

## Research Article

# Joint Management of Energy Harvesting, Storage, and Usage for Green Wireless Sensor Networks

**Xi Jin,<sup>1</sup> Fanxin Kong,<sup>2</sup> Peng Zeng,<sup>1</sup> Qingxu Deng,<sup>3</sup> and Huiting Xu<sup>3</sup>**

<sup>1</sup>Laboratory of Networked Control Systems, Shenyang Institute of Automation, Chinese Academy of Science, Shenyang 110016, China

<sup>2</sup>School of Computer Science, McGill University, QC, Canada H3A 0E9

<sup>3</sup>College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Xi Jin; [jinxia@sia.cn](mailto:jinxia@sia.cn) and Peng Zeng; [zp@sia.cn](mailto:zp@sia.cn)

Received 20 March 2014; Revised 16 May 2014; Accepted 29 May 2014; Published 25 June 2014

Academic Editor: Bulent Tavli

Copyright © 2014 Xi Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, energy harvesting has been emerging as a promising technique to prolong the lifetime for wireless sensor nodes. Most existing efforts address the design of energy harvesting and sensor node subsystem separately or ignore some real-world constraints. In this paper, we study how to codesign the two subsystems and how to jointly manage energy harvesting, storage, and usage. We first propose a novel system architecture for energy harvesting which employs several supercapacitors to eliminate the conflicts on charging and discharging among different system components. Then, we present a method to schedule their charging and discharging, which is proved to be able to guarantee zero waste of the harvested energy if the battery is not full. Third, we propose an optimal algorithm to minimize different components' capacity and two heuristic algorithms to maximize the system reward. We conduct extensive experiments based on real-life data traces. Results show that the proposed system architecture can harvest more energy compared to the state of the art, and the capacity optimization algorithm can choose the most suitable size for each system component.

## 1. Introduction

In recent years, more and more wireless sensor nodes have been integrating into real-world applications such as habitat study [1], environment or ecology monitoring [2], and structural health monitoring [3]. Traditional battery-powered sensor nodes are notorious for their limited lifetime. It is usually very expensive or even infeasible to replace their batteries after deployment. Energy harvesting has been emerging as a promising technique for such long-term applications [4–7]. This technique refers to harvesting energy from ambient energy sources (including solar, wind, thermal, and vibration) and converting it to usable electrical power to feed sensor nodes. This paper studies an important open problem of how to codesign the energy harvesting and sensor node subsystems and how to jointly manage energy harvesting, storage, and usage among different system components.

Most existing works address the design of the two subsystems separately. They either manage the power consumption of nodes without considering the scale or operations of

the harvesting subsystem (e.g., [8, 9]) or confine themselves to devising different architectures or implementations for energy harvesting with a little consideration of the management of nodes' power usage (e.g., [10–12]). This will cause energy under- or overprovisioning for the system, which results in problems such as energy inefficiency and lower utilization of the harvested energy. Some works have studied joint design of the two subsystems but ignored some real-world constraints for either one [13, 14]. For example, they ignore how fluctuations in energy harvesting impact the energy usage. This may cause system halts in real-world applications. In this paper, we present a holistic solution to solve these problems. We list our contributions as follows.

First, we propose a novel system architecture 3SC, which eliminates the charging-discharging conflicts among different system components. This paper considers photovoltaic (PV) solar energy harvesting. The converted electrical energy is buffered by energy storage components including supercapacitors and batteries. However, these storage components

cannot charge and discharge at the same time [15, 16]. If they all start charging together, the sensor node subsystem will halt due to no available power supply. To solve this problem, 3SC employs three supercapacitors between solar panel and batteries/sensor nodes. They behave as chargers (or dischargers) at different time instances.

Second, we present a charging and discharging scheduling algorithm named CDSA. CDSA makes wise decisions on how to choose energy storage component for the harvested energy to flow to and when should charge or discharge the component. The algorithm is proved to be able to guarantee zero waste of the harvested energy if the battery is not full.

Third, the design of energy harvesting sensor nodes is a three-player game among energy harvesting, storage, and usage. Energy storage components with small capacity may be enough to power up sensor nodes for a location with abundant solar resource and vice versa. On the other hand, to adapt to short-term variations of the harvested energy, we can also adjust the power needs of sensor nodes by tuning tasks' execution. We propose several algorithms to address the problem of how to achieve a match among solar resource, components' capacity, and power needs. An optimal algorithm COA is designed to minimize each component's capacity. Two heuristic algorithms (L2U and U2L) are developed to optimize the system reward if tasks are of different importance. We conduct extensive experiments using real-life data traces and results show the effectiveness of our design.

