3\}$, $(s_4, r_1) = \{t_1, t_3\}$ and $(s_4, r_2) = \{t_1, t_2, t_3\}$. The dotted lines in Figure 1 (b) show the additional targets covered by increasing the sensing range from r_1 to r_2. Note that a circular sensing area is not a requirement for our solution; we are just concerned with identifying which sensors cover each target.

In this paper, a sensor can be part of more than one cover set. Let us consider for this example $E = 2$, $e_1 = 0.5$, and $e_2 = 1$. Each set cover is active for a unit time of 1. One solution for the AR-SC problem uses the set covers illustrated in the Figure 2. This solution has five different set covers, and maximum lifetime 6, obtained for example with the following sequence of set covers: C_1, C_2, C_3, C_4, C_5, and C_4. After this sequence, the residual energy of each sensor becomes zero.

If sensor nodes do not have adjustable sensing ranges, then we obtain a lifetime 4 for a sensing range equal to r_2. Sensors can be organized in two distinct set covers, such as $\{(s_1, r_2), (s_2, r_2)\}$ and $\{(s_4, r_2)\}$, and each can be active twice. The number of times a set cover is active depends on the residual energy values. Therefore, this example shows a 50% lifetime increase when using adjustable sensing ranges.

IV. Solutions for the AR-SC Problem

In this section we present three heuristics for solving the AR-SC problem. In section IV-A we formulate the problem using integer programming and then solve it using *relaxation* and *rounding* techniques. In section IV-B we propose a greedy heuristic, where both centralized and distributed (localized) solutions are given for computing the set covers.

The centralized heuristics are executed at the BS. Once the sensors are deployed, they send their coordination to the BS. The BS computes and broadcasts back the sensor schedules. In the distributed and localized algorithm, each sensor node determines its schedule based on communication with one-hop neighbors.

A. Integer Programming based Heuristic

In this subsection we first formulate the AR-SC problem using integer programming in section IV-A.1 and then present the LP-based heuristic in section IV-A.2.

1. Integer Programming Formulation of the AR-SC Problem

Given:

- N sensor nodes s_1,..., s_N
- M targets t_1, t_2,..., t_M
- P sensing ranges r_1, r_2,..., r_P and the corresponding energy consumption e_1, e_2,..., e_P
- initial sensor energy E
- the coefficients showing the relationship between sensor, radius and target: $a_{ipj} = 1$ if sensor s_i with radius r_p covers the target t_j.

For simplicity, we use the following notations:

- i: i^{th} sensor, when used as index
- j: j^{th} target, when used as index
- p: p^{th} sensing range, when used as index
- k: k^{th} cover, when used as index

Variables:

- c_k, boolean variable, for $k = 1..K$; $c_k = 1$ if this subset is a set cover, otherwise $c_k = 0$.
- x_{ikp}, boolean variable, for $i = 1..N$, $k = 1..K$, $p = 1..P$; $x_{ikp} = 1$ if sensor i with range r_p is in cover k, otherwise $x_{ikp} = 0$.

Maximize $c_1 + ... + c_K$
subject to

$\sum_{k=1}^{K}(\sum_{p=1}^{P} x_{ikp}e_p) \leq E$	for all $i = 1..N$
$\sum_{p=1}^{P} x_{ikp} \leq c_k$	for all $i = 1..N$, $k = 1..K$
$\sum_{i=1}^{N}(\sum_{p=1}^{P} x_{ikp} * a_{ipj}) \geq c_k$	for all $k = 1..K$, $j = 1..M$
$x_{ikp} \in \{0, 1\}$ and $c_k \in \{0, 1\}$	

Remarks:

1) K represents an upper bound for the number of covers
2) The first constraint, $\sum_{j=1}^{K}(\sum_{p=1}^{P} x_{ikp}e_p) \leq E$ for any $i = 1..N$, guarantees that the energy consumed by each sensor i is less than or equal to E, which is the starting energy of each sensor.
3) The second constraint, $\sum_{p=1}^{P} x_{ikp} \leq c_k$ for any $i = 1..N$ and $k = 1..K$, assures that, if sensor i is part of the cover k then exactly one of its P sensing ranges are set.
4) The third constraint, $\sum_{i=1}^{N}(\sum_{p=1}^{P} x_{ikp} * a_{ipj}) \geq c_k$ for any $k = 1..K$ and $j = 1..M$, guarantees that each target t_j is covered by each set c_k.

