Energy Allocation for Stochastic Event Detection in Rechargeable Sensor Networks

Fanggai Ye
School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China;
Phone: +86 136 668 41534; e-mail: 13666841534@163.com

Xiuying Zhou
School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China;
e-mail: 201930604012@mails.zstu.edu.cn

Zhu Ren, Shan Yang
School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China;
e-mail: zhuren@zstu.edu.cn, 2769830276@qq.com

Corresponding author: zhuren@zstu.edu.cn

Wireless rechargeable sensor networks have been applied to all aspects of the real world today. Though sensors can collect energy from the environment, the energy collection cannot support sensors to work continuously as usual. Energy scheduling problems have to be solved. In this paper, we study the energy allocation problem of a rechargeable sensor network that can monitor multiple random events. It is assumed that each event follows a Poisson process, the energy received by the sensor is random, and each sensor has a chance to be assigned to detect one or more events. In the paper, we also introduce multi-objective nonlinear programming to solve the problems of nonlinearity and energy. Two algorithms are also proposed to obtain the programming’s Pareto optimal solution. At last, we conduct a number of practical simulations to verify our results.

KEYWORDS: Rechargeable sensor network, Multiple events, Pareto optimal, Nonlinear programming, Algorithms, Cybernetics.
1. Introduction

Wireless sensor network (WSN) is one of the hot-spots in the field of information, which is widely used technically in real life. WSN can be deployed in various spaces including the most hazardous working environment, and plays a positive role in agricultural production assistance, eco-environmental monitoring, and military. The sensor node can be used to replace part of the staff for tasks in a hazardous environment. WSN’s energy scheduling is still an important issue.

In this paper, we study a rechargeable wireless sensor network. Due to the special application environment of certain sensors, the maintenance cost of the sensor network is very expensive, requiring the sensors to collect energy from the environment and be self-sufficient. However, the energy collection is sufficient to support the continuous work of the sensor in most cases. In this case, we should dispatch the sensor network and get the best sensor network scheduling strategy.

We assume that each event follows a Poisson process. Multiple rechargeable sensors are randomly deployed in the area. Taking the variety of the geographical environment into consideration, each sensor can be assigned to detect one or more events.

As the ambient energy changes with time, the energy received by the sensor is also random. Our objective is to dispatch the energy of the network so that the quality of monitoring (QoM) can be maximized.

Our contributions are summarized as follows:

1. We consider an energy distribution problem for a rechargeable sensor network where multiple stochastic events are monitored.

2. The problem of nonlinearity and energy constraints are discussed, and multi-objective nonlinear programming is introduced to solve this problem. Considering the relation between energy and detection rates, we propose two heuristic algorithms which are proven to be optimal methods.

3. We conduct several simulations to verify our results, especially the iterations in our proposed algorithms and the differences among several deployments of the sensors and events.

2. Related Work

First, we discuss the general wireless sensor network problem. A network coverage algorithm was studied by Li et al. [9] based on evidence theory, which calculates the direction of movement of the wireless sensor node and moves the wireless sensor node to an area with low perception probability. Singh et al. [18] proposed a sleep scheduling algorithm, namely, EC-CKN to balance the energy consumption and extended network life. Nguyen et al. [13] studied a more general target coverage and network connection problem, termed the Maximum Weighted Target Coverage and Sensor Connectivity with Limited Mobile Sensors (TAR-CC) problem. To solve the sub-problems of the TAR-CC, an approximate algorithm is proposed, i.e., the weighted-maximum-coverage-based algorithm (WMC BA) which is used as the basis to propose the Steiner-tree-based algorithm for the TAR-CC problem.

In this paper, we study the rechargeable wireless sensor network. Hung et al. [6] studied a distributed collaboration algorithm suitable for partially rechargeable mobile wireless sensor networks. The algorithm considers not only the energy consumption of the mobile node but also the resident energy of the mobile node. Besides, it cooperates with neighbors to extend the life cycle of environmental monitoring. Han et al. [5] introduced wireless mobile chargers to supplement energy for nodes to solve the problem of energy limitation in wireless sensor networks fundamentally. A joint energy supplement and data acquisition algorithm for WRSNs is proposed. Deng et al. [2] studied the problem of maximizing network utility in a static route rechargeable sensor network with link and battery capacity constraints, and proposed a method named decouple spatiotemporally-coupled constraint algorithm. Zhu et al. [26] proposed a new type of routing tree, namely, event detection tree to achieve energy-efficient composite event detection, thereby achieving a tradeoff between them to minimize the overall energy consumption. Zou et al. [27] proposed an optimal reader power for balanced energy charging and transmission collision. Wang et al. [22] presented an Improved Cuckoo Search (ICS) algorithm which redefines its step factor based on the traditional cuckoo search algorithm (CS). It then uses the mutation factor to change the nesting position of the host bird to update the nest position before utilizing ICS to find those available to maximize the reception of the sensor node’s power, and the best solution to minimize the number of charger nodes. Han et al. [4] proposed a grid joint routing and charging algorithm for industrial wireless charging sensor networks.