The rest of paper is organized as follows. Section 2 introduces the related work. Section 3 describes the novel system architecture for energy harvesting sensor nodes. Section 4 proposes a charging and discharging scheduling algorithm based on the proposed architecture. Sections 5 and 6 present several algorithms for system components' capacity optimization and system reward maximization. Section 7 shows experimental results. Section 8 concludes the paper.

## 2. Related Work

*2.1. System Architectures and Charging/Discharging Scheduling Algorithms.* There have been a number of works on the design of energy harvesting system architectures and charging/discharging scheduling methods. The authors of [10] present a system named *Prometheus* that combines positive attributes of the supercapacitor in terms of charging/discharging cycles and the lithium batteries for their high energy density and low leakage. In addition, a charging/discharging scheduling driver for *Prometheus* is proposed to prolong its lifetime. In [11], an autonomous energy harvesting platform named *AmbiMax* with a battery and a reservoir capacitor array is proposed and a simple charging/discharging management is designed according to voltages of capacitors and batteries. In [12], a battery-supercapacitor hybrid accumulator architecture is proposed and a charging/discharging management algorithm based on finite state machine is designed to extend the lifetime of batteries. Reference [13] proposes a unified joint rate control and power management framework. Reference [14]

proposes a hybrid energy storage system with one Li-Ion battery and two separate supercapacitors connected by a DC bus and then designs a budgeting heuristic algorithm to assign energy for executing tasks and charging battery. In [17], *hypoenergy* (hybrid supercapacitor-battery power-supply optimization for energy efficiency) is introduced, which aims at maximizing the battery's lifetime. Reference [18] proposes a photovoltaic-thermoelectric hybrid power system, in which the energy management is to improve the energy efficiency and reliability. Reference [19] designs a real-time task scheduling framework based on the energy harvesting system, which contains a solar panel and a supercapacitor. And there are some other researches, for example, [20–24]. But these previous works do not consider how to eliminate the conflicts on charging and discharging among different system components and how to avoid wasting harvested energy.

*2.2. Minimizing System Components' Capacity and Maximizing the System Reward.* For minimizing components' capacity in energy harvesting systems, the related work is as follows. The authors of [17] design a heterogeneous supercapacitor bank and choose the supercapacitors' capacity so as to minimize the supercapacitor leakage. In [25], they system designers sweep the components' capacity to determine the system architecture that meets their requirements. Reference [26] focuses on stand-alone PV-H<sub>2</sub> systems and develops a systematical method to select the sizes of the PV panel, the electrolyzer, H<sub>2</sub> tanks, the fuel cell, and the battery. Reference [27] presents that the size of the battery should be overrated with a factor to extend battery lifetime and analyzes the effect on the factor for the volume and mass of system components. In [28], in order to satisfy system requirements, the designer chooses the sizes of energy system components based on the Pareto curve that relates energy storage to energy generation.

Reference [29] focuses on the reward of the system with multiversion tasks, in which a task has multiple versions and each version corresponds to a reward value that shows the importance for systems. Then a heuristic algorithm is proposed to maximize the system reward under time and energy constraints. Reference [30] is the first work to maximize the system reward for energy harvesting systems with predictable energy sources. The authors design polynomial-time algorithms to solve this problem for systems with continuous service levels and prove that the polynomial-time algorithms are optimal. Then, in [31, 32], the same authors extend these algorithms to the system with discrete service levels and more application scenarios. Reference [33] is based on decomposable and combinable tasks. The reward optimization problem is modeled as the bounded Knapsack problem, which is solved using dynamic programming.

However, the system models of previous works are not the same as ours; therefore the related work about minimizing components' capacity and maximizing system reward cannot be used without modification in this paper.

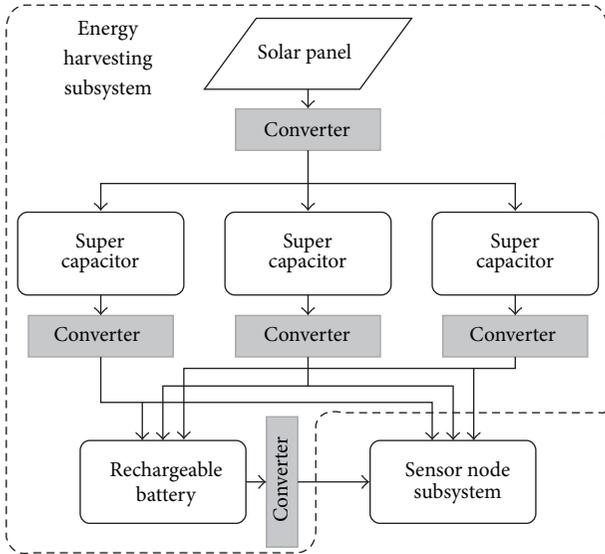


FIGURE 1: Our proposed system architecture 3SC.

