2014

On complete target coverage in wireless sensor networks with random recharging rates

Changlin Yang
University of Wollongong, cy116@uowmail.edu.au

Kwan-Wu Chin
University of Wollongong, kwanwu@uow.edu.au

Publication Details

C. Yang & K. Chin, "On complete target coverage in wireless sensor networks with random recharging rates," IEEE Wireless Communications Letters, vol. 4, (1) pp. 50-53, 2015.

On complete target coverage in wireless sensor networks with random recharging rates

Abstract

A fundamental problem inWireless Sensor Networks (WSNs) is determining the minimum active time of a set of sensor nodes such that they monitor all targets for the maximum time. However, existing solutions do not consider random recharging rates and staled battery level information, resulting in an activation schedule that is not realizable by sensor nodes. Henceforth, we propose a Stochastic Programming (SP) based approach that considers random battery levels. Experimental results show our SP approach achieves 80% of the theoretically achievable coverage lifetime.

Keywords

Wireless Sensor Networks, Target Coverage, Energy Harvesting, Stochastic Programming

Disciplines

Engineering | Science and Technology Studies

Publication Details

C. Yang & K. Chin, "On complete target coverage in wireless sensor networks with random recharging rates," IEEE Wireless Communications Letters, vol. 4, (1) pp. 50-53, 2015.

On Complete Target Coverage in Wireless Sensor Networks with Random Recharging Rates

Changlin Yang and Kwan-Wu Chin [*][†]

Abstract

A fundamental problem in Wireless Sensor Networks (WSNs) is determining the set of sensor nodes and their active duration such that all targets are monitored by at least one sensor node at all times. Moreover, we want these targets to be monitored for the longest possible time. However, existing solutions for this problem do not consider random recharging rates and staled battery level information, resulting in an activation schedule that is not realizable by sensor nodes. Henceforth, we propose a Stochastic Programming (SP) based approach that considers random battery levels. Experimental results show our SP approach achieves 80% of the theoretically achievable coverage lifetime.

Wireless Sensor Networks, Target Coverage, Energy Harvesting, Stochastic Programming

1 Introduction

Energy harvesting Wireless Sensor Networks (WSNs) have received considerable attention in the past few years. These WSNs are able to operate perpetually if they have energy neutral operation. Consequently, they are attractive for use in many applications. In this respect, our aim is to address a fundamental problem in surveillance applications. Namely, the Maximum Lifetime Coverage with Energy Harvesting (MLCEH) problem, which calls for a solution that determines an activation schedule for sensor nodes such that *all* targets are monitored by at least one sensor node for maximum time. Here, *coverage lifetime* is defined as the operating start time of a WSN until a target fails to be monitored by a sensor node.

Past works are mostly focused on maximizing events detection probability at target locations. For example, the algorithm proposed in [1] aims to derive a duty cycle that maximizes events detection. In [2] and [3], the authors use the probability of an event occurring at a target's location. On the other hand, Kar et al. [4][5] propose to dynamically activate sensor nodes to maintain

[*] Authors Yang and Chin are with the School of Electrical, Computer, and Telecommunication Engineering, University of Wollongong, NSW, Australia, e-mail: cy116@uowmail.edu.au, kwanwu@uow.edu.au

[†]

a certain coverage level. Critically, these solutions do not consider *complete* target coverage. In this respect, only the authors of [6] have addressed the MLCEH problem by proposing two algorithms: one uses a Linear Program (LP) and the other is a greedy algorithm that maximizes coverage utility. The main idea is to provide sensor nodes with sufficient time to recharge. Although the proposed greedy algorithm preferentially activates nodes with a full battery so that they do not forego any recharging opportunities, the LP solution, however, does not have such consideration.