We focus on the dynamic activation of the sensor. Yin et al. [24] studied the performance of a simple threshold activation strategy, and the optimal thresh-
old strategy can be used to achieve at least 3/4 optimization of the situations in which the sensor coverage area is completely overlapped. Rout et al. [17] proposed a handover algorithm based on the Markov decision process to find the best handover strategy for sensor nodes. While reducing energy consumption in the network, it also uses real-time sensor flow patterns to analyze energy consumption. Zhang et al. [25] considered the data sensing and data transmission, optimized the network utility data acquisition, and designed the dynamic sensing and routing data acquisition optimization algorithm. Liu et al. [10] proposed two reasonable charging strategies and a variable-step size adaptive algorithm to optimize the entire wireless rechargeable sensor network. Liu et al. [11] aimed to jointly optimize the number of dead zone sensors and energy efficiency in this multi-node, they also proposed a multi-node temporal-spatial partial-charging algorithm (MTSPC) to solve the conflict between minimizing the number of dead zone sensors and energy efficiency due to partial charging mechanisms. Malebary [12] introduced the optimization (WMCEO) algorithm to achieve enhanced energy efficiency and network life by optimizing the movement trajectory and charging time of WMC at each stay position. Tang et al. [19] proposed an optimization algorithm for both the charging process and routing process. To balance the network energy of the charging part, the charging efficiency of the node is balanced by dynamically planning the location of the charging point, and the charging time is allocated according to the energy consumption rate of the node. Jiang et al. [8] studied the use of mobile chargers with wireless rechargeable sensors to achieve maximum coverage for on-demand scheduling. Ren et al. [15] considered to detect single-event problems and used dynamic control theory to monitor events after the update process. Wu et al. [23] introduced the concept of virtual time in Heterogeneous Wireless Rechargeable Sensor Network (HWRSN), and then proposed a new online charging algorithm named VTMT. Ge et al. [3] developed a new dynamic event-triggered transmission scheme (ETS) to schedule the transmission of each sensor’s local measurement. Tomar et al. [20] studied a wireless and rechargeable sensor network with multiple chargers, and used fuzzy logic mixing various network attributes to formulate a new WRSNs on-demand charging scheduling strategy. Wang et al. [21] designed a time-varying filter so that both H∞ requirements and the variance constraints are guaranteed over a given finite-horizon against the random parameter matrices. Ouyang et al. [14] proposed an important differential charging scheduling (IDCS) strategy based on matroid theory to improve charging utility and reduce data loss.

3. Problem Formulation

Multiple rechargeable sensors are deployed randomly in a situation for detecting important events. We use \( S_i \) \( i = 1, 2, ..., N \) to denote each sensor and \( I_j \) \( j = 1, 2, ..., M \) to denote events. Owing to the geographical environment effect, when event \( I_j \) is in \( S_i \)’s sensor coverage area, we say \( I_j \) can be monitored by \( S_i \) (see Figure 1). Since environmental energy changes over time (e.g., from solar irradiation, vibration), so does the number of energy arrivals. Consequently, as presented in Jaggi et al. [7], each sensor’s recharge process is modelled as a Poisson process: in each time slot, sensor \( S_i \) will receive units of energy with probability \( 0 < q < 1 \). Then, the recharge rate of \( S_i \) is \( q_c \). The sensor expends a charge of \( \delta \cdot c \) energy when it is in an active state and no energy when it is in a dormant state in each time slot. For event \( I_j \) we use a Poisson process with parameter \( \lambda_j \) to denote its randomness. When sensor \( S_i \) spent \( 0 < \alpha_{ij} < 1 \) ratio energy on event \( I_j \), the total power cost by all sensors on this event is \( \sum_{S_i \in A_j} \alpha_{ij} q_c \), where \( A_j = \{ S_i \mid \text{sensor } S_i \text{ can detect event } I_j \} \). Let \( \alpha = \{ \alpha_{ij} \mid i = 1, ..., M, j = 1, ..., N \} \). We call \( \alpha \) the network’s energy distribution strategy. From Ren et al. [16], we can calculate that the capture probability for \( I_j \) under \( \alpha \) is

\[
D_j(\alpha) = \begin{cases} 
\sum_{i \in A_j} \alpha_{ij} q_c / \delta, & \sum_{i \in A_j} \alpha_{ij} q_c < \delta, \\
1, & \sum_{i \in A_j} \alpha_{ij} q_c \geq \delta.
\end{cases}
\]

Figure 1

The system model of a rechargeable wireless sensor network
In this paper, we study the following problem

**Problem 1**

\[
\max_{\alpha} D_{\lambda,\omega}(\alpha) = \sum_{i=1}^{N} \gamma_i D_i(\alpha),
\]

s.t. \( \gamma_i \geq 0, \sum_{i=1}^{N} \gamma_i = 1, \sum_{j=1}^{\infty} \alpha_{i,j} \leq 1, i = 1, 2, \ldots, N, \)

\[
\alpha_{i,j} = 0, \quad \text{if} \quad I_j \notin \Lambda_i,
\]

\[
\alpha_{i,j} \geq 0
\]

where \( \Lambda_i \) denote all the events that can be detected by sensor \( S_i \).