### 3. System Architecture

In this paper, we adopt the harvest-store-use architecture, which is the most popular architecture for sensor nodes [5]. We choose the solar energy as our energy harvesting source. Rechargeable batteries and supercapacitors are two choices for energy storage. Rechargeable batteries have a higher energy density than supercapacitors, while supercapacitors have a higher power density and more charging/discharging cycles than batteries. Recent research works have illustrated that the battery-supercapacitor hybrid energy system is the best choice for energy harvesting sensor nodes. So we adopt this battery-supercapacitor hybrid architecture in our system. However, existing battery-supercapacitor hybrid architectures (e.g., [12, 14]) cannot eliminate the charging-discharging conflicts among different system components, which will cause the intermittency of power supply. To address this problem, we design a battery-supercapacitor hybrid system architecture, which we name 3SC (shown in Figure 1).

3SC is composed of one rechargeable battery, one solar panel, one sensor node subsystem, and three supercapacitors. Supercapacitors store electrical energy generated by the solar panel and transmit electrical energy to the sensor node or the battery. Each supercapacitor can play multiple roles: (1) receiving energy from the solar panel, (2) charging the battery, and (3) powering the sensor node subsystem. But a supercapacitor cannot play more than one role at the same time. The roles of several supercapacitors can switch with each other. If the energy stored in supercapacitors is not enough to power the sensor node subsystem, the battery will supply energy to power it. We can see that supercapacitors are related to all of other system components. Moreover, (1), (2), and (3) may occur simultaneously. But supercapacitors cannot charge and discharge at the same time. So the number of supercapacitors must be at least 3. In the case of existence of only two supercapacitors, only two of three roles can be

supported at the same time, and the remaining one must wait. Then the following situations may occur.

- (i) Although the solar panel generates electrical energy and energy storage components are not full, the system cannot harvest energy, since two supercapacitors are assigned to the sensor node subsystem and the battery, respectively.
- (ii) In order to harvest solar energy uninterruptedly, one supercapacitor must be used to receive energy generated by the solar panel and another supercapacitor charges the battery or powers the sensor node subsystem. But the charging time will affect the sensor node subsystem performance. In this case, how to satisfy the performance constraint of the sensor node system is an unsolved problem.

Three supercapacitors with an advanced charging/discharging scheduling algorithm can eliminate the conflicts on charging and discharging between energy components, since the three supercapacitors can be seen as an energy buffer with three bidirectional interfaces, which can make (1), (2), and (3) execute in parallel. We propose the system architecture as shown in Figure 1, where DC-DC converters serve the purpose of voltage conversions between components. Each supercapacitor is connected to the solar panel, the battery, and the sensor node subsystem by converters. The charging/discharging scheduling algorithm is described in the next section.

Comparing with the architecture containing two supercapacitors [12], our system architecture adds extra cost, but it is small. The price of the supercapacitor with capacitances on the order of tens of farads is about several dollars. (The suggestion about selecting supercapacitors is shown in Section 5.) For example, the 10F, 30F, 50F, and 70F supercapacitors produced by Panasonic Corporation are \$2, \$4, \$4, and \$6, respectively. And the price of the converter controller TPS4000 is \$5. In the DC-DC converter, all other components' price is less than \$1. So the extra cost of our architecture is less than \$15.

### 4. Charging/Discharging Scheduling

The key notations used in this paper are summarized in Notations section.

