Received 6 July 2014; revised 9 October 2014; accepted 13 November 2014. Date of publication 25 November, 2014; date of current version 6 March, 2015.

Digital Object Identifier 10.1109/TETC.2014.2371543

Quality-Aware Target Coverage in Energy Harvesting Sensor Networks

XIAOJIANG REN[1], (Student Member, IEEE), WEIFA LIANG[1], (Senior Member, IEEE), AND WENZHENG XU[1,2]

[1]Research School of Computer Science, Australian National University, Canberra, ACT 0200, Australia
[2]School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510006, China
CORRESPONDING AUTHOR: W. Liang (wliang@cs.anu.edu.au)

This work was supported in part by the Actew/ActewAGL Endowment Fund the ACT Government of Australia.

ABSTRACT Sensing coverage is a fundamental problem in wireless sensor networks for event detection, environment monitoring, and surveillance purposes. In this paper, we study the sensing coverage problem in an energy harvesting sensor network deployed for monitoring a set of targets for a given monitoring period, where sensors are powered by renewable energy sources and operate in duty-cycle mode, for which we first introduce a new coverage quality metric to measure the coverage quality within two different time scales. We then formulate a novel coverage quality maximization problem that considers both sensing coverage quality and network connectivity that consists of active sensors and the base station. Due to the NP-hardness of the problem, we instead devise efficient centralized and distributed algorithms for the problem, assuming that the harvesting energy prediction at each sensor is accurate during the entire monitoring period. Otherwise, we propose an adaptive framework to deal with energy prediction fluctuations, under which we show that the proposed centralized and distributed algorithms are still applicable. We finally evaluate the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed solutions are promising.

INDEX TERMS Sensing coverage, utility functions, renewable sensor networks, target quality monitoring, dynamic framework, energy replenishment.

I. INTRODUCTION

The limited lifetime of conventional, battery-powered sensor networks has hindered their wide deployments for many applications that need long-term network operations. A promising solution to address this energy shortage is enabling sensor nodes to harvest renewable energy from their surroundings [13]. In addition to environmental friendliness of renewable energy, sensors powered by renewable energy allow the sensor network to operate perpetually with proper energy management. As sensing coverage is a fundamental problem in wireless sensor networks, in this paper, we consider the sensing coverage problem in an energy harvesting sensor network, which can be stated as follows. Given a set of targets (e.g., some critical facilities) in a monitoring region, a sensor network that consists of a set of heterogeneous sensors powered by renewable energy and a base station used to monitor the set of targets for a specified period, where sensors transmit their sensing data to the base station in a real-time manner. The problem is to activate sensors such that the target coverage quality is maximized, subject to that (i) the amount of energy consumed by each sensor is no more than that it has been charged during this monitoring period; and (ii) the communication network induced by the active sensors and the base station at each time point is connected. One such an application scenario is an energy harvesting sensor network deployed for forest fire monitoring.

Sensing coverage in conventional sensor networks has been extensively studied in the past decade. Most studies focused on the network lifetime prolongation. To maximize the network lifetime, various strategies of sensor activity scheduling have been proposed. Among them, a popular one is the adoption of duty-cycles, that is, each sensor works either in active or sleep mode [3], [7], [8], [12], [14], [15], [23]. In comparison with conventional sensor networks, network lifetime of energy harvesting sensor networks is no longer a main issue since sensors can be recharged repeatedly by renewable energy sources. This results in the research focus shift from the network lifetime maximization to scheduling sensor activities to keep them survival through accurate energy harvesting predictions. For the latter, several studies on target coverage have been conducted with the aim of optimizing the coverage performance [6], [19], [20], [24]. These mentioned studies however did not consider the connectivity of the communication network induced by the activated sensors and the base station. It is well known that both sensing coverage and network connectivity are the fundamental performance metrics for wireless sensor networks, where the coverage quantifies the quality of monitoring while the network connectivity indicates the accessibility from the base station to sensory data.

In this paper, we study the coverage maximization problem in a renewable sensor network, and focus on devising efficient centralized and distributed algorithms for scheduling sensor activities such that the target coverage quality is maximized, subject to that the communication network induced by the activated sensors and the base station at each time point is connected. Unlike most existing studies on conventional sensor networks that the energy of each sensor decreases monotonically over time, the energy consumption at each sensor in renewable networks can be well managed. In contrast, the energy harvesting rate of each sensor in energy harvesting sensor networks varies over time, and the energy of each sensor can be replenished if needed. However, the energy consumption at each sensor must be carefully managed. On one hand, if there is enough amount of harvested energy available in the near future, we must fully make use of the harvested energy for maximizing target coverage; otherwise, the conservative use of the harvested energy may miss the next recharging opportunity. On the other hand, if the energy charging chances of a sensor in the near future is predictably small, its energy should not be used carelessly despite that the sensor may still have plenty of energy. Otherwise, the sensor will expire very soon, and its coverage quality will severely decrease. In summary, time-varying characteristics of renewable energy sources in energy harvesting sensor networks makes sensor activity scheduling become very difficult, not to mention ensuring that all activated sensors and the base station must be connected.

In this paper we approach the coverage maximization problem for a given monitoring period by adopting a general strategy. That is, we start by dividing the entire monitoring period into L equal numbers of time slots. We then perform sensor activation or inactivation scheduling in the beginning of each time slot. The challenges to solve the problem are as follows. (1) At which time slots among the L time slots, a sensor should be activated or deactivated, as the amount of harvested energy (of consumed energy) at a sensor depends on not only different scheduling strategies but also the availabilities of time-varying energy harvesting sources in the entire monitoring period. (2) How to make sure that all activated sensors and the base station form a connected component at each time slot. (3) How to devise an efficient sensor scheduling algorithm whose solution will guarantee that the target coverage quality for the entire monitoring period is maximized.

The novelty of our work lies in two aspects. We are the first to introduce a new coverage metric to accurately measure the target coverage quality. This new metric enables to model the coverage quality of each target within two different time scales: One is within each time slot, in which the coverage quality of the target is modeled by a sub-modular function of the number of sensors covering it, which implies that the margin gain of the coverage quality of the target decreases with the number of sensors it is covered in the time slot. Another is within the entire monitoring period, the coverage quality of a target is measured by the number of time slots it is covered, this metric is also modeled by a sub-modular function that may be different from the one within each time slot, which implies that the more the number of time slots the target is covered, the higher the coverage quality of the target will be. The overall coverage quality of a target for the entire monitoring period then is a weighted linear combination of these two sub-modular functions. Not only do we introduce this new coverage quality metric, but also do we devise novel centralized and distributed algorithms for the coverage maximization problem in a renewable sensor network, in which sensors are powered by time-varying harvesting energy sources. Also, we propose an adaptive framework for the problem under both network connectivity and harvesting energy prediction fluctuation constraints.

The main contributions of this paper are as follows. We first consider quality-aware target coverage in an energy harvesting sensor network by introducing a new coverage metric that can measure the coverage quality accurately, and formulating a novel coverage maximization problem that takes both sensing coverage quality and network connectivity into consideration. As the problem is NP-hard, we then devise efficient centralized and distributed algorithms for it, provided that the amount of harvested energy of each sensor for a given monitoring period can be accurately predicted. Otherwise, we propose an adaptive framework to handle energy prediction fluctuations during the monitoring period. We finally conduct extensive experiments by simulations to evaluate the performance of the proposed algorithms. Experimental results show that the solutions delivered by the proposed algorithms are very promising.

The rest of the paper is organized as follows. Section II surveys related works. Section III introduces basic models, defines the coverage maximization problem, and shows its NP-hardness. A centralized heuristic algorithm and its distributed implementation are given in Sections IV and V, respectively. An adaptive framework dealing with energy prediction fluctuation is proposed in Section VI. Section VII presents the simulation results, and Section VIII concludes the paper.

II. RELATED WORK

Sensing coverage problems in conventional sensor networks have been extensively investigated in the past [1], [3], [4], [8], [15], [22]. One efficient method is to partition sensors in a sensor network into multiple subsets (*sensor covers*) such that the sensors in each subset can cover all targets. Thus, only one sensor cover at each time slot is activated for a fractional of the entire monitoring period and only the sensors in the active sensor cover are in active mode, while the others are in sleep mode to save their energy [3]. In terms of connected coverage problem, Gupta *et al.* [8] proposed the minimum connected sensor cover problem to find a minimum number of sensors to achieve a full coverage while the communication graph induced by the sensors is connected. They presented a greedy algorithm with a guaranteed performance ratio, assuming that each sensor can adjust its transmission range dynamically. Wu *et al.* [23] recently presented an improved approximation algorithm for it. Liu and Liang [15] studied the connected coverage problem with a given coverage guarantee. They introduced the partial coverage concept, and presented a centralized heuristic algorithm which takes both partial coverage and sensor connectivity into account simultaneously. They also considered the full coverage and sensor connectivity by partitioning the lifetime of a sensor into several equal intervals and finding a collection of connected sensor covers such that the network lifetime is maximized [16]. Ammari and Das [1] addressed the k-coverage problem that within each scheduling round, every location in a monitoring field is covered by at least k active sensors while keeping all active sensors connected. They proposed several heuristic algorithms for the problem.