A key observation, which motivated our research, is that the solutions in [6] assume the sink, where the coverage algorithm is run, knows the exact battery level information of all nodes. This assumption, however, is not necessarily valid. As shown in [5], sensor nodes have random recharging rates. This means they will have varying battery levels over time. Consequently, upon receiving an activation schedule, a node may have insufficient energy to implement the schedule. Conversely, a sensor node may experience a temporary but "high" recharging rate that allows it to recharge fully. In this case, we need the node to expend its energy in order to take advantage of future recharging opportunities that in turn help prolong coverage lifetime. We remark that more accurate information can be obtained if nodes coordinate their updates and send them frequently to the sink. This, however, is at the expense of precious energy, especially by nodes near the sink, which could have been used for monitoring targets. Hence, a key research question is whether we can conserve energy by reducing the frequency of updates whilst accounting for the resulting increase in uncertainty.

To this end, this paper contains a number of contributions. First, to account for battery level uncertainty, we propose a stochastic program (SP) based Uncertain Maximum Lifetime Coverage algorithm; also called SP-UMLC (see Section 4). We remark that the problem (see Section 3) is new. We solve the problem via a two-stage SP with the goal of minimizing the activation time of sensor nodes. We then solve the SP in the Sample Average Approximation (SAA) framework due to the exponential number of scenarios [7]. Secondly, we modify the LP-MLCEH algorithm of [6] to incorporate a penalty for nodes with a high battery level; the new formulation is denoted as LP-MLCEH-P. In experiments where LP-MLCEH-P uses accurate battery level information, a theoretical benchmark that requires the sink to take a snapshot of the current battery level at each node at a time point, SP-UMLC achieves 80% of the coverage lifetime attained by LP-MLCEH-P.

2 Network Model

We model a WSN as a sensor-target bipartite graph (S, Z, E, W). Here, S is the set of sensors, Z is the set of targets, and E is the set of edges connecting a sensor $s_i \in S$ to one or more targets in Z. Note, we will use s_i and z_j to index sensors and targets, where $i = 1 \ldots |S|$ and $j = 1 \ldots |Z|$. Let $Z(s_i)$ and $S(z_j)$ be a function that returns the set of targets covered by sensor s_i and the set of sensors covering target z_j respectively. We divide time into intervals, indexed by t. We refer to each interval as a *time slot*. Define $C_t \subseteq S$ to be the set of nodes providing complete coverage at time slot t. With a slight abuse of notation, let $Z(C_t)$ be a function that returns the set of targets covered by sensors C_t. Let $\phi(C_t, z_j)$ be a coverage mapping function that returns one if target z_j is covered by C_t, otherwise it returns zero. Also, $E(C_t)$ is an indicator function that returns one if the residual energy of all sensors in C_t is sufficient to cover time slot t. Let E_i (Joules) denote the level of sensor node s_i's rechargeable battery, which is bounded by B_{max}. To safeguard against imprecise schedule, explained later, each sensor node i has a fixed non-rechargeable battery reserve, denoted as R_i. This reserve is only drawn upon if there is a short fall in energy.

In the following sections, we will refer to E_i^t as the *current* battery level of sensor node s_i, and a subsequent update as E_i^{t+1}. We model the uncertainty in battery level as follows. Let u represent the variation in recharging rates, and $\gamma(u)$ be a random value generated from a standard normal distribution in the range of $1-u$ to $1+u$. At E_i^{t+1}, the battery level of node i is

$$E_i^{t+1} = E_i^t - E_i^c x_i^t + E_i^r(1-x_i^t)(1+\gamma(u)) \quad (1)$$

where E_i^r is the recharging rate of sensor node s_i, which is governed by a known probability distribution. The term E_i^c and x_i^t refer to s_i's consumption rate when active and its activation time at time slot t. We assume that sensor nodes are able to sense omni-directionally and thus monitor one or more targets with equal energy consumption rate. In subsequent sections, in terms of battery level information, we will refer to E^{t+1} as *accurate*, which is the battery level at sensor nodes. The information at the sink, however, is *staled*, denoted as E^t.