The solution for **Problem 1**, denoted as \( \alpha' \), is an optimal strategy that maximizes the weighted objective \( D_{\lambda,\omega}(\alpha) \). From Deb [1], \( \alpha' \) is a Pareto optimal solution. That is, there does not exist another solution \( \alpha'' \) such that \( D_i(\alpha') \geq D_i(\alpha'') \) for all events and \( D_i(\alpha') > D_i(\alpha'') \) for at least one \( I_r \).

The commonly used symbols are listed in **Table 1**.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_i )</td>
<td>The energy of sensor ( i )</td>
</tr>
<tr>
<td>( \lambda_i )</td>
<td>The probability of occurrence of event ( j )</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>The network's energy distribution strategy</td>
</tr>
<tr>
<td>( D_i(\alpha) )</td>
<td>The capture probability for events ( I_i ) under ( \alpha )</td>
</tr>
<tr>
<td>( D_{net}(\alpha) )</td>
<td>The weighted capture probability for the network under ( \alpha )</td>
</tr>
</tbody>
</table>

### 4. Unconstrained Case

Since the nonlinearity of Equation (1), **Problem 1** cannot be solved directly. However, from the intuition, we should improve \( D_i(\alpha) \) with the highest weights at the outset in order to improve \( D_{net}(\alpha) \). In other words, each sensor should give more energy to these events with the highest weight. The difficulty in **Problem 1** is that, each event's total received energy would not be larger than \( \delta \). The extra energy can be given to some events with lower weights. Next, following the above intuition, we describe a method we use to solve **Problem 1**.

**Definition 1** To each event pair \( I_i, I_j \), we say vector \( \text{Chain}_{\alpha_i, \alpha_j} = (I_{\alpha_i}, \alpha_{i,j}, S_{\alpha_i}, S_{\alpha_i}, S_{\alpha_i}, I_{\alpha_j}) \) is a chain from \( I_{\alpha_i} \) to \( I_{\alpha_j} \) if, \( I_{\alpha_i}, \alpha_{i,j} \in \Lambda_{S_i}, j = 1, \ldots, m \).

From Chain \( \alpha_i, \alpha_j \), we can see that the energy given to event \( I_{\alpha_i} \) (i.e., \( \sum_{S \in \Lambda_{S_i}} \alpha_{i,j} q_{ij} \)) can be transferred to \( I_{\alpha_j} \) and other events' energy in the chain has no changes. The manipulation is to select a small value \( d_e > 0 \). Then let

\[
\alpha_{i_a, i_b} \leftarrow \alpha_{i_a, i_b} - d_e / (q_{i_a, i_b} c),
\]

\[
\alpha_{i_a, i_b} \leftarrow \alpha_{i_a, i_b} + d_e / (q_{i_a, i_b} c),
\]

\[
\alpha_{i_a, i_b} \leftarrow \alpha_{i_a, i_b} - d_e / (q_{i_a, i_b} c),
\]

\[
\alpha_{i_a, i_b} \leftarrow \alpha_{i_a, i_b} + d_e / (q_{i_a, i_b} c).
\]

The energy can be transferred if all \( \alpha_i \)’s under this manipulation are not less than 0. As a result, event \( I_{\alpha_i} \) has a loss of \( d_e \) energy while event \( I_{\alpha_j} \) receives \( d_e \) energy.

Let us discuss when we shall transfer energy from \( I_{\alpha_i} \) to \( I_{\alpha_j} \). Obviously, if \( \gamma_{\alpha_i} > \gamma_{\alpha_j} \), from the formation of \( D_{net}(\alpha) \), we can transfer energy to increase \( D_{net}(\alpha) \). If \( \gamma_{\alpha_i} < \gamma_{\alpha_j} \), the energy must be transferred only when \( \left( \sum_{j=1}^{N} \alpha_{i,j} q_{ij} \right) > \delta \). That is to say, event \( I_{\alpha_i} \) has redundant energy \( \left( \sum_{j=1}^{N} \alpha_{i,j} q_{ij} \right) - \delta \). However, when there exists one event whose weight is larger than \( I_{\alpha_i} \) and the detective rate is less than 1, the energy must be given to this event firstly for the reason that the gain from this event is greater than \( I_{\alpha_j} \). What’s more, if the chain contains a loop, e.g., \( I_{\alpha_i} = I_{\alpha_j} \), when energy is transferred, \( \alpha_{i_a, i_b} \leftarrow \alpha_{i_a, i_b} - d_e / (q_{i_a, i_b} c), \alpha_{i_b, i_a} \leftarrow \alpha_{i_b, i_a} + d_e / (q_{i_a, i_b} c) \). After these two operations, the energy input to \( I_{\alpha_i} \) is not changed. Thus, the chain we want to find should not contain loops.