Compared with the studies on sensing coverage in conventional sensor networks, a very few attentions have been paid to the sensing coverage problem in energy harvesting sensor networks. Tang *et al.* [20] studied the problem and proposed an approximation algorithm with an approximation ratio 1/2, by assuming that the coverage quality is characterized by a sub-modular function and the communication graph induced by the active sensors and the base station may be disconnected. They [21] also extended their work by proposing distributed sensing schedule algorithms with provable convergence and performance bound by fixing the duty cycle of each sensor. Dai *et al.* [6] considered a similar problem for stochastic event capture by formulating a coverage optimization problem and presenting an approximation algorithm with an approximation ratio 1/2. Yang and Chin [24] considered the problem of maximizing the network lifetime while ensuring all targets are continuously monitored by at least one sensor. They formulated a linear programming solution to determine the activation schedule of sensors, where one subset of sensors is active while the rest of sensors keep in sleep modes to conserve energy. However, none of these mentioned works takes into consideration of the connectivity of active sensors and the base station. Consequently, the sensing data generated by active sensors may not be able to relayed to the base station immediately. In practice, many critical real-time applications do need the sensed data to be collected in a real-time manner. Consider that the transmission energy consumption of each sensor in most real applications is the dominant one among its energy consumptions in sensing, computation and communications, its sensing data must be relayed to the base station through multiple relays to reduce its energy consumption. The connectivity among active sensors and the base station thus is necessitated to ensure such real-time data transfer. This connectivity requirement thus poses great challenges in the design of approximation algorithms for the problem. That is why none of approximation algorithms for the problem under the connectivity constraint with an optimization objective expressed by a sub-modular function has ever been developed. Orthogonal to these existing studies, in this paper, we take the network connectivity into consideration, and focus on developing centralized and distributed heuristic for the coverage maximization problem. We will propose a more accurate quality coverage model that measures the coverage quality of each target within two different time scales: the number of sensors the target is covered in each time slot; and the duration the target has been covered for the monitoring period.

III. MODELING AND PROBLEM FORMULATION

A. SYSTEM MODEL

We consider an energy harvesting sensor network $G = (V \cup \{s\}, E)$ consisting of $|V|$ heterogeneous stationary sensors and a base station s, which is deployed to monitor m targets $O = \{o_1, o_2, \ldots, o_m\}$ in a 2D region of interest. Each sensor $v \in V$ is powered by renewable energy source such as solar energy, and has a fixed transmission and sensing ranges. There is an edge in E between two sensors or a sensor and the base station if they are within the transmission range of each other. For each sensor $v \in V$, let C_v be the set of targets within its sensing range. For each target $o \in O$, let S_o be the set of maximum number of active sensors covering it.

B. ENERGY HARVESTING BUDGET MODEL

Following a widely adopted renewable energy replenishment assumption [13], [18], we assume that the energy replenishment rate of each sensor is much slower than its energy consumption rate, and the amount of energy harvested by the sensor in a future time period is uncontrollable but predictable, based on its source type and its historic energy harvesting profile. Assume that time is divided into equal time slots. Let L be the number of time slots after which the next recharging pattern will be repeated, where *a recharging pattern* of solar energy depends on the weather conditions accordingly (e.g., 24 hours on default). Assume that the L time slots are indexed by $1, 2, \ldots, L$. To estimate the amount of energy harvested of each sensor at a recharging pattern, several prediction algorithms are available [2], [9], e.g., the Exponentially Weighted Moving-Average (EWMA) algorithm by Kansal *et al.* [9]. Specifically, let $\overline{Q}(t)$ be the prediction of the amount of harvested energy of sensor $v_i \in V$ at time slot t with $1 \leq t \leq L$. The value of $\overline{Q}(t)$ is calculated as follows.

$$\overline{Q}(t) = w \cdot \overline{Q}(t') + (1 - w) \cdot Q(t'), \tag{1}$$

where w is a given weight with $0 < w < 1$, t' $(= t - L)$ is the Lth time slot in the previous recharging pattern, and $Q(t')$ is the actual amount of energy harvested at time slot t'. With the knowledge of its harvesting energy prediction, the energy budget $P(v_i)$ of sensor $v_i \in V$ in the next L time slots is defined as

$$P(v_i) = \min\{B(v_i), RE(v_i) + \sum_{t=1}^{L} \overline{Q}(t)\}, \tag{2}$$

where $B(v_i)$ and $RE(v_i)$ are the battery capacity and the residual energy of sensor v_i in the beginning of the previous recharging pattern, $1 \le i \le |V|$.

C. ENERGY CONSUMPTION MODEL

Recall that each sensor $v_i \in V$ at each time slot operates in either active or sleep (or inactive) mode. Let e_i^{active} and e_i^{sleep} be the energy consumptions of sensor v_i in active and sleep modes at each time slot, respectively. Assume that $e_i^{sleep} \ll e_i^{active}$ and the energy consumption of sensor v_i in sleep mode is negligible. The base station will determine the schedule of sensors in the beginning of every L time slots, according to the energy budget of each sensor. By the energy neutral operation theory [9], to support continuous monitoring services, sensors should not consume more energy than that they harvested at any period. The activation of a sensor thus is constrained by the actual amount of energy it harvested. Let $b_i = \lfloor \frac{P(v_i)}{e_i^{active}} \rfloor$ be *the time slot budget* of sensor $v_i \in V$ for a monitoring period of L time slots. Then, sensor v_i cannot be activated more than b_i time slots for a monitoring period of L time slots, where $P(v_i)$ is the energy budget of sensor v_i.

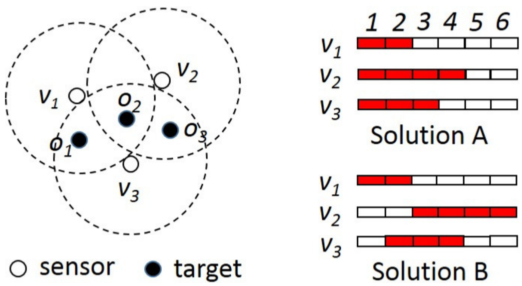


FIGURE 1. A simple motivation example for measuring the coverage quality of targets.

D. COVERAGE QUALITY

In each time slot, a different subset of sensors will be activated, which leads to a different subset of targets to be covered. Also, the more time slots in which a target is covered, the higher the coverage quality of the target will be. To measure the coverage quality of targets, we here consider the target coverage quality within two different time scales, which is illustrated by a simple motivation example in Fig. 1, where sensors v_1, v_2, and v_3 are deployed to monitor targets o_1, o_2, and o_3 for a monitoring period of 6 time slots. Assuming that the time slot budgets of sensors v_1, v_2, and v_3 are 2, 4, and 3, respectively. There are two different solutions A and B for sensor activation in a given monitoring period. Targets in solution A are covered by more sensors in each time slot but for less time slots, e.g., target o_1 is covered by both sensors v_1 and v_3 in time slots 1 and 2, but it is only covered by 3 time slots among the monitoring period of 6 time slots. Targets in solution B are covered by more time slots but by less sensors in each time slot, e.g., target o_1 is covered by 4 time slots, but it is only covered by a single sensor at time slots 1, 3, and 4, respectively. From these two different solutions, it can be seen that the coverage quality of each target o is determined by not only the number of time slots it is covered but also the number of sensors it is covered within each time slot.

In the following we first adopt a utility metric similar to the one in [12], where the coverage quality of a target is measured by the number of time slots in which the target is covered. Specifically, for each target $o \in O$ at each time slot t with $1 \le t \le L$, let $N_1(o,t) = \{t\}$, which is a set containing the index t of time slot t if target o is covered by an active sensor in time slot t; $N_1(o,t) = \emptyset$ otherwise. Let N_c^o be the set of time slots in which target o is covered, then $N_c^o = \cup_{t=1}^{L} N_1(o,t)$. Clearly, N_c^o is a subset of the set of all time slots $\{1, 2, \ldots, L\}$. Let $U_1(o) = f_1(N_c^o)$ represents the coverage quality of target o, by counting the number of time slots the target being covered during a monitoring period of L time slots, where f_1 is a sub-modular function whose definition is as follows. $f_1 : 2^A \mapsto R^{\ge 0}$ satisfies the following three properties:

$$(1)\ f_1(\emptyset) = 0; \tag{3}$$
$$(2)\ f_1(A_1) \le f_1(A_2) \quad \text{where } A_1 \subseteq A_2 \subseteq A \tag{4}$$
$$\text{and } A \text{ is a finite ground set;} \tag{5}$$
$$(3)\ f_1(A_1 \cup \{a\}) - f_1(A_1) \ge f_1(A_2 \cup \{a\}) - f_1(A_2) \tag{6}$$
$$\text{where } A_1 \subseteq A_2 \subseteq A \text{ and } \exists a \in A \setminus A_1 \cup A_2. \tag{7}$$

The rationale behind the adoption of the sub-modular function f_1 (sometimes it is also referred to as a utility function) is that f_1 is a monotonic increasing function, whose marginal utility decreases with the increase of the number of time slots. In other words, for each target $o \in O$, the more time slots it is covered, the higher coverage quality it will have. However, with the further increase on the number of time slots it is covered, the net gain of its coverage quality becomes diminishing.

The use of coverage metric $U_1(\cdot)$ to measure the target coverage quality however is biased. Under this metric, for a given target, it cannot be distinguished whether the target is covered by only a single sensor or by multiple sensors at a given time slot. For example, in event detection applications, the more the sensors an event is detected, the higher probability the event can be discovered [25]. To capture the coverage quality of each target both in each time slot and for the entire monitoring period, we then introduce a new coverage quality metric within two different time scales that takes into account not only the number of sensors covering a target at each given time slot but also the number of time slots the target is covered for the monitoring period of L time slots, through two non-decreasing sub-modular functions $f_1(\cdot)$ and $f_2(\cdot)$, respectively. Specifically, for each target $o \in O$ at each time slot t, let $U_2(o, t) = f_2(S_o^t)$ represents the coverage quality of target o at time slot t, where $S_o^t \subseteq S_o$ is the set of active sensors covering target o at time slot t. The coverage quality of target o for L consecutive time slots thus is

$$U(o) = \alpha \cdot U_1(o) + (1 - \alpha) \cdot \sum_{t=1}^{L} U_2(o, t), \tag{8}$$

where α is a given utility weight with $0 \leq \alpha \leq 1$. When $\alpha = 0$, this means we only consider the coverage quality caused by the number of sensors covering target o, while $\alpha = 1$ means we only consider the coverage quality by the number of time slots target o being covered during the entire monitoring period. Hence, the overall coverage quality achieved for the L time slots is $\sum_{o \in O} U(o)$.