2. *LP-based Heuristic*

In this subsection we propose a heuristic to solve the AR-SC problem. In section IV-A.1 we presented the Integer Programming (IP) based formulation. Since IP is NP-hard, we propose to use a relaxation and rounding mechanism. We first *relax* the IP to Linear Programming (LP), solve the LP in polynomial time, and then *round* the solutions in order to get a feasible solution for the IP.
Relaxed Linear Programming:

Maximize $c_1 + ... + c_K$
subject to

$\sum_{k=1}^{K}(\sum_{p=1}^{P} x_{ikp}e_p) \leq E$	for all $i = 1..N$
$\sum_{p=1}^{P} x_{ikp} \leq c_k$	for all $i = 1..N$, $k = 1..K$
$\sum_{i=1}^{N}(\sum_{p=1}^{P} x_{ikp} * a_{ipj}) \geq c_k$	for all $k = 1..K$, $j = 1..M$
$0 \leq x_{ikp} \leq 1$	for all $i = 1..N, k = 1..K$, and $p = 1..P$
$0 \leq c_k \leq 1$	for all $k = 1..K$

LP-based Heuristic

1: solve the LP and get the optimal solution $\bar{x}_{ikp}$ and $\bar{c}_k$
2: set $\bar{x}'_{ikp} = 0$ and $\bar{c}'_k = 0$ for all $i = 1..N, k = 1..K, p = 1..P$
3: sort $\bar{c}_k$ in nonincreasing order $\bar{c}_1, \bar{c}_2, ..., \bar{c}_K$
4: **for all** variable $\bar{c}_k$ taken from the list in nonincreasing order **do**
5:   **if** $\bar{c}_k > 0$ **then**
6:     $/*$ try to build a set cover if $\bar{c}_k > 0$ $*/$
7:     sort $\bar{x}_{ikp}$, $i = 1..N$, $p = 1..P$ in nonincreasing order
8:     **for all** $\bar{x}_{ikp}$ **do**
9:       **if** $\bar{x}_{ikp}$ covers new targets and sensor i has at least e_p energy at the beginning of setting up the cover $\bar{c}'_k$ **then**
10:         set up the range of sensor i to r_p, $\bar{x}'_{ikp} = 1$
11:       **else**
12:         $\bar{x}'_{ikp} = 0$
13:       **end if**
14:     **end for**
15:     **if** all targets are covered by $\bar{x}'_{ikp}$ having value 1 **then**
16:       $/*$ we formed a valid set cover $*/$
17:       set $\bar{c}'_k = 1$
18:       update residual energy of any sensor i with range r_p in $\bar{c}'_k$: $E_i = E_i - e_p$
19:     **else**
20:       set $\bar{c}'_k = 0$ and reset $\bar{x}'_{ikp} = 0$ for any $i = 1..N$ and $p = 1..P$
21:     **end if**
22:   **end if**
23: **end for**
24: return the total number of set covers $\sum_{k=1}^{K} \bar{c}'_k$

The heuristic starts in line 1 by solving the relaxed LP that outputs the optimal solution $\bar{x}_{ikp}$ and $\bar{c}_k$. We round this solution in order to get a feasible solution $\bar{x}'_{ikp}$ and $\bar{c}'_k$ for the IP. We use a greedy approach, by giving priority to the set covers with a larger $\bar{c}_k$. When adding sensors to a cover $\bar{c}'_k$, priority is given to the sensors with larger $\bar{x}_{ikp}$. We sort values $\bar{c}_k$ in the nonincreasing order. In lines 8..14, we add sensors to the current set cover k, by adding first the sensors with higher $\bar{x}'_{ikp}$ values. If, later, the same sensor with a larger range is encountered, the new range setting is used if new targets are covered and if the sensor has sufficient energy resources for this setting. If all the targets are covered by the selected sensors in this set cover, then we set $\bar{c}'_k = 1$. Otherwise, forming the current set cover was unsuccessful, $\bar{c}'_k = 0$, and all of set k's members are removed ($\bar{x}'_{ikp} = 0$ for any $i = 1..N$ and $p = 1..P$).