Our proposed charging/discharging scheduling algorithm (shown in Algorithm 1) manages energy conversions according to the stored electrical energy in system components. We assume that all of the energy storage components are ideal. SP,  $SC_i$ , BA, and SN denote the solar panel, the  $i$ th supercapacitor, the battery, and the sensor node subsystem, respectively.  $E_{SC_i}$  is the stored energy in the supercapacitor  $SC_i$ . We use  $\xi_{SP}$  to present the supercapacitor that is being charged by the solar panel SP. Similarly,  $\xi_{BA}$  is the supercapacitor that charges the battery BA and  $\xi_{SN}$  is the supercapacitor or the battery that supplies energy to the sensor node subsystem SN. The threshold energy  $E_{th}$  is the minimum energy to power a sensor node.  $i$ ,  $j$ , and  $k$  are variables.  $SC_i$ ,  $SC_j$ , and  $SC_k$  denote supercapacitors. We use

```

Require:  $E_{SC_i}, \xi_{SP}, \xi_{BA}$  and  $\xi_{SN}$ 
Ensure:  $\xi_{SN}, \xi_{BA}$  and  $\xi_{SN}$ 
(1) if  $E_{\xi_{SN}} == 0$  then
(2)    $\xi_{SN} = \arg(\max_{VSC_i, E_{SC_i} \geq E_{th}} \{E_{SC_i}\});$ 
(3)    $\xi_{SP} = \arg(\min_{VSC_j, SC_j \neq \xi_{SN}} \{E_{SC_j}\});$ 
(4)   if  $\xi_{SN} == \phi$  then
(5)      $\xi_{SN} = BA;$ 
(6)   else if  $(E_{SC_k} \geq E_{th}) \&\& (SC_k \neq \xi_{SP}) \&\& (SC_k \neq \xi_{SN})$  then
(7)      $\xi_{BA} = SC_k;$ 
(8) if  $(\xi_{SN} == BA) \&\& (E_{\xi_{SP}} \geq E_{th})$  then
(9)    $\xi_{SN} = \xi_{SP};$ 
(10)   $\xi_{SP} = \arg(\min_{VSC_j, SC_j \neq \xi_{SN}} \{E_{SC_j}\});$ 
(11) if  $\xi_{SP}$  is charged fully then
(12)   $\xi_{SN} = \xi_{SP};$ 
(13)   $\xi_{SP} = \arg(\min_{VSC_j, SC_j \neq \xi_{SN}} \{E_{SC_j}\});$ 
(14) if  $(E_{SC_k} \geq E_{th}) \&\& (SC_k \neq \xi_{SP}) \&\& (SC_k \neq \xi_{SN})$  then
(15)   $\xi_{BA} = SC_k;$ 

```

ALGORITHM 1: Charging/discharging scheduling algorithm.

different variables to distinguish different supercapacitors. The algorithm is driven by three events.

The first event is between lines (1) and (7). If there is no energy in the supercapacitor that is powering the sensor node subsystem (line (1)), then the energy source of the sensor node subsystem is assigned as the supercapacitor that can power the sensor node subsystem and stores the maximum energy (line (2)). At the same time, the supercapacitor with the minimum energy is charged by the solar panel (line (3)). In this case, if no supercapacitor can start the sensor node subsystem (line (4)), the battery supplies energy to the sensor node subsystem (line (5)). Otherwise, the energy that is stored in the idle supercapacitor (neither the supercapacitor powering the sensor node subsystem nor the supercapacitor charged by the solar panel) is charged to the battery if it exists (lines (6) and (7)). If the battery is full, the redundant energy is discarded.

The second event is between lines (8) and (10). When the amount of one supercapacitor's energy is enough to power the sensor node subsystem (line (8)), the energy source is changed from the battery to the supercapacitor (line (9)) and the supercapacitor with the minimum energy is charged by the solar panel (line (10)). The premise of this event is that the battery is the energy source of the sensor node subsystem; that is, the energy stored in any supercapacitor is less than  $E_{th}$ . Then one supercapacitor is charged by the solar panel, until its energy is higher than  $E_{th}$  and the second event occurs. The energy stored in the other two supercapacitors is also less than  $E_{th}$ . So they cannot charge the battery; that is, no supercapacitor is assigned to the battery in this event.

The third event is between lines (11) and (15). When the supercapacitor is charged fully by the solar panel, the third event occurs (line (11)). This supercapacitor is assigned as the energy source (line (12)). Similarly, the supercapacitor with the minimum energy is charged (line (13)), and the energy

stored in the idle supercapacitor is charged to the battery (lines (15)).

To sum up, the design for CDSA is based on the following considerations.

- (i) To reduce the frequency of running CDSA, the supercapacitor with the maximum energy is chosen as the power source of the sensor node subsystem SN (line (2)) and the supercapacitor with the minimum energy is charged by the solar panel SP (lines (3), (10) and (13)).
- (ii) To reserve enough capacity to store the energy generated by the solar panel, the energy in the idle supercapacitor is converted to the battery BA (lines between (6) and (7) and between (14) and (15)).

We show the following example to explain our algorithm CDSA.

- (1) We assume that the battery is half full and the three supercapacitors are empty. In this case,  $\xi_{SN} = BA$  and  $\xi_{SP} = SC_1$ .
- (2) When  $E_{SC_1} \geq E_{th}$ , the second event occurs. Then  $\xi_{SN} = SC_1$  and  $\xi_{SP} = SC_2$ .
- (3) The supercapacitor  $SC_1$  supplies energy to the sensor node subsystem until it becomes empty. Then the first event occurs, and  $\xi_{SN} = SC_2$  and  $\xi_{SP} = SC_1$ .
- (4) If sunlight is sufficient,  $SC_1$  is charged fully before the energy of  $SC_2$  becomes less than  $E_{th}$ . In this case, the third event occurs. Then  $\xi_{SN} = SC_1$ ,  $\xi_{SP} = SC_3$ , and  $\xi_{BA} = SC_2$ .