Now, we propose an energy allocation algorithm (Algorithm 1) based on the above discussion. The principle is that each sensor gives the energy to the event with the largest weight at first. If this sensor has redundant energy, it gives the rest to the event with the second-largest weight. The procedure goes on until all the energy is assigned.
Algorithm 1. Energy Allocation Algorithm.
1: Input: All sensors and events’ parameters
2: The topology of the sensors and events;
3: Each sensor’s parameter \( q_i, c, \delta, i = 1,..., N \);
4: Each event’s occurrence probability \( p_j \) and
5: its weight \( \gamma_j, j = 1,..., M \).
6: Function INCREASE, which is given in the
7: Appendix A.
8: Output: The energy allocation policy
9: \( \alpha^* = \{\alpha^*_j, i = 1,..., N, j = 1,..., M \} \).
10: Assume \( \alpha^{(0)} \) is a feasible solution and
11: \( \sum_{j \in S_k} \alpha_j^{(0)} = 1, \forall S_k \).
12: Assume \( \gamma_j \geq \gamma_{j-1} \geq \gamma_{j-i} \)
13: for \( k = 1 \rightarrow M \) do
14: \( \alpha^{(k)} = \text{INCREASE}(J_{\alpha}, \alpha^{(k-1)}) \)
15: end for
16: \( \alpha^* = \alpha^{(M)} \).

Algorithm 1 is illustrated in the following steps:

**Step 1.** Initially, we know the parameters of each sensor \((q_i, c, \delta)\), and the probability of each event \(p_j\), weight of each event \( \gamma_j \). 

**Step 2.** Assume that each event weight obeys the following inequality: \( \gamma_j \geq \gamma_{j-1} \geq \gamma_{j-i} \), and \( \alpha^{(0)} \) is a feasible solution.

**Step 3.** In each energy distribution strategy of \( \alpha \), the INCREASE function (see Appendix A) is called, which is mainly to traverse all events and sensors to find whether the weight is greater than the current weight. If \( \gamma_j \leq \gamma_{j-i} \) not exist, it will provide energy for the current sensor event. Otherwise, it will find all the chains whose weight is greater than the current value and provide energy for these events through \( T(\text{Chain}^{\alpha}_{\gamma_j}, \alpha) \).

We can prove the optimality of Algorithm 1.

**Theorem 1.** The allocation policy \( \alpha^* \) derived from Algorithm 1 is an optimal solution for Problem 1.

**Proof.** Proof by contradiction. Assume that there exists policy \( \alpha' \) such that \( D_{\text{Net}}(\alpha') > D_{\text{Net}}(\alpha^*) \). Without loss of generality, assume that \( \gamma_{j-1} \geq \gamma_{j-2} \geq \gamma_{j-3} \). From the structures of \( D_{\text{Net}}(\alpha') \) and \( D_{\text{Net}}(\alpha^*) \), we know there exists one event \( I_j \) such that \( 1 \geq D_{\text{Net}}(\alpha') > D_{\text{Net}}(\alpha^*) \). Then \( D_{\text{Net}}(\alpha') < 1 \).

Among all the sensors in \( \Lambda_j \), some of them can detect events whose weights are larger than \( I_j \) and the others are less than \( I_j \). Let \( Sen = \{S_j | S_j \in \Omega_j \} \) and \( Eve = \{I_i \in \Omega_j, S_j \in \text{Sen}, I_j \neq I_i \} \). Then the events in the set \( Eve \) can be divided into two parts: \( Eve_{\text{low}} = \{I_i | \gamma_i < \gamma_j, I_i \in Eve \} \), \( Eve_{\text{high}} = \{I_i | \gamma_i > \gamma_j, I_i \in Eve \} \). We discuss the proof in the following three cases.

First, \( Eve = Eve_{\text{low}} \), \( Eve_{\text{high}} = \emptyset \). As is shown in Figure 2(a), \( S_j \in \text{Sen}_j \), \( I_j \in Eve_{\text{low}} \). From Algorithm 1, all the energy in \( S_j \) is given to \( I_j \), and none to \( I_j \). Thus, \( D_{\text{Net}}(\alpha') > D_{\text{Net}}(\alpha^*) \) is impossible.

