Sustainable Computing: Informatics and Systems 18 (2018) 1–13

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Maximum lifetime coverage problem with battery recovery effect☆

Norie Fu[a,b], Naonori Kakimura[c,b], Kei Kimura[e,b], Vorapong Suppakitpaisarn[d,b,*]

[a] National Institute of Informatics, Tokyo, Japan
[b] JST, ERATO Kawarabayashi Large Graph Project, Japan
[c] Keio University, Kanagawa, Japan
[d] The University of Tokyo, Tokyo, Japan
[e] Toyohashi University of Technology, Aichi, Japan

ARTICLE INFO

Article history:
Received 15 February 2017
Received in revised form 16 October 2017
Accepted 15 February 2018
Available online 17 February 2018

Keywords:
Wireless sensor networks
Maximum lifetime coverage problem (MLCP)
Approximation algorithms
Battery recovery effect

ABSTRACT

Scheduling sensors to prolong the lifetime of target coverage is one of the central problems faced in wireless sensor networks. This problem, called the maximum lifetime coverage problem (MLCP), can be formulated as a linear program with exponential size and has a polynomial-time approximation scheme (PTAS). In reality, however, sensor batteries are subject to the recovery effect, which means that the deliverable energy in a battery can replenish itself if it is left idle for a sufficient duration. Thanks to this effect, we can obtain much longer sensor lifetime if each sensor is intermittently forced to turn off for some interval. In this study, we introduce two models that extend the MLCP to incorporate the battery recovery effect. The first model, called as duty cycle model, represents the battery recovery effect in a deterministic way. The second one, called as linear recovery model, uses a probabilistic model to imitate this effect. We propose two efficient algorithms that work for both models, adapting greedy and Garg–Könemann-based algorithms for the original MLCP. In our numerical experiments, our greedy algorithm performs best in the duty cycle model, while our Garg–Könemann-based algorithm performs best in the linear recovery model. For each network, we compare the longest lifetime obtained from our algorithms with the longest lifetime obtained from algorithms for the original MLCP. As a result, we found that our lifetime is 10–40% longer.

1. Introduction

One of the most important applications of wireless sensor networks is to monitor the environment of given targets. Unlike IP networks, the amount of energy in each sensor is very limited. Hence, many methods have been proposed to optimize the monitoring under energy limitations [2–4]. In addition to energy limitations, the area that each sensor can monitor is also limited; that is, the sensor can observe only a small region of the target field. In this paper, we assume that the region is a unit circle. Also, we assume that every target can be monitored by at least one sensor.

Sensors have two modes, namely, idle and active. They consume significantly more energy in active mode while cannot observe the environment in idle mode (see [5] for example). We would like to monitor every target area by at least one active sensor as long as possible. The duration of monitoring every target area is called as the *network lifetime*. The constraint is satisfied when all the sensors are always in an active mode. In this case, the network lifetime is equal to the sensor battery lifetime. However, a small subset of the sensors is usually enough to monitor all the targets, as the number of sensors placed on the field is large compared to the number of targets. We can prolong the network lifetime by switching an appropriate subset of the sensors to the active mode for a proper length of time and then replacing it by another subset.

The problem of finding a switching method that can maximize the network lifetime is called as the *maximum lifetime coverage problem* (MLCP). MLCP has been studied extensively in the literature, where various methods are proposed to solve it. Those methods include the greedy heuristic algorithms [6], algorithms based on the general framework for packing linear programs of Garg and Könemann in [7]. The Garg and Könemann framework includes a $(1+\ln n)$-approximation algorithm for a network with n sensors in [8], a $(4+\varepsilon)$-approximation algorithm for an arbitrary small $\varepsilon>0$ in [9], a constant-factor approximation based on lin-

☆ This paper is an extended version of work published in [1].

* Corresponding author at: Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, 113-0033 Tokyo, Japan.
E-mail addresses: funorie@nii.ac.jp (N. Fu), kakimura@math.keio.ac.jp (N. Kakimura), kimura@cs.tut.ac.jp (K. Kimura), vorapong@is.s.u-tokyo.ac.jp (V. Suppakitpaisarn).

https://doi.org/10.1016/j.suscom.2018.02.007
2210-5379/

ear programming (LP) relaxation in [10], and a polynomial-time approximation scheme (PTAS) in [11].

In practical battery usage, there is a phenomenon called *battery recovery effect*, which refers to the process by which the deliverable energy in a battery can replenish itself if it is left idle for a sufficient duration. However, to the best of our knowledge, there is no work applying the recovery effect to a practical sensor network setting. This motivates us to incorporate the effect in MLCP, one of the well-studied practical problems for sensor networks.

Two mathematical models have been proposed to capture the characteristics of the battery recovery effect. The first one is the kinetic battery model [12,13], which attempts to model the details of chemical reactions and diffusion process of a battery through a set of partial differential equations. These models aim to fully capture the nonlinear dynamics of a battery. However, it is hard to include this complicated model in our coverage problem because the problem becomes finding solutions for a system of nonlinear differential equations, which is computationally hard. Another model is the stochastic battery model [14–17], which captures the battery dynamics using a probabilistic Markovian model. This model is simpler, but cannot capture battery behaviors as well as the first one can.