We assume that 10 percent of the battery's energy is reserved to deal with the unpredictable solar radiation, supply energy to CDSA, and handle some other unexpected events.

So the energy leakage of supercapacitors and the energy used by CDSA are ignored in the following sections of the paper.

In our system, the output power of the solar panel with the area  $a$  ( $\text{cm}^2$ ) is  $P_{\text{SP}} = a \cdot P_{\text{cm}}$ , where  $P_{\text{cm}}$  ( $\text{mW}/\text{cm}^2$ ) is the power density.  $P_{\text{cm}}^{\text{max}}$  is the maximum  $P_{\text{cm}}$ . The voltage of the battery is represented by  $v_{\text{BA}}$  and the cut-off voltage  $v_{\text{BA}}^{\text{min}}$  is the voltage at which the battery is considered to be fully discharged. To extend the lifetime of the battery, our system uses the constant current  $\alpha C$  ( $0 < \alpha < 1$ ) to charge the battery, even if this method will lose a part of the capacity of the battery (20%–30%) [12]. So the battery with 80% capacity is defined as full battery.  $C$  denotes the charging and discharging rate (see [34]), and  $\alpha$  is its coefficient. Based on the above statements, we prove the effectiveness of CDSA as follows.

*Property 1.* The energy storage components can harvest all output power of the solar panel if the battery is not full and

$$a \cdot P_{\text{cm}}^{\text{max}} \leq \alpha \cdot C \cdot v_{\text{BA}}^{\text{min}}. \quad (1)$$

*Proof.* To prove Property 1, the following two parts must be proved.

- (1) If inequality (1) can be satisfied, then the energy conversion channel from the solar panel to the battery (or to the sensor node subsystem) will not be blocked.
- (2) Our algorithm CDSA can guarantee that at any time there is at least one nonfull supercapacitor that can be charged by the solar panel.

For part (1), we use  $P_{\text{out}}$  and  $P_{\text{in}}$  to denote the output and input power of the supercapacitor, respectively. If  $P_{\text{out}} \geq P_{\text{in}}$ , the energy conversion channel will not be blocked. The power density of the supercapacitor is quite high, so the output power and input power of the supercapacitor are determined by the input power of the battery (or the sensor node subsystem) and the output power of the solar panel, respectively. The sensor node subsystem is also powered by the battery; thus its input power is not higher than that of the battery. Therefore, if the battery is not full, the output power of the supercapacitor is equal to the input power of the battery. Thus,

$$\begin{aligned} P_{\text{out}} &= \alpha \cdot C \cdot v_{\text{BA}} \geq \alpha \cdot C \cdot v_{\text{BA}}^{\text{min}}, \\ P_{\text{in}} &= a \cdot P_{\text{cm}} \leq a \cdot P_{\text{cm}}^{\text{max}}. \end{aligned} \quad (2)$$

Therefore, if

$$a \cdot P_{\text{cm}}^{\text{max}} \leq \alpha \cdot C \cdot v_{\text{BA}}^{\text{min}}, \quad (3)$$

then  $P_{\text{out}} \geq P_{\text{in}}$ .

For part (2), we assume by contradiction that all supercapacitors are full and the last charged supercapacitor is charged fully in the time interval between  $q_1$  and  $q_2$ . Thus, at the time  $q_1$ , the other two supercapacitors are full. In this situation, nothing but the third event can occur, and this event provides that one of two supercapacitors is chosen to charge the battery (line (15)). Therefore, at the time  $q_2$ , the supercapacitor  $\xi_{\text{BA}}$  must not be full, which contradicts the assumption.  $\square$

*Property 2.* If the energy storage components are not empty, the sensor node subsystem can be powered uninterruptedly.

*Proof.* As shown in Algorithm 1, in all of three events, a nonempty supercapacitor (lines (2), (9), and (12)) or the battery (line (5)) is assigned to the sensor node subsystem as its power source.  $\square$

## 5. Capacity Optimization for System Components

The capacity selection is a major factor affecting the system performance. If the capacity is too small, there is not enough energy to drive the sensor node subsystem. If the capacity is very large, although there is enough energy to drive it, the remaining resource cannot be used effectively and the system cost is wasted. Therefore, in this section, we optimize the capacity of each system component according to system power needs and the historic values of the solar radiation. Recall that we reserve 10 percent of the battery's energy to deal with the unpredictable solar irradiation. This is also because the value of the solar radiation is different for different years. The power required by the sensor node depends on the tasks running on it. So we will first introduce our system model in the following subsection.