The complexity of this algorithm is dominated by the linear programming solver. The best performance is $O(n^3)$ using Ye's algorithm [15], where n is the number of variables. In our case $n = K(1 + NP)$, where P usually a small number.

B. *Greedy based Heuristics*

In this subsection we propose two greedy solutions for the AR-SC problem. The centralized solution is given in subsection IV-B.1 followed by a distributed and localized solution in subsection IV-B.2.

1. *Centralized Greedy Heuristic*

In this subsection we present a centralized greedy heuristic. We use the following notations:

- T_{ip}: the set of uncovered targets within the sensing range r_p of sensor i.

- B_{ip}: the contribution of sensor i with range r_p. $B_{ip} = |T_{ip}|/e_p$.
- ΔB_{ip}: the incremental contribution of the sensor i when its sensing range is increased to r_p. $\Delta B_{ip} = \Delta T_{ip}/\Delta e_p$, where $\Delta T_{ip} = |T_{ip}| - |T_{iq}|$ and $\Delta e_p = e_p - e_q$. The range r_q is the current sensing range of the sensor i, thus $r_p > r_q$. Initially, all the sensors have assigned a sensing range $r_0 = 0$ and the corresponding energy is $e_0 = 0$.
- C_k: the set of sensors in the kth cover.
- $\overline{T}_{C_k}$: the set of targets uncovered by the set C_k.

The algorithm selects sensors in a greedy fashion, based on their contribution values. A contribution parameter B_{ip} is associated with each (sensor, range) pair. For brevity, in cases of no ambiguity, we write (i, p) instead of (s_i, r_p). Intuitively, a sensor that covers more targets per unit of energy should have higher priority in being selected in a sensor cover. We are using the incremental contribution parameter ΔB_{ip}, defined at the beginning of this subsection, as the selection decision parameter.

In our algorithm, we are concerned not only with selecting the sensors of each set cover, but also with determining their sensing ranges. Intuitively, a smaller sensing range is preferable as long as the target coverage objective is met, since energy resources are conserved, allowing the sensor to be operational longer.

Our algorithm repeatedly constructs set covers, as long as each target is covered by at least one sensor with enough energy resources. In forming a set cover, sensors are selected repeatedly, giving priority to the sensors with highest contribution. We assume that initially all the sensors have been assigned the range $r_0 = 0$. If a sensor i is selected based on its contribution ΔB_{ip}, its sensing range is increased to r_p. Once the set cover is formed (e.g. all targets are covered by the selected set of sensors), the sensors with a sensing range greater than zero form the set of active sensors, while all other sensors with sensing range r_0 will be in sleep mode.

Assume that a sensor (i, p) with the highest contribution ΔB_{il} is selected to be added to the current set cover. Then the sensor i updates its sensing range from r_p to r_l. For each sensor node s_x that covers at least one target in T_{il}, we update $T_{xu} = T_{xu} - T_{il}$ and ΔB_{xu} for any range r_u greater than the current sensing range of s_x. Note that although there are P sensing ranges for each sensor, we maintain contribution values only for those sensing ranges for which sufficient residual energy is available. For example, if the residual energy E_x of the sensor s_x satisfies the relation $e_q \leq E_x < e_{q+1}$, then we consider only the contributions ΔB_{xu} for $u \leq q$.

We present next the **Centralized Greedy Algorithm** that repeatedly constructs set covers as long as each target is covered by at least one sensor node with sufficient residual energy.

Centralized Greedy Algorithm

1: set the residual energy of each sensor s_i to E, $E_i = E$
2: assign to each sensor s_i a range $r_0 = 0$ having the corresponding energy $e_0 = 0$
3: $k = 0$
4: **while** each target is covered by at least on sensor (i, p) and $E_i > e_p$ **do**
5:   /* a new set cover will be formed */
6:   $k = k + 1$
7:   $\overline{T}_{C_k} = \{t_j | j = 1..m\}$
8:   for each sensor s_i compute ΔB_{ip} and T_{ip}, for all sensing ranges that can be set up with the current residual energy
9:   **while** $\overline{T}_{C_k} \neq \emptyset$ **do**
10:     /* more targets have to be covered */
11:     select the sensor (i, p) with the highest contribution value ΔB_{il}
12:     increase sensor's s_i sensing range from r_p to r_l
13:     $\overline{T}_{C_k} = \overline{T}_{C_k} - T_{il}$
14:     **for all** (x, u) such that $T_{xu} \cap T_{il} \neq \emptyset$ **do**
15:       /* update the uncovered target set and the incremental contribution */
16:       update $T_{xu} = T_{xu} - T_{il}$
17:       update $\Delta B_{xu} = \Delta T_{xu}/\Delta e_u$
18:     **end for**
19:   **end while**
20:   **for all** $(i, p) \in C_k$ **do**
21:     update the residual energy of sensor s_i, $E_i = E_i - e_p$
22:   **end for**
23: **end while**
24: output the number of set covers k