3 Problem Statement

We first describe the *deterministic* version of the complete target coverage problem. The goal is to determine the maximum coverage time T, where $T \in [0, \infty]$, that satisfies the following constraints: (i) $E(C_t) = 1$, and (ii) $\phi(C_t, Z) = 1$. We remark that the problem becomes NP-hard, see [8], if the aim is to determine the minimum number of sensor nodes that covers all targets. However, our problem seeks the minimum activation time for sensor nodes such that all targets are covered, whilst affording them ample time to recharge. Mathematically, we have the following Linear Program (LP), with the objective is to minimize each sensor node i's active time to monitor a target j; i.e., x_{ij}.

$$\text{MIN} \sum_{i \in S} \sum_{j \in Z} x_{ij} \quad (2)$$

Subject to:

$$\sum_{i \in S(z_j)} x_{ij} \geq 1, \quad \forall z_j \in Z \quad (3)$$

$$\sum_{j \in Z(s_i)} x_{ij} E_i^c \leq E_i, \quad \forall i \in S, \quad (4)$$

Constraint (3) ensures each target is watched for at least one time slot. Constraint (4) ensures the total energy expenditure is within limit. Recall that each sensor node i is able to sense omni-directionally. However, term $\sum_{j \in Z(s_i)} x_{ij}$ does not take this fact into account. To this end, in our implementation, we divide the term $\sum_{j \in Z(s_i)} x_{ij}$ and $\sum_{j \in Z} x_{ij}$ by $|Z(s_i)|$ to yield the correct activation time.

Notice that a key assumption of constraint (4) is that the scheduler/sink is aware of the current energy level of each node. As mentioned in Section 1, due to random recharging rates, when sensor nodes receive their respective x_{ij} value, they may find that the computed x_{ij} value to be infeasible because the scheduler/sink used staled information.

4 The Approach

We now outline our SP based approach. We first provide a brief introduction to two-stage SP [9]. In the first stage, a decision is made based on the "current" battery level of nodes. In the second stage, actual battery levels become available, which require recourse actions to be carried out if the decision made in the first stage is inadequate; e.g., the scheduled active time exceeds a node's energy constraint, and thus it has to draw energy from its reserve as a recourse. Mathematically, in the first stage, we have,

$$\min_{x \in X} \{ g(x) := c^T x + \mathbb{E}[Q(x, \xi)] \} \tag{5}$$

Then, given the first stage decision x and random vector $\xi = (q, T, W, h)$, the second stage problem is as follows,

$$Q(x, \xi) = \min_{y} \{ q^T y \mid Tx + Wy \leq h \} \tag{6}$$

Here, the decision variable y is the recourse action to be undertaken in order to meet the budgetary constraint h. Note, the actual value and interpretation of the components in ξ, which can be fixed or random, are application specific.

In our problem, in the first stage, the scheduler first determines the set of sensor nodes and their active time based on E_i^t. The second stage uses E_i^{t+1}, which is governed by random recharging rates. Hence, we aim to minimize the expected recourse cost. In order to ensure the scheduler/sink preferentially activates sensor nodes with full battery, we add a penalty coefficient ω_i to each variable x_{ij} in the objective function. This coefficient conversely reflects the i-th node's residual energy level. For example, if sensor node i's battery is at $100\%, 90\%, \ldots, 0\%$ capacity, then ω_i will be set to $1, 2, \ldots, 10$ respectively.

We now rewrite our earlier LP formulation for the problem, see Section 3, to consider random battery levels and recharging opportunities. In the first stage, we have,

$$\text{MIN} \sum_{i \in S} \sum_{j \in Z} \omega_i x_{ij} + \mathbb{E}_\rho[Q(x_{ij})] \tag{7}$$

Subject to:

$$\sum_{i \in S(z_j)} x_{ij} \geq 1, \quad \forall z_j \in Z \tag{8}$$

$$\sum_{j \in Z(s_i)} x_{ij} E_i^c \leq E_i^{t-1}, \quad \forall i \in S, \tag{9}$$

$$\sum_{j \in Z(s_i)} x_{ij} \geq 0, \quad \forall i \in S, \tag{10}$$

The main changes are to (i) the objective function, which now considers the uncertainties caused by the varying battery levels, as described by the probability distribution ρ, and (ii) constraint (9), which reflects the sink's record of the nodes' *current* battery level.