*5.1. System Model.* In the sensor node subsystem, there are a total of  $n$  tasks. The task set is denoted by  $T = \{t_1, \dots, t_n\}$ . Each task  $t_i$  ( $i \in [1, n]$ ) is characterized by  $t_i = \langle e_i, E_i^t, r_i, d_i, d_i^u, d_i^l \rangle$ , where  $e_i$ ,  $E_i^t$ ,  $r_i$ , and  $d_i$  denote the worst-case execution time, energy requirement, reward, and period, respectively.  $d_i^u$  ( $d_i^l$ ) represents the upper (lower) bound of  $d_i$ . Each task must be complete within its period; that is, the relative deadline is equal to the period  $d_i$ . The power required by a sensor node subsystem is represented as

$$P^t = \sum_{i=1}^n \frac{E_i^t}{d_i}. \quad (4)$$

There are  $m$  batteries that can be selected. The selectable battery set is denoted by  $B = \{b_1, \dots, b_m\}$ . Each battery  $b_i$  ( $i \in [1, m]$ ) is characterized by  $b_i = \langle C_i, v_i^{\text{min}}, E_i^b \rangle$ , where  $C_i$ ,  $v_i^{\text{min}}$ , and  $E_i^b$  denote the charging and discharging rate, the cut-off voltage, and maximum available energy (excl. the reserved energy and unchargeable capacity), respectively. We use  $b_x$  ( $x \in [1, m]$ ) to denote the selected battery. The cut-off voltage is highly dependent on the type of the battery instead of the capacity, and the charging and discharging rate  $C_i$  is proportional to  $E_i^b$ . So we choose  $C_i$  as the optimized capacity of batteries. For the solar panel, the optimized capacity is the area  $a$ , which is less than or equal to the maximum acceptable area  $A$ .

In our proposed system, the supercapacitor is designed as the energy buffer. The supercapacitor's capacity cannot affect energy harvesting (Property 1). Therefore, we do not optimize it in this paper. But the capacity of the supercapacitor cannot be too large or too small, since the supercapacitor with small capacity will lead to invoke CDSA frequently. While the large

supercapacitor is not the best alternative, since the larger the supercapacitor, the higher the leakage power, we suggest selecting supercapacitors with capacitances on the order of several to tens of farads.

We process the solar energy on a daily basis. This is because the daily cycle of sunlight leads to the charging/discharging cycle of the energy storage system. Moreover, this method can balance the complexity and efficiency of our algorithm. We use  $h$  to represent the number of days of the historic values and  $E_i^d$  to denote the generated energy per unit area of the solar panel at the  $i$ th day.

**5.2. Problem Formulation.** Given the task set  $T$ , the battery set  $B$ , and the historic values  $E_i^d$  of  $h$  days, our objective is to find  $x$  and  $a$  such that

$$\min w_{\text{SP}} \cdot a + w_{\text{BA}} \cdot C_x, \quad (5)$$

where  $w_{\text{SP}}$  and  $w_{\text{BA}}$  are the given weight coefficients.

The capacity optimization should respect the following constraints.

- (i) *Area Constraint.* The area  $a$  cannot exceed the maximum acceptable area  $A$ :

$$0 < a \leq A. \quad (6)$$

- (ii) *Power Conversion Constraint.* According to Property 1, inequality (1) should be satisfied, which is rewritten as follows:

$$a \cdot P_{\text{cm}}^{\max} \leq \alpha \cdot C_x \cdot v_x^{\min}, \quad (7)$$

where  $\alpha$  is a constant between 0 and 1.

- (iii) *Energy Constraint.* At any time, the available energy in energy storage components cannot be less than zero:

$$\min_{\forall i \in [1, h]} \{F_i^d\} - T_{\text{day}} \cdot P^t + \frac{T_{\text{day}} \cdot P^t}{a \cdot P_{\text{cm}}^{\max}} \cdot P^t \geq 0, \quad (8)$$

where  $T_{\text{day}}$  is the number of seconds in a day and  $F_i^d$  is the lower bound of available energy in the battery at the end of the  $i$ th day, which is shown in Lemma 1. The proof of inequality (8) is shown in Theorem 2.