E. PROBLEM STATEMENT

Given an energy harvesting sensor network $G = (V \cup \{s\}, E)$ deployed for monitoring a set of targets O for a period of L consecutive time slots, and the time slot energy budget b_i of each sensor $v_i \in V$, *the coverage maximization problem* in G is to activate a subset of sensors V_t ($V_t \subseteq V$) at each time slot t with $1 \leq t \leq L$ such that the overall coverage quality for the monitoring period $\sum_{o \in O} U(o)$ is maximized, where

$$\sum_{o \in O} U(o) = \alpha \sum_{o \in O} U_1(o) + (1 - \alpha) \sum_{o \in O} \sum_{t=1}^{L} U_2(o, t) \tag{9}$$

$$= \alpha \sum_{o \in O} f_1(\cup_{t=1}^{L} N_1(o, t)) + (1 - \alpha) \sum_{o \in O} \sum_{t=1}^{L} f_2(S_o^t), \tag{10}$$

$$N_1(o, t) = \begin{cases} \emptyset & \text{if } \nexists v \in V_t \text{ s.t. } o \in C_v \\ \{t\} & \text{if } \exists v \in V_t \text{ s.t. } o \in C_v, \end{cases} \tag{11}$$

and

$$S_o^t = \begin{cases} \emptyset & \text{if no sensor node in } V_k \text{ covers target } o \\ \{v \mid v \in V_t,\ o \in C_v\} & \text{otherwise,} \end{cases} \tag{12}$$

subject to the following two constraints:

1) the induced communication subgraph by activated sensors in V_t and the base station is connected, i.e., $G[V_t \cup \{s\}]$ is a connected graph for each time slot t with $1 \leq t \leq L$. Thus, the sensing data of activated sensors in V_t can be relayed to the base station in real time.
2) For each sensor $v_i \in V$, the number of time slots in which it is activated is no more than its time slot budget b_i so that none of the sensors will run out of its budgeted energy, i.e., $\sum_{t=1}^{L} I(V_t, v_i) \leq b_i$, where $I(V_t, v_i)$ is an indicator function, which is defined as $I(V_t, v_i) = 1$ if $v_i \in V_t$ and $I(V_t, v_i) = 0$ otherwise.

The coverage maximization problem defined is NP-hard. It is easy to verify that the dynamic activation schedule problem in [20] is a special case of the problem, where each sensor can communicate with the base station directly, and the utility weight α is 1. Even for this special case, it has been shown to be NP-hard, which implies the NP-hardness of the coverage maximization problem.

IV. HEURISTIC ALGORITHM

Due to the NP-hardness of the coverage maximization problem, we here propose a greedy heuristic for it, assuming that the energy budget of each sensor for a monitoring period of L time slots is given in advance. In general, for each time slot t with $1 \leq t \leq L$, we assume that there is a corresponding tree rooted at the base station consisting of all activated sensors at time slot t. Initially, there is a forest consisting of L trees with each tree containing only the tree root - the base station. Recall that b_i is the time slot budget of sensor $v_i \in V$ in the beginning of a monitoring period of L time slots. Then, sensor v_i can join no more than b_i trees in the forest; otherwise, its energy budget is not enough to support its operation.

The construction of the forest proceeds iteratively. Within each iteration, a sensor node is added to one of the L trees in the forest if it results in the maximum utility gain in terms of the coverage quality by (9). This procedure continues until either no more sensors can be added to the trees, or no more utility gain on the coverage quality can be achieved. Note that none of the sensor nodes is added to a single tree twice.

A. ALGORITHM

Given the time slot budget $b_i \geq 0$ of sensor $v_i \in V$ for all i with $1 \leq i \leq |V|$, we first construct an auxiliary graph $G' = (V' \cup \{s_1, s_2, \ldots s_L\}, E')$ from the energy harvesting sensor network $G = (V, E)$ as follows.

For the base station s, there are L corresponding copies $s_1, s_2, \ldots, s_L$ in G' with each being the root of a tree T_j, $1 \leq j \leq L$. These L trees form a forest. For each sensor $v_i \in V$, there are b_i corresponding node copies $v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(b_i)}$ in V' with each corresponding an activation of sensor v_i in one of up to b_i time slots, assuming that $b_i \ll L$. For each edge $(v_i, s) \in E$ that corresponds that the base station and sensor v_i are within the transmission range of each other, there are $b_i \times L$ corresponding edge copies $(v_i^{(1)}, s_1), \ldots, (v_i^{(b_i)}, s_1), \ldots, (v_i^{(1)}, s_L), \ldots, (v_i^{(b_i)}, s_L)$ in E'. For each edge $(v_i, v_j) \in E$ that corresponds that sensors v_i and v_j are within the transmission range of each other, there are $b_i \times b_j$ corresponding edge copies $(v_i^{(1)}, v_j^{(1)}), \ldots, (v_i^{(b_i)}, v_j^{(1)}), \ldots, (v_i^{(1)}, v_j^{(b_j)}), \ldots, (v_i^{(b_i)}, v_j^{(b_j)})$ in E'.

Fig. 2(b) is an illustrative construction of graph G' of the original energy harvesting sensor network $G = (V \cup \{s\}, E)$ in Fig. 2(a), where the time slots are indexed by $1, 2, \ldots, L$ with $L = 6$ and the sensor set $V = \{v_1, v_2, v_3, v_4, v_5\}$. Let $b_i = i$ be the time slot energy budget of each sensor $v_i \in V$ for a given monitoring period of L time slots, $1 \leq i \leq 5$.

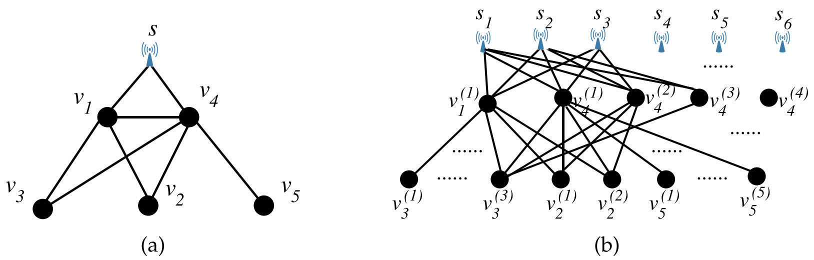


FIGURE 2. An example: $L = 6$ and an energy harvesting sensor network $G = (V \cup \{s\}, E)$ with the set of sensors $V = \{v_1, v_2, v_3, v_4, v_5\}$ and $b_i = i$ for all i with $1 \le i \le 5$. (a) $G = (V \cup \{s\}, E)$. (b) $G' = (V' \cup \{s_1, s_2, \dots, s_L\}, E')$.

The forest consists of L trees $T_1, T_2, \dots, T_L$, which is constructed as follows. Initially, each tree T_j contains only the root node s_j, $1 \le j \le L$. We add the other copies of sensor nodes in V' to the trees iteratively. Within each iteration, a node is added to the forest if it leads to the maximum utility gain of the coverage quality. Specifically, for each node $v_i^k \in V'$ with $1 \le k \le b_i$, let $v_i \in V$ be its corresponding sensor and $V(v_i^k) = \{v_i^{(1)}, v_i^{(2)}, \dots, v_i^{(b_i)}\}$ the set of copies of v_i in G'. Recall that C_{v_i} is the set of targets within the sensing range of v_i. We set $C(v_i^k) = C_{v_i}$ for each node v_i^k, which is the set of targets covered by node v_i^k. For each tree T_j rooted at node s_j, let $V(T_j) \subseteq V'$ be the set of nodes in tree T_j and $C(T_j) \subseteq O$ the set of targets covered by the sensor nodes in $V(T_j)$ with $1 \le j \le L$. Recall that N_c^o is the subset of time slots in which target o is covered for the monitoring period of L time slots, where $N_c^o = \{j \mid \exists j \; s.t. \; o \in C(T_j), 1 \le j \le L\}$. For each node $v_i^k \in V'$ that has not been contained by any tree and one of its adjacent nodes in G' is in tree T_j, we can calculate the potential utility gain of the coverage quality ΔU_{ij} if node v_i^k is added to T_j by Eq.(13),

$$\Delta U_{ij} = \begin{cases} 0 & V(v_i^k) \cap V(T_j) \neq \emptyset \text{ implies that another copy of } v_i \text{ has been contained by tree } T_j, \\ \alpha \cdot \sum\limits_{o \in \{C(v_i^k) - C(T_j)\}} (f_1(N_c^o \cup \{j\}) - f_1(N_c^o)) + (1-\alpha) \cdot \sum\limits_{o \in C(v_i^k)} (f_2(S_o^j \cup \{v_i\}) - f_2(S_o^j)) & \text{otherwise,} \end{cases} \tag{13}$$

where $V(v_i^k) \cap V(T_j) \neq \emptyset$ represents that sensor v_i has already been activated at time slot j.