The second stage problem, i.e., $Q(x_{ij})$, is similar. Let y_{ij} be the activation time taken as a recourse in solving the second stage problem, and also corresponds to a sensor node drawing from its battery reserve. Hence, in order to discourage its use, we add a high penalty ω' to each

y_{ij}, where $\omega' \gg 10$. Specifically,

$$Q(x_{ij}) = \text{MIN} \sum_{i \in S} \sum_{j \in Z} \omega' y_{ij} \tag{11}$$

Subject to:

$$\sum_{j \in Z(s_i)} (x_{ij} - y_{ij}) E_i^c \leq E_i^t, \quad \forall i \in S, \tag{12}$$

$$\sum_{j \in Z(s_i)} y_{ij} \leq R_i, \quad \forall i \in S, \tag{13}$$

$$\sum_{j \in Z(s_i)} y_{ij} \geq 0, \quad \forall i \in S, \tag{14}$$

Note that x_{ij} is determined by the first stage problem. The term E_i^t in constraint (12) is a realization of sensor node i's battery level at the sink. Each realization is generated from a probability distribution function. Also, the term y_{ij} models the recourse taken given x_{ij} and E_i^t. Constraint (13) ensures recourse actions are limited by nodes' battery reserve. In our experiments, if y_{ij} exceeds node i's battery reserve, then the simulation ends and we record the resulting lifetime.

The main difficulty in solving the SP problem is the number of battery levels each node has; so called 'scenarios'. Assuming b discrete battery levels for each node, then a WSN with 50 nodes has a total number of b^{50} scenarios! To this end, we apply the sample average approximation (SAA) method, which uses Monte Carlo simulation [7] to yield a sample average estimate of the expected recourse cost. In particular, we estimate $\mathbb{E}_\rho[Q(x_{ij})]$ as follows,

$$\frac{1}{N} \sum_{j=1}^{N} Q(x_{ij}, \xi^j) \tag{15}$$

where ξ^j is a generated sample represented as a vector of dimension $|S|$ with component E_i^t, and N is the total number of required samples; explained further below.

In words, SAA requires solving (11)-(14) for each sample ξ^j, with each result weighted 1/N. To ensure the second stage always has a solution, which is a precondition for applying SAA, see [7], we set y_{ij} to be unbounded.

To measure the quality of the solution generated by SAA, we employ the method developed in [7]. Specifically, given a solution $\hat{x}^*$, the optimality gap is defined as,

$$\hat{z}_{N'}(\hat{x}^*) - \bar{z}_N \tag{16}$$

We now proceed to define $\hat{x}^*$, $\hat{z}_{N'}(.)$ and $\bar{z}_N$. Let $\bar{z}_N$ denote a solution to our SP problem computed using SAA. We proceed by generating M candidate solutions, and denote the k-th objective value as $\bar{z}_N^k$ and the corresponding vector of solutions, i.e., x_{ij} by $\hat{x}^k$. The average of these M solutions is,

$$\bar{z}_N = \frac{1}{M} \sum_{m=1}^{M} z_N^m \tag{17}$$

Next, for a given solution $\hat{x}$, i.e., nodes' wake-up time, we set $\hat{z}_{N'}(\hat{x})$ as follows,

$$\hat{z}_{N'}(\hat{x}) = c^T \hat{x} + \frac{1}{N'} \sum_{j=1}^{N'} Q(\hat{x}, \xi^j) \tag{18}$$

where c is a vector of all ones, and $N' \gg N$. Lastly, $\hat{x}^*$ is defined as,

$$\hat{x}^* = \underset{x^k,\, k \in [1,M]}{\arg\min} \{\hat{z}_{N'}(x^k)\} \tag{19}$$

In our experiments, we discretize nodes' battery to 100 levels and pick a M and N value that ensures the gap, see Equation (16), is within 1% of the average objective value $\bar{z}_N$.