The complexity of Centralized Greedy Algorithm is $O(MN^2P\frac{E}{e_1})$. The number of iterations of the while loop (lines 4..23) is upper-bounded by $N\frac{E}{e_1}$, corresponding to the case when all the targets are covered by all sensors with range r_1. The complexity of the inner while loop (lines 9..19) is upperbounded by MNP.

2. Distributed and Localized Heuristic

In this subsection, we extend the algorithm introduced in subsection IV-B.1 to a distributed and localized version. We use the notations introduced in the previous subsection. By "distributed and localized" we refer to a decision process at each node that makes use of only information for a neighborhood within a constant number of hops. A distributed and localized algorithm is desirable in wireless sensor networks since it adapts better to dynamic and large topologies.

The distributed greedy algorithm runs in rounds. Each round begins with an initialization phase, where sensors decide whether they will be in an active or sleep mode during the current round. The initialization phases takes W time, where W is far less than the duration of a round. Each sensor maintains a waiting time, after which it decides its status (sleep or active) and its sensing range, and then it broadcasts the list of targets it covers to its one-hop neighbors. The waiting time of each sensor s_i depends on s_i's contribution, and is set up initially to $W_i = (1 - \frac{B_{iP}}{B_{\max}}) \times W$ where $B_{\max}$ is the largest possible contribution, defined as $B_{max} = M/e_1$, where M is the number of targets.

The waiting time can change during the initialization phase, when broadcast messages are received from neighbors. If a sensor s_i receives a broadcast message from one of its neighbors, then s_i updates the set of uncovered targets T_{iP} and sets up its sensing range to the smallest value r_u needed to cover this set of targets. The sensor contribution value is also updated to B_{iu}. If all s_i's targets are already covered by its neighbors, then s_i sets up its sensing range to $r_0 = 0$. The waiting time W_i of the sensor s_i is also updated to $(1 - \frac{B_{iu}}{B_{\max}}) \times W$. At the end of its waiting time, a sensor broadcasts its status (active or sleep) as well as the list of targets it covers. If its sensing range is r_0 then this sensor node will be in the sleep mode, otherwise it will be active during this round.

As different sensors have different waiting times, this serializes the sensors' broadcasts in their local neighborhood and gives priority to the sensors with higher contribution. These sensors decide their status and broadcast their target coverage information first. In this algorithm we use a discrete time window, where d is the length of the time slot. Thus, the time window W has $\frac{W}{d}$ time units. If the waiting times of two sensor s_i and s_j are too close, i.e. $|W_i - W_j| < d$, then the sensors that are neighbors to both s_i and s_j cannot tell from whom the message was received, thus they will not update their uncovered target set.

We assume sensor nodes are synchronized and the protocol starts by having the base station (BS) broadcast a start message. If, after the initialization phase, a sensor s_i cannot cover one of the targets in the set T_{iP} and its waiting time reached the value zero, then s_i sends this failure information to BS. In our algorithm, we measure the network lifetime as the time until BS detects the first failure.

Next we present the **Distributed Greedy Initialization**, that is run by each sensor s_i, $i = 1..N$ during the initialization phase:

Distributed Greedy Initialization (s_i)

compute the waiting time W_i and start timer t
while $t \le W_i$ and $T_{iP} \neq \emptyset$ **do**
  if message from neighbor sensor is received **then**
    update T_{iP} and set-up the sensing range to the smallest value r_u needed to cover T_{iP}
    if $T_{iP} == 0$ **then**
      set s_i's sensing range to r_0
      break
    end if
    update s_i's contribution to B_{iu}
    update the waiting time W_i to $(1 - \frac{B_{iu}}{B_{\max}}) \times W$
  end if
end while
/* assume s_i's sensing range was set up to r_u */
if $r_u == r_0$ **then**
  s_i broadcasts its sleep state decision
  return
end if
if $E_i < e_u$ **then**
  s_i reports failure to BS, indicating the targets it cannot cover due to the energy constraints
end if
s_i broadcasts information about the set of targets T_{iu} it will monitor during this round
return

The complexity of the Distributed Greedy Initialization procedure is $O(\frac{W}{d}NMP)$. This corresponds to the case when s_i receives messages from N neighbors, each d time. The updates for each message take $O(MP)$.