**Lemma 1.** *The lower bound of available energy at the end of the  $i$ th day is*

$$F_i^d = \min \{F_{i-1}^d + a \cdot E_i^d - H_i \cdot P^t, E_x^b\} - (T_{\text{day}} - H_i) \cdot P^t, \quad (9)$$

where

$$H_i = \frac{a \cdot E_i^d}{a \cdot P_{\text{cm}}^{\max}} = \frac{E_i^d}{P_{\text{cm}}^{\max}}. \quad (10)$$

*Proof.* Earlier in the paper, we described the supercapacitor (several to tens of farads), whose capacity is far less than the capacity of the battery. So we ignore it.

If the capacity of the battery is unlimited, then  $F_i^d = F_{i-1}^d + a \cdot E_i^d - T_{\text{day}} \cdot P^t$ . But the capacity of the battery is up to  $E_x^b$ . It limits the peak of the stored energy. When the battery is full, the incoming harvested energy cannot be stored anymore and the superfluous energy has to be discarded if the energy required by the sensor node subsystem has been met. Therefore, although the generated energy  $a \cdot E_i^d$  has been known, not all of them can be stored in the battery or used by the sensor node subsystem.

If the battery is full and the power supplied by the solar panel is the maximum, then the discarded energy is the maximum and the harvested energy is the minimum. So the lower bound of available energy occurs in the situation where  $E_i^d$  is harvested within the shortest time  $H_i$ . At time  $H_i$ , if we do not consider the capacity limit of the battery, the stored energy is  $F_{i-1}^d + a \cdot E_i^d - H_i \cdot P^t$ , where  $H_i \cdot P^t$  denotes the consumed energy by tasks. But due to the capacity limit of the battery the available energy is  $\min\{F_{i-1}^d + a \cdot E_i^d - H_i \cdot P^t, E_x^b\}$ . From the time  $H_i$  to the end of the  $i$ th day, there is no harvested energy and the energy  $(T_{\text{day}} - H_i) \cdot P^t$  is supplied to tasks. Therefore, the lower bound  $F_i^d$  holds.  $\square$

**Theorem 2.** *In  $h$  days, the lower bound of available energy is*

$$\min_{\forall i \in [1, h]} \{F_i^d\} - T_{\text{day}} \cdot P^t + \frac{T_{\text{day}} \cdot P^t}{a \cdot P_{\text{cm}}^{\max}} \cdot P^t. \quad (11)$$

*Proof.* We assume that  $j = \arg(\min_{\forall i \in [1, h]} \{F_i^d\})$  and the minimum available energy appears at the time  $q$  of the  $k$ th day; then  $F_j^d \leq F_k^d$ . Since  $E_{k+1}^d \geq 0$ , at the time  $q$ , the available energy

$$F_q \geq F_k^d - T_{\text{day}} \cdot P^t \geq F_j^d - T_{\text{day}} \cdot P^t. \quad (12)$$

But it is impossible to see the value of  $F_j^d - T_{\text{day}} \cdot P^t$  in this system. If it appears, then there must be  $g$  that  $F_g^d = F_j^d$  and  $E_{g+1}^d = 0$ . Thus  $F_{g+1}^d < F_j^d$ . It contradicts the assumption. To guarantee the assumption for  $j$ , at least the time

$$\frac{T_{\text{day}} \cdot P^t}{a \cdot P_{\text{cm}}^{\max}} \quad (13)$$

is used to recover the available energy; then

$$F_q \geq F_j^d - T_{\text{day}} \cdot P^t + \frac{T_{\text{day}} \cdot P^t}{a \cdot P_{\text{cm}}^{\max}} \cdot P^t. \quad (14)$$

Therefore, the lower bound of  $F_q$  is

$$\min_{\forall i \in [1, h]} \{F_i^d\} - T_{\text{day}} \cdot P^t + \frac{T_{\text{day}} \cdot P^t}{a \cdot P_{\text{cm}}^{\max}} \cdot P^t. \quad (15)$$

$\square$

**5.3. An Optimal Algorithm.** Now we present our capacity optimization algorithm COA (shown in Algorithm 2), which

```

Require:  $B, P^t, P_{cm}^{\max}, A, \alpha, w_{SP}, w_{BA}, E_i^d (i \in [1, h])$ 
Ensure:  $x, a$ 
(1)  $x = a = obj = +\infty$ ;
(2) for each  $i \in [1, m]$  do
(3)  $left = 1; right = \min \left\{ A, \frac{\alpha \cdot C_i \cdot v_i^{\min}}{P_{cm}^{\max}} \right\}$ ;
(4) while  $right - left > 1$  do
(5)  $mid = \left\lfloor \frac{right + left}{2} \right\rfloor$ ;
(6) if  $mid$  satisfies Inequality (8) then
(7)  $right = mid$ ;
(8) else
(9)  $left = mid$ ;
(10) if  $left$  satisfies Inequality (8) then
(11)  $a' = left; obj' = w_{SP} \cdot left + w_{BA} \cdot C_j$ ;
(12) else if  $right$  satisfies Inequality (8) then
(13)  $a' = right; obj' = w_{SP} \cdot right + w_{BA} \cdot C_j$ ;
(14) else
(15)  $obj' = +\infty$ ;
(16) if  $obj' < obj$  then
(17)  $x = i; a = a'; obj = obj'$ ;
(18) return  $x, a$ ;