5 Evaluation

We study the performance of the proposed two stage SP-UMLC algorithm with different uncertainty level $\pm u$; see Equation (1). Our experiments use the parameters of WaspMote [10], which consumes 60 mW when active and 0.2 mW when in sleep mode. All sensor nodes are equipped with an Enocean ECS310 solar cell [11]. It has a conversion rate of 10% and a recharging efficiency of 50%, which is conservative as compared to other technologies [12]. In addition, we use real solar irradiance data retrieved from Southwest Solar Research Park, Phoenix, Arizona, USA [13] on the 16-th of April 2013. Hence, for each sensor node, its recharging rate is a sinusoidal function that peaks at 12 o'clock in every 24 hours period. Other parameter values are as follows: (i) battery size, 1100 mA, (ii) consumption rate, 3.6 Joules/hour, (iii) voltage, 4V, (iv) solar panel conversion rate, 10%, and (v) recharging efficiency, 50%. For the SP-UMLC algorithm, we allocate 10% of the battery capacity of sensor nodes as non-rechargeable back-up at the start of each experiment.

We compare SP-UMLC to LP-MLCEH [6], a theoretical approach that has accurate battery level information; we assume an oracle exists that could gather this information without energy cost. Also, as mentioned in Section 1, LP-MLCEH neglects recharging opportunities. We thus add a penalty to each x_{ij} in the objective function of LP-MLCEH, similar to the SP in Section 4, so that the LP solver preferentially activates nodes with a full battery. We call the revised LP as LP-MLCEH with penalty or LP-MLCEH-P. For comparison against SP-UMLC, we use LP-MLCEH-P on staled information, and label the resultant coverage lifetime as LP-MLCEH-P2.

In our experiments, sensor nodes are dispersed within a $100 \times 100\ m^2$ sensing field. All sensor nodes also have a uniform sensing range of 50 meters and a maximum 76 hours worth of energy. We then set each sensor node to have a different average recharging rate, which is reasonable as the recharging rate of sensor nodes is dependent on their location; e.g., sensor nodes obstructed by foliage will inevitably have a lower recharging rate [14]. Also, we set both the number of samples and scenarios to five, which we found sufficient to yield an optimality gap of less than 1%.

5.1 Results

We first compare the *average* coverage lifetime of LP-MLCEH-P, LP-MLCEH-P2 and SP-UMLC when uncertainty is $u = 0.1$, $u = 0.4$ and $u = 1$. We fix the number of targets to 20 and vary the number of sensor nodes from five to 15. The results are an average of 200 runs, each with a different randomly generated topology. Referring to Figure 1, the coverage lifetime of LP-MLCEH-P and SP-UMLC increases rapidly from 200 hours to more than 3000 hours. The reason is because sensor nodes have more opportunities to be in the sleep state and harvest energy. On the other hand, LP-MLCEH-

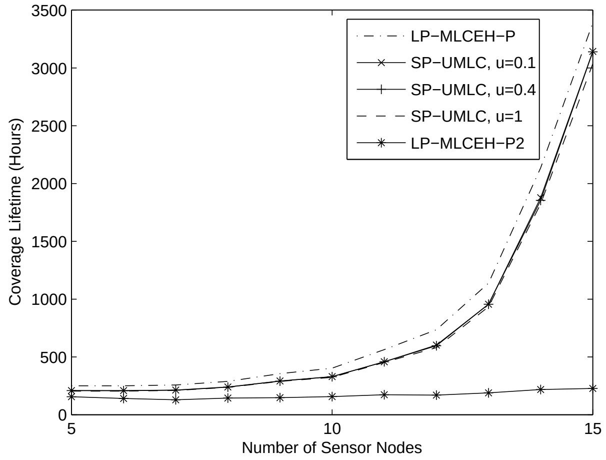

Figure 1: Sensor node density versus coverage lifetime

P2, which activates sensor nodes using staled information, has poor coverage lifetimes. Indeed, SP-UMLC outperforms LP-MLCEH-P2 and achieves 80% of the average coverage lifetime attained by LP-MLCEH-P even though it uses staled information. Another observation is that the average coverage lifetime of SP-UMLC when $u = 0.1$ and $u = 0.4$ is very close but reduces by 350 hours when uncertainty is one. This is due to the significant variation in battery levels, which leads to unnecessarily long active times, leading to energy wastage. Next, we investigate the variation in coverage lifetimes. We plot the Probability Density Function (PDF) of coverage lifetimes when the number of sensor nodes is 12; see Figure 2. The result is similar for other node numbers. We see that 90% of the recorded lifetimes are within 450 hours to 470 hours when uncertainty is 0.1. However, this percentage reduces to 20% when $u = 1$.