We then choose a node $v_{i'} \in V'$ with the maximum utility gain of the coverage quality $\Delta U_{i'j'}$, and add $v_{i'}$ to tree $T_{j'}$ if this results in the maximum gain of the coverage quality. This procedure continues until all nodes are added to the forest or no further improvement in the coverage quality can be achieved. That is, either all nodes in G' have been added to the trees in the forest, or no node addition results in a positive utility gain of the coverage quality. As a result, trees $T_1, T_2, \dots, T_L$ rooted at nodes $s_1, s_2, \dots, s_L$ are obtained, where the nodes in tree T_j rooted at s_j represent that their corresponding sensors in G will be activated at time slot j, and these sensors and the base station will be connected, $1 \le j \le L$. Notice that it is very likely there are some trees in the forest containing the root node only. If this is the case, it implies that none of the sensors in the network at the corresponding time slot of this tree is active. The detailed description of the proposed algorithm is given in `Algorithm` 1.

Theorem 1: Given an energy harvesting sensor network $G = (V \cup \{s\}, E)$ deployed for monitoring a set of targets in the region for a period of L time slots, there is an algorithm `Greedy_Heuristic` for the coverage maximization problem, which takes $O(b_{max}^3 \cdot |V|^2 \cdot |E| + b_{max} \cdot d_{max} \cdot L)$ time, where $|V|$ is the number of sensors, $b_{max} = \max_{v_i \in V}\{b_i\}$, $d_{max} = |N(v)|$, and $N(v)$ is the set of neighbors of node v in G. Notice that d_{max} usually is a constant, while b_{max} is a constant and even if it is not, then $b_{max} \ll L$.

Proof: We first show that the algorithm is correct. That is, each sensor will not run out of its energy budget. As there are b_i nodes for sensor v_i in G' with each corresponding its energy consumption at one time slot. Thus, v_i will not run out of its energy budget as it can only join at most b_i trees. Following the construction of the trees, each of the b_i copies of v_i can appear in a tree only once. Also, within the time slot to which a tree corresponds, all sensors in the tree will be activated, and the activated sensors and the base station are in the same connected component. Thus, the solution delivered by algorithm `Greedy_Heuristic` is a feasible solution to the coverage maximization problem.

We then analyze the time complexity of the proposed algorithm `Greedy_Heuristic` in the following. The auxiliary graph G' contains at most $|V| \cdot b_{max}$ nodes since there are at most b_{max} copies in G' of each node in G. The number of edges in G', $|E'|$, is no more than $d_{max} \cdot b_{max} \cdot L + \sum_{e \in E} b_{max}^2 = b_{max} \cdot d_{max} \cdot L + b_{max}^2 \cdot |E|$ edges. Thus, the construction of G' takes $O(b_{max} \cdot d_{max} \cdot L + |V| \cdot b_{max} + b_{max}^2 |E|)$ time. Within each iteration, for each unscheduled node $v_i^k \in V'$, let $N_{G'}(v_i^k)$ be its neighbor set in G', we need to calculate the incremental coverage quality ΔU_{ij} for each $v' \in V(T_j) \cap N_{G'}(v_i^k)$ with tree root s_j, and choose a node $v_{i'}^{k'}$ with the maximum incremental coverage quality among the unscheduled nodes in V', this takes $O(\sum_{v_i^k \in V'} |N_{G'}(v_i^k)| \cdot |V'| \cdot C_{max}) = O(b_{max}^2 \cdot |V| \cdot |E| \cdot C_{max}) = O(b_{max}^2 |V||E|)$ time, where C_{max} is the maximum

Algorithm 1 `Greedy_Heuristic`

Input: An energy harvesting sensor network $G = (V \cup \{s\}, E)$, a set of targets O, and time slots that are indexed by $1, 2, \ldots, L$. For each sensor $v_i \in V$, its energy budget $P(v_i)$ in L time slots is given.

Output: For each time slot j, a set of sensors $V_j \subseteq V$ which will be activated at time slot j with $1 \leq j \leq L$.

Calculate its time slot budget b_i by its energy budget $P(v_i)$ for each sensor $v_i \in V$;
Construct an auxiliary graph $G' = \{V' \cup \{s_1, s_2, \ldots, s_L\}, E'\}$;
Construct a forest in G' consisting of L trees $T_1, T_2, \ldots, T_L$ rooted at nodes $s_1, s_2, \ldots, s_L$, respectively;
$T_j \leftarrow (\{s_j\}, \emptyset)$ initially, $1 \leq j \leq L$;
$W \leftarrow V'$; /* The nodes in W have not been examined */
/* Add the nodes in W to the L trees one by one */
$zero\_gain \leftarrow' true'$;
while (there is a node in W that has not been contained by any tree) and $zero\_gain$ **do**
  Calculate the gain of the coverage quality ΔU_{ij} for each node $v_i^k \in W$ and one of its adjacent nodes in a tree T_j rooted at s_j for each of these adjacent nodes in the adjacent list of v_i^k;
  Identify a node $v_{i'}^{k'}$ with the maximum $\Delta U_{i'j'}$ among the nodes in W;
  if $\Delta U_{i'j'} == 0$ **then**
    $zero\_gain \leftarrow' false'$; /* No further improvement in the coverage quality is achieved */
  else
    $V(T_{j'}) \leftarrow V(T_{j'}) \cup \{v_{i'}^{k'}\}$; /* Add node $v_{i'}^{k'}$ to tree $T_{j'}$ */
    $W \leftarrow W \setminus \{v_{i'}^{k'}\}$;
  end if
end while
Construct V_j from $V(T_j)$ by adding the corresponding sensor of a copy of a sensor in $V(T_j)$;
return The set of active sensors at time slot j is V_j for all j with $1 \leq j \leq L$.

number of targets covered by a sensor, which usually is a constant in practice. It is easy to verify that the number of iterations of the proposed algorithm is bounded by $|V'|$. The algorithm thus takes $O(b_{max} \cdot |V| \cdot b_{max}^2 \cdot |V| \cdot |E| + b_{max} \cdot d_{max} \cdot L) = O(b_{max}^3 |V|^2 |E| + b_{max} \cdot d_{max} \cdot L)$ time. □

V. DISTRIBUTED IMPLEMENTATION OF THE PROPOSED ALGORITHM

As real sensor networks are distributive, it is desirable that algorithms for sensor networks are distributed algorithms, whereas the solution obtained by the centralized algorithm usually serves as the benchmark of the solutions obtained by distributed algorithms. In this section, we propose a distributed implementation of the proposed centralized algorithm `Greedy_Heuristic`. Following most common assumptions in the design of distributed algorithms, we assume that the amount of energy consumed for finding a distributed solution can be neglected, in comparison with the amount of energy consumed for sensing coverage, local computation and sensing data transmission.

The idea behind the distributed implementation is that we treat the original network G as a *host graph*, and the constructed auxiliary graph G' as a *guest graph*. We "embed" the guest graph into the host graph. Each node v_i in the host graph G simulates its b_i copies in the guest graph G'. Each link (v_i, v_j) in the host graph G simulates its corresponding $b_i \cdot b_j$ links in the guest graph G' between the copies of nodes v_i and v_j. In the host graph G, there is a broadcast tree which is dynamically constructed. The broadcast tree will be used for tree information broadcasting of the L trees constructed from G', it also serves as collecting "joining-tree request" messages from non-tree nodes in G'. In the guest graph G', there is a forest consisting of the L trees with the sensors in each tree corresponding to the activated sensors at one time slot among the L time slots in the monitoring period. The base station contains the L trees of the forest with each tree T_j having a tree root at s_j and spanning all activated sensors at time slot j, $1 \leq j \leq L$. Assume that the broadcast tree in G contains the base station only initially.

The construction of the forest $\mathcal{F}$ consists of the L trees $T_1, T_2, \ldots, T_L$ proceeds iteratively. Within each iteration, some nodes in V' join some of the L trees in the forest, and their "joining-tree request" messages will be propagated to the base station along the links of the broadcast tree. The base station then calculates the coverage quality and broadcasts the L tree messages to those unjoined nodes which are close to the tree nodes, i.e., there is an edge in G' between a tree node and an unjoined node. This procedure continues until either all the nodes in V' have joined the trees in the forest, or there is no improvement on the utility gain of the coverage quality. In the following, we detail the distributed implementation of the proposed algorithm at iteration t.

Within iteration t, let $V_t(\mathcal{F})$ be the set of nodes in the forest and $W_t = V' \setminus V_t(\mathcal{F})$ the set of nodes that are not in the forest yet. Assume that each node in $V_t(\mathcal{F})$ is labelled as a *tree node* which contains the following information: *its tree root, the set of members in the tree, and the value of the coverage quality*. Let $E_t = E' \cap (V_t(\mathcal{F}) \times W_t)$ be the set of edges in G' across the two sets $V_t(\mathcal{F})$ and $V' \setminus V_t(\mathcal{F})$. For each unlabeled node in $v \in W_t$, let $(v, u_1), (v, u_2), \ldots, (v, u_l)$ be its incident edges in E_t. These l nodes $u_1, u_2, \ldots, u_l$ forms a set, which is then partitioned into l' subsets, where all the nodes in the same tree in $\mathcal{F}$ belong to the same subset. Discard these subsets in which the trees contain a copy of v already. Denote by l'' the remaining subsets (or trees). Clearly $l'' \leq l' \leq l$. Compute the utility gain of the unlabeled node v if it is added to one of the l'' trees, identify a tree with the maximum gain of the utility, and v then sends a "joining-tree request" to the tree node and puts it as a candidate of joining that tree. All tree nodes send their received "joining-tree request" messages to the base station.