Second, \( Eve = Eve_{\text{high}} \), \( Eve_{\text{low}} = \emptyset \), i.e., as is shown in Figure 2(b), \( S_j \in \text{Sen}_j \), \( I_j \in Eve_{\text{high}} \). From Algorithm 1, all the energy in \( S_j \) is assigned to \( I_j \). When \( D_{\text{Net}}(\alpha') = 1 \), extra energy is given to \( I_j \). Assume that all the sensors in \( Sen \), give \( e \) energy to \( I_j \). Then \( D_{\text{Net}}(\alpha') = e/\delta \). To \( \alpha' \) assume \( D_{\text{Net}}(\alpha') = e'/\delta \), \( e' = e' + e, e', e/0 \), Since this \( e \) is from \( Sen \), which will give energy to \( Eve \), we have \( e = \sum_{j \in Eve} e_j \), where \( e_j = \sum_{r \in Sen} q_r c(\alpha^*_r - \alpha^*_r) > 0 \) i.e., the extra energy \( I_j \) received under \( \alpha' \). Then

\[
\frac{\gamma_j D_{\text{Net}}(\alpha') + \sum_{r \in \text{Sen}} q_r (\alpha_r - \alpha^*_r) - \sum_{r \in \text{Sen}} q_r (\alpha_r - \alpha^*_r)}{\delta} = \frac{\gamma_j e/\delta + \sum_{r \in \text{Sen}} q_r (\alpha_r - \alpha^*_r)}{\delta} \geq 0
\]

where Equation (2) is determined by Algorithm 1, and Equation (3) is derived by \( \sum_{r \in \text{Sen}} q_r c(\alpha^*_r - \alpha^*_r) = -c_j > 0, \gamma_j > \gamma_j \). Then, it is impossible that \( D_{\text{Net}}(\alpha') > D_{\text{Net}}(\alpha^*) \).

At last, \( Eve_{\text{low}} = \emptyset \), \( Eve_{\text{high}} = \emptyset \) (see Figure 2(c)). Each sensor which belongs to \( \text{Sen}_p \) will connect \( I_j \), or \( I_j \). According to Algorithm 1, sensors in \( \text{Sen}_p \) will not give energy to \( Eve_{\text{low}} \). Note that \( D_{\text{Net}}(\alpha') < 1, D_{\text{Net}}(\alpha') > D_{\text{Net}}(\alpha^*) \) means that \( Eve_{\text{high}} \) receive more energy under policy \( \alpha' \) than \( \alpha^* \). Analogous to the previous case, we have
5. Constrained Problem

From Theorem 1, the disadvantage of Problem 1 is that, it cannot guarantee that all the events can receive energy. Some events with high weights will receive enough energy, however, the events with low weights may have no energy and cannot be monitored by any sensor. To avoid this extreme scenario, we introduce \( \eta_j \) to denote event \( I_j \)'s lower bound. Then, we have the next problem.

**Problem 2**

\[
\begin{align*}
\max_{\alpha} D_{\text{net}}(\alpha) &= \sum_{j=1}^{M} \gamma_j D_{I_j}(\alpha), \\
\text{st.} & \quad D_{I_j}(\alpha) \geq \eta_j, \\
& \quad \sum_{j=\lambda_k}^{x} \alpha_{j} \leq 1, \quad i = 1, 2, \ldots, N, \\
& \quad \alpha_{j} = 0, \quad \text{if} \quad I_j \notin \Lambda_k, \\
& \quad \alpha_{j} \geq 0
\end{align*}
\]

where \( 0 \leq \eta_j \leq 1 \). Inequality (4) is the new added constraint. This denotes that the capture probability of event \( I_j \) must be larger than \( \eta_j \). By a simple analysis, we can know that the solution of Problem 2 is also a Pareto optimal. Next, we present an energy-constrained allocation algorithm (Algorithm 2) to solve the new problem.

**Algorithm 2.** Energy-Constrained Allocation Algorithm.

1: **Input:** All sensors and events’ parameters
2: The topology of the sensors and events;
3: Each sensor’s parameter \( q, c, \delta, i = 1, \ldots, N; \)
4: Each event’s occurrence probability \( p_j \) and its
5: weight \( \gamma_j, j = 1, \ldots, M; \)
6: Each event’s constraint \( \eta = \{\eta_j, j = 1, \ldots, M\}. \)
7: Function CHECK, which is given in the
8: AppendixA.
9: **Output:** The energy allocation policy
10: \( \alpha^* = \{\alpha_i^*, i = 1, \ldots, N, j = 1, \ldots, M\}. \)
11: Execute Algorithm 1.
12: Assume \( \gamma_{h_{1}} \geq \gamma_{h_{2}} \cdots \geq \gamma_{h_{N}} \).
13: Call function CHECK(\( \eta \)).