V. Simulation Results

In this section, we evaluate the performance of LP-based and greedy-based heuristics. We simulate a stationary network with sensor nodes and targets randomly located in a $100m \times 100m$ area. We assume sensors are homogeneous and initially have the same energy. In the simulation, we consider the following tunable parameters:

- N the number of sensor nodes. In our experiments we vary N between 25 and 250.
- M the number of targets to be covered. It varies between 5 to 50.
- P sensing ranges $r_1, r_2,...,r_P$. We vary P between 1 and 6, and the sensing range values between $10m$ and $60m$.
- Energy consumption model $e_p(r_p)$. We evaluate network lifetime under linear ($e_p = \Theta(r_p)$) and quadratic ($e_p = \Theta(r_p^2)$) energy consumption models.
- Time slot d in the distributed greedy heuristic. d shows the impact of the transfer delay on the performance of the distributed greedy heuristic. We vary d between 0.2 and 0.75.

In the first experiment in Figure 3, we compare the network lifetime computed by LP-based, centralized greedy and distributed greedy heuristics when we vary the number of sensors. We consider 10 targets randomly deployed, and we vary the number of sensors between 25 and 100 with an increment of 5. Each sensor has two adjustable sensing ranges, $30m$ and $60m$. The energy consumption model is linear.

Network lifetime results returned by the 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.

In the second experiment in Figure 4, we study the impact of the number of adjustable sensing ranges on network lifetime. We consider 40 targets randomly distributed and we vary the number of sensors between 120 and 250 with an increment of 10. We let the largest sensing range equal to $60m$ for all cases. We compare the network lifetime when sensors support up to 6 sensing range adjustments: $r_1 = 60m$, $r_2 = 50m$, $r_3 = 40m$, $r_4 = 30m$, $r_5 = 20m$, and $r_6 = 10m$. A case with P sensing ranges, where $P = 1..6$, allows each sensor node to adjust P sensing ranges r_1, r_2,..., r_P. Note that $P = 1$ is the case when all sensor nodes have a fixed sensing range with value $60m$.