The base station then updates the members of the trees in the forest $\mathcal{F}$, by adding the new members to the trees and updating their utility values. For a given tree (e.g., T_j), there may have multiple joining-tree requests such as (v, u) and (v', w) where $u, w \in V(T_j)$. If both v and v' are different copies of the same sensor, only one of them will join the tree. Or, if there is no positive gain for all trees or all the nodes in V' have been included in forest $\mathcal{F}$, the procedure terminates. Otherwise, the base station broadcasts the updated information of the L trees along the links of the broadcast tree. Each unlabeled node in G' that has sent a "joining-tree request" message will check whether it becomes a member in its requested tree. If yes, label itself as a tree node, and check whether its host node is included in the broadcast tree already. If not, set the host node as a tree node in the broadcast tree, and send a message to its parent host node. The parent host node then sets the host node as one of its children in the broadcast tree.

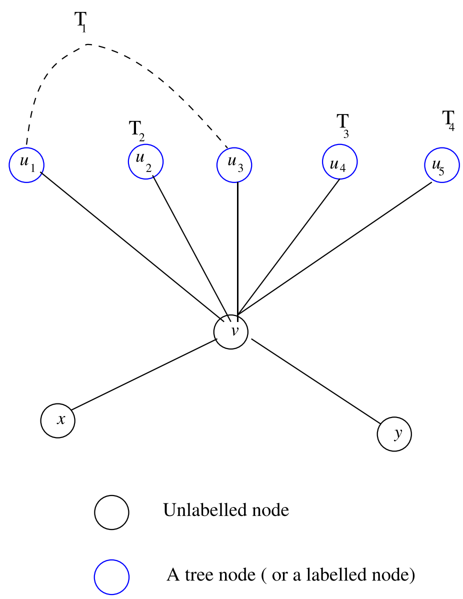


FIGURE 3. An illustration of an unlabeled node *v* joining one of the *L* trees.

We here use an example to illustrate the procedure of node joining the trees (see Fig. 3). Assume that an unlabeled node v has 5 tree neighboring nodes $u_1, u_2, \ldots, u_5$, and two unlabeled neighboring nodes x and y. We further assume that u_1 and u_3 are in the same tree in the forest and denote by this tree as T_1. Nodes u_2, u_4 and u_5 are in trees T_2, T_3 and T_4, respectively. We further assume that tree T_3 contains a copy of sensor v already. Thus, in this case $l = 5, l' = 4$ and $l'' = 3$. Node v can join either of trees T_1, T_2, and T_4. Assume that v joining T_2 will result in the maximum utility gain of coverage quality utility, then node v sends a "joining-tree request" to the tree node u_2 for joining T_2. The base station then updates each of the L tree information once it receives all "joining tree request" messages from its tree nodes. Assume that it updates tree T_2, if there is no other messages from the other unlabeled nodes that are the copies of the same sensor as node v, then v is added to T_2 as a new member. Otherwise, the base station chooses one of different copies of the same sensor to admit, and broadcasts all updated tree information to each tree node through the broadcast tree. When v received the updated message, it checks whether it has been admitted. If yes, set itself as a tree node, and also check whether its host node is in the broadcast tree. Otherwise, set the host node as a tree node in the broadcast tree, and send its parent in the tree a message that it will one its child, and its parent node sets it as one of its children.

Now, we estimate the utility gain delivered by the proposed distributed algorithm. Consider a tree T_j at iteration t, assume that the member set of T_j is $V_t(T_j)$ prior to iteration t. Let $v_1, v_2, \ldots, v_k$ be the nodes added to T_j after iteration t, then the estimated gain of the utility in T_j is $\sum_{i=1}^{k} U(V_t(T_j) \cup \{v_i\})$ when these nodes joined it. The actual increase on the utility gain in tree T_j however is $U(V_t(T_j) \cup \{v_i \mid 1 \leq i \leq k\}) \leq \sum_{i=1}^{k} U(V_t(T_j) \cup \{v_i\})$. The detailed implementation of Algorithm `Distributed_Implement` consists of two subroutines `Distributed_Implement_Base_Station` as Algorithm 2 and `Distributed_Implement_Sensor` as Algorithm 3.

Algorithm 2 `Distributed_Implement_Base_Station`

```
Broadcast an initial message which contains the following information: L trees with each having root at it, its coverage quality utility value, and its members;
while Receive "joining-tree request" messages from its broadcast tree nodes do
    if No "joining-tree request" messages are received or all nodes are included in the forest then
        Terminate; /*The sensor schedules are finalize*/
    else
        Process received requests by removing redundancies. That is, for a given tree T_j, there may have multiple joining requests originated from the same sensor, then only one of them will join;
        Broadcast the updated broadcast message which contains the updated tree nodes and the value of coverage quality along the broadcast tree edges to each tree node; /* Start next iteration */
    end if
end while
```

Lemma 1: Algorithm `Distributed_Implement` delivers a feasible solution to the coverage maximization problem.

Proof: Since algorithm `Distributed_Implement` consists of a number of iterations, we show that the final L trees in the forest is a feasible solution to the problem by induction on the number of iterations. At iteration $t = 0$, there are L trees with each containing a root node only. It is a feasible solution. Let $\mathcal{F}_t$ be the forest of the L trees constructed so far by iteration $t - 1$, in which each tree meets the following conditions: (1) there is no more than one copy

Algorithm 3 `Distributed_Implement_Sensor`

```
while Receive a broadcast message from its neighbor nodes or the base station do
    if It is already a tree node then
        Broadcast this message to its children nodes or other neighbor nodes;
    else if Its "joining-tree request" in the previous round has been admitted then
        Label itself as a tree node;
        Broadcast this message to its neighbor nodes;
    else
        Identify which tree that it should join through computing the utility gain of the coverage quality if it is added to the tree, and choose a tree with the maximum gain of the utility;
        Send a "joining-tree request" message to its parent node;
    end if
end while
while Receive "joining-tree request" messages from other neighbor nodes or its children nodes do
    Forward the received messages along its tree paths towards its parent nodes;
end while
```

of each sensor in each tree; (2) the communication subgraph induced by the sensor nodes in each tree and the base station (the tree root) is connected. We now deal with iteration t. Within iteration t, some unlabeled nodes (or non-tree nodes) join the trees in $\mathcal{F}_t$. Clearly, if another copy of a joining node is already in a tree, it will not be added to the tree. Or, if there are multiple copies of a sensor seeking to join a tree, only one of them will succeed. Also, there must have an edge in G' between a tree node and the joining node. Thus, the resulting forest $\mathcal{F}_{t+1}$ is still feasible. When no positive utility gain of the coverage quality can be obtained at iteration t, this implies that the trees containing the neighbors of each node $v \in W_t$ have already contained another copy of the sensor that node v is one of its copies. The lemma then follows. □

Theorem 2: Given an energy harvesting sensor network $G = (V \cup \{s\}, E)$ deployed to monitor a set of targets for a period of L time slots, there is a distributed algorithm `Distributed_Implement` for the coverage maximization problem, which takes $O(L|V|+|V|^2)$ time and $O(L|V|^2+|E|)$ messages, where $|V|$ is the number of sensors and $|E|$ is the number of links in G.

Proof: Following Lemma 1, it can be seen that algorithm `Distributed_Implement` will deliver a feasible solution to the coverage maximization problem. Assume that there are l iterations of the entire algorithm. Within iteration i, the amount of time spent for the message broadcasting of the L trees is $\max\{L, t_i\}$ by broadcasting the L tree messages along the tree edges of the broadcast tree in a pipeline manner, where t_i is the longest one among the shortest distances between the base station and a node in W_t at iteration i, clearly $t_i \leq |V|$, $1 \leq i \leq l$. The time for collecting the "joining-tree request" messages from joining nodes in W_t through the tree edges is t_i, The number of messages needed for iteration i thus is $m_i = O(L(n_i - 1) + |E_i|) = O(L|V| + |E_i|)$, where n_i is the number of nodes in the broadcast tree of the host graph at iteration i. There are l iterations of the distributed implementation of the proposed algorithm, thus, the time complexity of the distributed implementation of the proposed algorithm is $O(\sum_{i=1}^{l} \max\{L, t_i\}) = O(\sum_{i=1}^{l} \max\{L, |V|\}) = O(\max\{L|V|, |V|^2\}) = O(L|V| + |V|^2)$ since $l \leq |V|$. Similarly, the number of messages needed by the distributed implementation of the proposed algorithm is $O(\sum_{i=1}^{l} m_i) = O(\sum_{i=1}^{l}(L|V|+|E_i|)) = O(L|V|^2+\sum_{i=1}^{l} |E_i|) = O(L|V|^2 + |E|)$ since $\sum_{i=1}^{l} |E_i| = |E|$. The theorem then follows. □

VI. DYNAMIC OPTIMIZATION FRAMEWORK FOR ENERGY PREDICTION FLUCTUATION

The proposed centralized and distributed algorithms so far for the coverage maximization problem are based an assumption. That is, the energy budget of each sensor for the entire monitoring period of L time slots can be accurately predicted. In reality, the accuracy of energy prediction however depends heavily on weather conditions and the prediction duration. Particularly, a longer period prediction usually is less accurate. The assumption thus is problematic in realistic applications, and especially for sensors whose actual amounts of harvested energy are significantly less than their predicted amounts, they may not have enough energy to maintain their scheduled activities for the monitoring period. Moreover, other active sensors with sufficient energy may also be inversely affected by these sensors when they serve as relay nodes between the base station and the sensors with sufficient energy. Consequently, the overall coverage quality of the network will drastically degrade. To remove or eliminate this realistic assumption, in this section we propose an adaptive framework to deal with harvesting energy prediction fluctuations, and show that under this adaptive framework, the proposed centralized and distributed algorithms are still applicable.