Algorithm 2 is illustrated in the following steps:

**Step 1.** Same as the step 1 of Algorithm 1, we know \( q, c, \delta, p_j, \gamma(j = 1, \ldots, M) \). The difference is that we introduce each event’s constraint \( \eta = \{\eta_j, j = 1, \ldots, M\}. \)

**Step 2.** Execute Algorithm 1 that allocates energy to the higher-weighted ones.

**Step 3.** Assume that each weight obeys the following inequality:
\( \gamma_{h_{1}} \geq \gamma_{h_{2}} \cdots \geq \gamma_{h_{N}} \).

**Step 4.** After executing Algorithm 1, we need to traverse the \( M \) events through the CHECK function (see Appendix A) to find \( D_{I_j}(\alpha) < \eta_j \) that does not meet the constraint condition. Then we find the set whose weight is greater than the current event, that is \( I_{\text{gw}} = \{I_{h_{1}}, I_{h_{2}}, \ldots, I_{h_{x}} | \gamma_{h_{x}} \geq \gamma_{h_{y}} D_{I_j}(\alpha) > \eta_j, i = 1, \ldots, Q\}. \)

**Step 5.** Through the SEARCHWEIGHT function (see Appendix A), we find the high-weight event chain. Through the \( T(\text{Chain}_{h_{x}}, \alpha) \) function (see Appendix A), energy is transferred from high-weight events to low-weight events until \( D_{I_j}(\alpha) = \eta_{h_{x}} \) meets.

Algorithm 2 is based on Algorithm 1. The principle of it is to use Algorithm 1 at first and then check each event’s detective rate. If one event (e.g., \( I_j \)) does not satisfy the constraint, we must extract energy from other events to fill up this gap. Since Algorithm 1 has given energy to events with the highest weights, the extracted energy must be from events whose weights are higher than \( I_j \) until \( D_{I_j}(\alpha) = \eta_{h_{x}} \). Similar to Theorem 1, we can also prove the optimality of Algorithm 2.
Theorem 2: The allocation policy $a'$ derived from Algorithm 2 is an optimal solution for Problem 2.

6. Simulation

In this section, we assess the performance of the proposed power allocation algorithms. Where $\delta$ represents the energy consumption of the sensor, and $B$ refers to the capacity of the sensor that is scaled by energy unit. The values of $B$ and $\delta$ are both dependent on the hardware’s settings. Thus, it is assumed that the duration of the time slot is 60 seconds, the voltage is 3.3V, the working current is 3.3mA, the data packet transmission current is 20mA and the battery capacity is 100J (3V, 9.26mAh). In this case, unless otherwise stated, we use the following settings: the sensors and events are deployed as shown in Figure 1; the battery capacity of each sensor is $B=1000$ and costs $\delta=4$ energy when it is active; each sensor can receive $c=3$ energy with the probability $q=0.5$ at each slot where $q$ is related to the location of the sensor or the weather, e.g., sensor is blocked by leaves or clouds. We assume that the probability of each event is 0.3, and simulate the algorithm for several times to show the final average effect.

First verify Algorithm 1. Assume that the distribution of sensors and events is shown in Figure 1. The probability of receiving the energy of each sensor is: $q_1 = 0.7, q_2 = 0.8, q_3 = 0.5, q_4 = 0.8, q_5 = 0.5, q_6 = 0.8, q_7 = 0.7$. The weight of each event is $0.05, 0.025, 0.125, 0.175, 0.05, 0.025, 0.225, 0.15, 0.075, 0.1$. After the initialization, the value of $\alpha_{1,1}, \alpha_{2,2}, \alpha_{3,3}, \alpha_{4,4}, \alpha_{5,5}, \alpha_{6,6}, \alpha_{7,7}$ is 1, separately, other values are 0. Then the iteration is executed by Algorithm 1. Starting from the highest weight event $S_7$, the first iteration gets the chain $I_7 - S_4 - I_3$ which means that sensor $S_4$ can detect $I_7$. Obviously, the weight of $I_7$ is the highest among the events that the sensor $S_4$ can detect. So $S_4$ should be allocated first. It should be noted that initially all energy is assigned to $I_3$ ($\alpha_{4,3}$ is 1). Therefore, $I_3$ will be allocated all energy from $I_3$ through $S_4$. If $I_4 - S_4 - I_3$ can be obtained after $I_3$ get allocated. Considering the third highest weight event $I_9$, it does not need to be reassigned because it is the only event under $S_9$. Similarly, $I_9$ does not need to be reassigned. For event $I_9$, it can be detected by $S_1$ or $S_4$. The reason why $I_9$ can no longer get energy from $S_4$ is that the weight of $I_9$ is lower than $I_3$ and $I_9$,