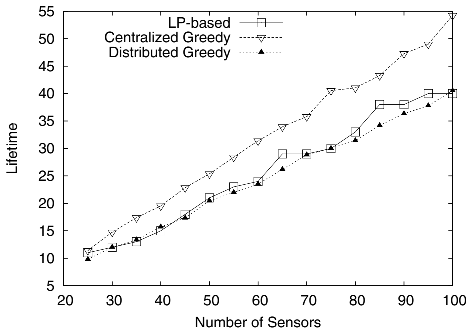


Fig. 3. Network lifetime with number of sensors

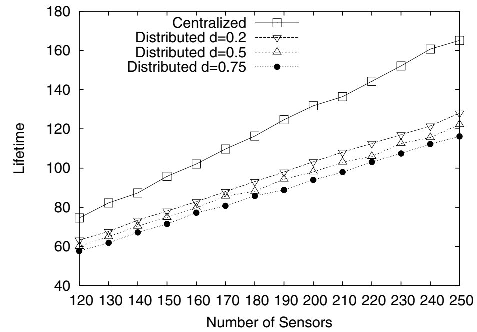


Fig. 5. Network lifetime for different values of the time slot d

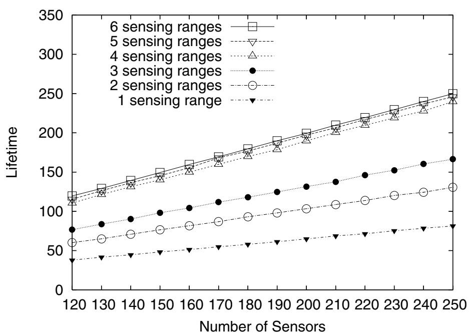


Fig. 4. Network lifetime for different sensing range values

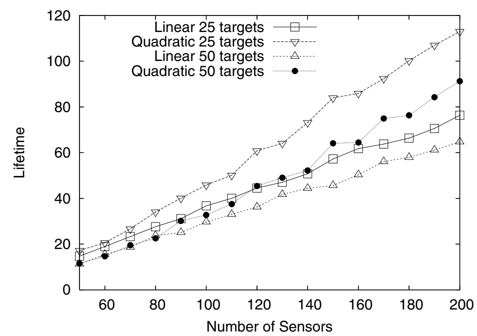


Fig. 6. Linear and quadratic energy models

Simulation results indicate that adjustable sensing ranges have great impact on network lifetime, especially when increasing P from 1 to 2, 3 or 4. When increasing P from 4 sensing ranges to 5 or 6 sensing ranges, the network lifetime increases at a lower rate. From $P = 1$ to $P = 2$, the network lifetime increases with more than 20 set covers on average. This simulation results also justify the contribution of this paper, showing that adjustable sensing ranges can greatly contribute to increasing the network lifetime.

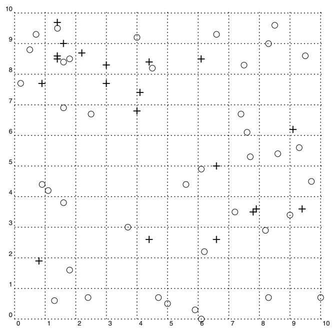
In Figure 5 we compare the network lifetime produced by centralized and distributed greedy algorithms. We measure the network lifetime when the number of sensors varies between 120 and 250 with an increment of 10 and the number of targets is 50. Each sensor has 6 sensing ranges with value $10m$, $20m$, $30m$, $40m$, $50m$, and $60m$. The energy consumption model is linear. We change the length of the time slot d in the distributed greedy algorithm to $d = 0.2$, 0.5, and 0.75.

Network lifetime produced by the centralized algorithm is longer than that produced by the distributed algorithm. This happens because the centralized greedy heuristic has global information and can always select the sensor with the greatest contribution. Also, if there is a tie between the contribution of different sensors, the centralized greedy heuristic can break the tie arbitrarily, without any additional cost.

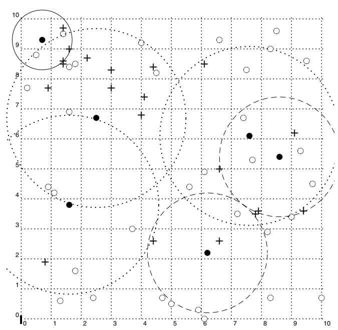
In the distributed heuristic, breaking a tie is at the expense of backoff time, and there is also no guarantee of no conflict. A conflict occurs when sensors broadcast at the same time based on their contributions. Then, there might be sensors that work instead of going to the sleep state, even if the targets within their sensing range are already covered. As illustrated in Figure 5, the transfer delay also affects the network lifetime. The longer the transfer delay is, the smaller the lifetime.

In Figure 6 we study the impact of two energy models on the network lifetime computed by the distributed greedy heuristic when we vary the number of sensors between 40 and 200, and the number of targets is 25 or 50. Each sensor has $P = 3$ sensing ranges with values $10m, 20m,$ and $30m$. The two energy models are the linear model $e_p = c_1 * r_p$, and quadratic model $e_p = c_2 * r_p^2$. In this experiment we defined constants $c_1 = E/2(\sum_{r=1}^{P} r_p)$ and $c_2 = E/2(\sum_{r=1}^{P} r_p^2)$, where $E = 10$ is the sensor starting energy. For both energy models, the simulation results are consistent and indicate that network lifetime increases with the number of sensors and decreases as more targets have to be monitored.