The basic idea is that we schedule sensor activities by a "dynamic interval" concept, where an interval consists of the number of consecutive time slots that is significantly less than L, while the length of an interval is adaptively determined by the energy prediction accuracy so far. Thus, the entire monitoring period of L time slots consists of a number of intervals, and the proposed algorithm `Greedy_Heuristic` or `Distributed_Implement` is applied within each of these intervals. The only modification to these algorithms is that we cannot fully make use of all predicted energy budget for this interval, as the sensors in future intervals may not be recharged again. Instead, we only use a fraction γ of the energy budget for the current interval, e.g., $0.4 \leq \gamma \leq 0.8$. Specifically, let $|I_i|$ be the number of time slots in an interval I_i. In the beginning of interval I_i, we first compute the amount of predicted energy of each sensor in this interval, by applying a given prediction algorithm EWMA in [9]. We then schedule sensor activities within the interval by applying algorithm `Greedy_Heuristic` (or algorithm `Distributed_Implement`). Given an interval I_i, let $V(I_i)$ be the set of active sensors in I_i. The energy prediction accuracy of a sensor $v \in V(I_i)$ in I_i, $\theta_i(v)$ is defined as $\theta_i(v) = \frac{|Q_v - \bar{Q}_v|}{Q_v}$, where Q_v and $\bar{Q}_v$ are the actual and predicted amounts of harvested energy of sensor v in I_i. Denote by $\theta_i = \sum_{v \in V(I_i)} \theta_i(v)/|V(I_i)|$ *the energy prediction accuracy* of interval I_i, which is the average energy prediction accuracy among active sensors in this interval. We adaptively adjust the number of time slots $|I_{i+1}|$ for the next interval I_{i+1} by the energy prediction accuracy θ_i in I_i, and the number of time slots $|I_{i+1}|$ for the next interval I_{i+1} is defined as follows.

$$|I_{i+1}| = \begin{cases} \max\{1,\ \lfloor |I_i| \cdot \beta \rfloor\} & \theta_i \geq \epsilon \\ \min\{L_{ini},\ \lfloor \frac{|I_i|}{\beta} \rfloor,\ L'\}, & \text{otherwise} \end{cases} \tag{14}$$

where β is a tuning rate with the default value of 0.5 in the rest of paper with $0 < \beta \leq 1$, L_{ini} is a given initial value with the default value of $\lceil 0.2 \cdot L \rceil$, and $L' \leq L$ is the remaining available number of time slots for a monitoring period of L time slots, i.e., $L' \leq L$. That is, when the energy prediction in interval I_i is quite accurate (i.e., the value of θ is less than a given threshold ϵ), the number of time slots $|I_{i+1}|$ is increased for the next interval I_{i+1} by setting $|I_{i+1}| = \frac{|I_i|}{\beta}$ until it is either L_{ini} or L'; otherwise, the number of time slots is decreased by setting $|I_{i+1}| = |I_i| \cdot \beta$ until it decreases to 1. Thus, the entire monitoring period of L time slots consists of a number of variable-length intervals. This procedure continues until all the L time slots have been scheduled. The detailed adaptive optimization framework for the quality coverage maximization problem is described in `Algorithm` 4.

Notice that in terms of the energy budget allocation to the current interval I_k in Algorithm `Adaptive_Framework`, only a fraction of the energy budget $P^k(v_i)$ of each sensor $v_i \in V$ is allocated to interval I_k. The rationale behind is that we need to keep some residual energy of the sensor for later intervals if no further energy can be harvested in future intervals (such as obtaining the solar energy in the middle of night).

Theorem 3: Given an energy harvesting sensor network $G = (V \cup \{s\}, E)$ deployed to monitor a set of targets in the region for a period of L time slots, there is an algorithm `Adaptive_Framework` for the coverage maximization problem, which takes $O(b_{max}^3|V|^2|E| + d_{max}b_{max}L)$ time, where $|V|$ is the number of sensors, where $b_{max,i} = \max_{v_j \in V}\{b_j\}$ at interval I_i, $b_{max} = \sum_{i=1}^{l} b_{max,i}$ and $d_{max} = |N(v)|$ and $N(v)$ is the set of neighbors of node v in G, assuming that there are l intervals to cover the entire monitoring period of L time slots. Notice that d_{max} usually is a constant while b_{max} is a constant and even if it is not, then $b_{max} \ll L$.

Algorithm 4 `Adaptive_Framework`

Input: An energy harvesting sensor network $G = (V \cup \{s\}, E)$, a set of targets O, and time slots that are indexed by $1, 2, \ldots, L$.

Output: Schedule sensor activities in entire L time slots.

$\beta \leftarrow 0.5$; $L_{ini} \leftarrow \lceil 0.2 \cdot L \rceil$; /* These settings can be changed according to specific requirements */
$|I_1| \leftarrow L_{ini}$; /* Initial the first interval */
$L' \leftarrow L$; /* The remaining number of time slots for the entire of L time slots */
/* Schedule sensors' activities interval by interval */
/*Assume that the current interval is I_k with $k \geq 1$*/
while $L' > 0$ **do**
  for each sensor $v_i \in V$ **do**
    Predict the amount of energy harvested of v_i in the current interval I_k;
    Compute its energy budget $P^k(v_i)$ by Eq. (2);
    The amount energy budget allocated for the current interval I_k is $\gamma B^k(v_i)$ where γ is a constant with $0.4 \leq \gamma < 1$, e.g., $\gamma = 0.5$
  end for;
  Schedule sensor activities within the current interval I_k by invoking algorithm `Greedy_Heuristic` (or algorithm `Distributed_Implement`). Notice that in the construction of the auxiliary graph, instead of L trees rooted at s_j with $1 \leq j \leq L$, there are $|I_k|$ trees rooted at s_j^k with $1 \leq j \leq |I_k|$, the budget of each sensor v_i now is b_i^k in the current interval I_k.
  $L' \leftarrow L' - |I_k|$; /* Update the remaining available number of time slots */
  /* In the end of the current interval, examine the energy prediction accuracy θ in the current interval; adjust the number of time slots in the next interval according to the energy prediction accuracy by Eq. (14) */
  if $\theta_k \geq \epsilon$ **then**
    $|I_{k+1}| \leftarrow \max\{1,\ \lfloor |I_k| \cdot \beta \rfloor\}$; /* decrease the number of time slots in the next interval */
  else
    $|I_{k+1}| \leftarrow \min\{L_{ini},\ L',\ \lfloor \frac{|I_k|}{\beta} \rfloor\}$; /* increase the number of time slots in the next interval */
  end if
end while.

Proof: Following Theorem 1, it can be seen that algorithm `Adaptive_Framework` will deliver a feasible solution to the coverage maximization problem. Assume that there are l intervals of the entire monitoring period of L time slots, denoted by $I_1, I_2, \ldots, I_l$, respectively. Let I_i be the ith interval with $1 \leq i \leq l$, i.e., $\sum_{i=1}^{l} |I_i| = L$. Let $b_{max,i}$ be the maximum number of energy budget among sensors at interval i. Thus, algorithm `Greedy_Heuristic` will be invoked l times, and the amount of time taken by each of its invoking is $O(b_{max,i}^3|V|^2|E| + d_{max}b_{max,i}|I_i|)$ for interval I_i. The algorithm `Adaptive_Framework` consists of l intervals, thus, its time complexity is

$$O(\sum_{i=1}^{l}(b_{max,i}^3|V|^2|E| + d_{max}b_{max,i}L))$$
$$= O(|V|^2|E|(\sum_{i=1}^{l} b_{max,i})^3 + d_{max}L(\sum_{i=1}^{l} b_{max,i}))$$
$$= O(b_{max}^3|V|^2|E| + d_{max}b_{max}L), \quad (15)$$

where $b_{max} = \sum_{i=1}^{l} b_{max,i}$. □

The distributed implementation of algorithm `Distributed_Implement` is similar to the one in the previous section, omitted.

VII. PERFORMANCE EVALUATION

In this section, we study the performance of the proposed algorithms through experimental simulation. We also investigate the impact of related parameters: network size, number of targets, tuning rate β, threshold ϵ, and parameter γ on the coverage quality.

A. EXPERIMENTAL ENVIRONMENT SETTING

We consider an energy harvesting sensor network consisting of 100 to 500 sensors randomly deployed in a $100m \times 100m$ square region, where a base station is randomly located. The targets in O are also randomly deployed in this square region. We consider a monitoring period of 24 hours with each time slot of 30 minutes, i.e., the monitoring period consists of $L = 48$ time slots. We adopt the energy consumption parameters of real radio CC2420 [5], which consumes $56.4mW$ and $0.06mW$ when it is in active and sleep modes, respectively. Each sensor is powered by a solar panel with a dimension $10mm \times 10mm$. The solar power harvesting profile is derived from the solar data profiles in The National Solar Radiation Data Base (NSRDB) in the States [17], which contains the most comprehensive collection of solar data. Specifically, for each different network topology for a one day monitoring period, each sensor node is assigned a solar data sequence of one day. Each data item in the sequence is the amount of energy harvested in that 30-minute time slot of that day. For the sake of convenience, we assume that both the base station and sensor nodes have identical transmission ranges of 20 and sensing ranges of 25 meters. We further assume that the given coverage quality weight α is 0.5 in the default setting. Denote by `LOG` a utility function which is the sum of two sub-modular functions: $f_1(N_c^o) = \log(|N_c^o| + 1)$ and $f_2(S_o^t) = \log(|S_o^t| + 1)$. Similarly, denote by `SQR` another utility function which is the sum of two sub-modular functions: $f_1(N_c^o) = \sqrt{|N_c^o|}$ and $f_2(S_o^t) = \sqrt{|S_o^t|}$. We will adopt these two different utility functions to measure the target coverage quality. Each value in figures is the mean of the results by applying each mentioned algorithm to 30 different network topologies with the same network size.