<table>
<thead>
<tr>
<th>Iteration No</th>
<th>Chain Found</th>
<th>$D_{\text{avg}}(a)$ after iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$I_7 - S_4 - I_3$</td>
<td>0.40</td>
</tr>
<tr>
<td>2</td>
<td>$I_4 - S_4 - I_7$</td>
<td>0.43</td>
</tr>
<tr>
<td>3</td>
<td>$I_3 - S_3 - I_4$</td>
<td>0.46</td>
</tr>
<tr>
<td>4</td>
<td>$I_9 - S_9 - I_9$</td>
<td>0.47</td>
</tr>
</tbody>
</table>

Assume that the weight of each event is $\gamma^{(l)} = 0.05, 0.025, 0.125, 0.175, 0.05, 0.025, 0.225, 0.15, 0.075, 0.1$. We use Algorithm 1 to calculate. The networks energy distribution of each sensor is shown in Figure 3. Event $I_9$ has the largest weight. Thus both

The networks energy distribution in Case 1
According to the event weight, we need to allocate 10 added. We use Algorithm 2 to calculate it. The networks energy distribution is shown in Figure 4. In contrast to 0.025, 0.125, 0.175, 0.05, 0.025, 0.075, 0.15, 0.0225, 0.1. distribution of each sensor is shown in Figure 3. Event 0.025, 0.125, 0.175, 0.05, 0.025, 0.225, 0.15, 0.075, 0.1.

Table 2

<table>
<thead>
<tr>
<th>S</th>
<th>D</th>
<th>S</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Assume that the weights of the events coincide with $\gamma^{(2)} = 0.05, 0.025, 0.125, 0.175, 0.05, 0.025, 0.075, 0.15, 0.0225, 0.1$. We use Algorithm 1 to calculate it. The networks energy distribution is shown in Figure 4. In contrast to Figure 3, here, the weight of $I_9$ is the largest, thus $S_6$ and $S_7$ allocate all the energy to $I_9$ in order to increase $D_{net}$.

Figure 4
The networks energy distribution in Case 2

- Assume that the weights of the events coincide with $\gamma^{(2)}$, while the constraint of $D_{net} \geq 0.4, D_{net} \geq 0.1$ is added. We use Algorithm 2 to calculate it. The networks energy distribution is shown in Figure 5. In order to satisfy the constraint, the sensors $S_6$ and $S_7$ reallocate the energy. It can be clearly seen that, for $S_6$, energy has to be redistributed to $I_9, I_8$. While for $S_7$, $I_9$ does not meet the requirements of constraint condition, therefore, we need to find an event with the lowest weight among all the events that have higher weight than $I_9$ in terms of energy transmission.

Different detection rates obtained for each event are shown in Figure 6. The probabilities of events $I_1, I_2, I_3, I_4$ in three cases remain unchanged. $I_7$ greatly reduces the weight in Case 2, then its detection rate goes down to 0. In contrast, the weight of $I_9$ is increased in Case 2 with a detection rate of 0.9. $I_{10}$ can only be detected by $S_7$, however, $S_7$ allocates all energy to $I_9$ in Case 2, so that the detection rate of $I_{10}$ is reduced to 0. Event $I_9$ is more complex. It can only be detected by $S_9$. In all three cases, its weight remains the same, but the detection rate is very different. In Case 1, $S_9$ allocates the remaining energy to $I_9$ and $I_8$. In Case 2, the weight of $I_9$ is lower than $I_8$, $S_9$ allocates all of the energy to $I_9$. In Case 3, in order to satisfy the constraint $D_{net} \geq 0.4$, $S_9$ assigns energy to $I_9$ and $I_7$. Then, the detection rate of $I_9$ is 0.
7. Conclusion

In order to solve the problem of sensor energy allocation, we consider the energy unconstrained cases as well as constrained cases. To improve $D_{\text{SET}}(a)$ when energy is not constrained, we propose Algorithm 1, which assigns sensor energy priority to events with higher weights. Additional energy is assigned to lighter events. We also demonstrate the algorithm's optimality. In the case of energy constraints, we propose Algorithm 2 based on Algorithm 1 to find a Pareto optimal solution. Finally, the optimality of the two algorithms is simulated by several cases. In future, the method proposed in this paper can also be applied to other network architectures, such as cluster networks and heterogeneous networks. In addition, authors also want to study the safety and fault issues with sensor networks.