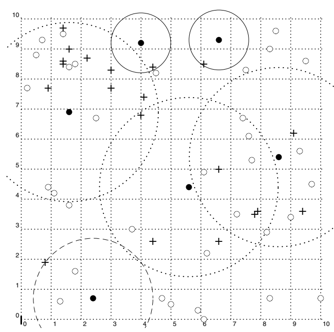
In Figure 7, we give an example of coverage produced by centralized and distributed heuristics. We assume a $100m \times 100m$ area, with 40 sensors and 20 targets. Each sensor has $P = 3$ sensing ranges with values $10m, 20m,$ and $30m$. We use solid lines to represent $r_1 = 10m$, dashed lines for $r_2 = 20m$ and dotted lines for $r_3 = 30m$. We used a linear energy model. The first graph represents the sensors' and targets' random deployment. Figure 7 (b) and (c) show set covers produced by the centralized and distributed greedy heuristics. The active sensors are blackened and the line type indicates the sensing range value.



(a)



(b)



(c)

Fig. 7. Set covers example, where $''\circ''$ and $''\bullet''$ are inactive (sleeping) and active sensors, respectively and $''+''$ are targets. (a) Sensors and targets deployment. (b) Set cover produced by the centralized greedy heuristic. (c) Set cover produced by the distributed greedy heuristic.

The simulation results can be summarized as follows:

- Given the number of targets and the sensing range values, the network lifetime output by our heuristics increases with the number of sensors deployed.
- Network lifetime increases with the number of adjustable sensing ranges. Greater impact is observed when increasing P from 1 to small values ($P \leq 5$). After that the increase in the network lifetime converges at a slower rate.
- Even if the two centralized solutions perform better than the distributed solution (longer network lifetime), using a distributed and localized heuristic is an important characteristic for a solution in wireless sensor networks environment.
- Transfer delay used for internode communication in the distributed greedy heuristic affects the network lifetime. Smaller transfer delays results in longer network lifetime.
- For both linear and quadratic energy models, network lifetime increases with the number of sensors and decreases as more targets have to be covered.

VI. Conclusions

In this paper we proposed scheduling models for the target coverage problem for wireless sensor networks with adjustable sensing range. The problem addressed in this paper is to determine maximum network lifetime when all targets are covered and sensor energy resources are constrained.

In this paper we introduced the mathematical model, proposed efficient heuristics (both centralized and distributed and localized) using integer programming formulation and greedy approaches, and verified our approaches through simulation.

In our future work we will integrate the sensor network connectivity requirement. Maintaining connectivity among the selected sensors has an advantage in facilitating the exchange of information between sensors and the base station.

Acknowledgment

This work is supported in part by NSF grants CCR 0329741, CNS 0434533, CNS 0422762, and EIA 0130806.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, A Survey on Sensor Networks, IEEE Communications Magazine, pp 102-114, Aug. 2002.

[2] M. Cardei, D.-Z. Du, Improving Wireless Sensor Network Lifetime through Power Aware Organization, ACM Wireless Networks, Vol 11, No 3, May 2005.

[3] M. Cardei, M. Thai, Y. Li, and W. Wu, Energy-Efficient Target Coverage in Wireless Sensor Networks, IEEE INFOCOM 2005, Mar. 2005.

[4] 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.

[5] J. Carle and D. Simplot, Energy Efficient Area Monitoring by Sensor Networks, IEEE Computer, Vol 37, No 2 (2004) 40-46.

[6] M. Cheng, L. Ruan, and W. Wu, Achieving Minimum Coverage Breach under Bandwidth Constraints in Wireless Sensor Networks, IEEE INFOCOM 2005, Mar. 2005.

[7] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, W. H. Freeman, 1979.

[8] C.-F. Huang and Y.-C. Tseng, The Coverage Problem in a Wireless Sensor Network, ACM MobiCom'03, pp 115-121, Sep. 2003.

[9] S. Megerian, M. Potkonjak, Wireless Sensor Networks, Wiley Encyclopedia of Telecommunications, Editor: John G. Proakis, Dec. 2002.

[10] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, Coverage Problems in Wireless Ad-Hoc Sensor Networks, IEEE Infocom 3, pp 1380-1387, 2001.

[11] Photoelectric Sensors, http://www.schneider-electric.ca/www/en/products/sensors2000/html/osiris.htm

[12] 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.

[13] 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.

[14] J. Wu and S. Yang, Coverage and Connectivity in Sensor Networks with Adjustable Ranges, *International Workshop on Mobile and Wireless Networking (MWN)*, Aug. 2004.

[15] Y. Ye, An $o(n^3 l)$ Potential Reduction Algorithm for Linear Programming, *Mathematical Programming*, Vol 50, pp 239-258, 1991.

[16] 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.