The battery models in this work are based on the research by Chau et al. [14]. Although their model is simpler and more analyzable than both models discussed above, it can capture several interesting battery characteristics. These include the fact that the amount of battery recovered over the whole usage cannot be larger than a specific number and the fact that the recovery in each idle period is usually small. These effects imply that we can maximize the battery recovery by periodically turning on and off our batteries for a specific and short period of time. We call such a switching strategy a *duty cycle*. The experimental results in [14] show that the lifetime of a battery can be increased up to 40% if the most appropriate duty cycle is used.

Another reason that we choose this stochastic model is its simplicity. The model is simple enough to explain a wide range of phenomena other than the battery recovery effect. Hence, we can also use our algorithm to solve MLCP while incorporating those phenomena, which include the energy-harvesting model [18].

1.1. Our contribution

We introduce two models based on the results in [14].

- *Duty cycle model:* Each sensor must be turned off after being active for a while. Consequently, because of the battery recovery effect, each sensor is guaranteed to have more battery lifetime.
- *Linear recovery model:* The battery recovery amount of each sensor is determined by the amount of idle time since the last discharge.

With the above two models, we aim to maximize the lifetime of the target coverage. The problem looks similar to MLCP, but is significantly harder. To illustrate the reason, suppose that we can use disjoint sets of sensors S_1 and S_2 to cover the target areas and all the sensors in S_1, S_2 can be used for 1 h. Consider the following two schedules:

Schedule 1:

1. Use S_1 for 1 h.
2. Use S_2 for 1 h.

Schedule 2: (repeat for 30 iterations)

1. Use S_1 for 2 min.
2. Use S_2 for 2 min.

In previous work where the battery recovery effect is not considered, none of the batteries can be used after either schedule is completed. All schedules that use S_1 and S_2 for 1 h in total have the same efficiency. Therefore, we can maximize the network lifetime just by deciding how long each set of sensors should be used.

In contrast, when the battery recovery effect is considered, batteries can be replenished during the idle periods. Let us assume that, for each idle period that is longer than 2 min, the battery lifetime is increased by 1 min. Hence, in Schedule 1, the recovery amount of all the batteries is 1 min. On the other hand, as the number of iteration in Schedule 2 is at least 30 and all batteries' lifetime is increased by 1 min at all iteration, the recovery amount of all batteries is at least 30 min. We can continue to use Schedule 1 for at least two more minutes, whereas we can use Schedule 2 for at least 60 more minutes. That is, the lifetime obtained from Schedule 2 is longer than the one obtained from Schedule 1. This example shows that, to maximize the network lifetime, it is not enough to just decide the duration for each subset of sensors. We also need to calculate an optimal usage sequence and the duration for each step in the sequence. Therefore, it is hard to modify the MLCP solution to solve the problem, and we cannot directly apply algorithms for MLCP to solve it.

In this paper, we devise three methods for both battery models by extending the algorithms for MLCP: greedy algorithms [6], the Garg–Könemann algorithm [7] and integer linear programming (ILP) formulations. The running times of all the proposed algorithms do not depend on the battery lifetime, and the asymptotic complexity is not larger than the algorithms for the original MLCP.

Through numerical experiments, we can see that our algorithms run as fast as those for the original MLCP, and the obtained solutions become much better. For the duty cycle model, the greedy heuristic algorithm works very efficiently. Its running time is usually less than 0.5 s. on average. The lifetime obtained from the greedy heuristic algorithm is about 40% longer than the one without the battery recovery effect. Furthermore, we show that the solution is close to the optimal solution obtained from the ILP formulation, as the difference between the two solutions is always less than 4.2%.

Because we assume that each battery can be recovered by 40% in our experiment, we know that the network lifetime cannot be prolonged by more than 40%. Because the experimental results show that our greedy heuristic algorithm can extend the lifetime by about 40%, the results indicate that it can almost fully utilize the battery recovery effect.

In contrast, the best algorithm for the linear recovery model is the algorithm based on the Garg–Könemann's framework. Our theoretical results show that the approximation ratio of the algorithm is at most $1.4+\varepsilon$ for any fixed $\varepsilon > 0$. In our numerical experiments, we can obtain a lifetime that is 10% longer compared to the greedy heuristic algorithm. The experimental results also indicate that the algorithm is relatively fast, as we can compute a sensor schedule in only 19.3 s on average.

Our approximation ratio for the MLCP with the battery recovery effect is slightly larger than the best approximation ratio for the original MLCP problem, which is $1+\varepsilon$ for any $\varepsilon > 0$.

Although our experimental results show a significant improvement in the network lifetime by using the battery recovery effect, the schedules obtained from our algorithms are more complicated than those of the original MLCP solutions. While each switch is turned on and off only a few times in the original MLCP, switches are turned on and off several hundred times in our schedules. Because our algorithms are centralized, we must send our schedule to each sensor. This may increase the communication cost, spending more energy of sensors. We show, however, in our experiments that it is not very critical. Because the size of our message to each sensor is