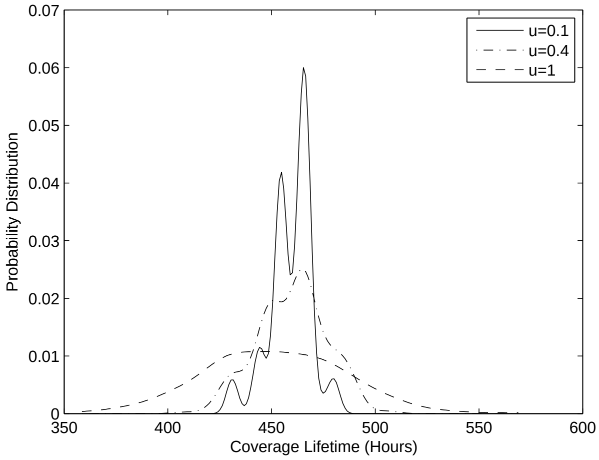

Figure 2: Coverage lifetime PDF of SP-UMLC under different uncertainties

6 Conclusion

This paper is the first to consider random recharging rates when solving the complete target coverage problem. Our stochastic programming based solution is shown to be within 80% of the theoretical coverage lifetime, and thus is a promising solution that addresses the tradeoff between uncertainties and energy consuming, frequent updates conducted to obtain accurate battery level information from all nodes.

References

[1] V. Pryyma, D. Turgut, and L. Bölöni, “Active time scheduling for rechargeable sensor networks,” *Computer Networks*, vol. 54, no. 4, pp. 631–640, 2010.

[2] Z. Ren, P. Cheng, J. Chen, D. K. Y. Yau, and Y. Sun, “Dynamic activation policies

for event capture with rechargeable sensors," in *32nd IEEE International Conference on Distributed Computing System*, (Macau, China), June 2012.

[3] N. Jaggi, K. Kar, and A. Krishnamurthy, "Rechargeable sensor activation under temporally correlated event," *Wireless Networks*, vol. 15, pp. 619–635, July 2009.

[4] T. Banerjee and A. A. Kherani, "Sensor node activation policies using partial or no information," in *IEEE WiOpt*, (Limassol, Cyprus), Apr. 2007.

[5] K. Kar and N. Jaggi, "Dynamic node activation in networks of rechargeable sensors," *IEEE/ACM ToN*, vol. 14, no. 1, pp. 15–26, 2006.

[6] C. Yang and K.-W. Chin, "Novel algorithms for complete targets coverage in energy harvesting wireless sensor networks," *IEEE Communications Letters*, vol. 18, no. 1, pp. 118–121, 2014.

[7] W.-K. Mak, D. P. Morton, and R. K. Wood, "Monte carlo bounding techniques for determining solution quality in stochastic programs," *Operations Research Letters*, vol. 24, pp. 47–56, 1999.

[8] B. Wang, *Coverage Control in Sensor Networks*, ch. Coverage Lifetime Maximization, pp. 65–95. Springer, 2010.

[9] D. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer, 1997.

[10] "Waspmote datasheet." `http://www.libelium.com/downloads/documentation/waspmote_datasheet.pdf`.

[11] "Enocean ECS 310." `http://www.enocean.com`.

[12] A. C. Valera, W.-S. Soh, and H.-P. Tan, "Energy-neutral scheduling and forwarding in environmentally-powered wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1202–1220, 2013.

[13] "Nrel: Southwest solar research park." `http://www.nrel.gov/midc/ssrp/`.

[14] M. Rahimi, H. Shah, G. Sukhatme, J. Heideman, and D. Estrin, "Studying the feasibility of energy harvesting in a mobile sensor network," in *IEEE International Conference on Robotics and Automation ICRA'03.*, vol. 1, (Taipei), pp. 19–24, IEEE, Sept 2003.