Appendix A

Functions for Algorithm 1

1: function INCREASE $(I_s, a)$

2: for all $S_y \in \Omega_{S_y}$ do

3:   for all $I_d \in \Omega_{S_y}$ do

4:     if $y_d < y_h$ then

5:       $a_{i,d} \leftarrow a_{i,d} + a_{i,d}$

6:       $a_{i,d} = 0$

7:     else if $y_d > y_h$ then

8:       let $I_{GW} = \{I_d | y_d \geq y_h, \sum_{S_y \in \Omega_{S_y}} q_c a_{c,x} \geq \delta\}$

9:       $I_{FP} = \{I_y | I_y \in I_{GW} \sum_{S_y \in \Omega_{S_y}} q_c a_{c,x} > \delta\}$

10:   end for

11: end for

12: end function

13: for all $I_d \in \Omega_{S_x}$ do

14:     $I_{GW} = \{I_d | y_d \geq y_h, \sum_{S_y \in \Omega_{S_y}} q_c a_{c,x} \geq \delta\}$

15: end for

16: function SEARCHWEIGHT

17: for $y = 1 \rightarrow Q$ do

18:   Find all the chains from $I_y$ to $I_x$:

19:   $Chain_{I_y,I_x}$, $r = 1,..., W$;

20:   The $r$-th chain is $Chain_{I_y,I_x}^r$ =

21:   $(I_{c_r}, S_{c_r}, I_{c_r}, S_{c_r},..., S_{c_r}, I_{c_r})$, where

22:   $I_{c_r} = I_{y_r}, S_{c_r} = S_{y_r}$

23:   $I_{c_r} = I_{x_r}, I_{c_r} = I_{x_r}$

24:   $t = 1,..., m$, and there is no loop in the chain;

25:   For each chain, call function $T(Chain_{I_y,I_x}^r, a)$. 

26: end function

27: function $T(Chain_{I_y,I_x}^r, a)$

28: In the $Chain_{I_y,I_x}^r$, let

29: $Pow^r = \sum_{S_y \in \Omega_{S_y}} q_c a_{c,x} - \delta$.

30: $Pow^r \leftarrow \max \{e | 0 < e \leq Pow^r\}$

31: $a_{y,c} - e / (q_c)$

32: for $t = 1 \rightarrow m$ do

33:   $a_{y,c} \leftarrow a_{y,c} - Pow^r / (q_c)$

34: end for

35: end function

Functions for Algorithm 2

1: function CHECK $(\eta)$

2: for $k = 1 \rightarrow M$ do

3:   if $D_{\eta}(a) < \eta_h$ then

4:     for all $S_y \in \Omega_{S_y}$ do

5:       $\gamma_d \geq y_h, D_{\eta}(a) > \eta_y$, $i = 1,..., Q$.

6:     Assume $\gamma_d \leq y_h, \gamma_y \leq \gamma_y$,

7:   end if

8:   end for

9:   if SEARCHWEIGHT() = 1 then
goto 2

10: end if

11: end for

12: end function

13: for $y = 1 \rightarrow Q$ do

14: Find all the chains from $I_y$ to $I_x$:

15: $Chain_{I_y,I_x}^r, r = 1,..., W$;

16: The $r$-th chain is $Chain_{I_y,I_x}^r$ =

17: $(I_{c_r}, S_{c_r}, I_{c_r}, S_{c_r},..., S_{c_r}, I_{c_r})$, where

18: $I_{c_r} = I_{y_r}, S_{c_r} = S_{y_r}$

19: $I_{c_r} = I_{x_r}, I_{c_r} = I_{x_r}$

20: $t = 1,..., m$, and there is no loop in the chain;

21: For each chain, call function $T(Chain_{I_y,I_x}^r, a)$. 

22: end function

23: function $T(Chain_{I_y,I_x}^r, a)$

24: In the $Chain_{I_y,I_x}^r$, let

25: $Pow^r = \sum_{S_y \in \Omega_{S_y}} q_c a_{c,x} - \delta$.

26: $Pow^r \leftarrow \max \{e | 0 < e \leq Pow^r\}$

27: $a_{y,c} - e / (q_c)$

28: for $t = 1 \rightarrow m$ do

29:   $a_{y,c} \leftarrow a_{y,c} - Pow^r / (q_c)$

30: end for

31: end function

Information Technology and Control
For the $r$-th chain, call function $T_r(\text{Chain}_r^{(m)}, \alpha)$. 
if $D_{\text{r}}(\alpha) \geq \eta_r$ then 
return 1 
end if
end for

$\text{Pow}^{(r)} \leftarrow \min\{e_1, e_2\}$

$\text{Pow}^{(0)} \leftarrow \max\{e|0 < e \leq \text{Pow}^{(0)}\}$
$\alpha_{\text{d}, i} \leftarrow e/(q_i c) \geq 0, t = 1, \ldots, m$
for $t = m \rightarrow 1$
da
$\alpha_{\text{d}, i} \leftarrow \alpha_{\text{d}, i} - \text{Pow}^{(0)}/(q_i c)$
$\alpha_{\text{d}, i+1} \leftarrow \alpha_{\text{d}, i} + \text{Pow}^{(0)}/(q_i c)$
end for

end function

References


Acknowledgement

This work was supported by the Zhejiang Provincial Natural Science Foundation of China, Grant NO. LQ19F030015 and the National Science Foundation of China, Grant NO. 61903338.