B. PERFORMANCE EVALUATION OF CENTRALIZED AND DISTRIBUTED ALGORITHMS ON THE COVERAGE QUALITY

We first investigate the proposed centralized algorithm `Greedy_Heuristic` and the distributed implementation `Distributed_Implement`, against a variant of an existing centralized algorithm in [8] `CPS_Cover` that finds such a connected sensor cover that maximizes the number of targets covered at each time slot. The number of sensors varies from 100 to 500, and the number of targets $|O|$ is set as 25 and 50, respectively.

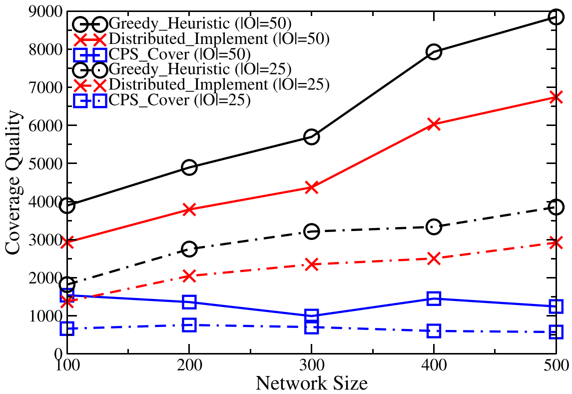
Fig. 4(a) clearly shows that in terms of the coverage quality function `SQR`, the centralized algorithm `Greedy_Heuristic` significantly outperforms algorithms `Distributed_Implement` and `CPS_Cover`, and algorithm `CPS_Cover` is the worst among all three mentioned algorithms. The coverage quality of algorithm `Greedy_Heuristic` is around 30% higher than that of algorithm `Distributed_Implement`, regardless of the number of targets $|O|$ is either 25 or 50. With the growth of network size, this performance gap is still stable. The coverage quality delivered by algorithms `Greedy_Heuristic` and `Distributed_Implement` is at least 100% more than that of algorithm `CPS_Cover`. For the coverage quality function `LOG`, Fig. 4(b) exhibits similar performance behaviors, and the coverage quality delivered by algorithm `Greedy_Heuristic` is about 50% higher than that by algorithm `Distributed_Implement`. With the increase of network size, it can be also seen from Fig. 4 that the coverage quality delivered by algorithms `Greedy_Heuristic` and `Distributed_Implement` increases accordingly. The coverage quality delivered by both algorithms increase too when the number of targets increases, while keeping the network size fixed.

C. IMPACT OF TUNING RATE β ON THE PERFORMANCE OF ALGORITHM DYNAMIC FRAMEWORK

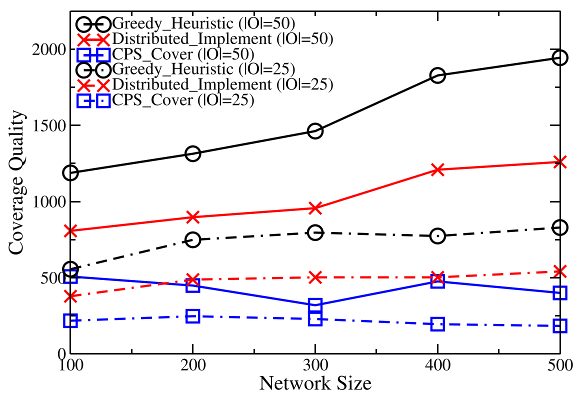
We then study the efficiency of the proposed dynamic optimization framework `Adaptive_Framework`, where algorithm `Greedy_Heuristic` is employed as its subroutine. We fix the threshold ϵ at 0.2 and the parameter γ at 0.5 while putting the tuning rate β as 0.2, 0.5, and 0.8, respectively.

Fig. 5 demonstrates that the coverage quality delivered by algorithm `Adaptive_Framework` is the highest in comparison with the other settings when the tuning rate $\beta = 0.5$. For example, when the number of targets is fixed at 25, for the coverage quality function `SQR` in Fig. 5(a), the coverage quality delivered by the algorithm when $\beta = 0.5$ is about 5% and 6% higher than that by the algorithm when $\beta = 0.2$ and $\beta = 0.8$, respectively. For the coverage quality function `LQG` in Fig. 5(b), the coverage quality delivered by the algorithm when $\beta = 0.5$ is about 9% and 8% higher than that by it when $\beta = 0.2$ and $\beta = 0.8$, respectively.

IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

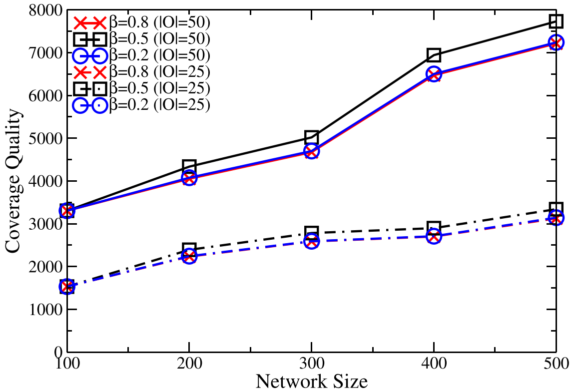


(a)

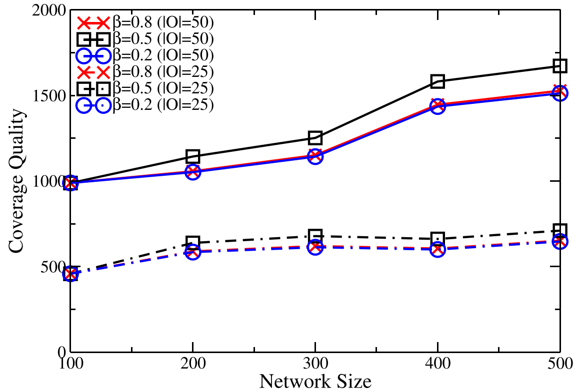


(b)

FIGURE 4. Performance of centralized algorithm `Greedy_Heuristic` and distributed algorithm `Distributed_Implement` under different quality measure functions *SQR* and *LOG*. (a) `SQR-metric`. (b) `LOG-metric`.

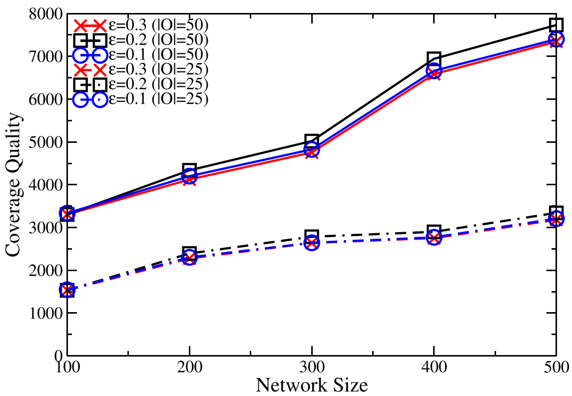


(a)

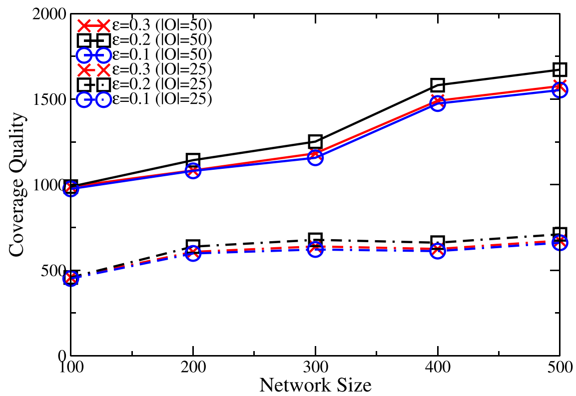


(b)

FIGURE 5. Impact of tuning rate β on the performance of algorithm `Adaptive_Framework` under different quality measure functions *SQR* and *LOG*. (a) `SQR-metric`. (b) `LOG-metric`.



(a)



(b)

FIGURE 6. Impact of threshold ϵ on the performance of algorithm `Adaptive_Framework` under different quality measure functions *SQR* and *LOG*. (a) `SQR-metric`. (b) `LOG-metric`.

D. IMPACT OF THRESHOLD ϵ ON THE PERFORMANCE OF ALGORITHM DYNAMIC FRAMEWORK

We thirdly evaluate the impact of threshold ϵ on the coverage quality delivered by the proposed framework `Adaptive_Framework`, in which the subroutine `Greedy_Heuristic` is employed. We set the threshold ϵ as 0.1, 0.2, and 0.3 while fixing the tuning rate β at 0.5 and parameter γ at 0.5.

Fig. 6(a) indicates that for the coverage quality function `SQR`, the coverage quality achieved by algorithm

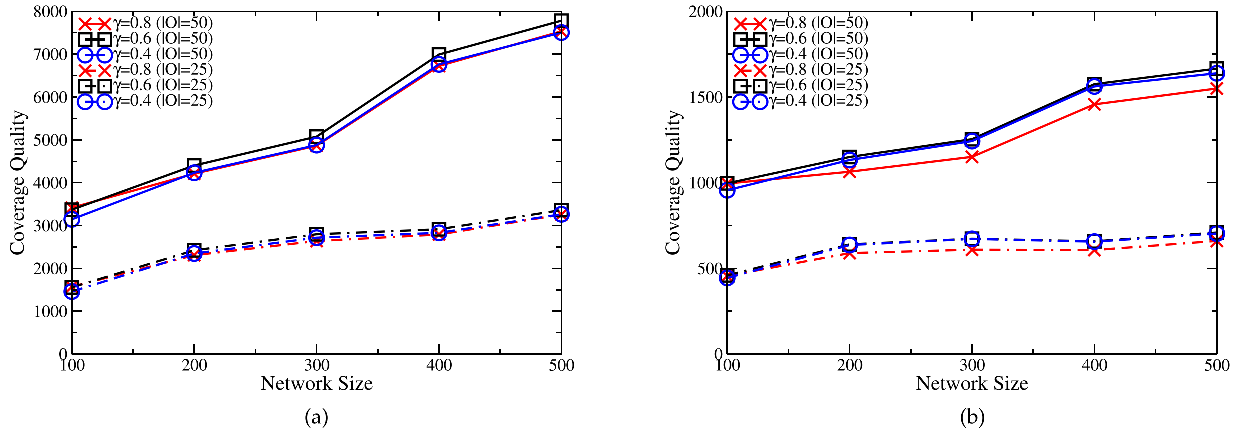


FIGURE 7. Impact of parameter γ on the performance of algorithm `Adaptive_Framework` under different quality measure functions *SQR* and *LOG*. (a) `SQR-metric`. (b) `LOG-metric`.

`Adaptive_Framework` is the highest compared with those of other settings when $\epsilon = 0.2$. Specifically, when the number of targets is fixed at 50, the coverage quality delivered by the algorithm with $\epsilon = 0.2$ is about 4% and 5% higher than those by it with $\epsilon = 0.1$ and $\epsilon = 0.3$, respectively. When the number of targets is fixed at 25, the coverage quality delivered by the algorithm with $\epsilon = 0.2$ is about 5% higher than that by it with $\epsilon = 0.1$ or $\epsilon = 0.3$. Fig. 6(b) exhibits the similar performance behaviors for the coverage quality function `LOG`, omitted.

E. IMPACT OF PARAMETER γ ON THE PERFORMANCE OF ALGORITHM DYNAMIC FRAMEWORK

We finally evaluate the impact of parameter γ on the coverage quality delivered by the proposed framework `Adaptive_Framework`, in which the subroutine `Greedy_Heuristic` is employed. We set parameter γ as 0.4, 0.6, and 0.8 while fixing the tuning rate β at 0.5 and the threshold ϵ at 0.2, respectively.

Fig. 7(a) implies that for the coverage quality function `SQR`, the coverage quality delivered by algorithm `Adaptive_Framework` with $\gamma = 0.6$ is higher than that by it with $\gamma = 0.4$ or $\gamma = 0.8$. Specifically, when the number of targets is fixed at 50, the coverage quality delivered by algorithm `Adaptive_Framework` with $\gamma = 0.6$ is about 3.5% higher than that by it with $\gamma = 0.4$ or $\gamma = 0.8$. When the number of targets is fixed at 25, the coverage quality delivered by the algorithm with $\gamma = 0.6$ is about 3% higher than that by it with $\gamma = 0.4$ or $\gamma = 0.8$. Fig. 7(b) exploits the performance behavior curves of algorithm `Adaptive_Framework` for the coverage quality function `LOG`. The coverage quality delivered by it with $\gamma = 0.4$ and $\gamma = 0.6$ is higher than or at the same level as that by the algorithm with $\gamma = 0.8$.

VIII. CONCLUSION

In this paper we studied the quality-aware target coverage problem in an energy harvesting sensor network deployed for monitoring a set of targets for a given monitoring period, where sensors are powered by renewable energy sources and operate in duty-cycle mode, for which we first introduced a new coverage quality metric that is a weighted linear combination of two utility sub-modular functions to measure the coverage quality within two different time scales. We then formulated a novel coverage maximization problem that takes both sensing coverage quality and network connectivity into consideration. Due to the NP-hardness of the problem, we instead devised efficient centralized and distributed algorithms, provided that the harvesting energy prediction of each sensor for the monitoring period is accurate. Otherwise, we proposed an adaptive framework to deal with energy prediction fluctuations. We finally evaluated the performance of the proposed algorithms through experimental simulations. Experimental results demonstrate that the proposed solutions are promising.

ACKNOWLEDGEMENTS

The authors appreciate the anonymous referees for their critical comments and constructive suggestions which have helped improve the quality and presentation of this paper significantly.

REFERENCES

[1] H. M. Ammari and S. K. Das, "Centralized and clustered k-coverage protocols for wireless sensor networks," *IEEE Trans. Comput.*, vol. 61, no. 1, pp. 118–133, Jan. 2012.

[2] A. Cammarano, C. Petrioli, and D. Spenza, "Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks," in *Proc. IEEE 9th Int. Conf. MASS*, Oct. 2012, pp. 75–83.

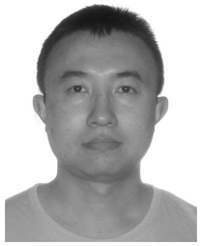
[3] M. Cardei, M. T. Thai, Y. Li, and W. Wu, "Energy-efficient target coverage in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2005, pp. 1976–1984.

[4] M. Cardei and J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," *Comput. Commun.*, vol. 29, no. 4, pp. 413–420, Feb. 2006.

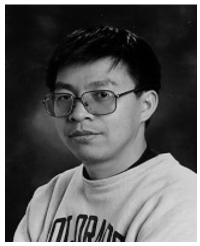
[5] *CC2420 RF Transceiver*. [Online]. Available: http://www.ti.com/lit/gpn/cc2420, accessed 2014.

[6] H. Dai, X. Wu, L. Xu, and G. Chen, "Practical scheduling for stochastic event capture in wireless rechargeable sensor networks," in *Proc. IEEE WCNC*, Apr. 2013, pp. 986–991.

[7] L. Ding, W. Wu, J. Willson, L. Wu, Z. Lu, and W. Lee, "Constant-approximation for target coverage problem in wireless sensor networks," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1584–1592.
[8] H. Gupta, S. R. Das, and Q. Gu, "Connected sensor cover: Self-organization of sensor networks for efficient query execution," in *Proc. ACM MobiHoc*, Jun. 2003, pp. 189–200.
[9] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, Sep. 2007, Art. ID 32.
[10] W. Liang, X. Ren, X. Jia, and X. Xu, "Monitoring quality maximization through fair rate allocation in harvesting sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1827–1840, Sep. 2013.
[11] C. Liu and G. Cao, "Spatial-temporal coverage optimization in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 4, pp. 465–478, Apr. 2011.
[12] C. Liu and G. Cao, "Distributed critical location coverage in wireless sensor networks with lifetime constraint," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1314–1322.
[13] R.-S. Liu, K.-W. Fan, Z. Zheng, and P. Sinha, "Perpetual and fair data collection for environmental energy harvesting sensor networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 947–960, Aug. 2011.
[14] H. Liu, X. Jia, P.-J. Wan, C.-W. Yi, S. K. Makki, and N. Pissinou, "Maximizing lifetime of sensor surveillance systems," *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, pp. 334–345, Apr. 2007.
[15] Y. Liu and W. Liang, "Approximate coverage in wireless sensor networks," in *Proc. IEEE Conf. LCN*, Nov. 2005, pp. 68–75.
[16] Y. Liu and W. Liang, "Prolonging network lifetime for target coverage in sensor networks," in *Proc. WASA*, LNCS, vol. 5258. 2008, pp. 212–223.
[17] *The National Solar Radiation Data Base*. [Online]. Available: http://rredc.nrel.gov/solar/old-data/nsrdb/1991-2010, accessed 2014
[18] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, to be published. [Online]. Available: http://dx.doi.org/10.1109/TC.2014.2349521
[19] V. Pryyma, D. Turgut, and L. Bölöni, "Active time scheduling for rechargeable sensor networks," *Comput. Netw.*, vol. 5, no. 4, pp. 631–640, Mar. 2010.
[20] S. Tang, M. Li, X. Shen, J. Zhang, G. Dai, and S. K. Das, "Cool: On coverage with solar-powered sensors," in *Proc. IEEE ICDCS*, Jun. 2011, pp. 488–496.
[21] S. Tang and J. Yuan, "DAMson: On distributed sensing scheduling to achieve high quality of monitoring," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 155–159.
[22] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Comput. Surveys*, vol. 43, no. 4, Oct. 2011, Art. ID 32.
[23] L. Wu, H. Du, W. Wu, D. Li, J. Lv, and W. Lee, "Approximations for minimum connected sensor cover," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 1187–1194.
[24] C. Yang and K.-W. Chin, "Novel algorithms for complete targets coverage in energy harvesting wireless sensor networks," *IEEE Commun. Lett.*, vol. 18, no. 1, pp. 118–121, Jan. 2014.
[25] D. K. Y. Yau, N. K. Yip, C. Y. T. Ma, N. S. V. Rao, and M. Shankar, "Quality of monitoring of stochastic events by periodic and proportional-share scheduling of sensor coverage," *ACM Trans. Sensor Netw.*, vol. 7, no. 2, Aug. 2010, Art. ID 8.



XIAOJIANG REN (S'12) received the B.E. and M.E. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Research School of Computer Science, Australian National University, Canberra, ACT, Australia. His research interests include wireless sensor networks, routing protocol design for wireless networks, and optimization problems.



WEIFA LIANG (M'99–SM'01) received the Ph.D. degree from the Australian National University, Canberra, ACT, Australia, in 1998, the M.E. degree from the University of Science and Technology of China, Hefei, China, in 1989, and the B.Sc. degree from Wuhan University, Wuhan, China, in 1984, all in computer science. He is currently an Associate Professor with the Research School of Computer Science, Australian National University. His research interests include design and analysis of energy-efficient routing protocols for wireless ad hoc and sensor networks, cloud computing, graph databases, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory.

WENZHENG XU received the B.Sc. and M.E. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008 and 2010, respectively, where he is currently pursuing the Ph.D. degree and is a visiting student with Australian National University, Canberra, ACT, Australia. His research interests include routing algorithms and protocols design for wireless ad hoc and sensor networks, approximation algorithms, combinatorial optimization, and graph theory.