```

ALGORITHM 2: Capacity optimization algorithm.

finds the area  $a$  and the battery  $b_x$  to minimize the objective equation (5). For each  $b_i \in B$  (line (2)), an area  $a'$  is searched to minimize the objective  $obj'$ . The **while** loop (lines between (4) and (9)) for searching  $a'$  is a binary search which reduces the search space by half in each iteration. At line (3), the upper bound  $right$  of the binary search is calculated according to inequalities (6) and (7). Then, lines between (6) and (9) determine to get rid off which half of the current interval  $[left, right]$ . Until the interval  $[left, right]$  cannot be compressed, the binary search finishes. Lines between (10) and (15) show that the bound satisfied inequality (8) is the result  $a'$  of the binary search. Then, at lines between (16) and (17), the battery  $b_x$  with the minimum  $obj'$  and its corresponding  $a'$  is selected as the optimized capacity of the battery and solar panel, respectively.

The number of iterations of **while** loop at line (4) is  $O(\log s)$  due to the binary search, where

$$s = \min \left\{ A, \frac{\alpha \cdot C_i \cdot v_i^{\min}}{P_{cm}^{\max}} \right\}. \quad (16)$$

The complexity of the verification of inequality (8) is  $O(h)$ . The number of iterations of **for** loop at line (2) is  $O(m)$ . Therefore, the complexity of Algorithm 2 is  $O(mh \log s)$ .

**Theorem 3.** *The algorithm COA can find the OPTIMAL capacities of the battery and the solar panel leading to the minimized equation (5) under the constraints inequalities (6), (7), and (8).*

*Proof.* For each battery, the binary search can find the optimal area of the solar panel to minimize the objective, since a linear relationship exists between the area and our

objective equation (5). Then COA finds the battery  $b_x$  with the minimum objective among the set  $B$ . The finding process traverses all of elements in  $B$ , so it is optimal. Therefore COA is optimal.  $\square$

## 6. System Reward Maximization

Each task has a reward which is a measure of task's importance. The task with a higher reward is considered more important and should be executed frequently. The task's importance is based on not only the system resource, but also the user requirements. So the task reward is assigned by the user. In this section, based on the system model shown in Section 5.1, we maximize the system reward according to the given system components' capacity.

**6.1. Problem Formulation.** Given the task set  $T$ , the historic values, and system components' capacity which satisfy inequalities (6) and (7), our objective is to assign the period  $d_i$  for each task  $t_i$  such that

$$\max \sum_{i=1}^n \frac{r_i}{d_i}. \quad (17)$$

The system reward maximization should respect the following constraints.

- (i) *Microprocessor Utilization Constraint.* The utilization of the microprocessor in the sensor node is not more than the utilization bound for EDF scheduling:

$$\sum_{i=1}^n \frac{e_i}{d_i} \leq 1. \quad (18)$$

```

Require:  $T, B, x, a, E_i^d$  ( $i \in [1, h]$ ) and an empty set  $\Omega$ 
Ensure:  $\{d_1, \dots, d_n\}$ 
(1) for each  $t_i \in T, d_i = d_i^u$ ;
(2) if  $\{d_1, \dots, d_n\}$  cannot satisfy Inequality (8) and (18) then
(3)   exit: the input parameters are not feasible;
(4) for each  $t_i \in T, d_i = d_i^l$ ;
(5) if  $\{d_1, \dots, d_n\}$  satisfies Inequality (8) and (18) then
(6)   return  $\{d_1, \dots, d_n\}$ ;
(7) while  $T \neq \Phi$  do
(8)   repeat
(9)      $t_j = \arg(\min_{\forall t_i \in T} \{\delta_i\})$ ;
(10)     $d_j ++$ ;
(11)    if  $d_j == d_j^u$  then
(12)       $T \leftarrow T - \{t_j\}$ ;
(13)      if  $T == \Phi$  then
(14)        break;
(15)    until  $\{d_1, \dots, d_n\}$  satisfies Inequality (8) and (18);
(16)    if  $T \neq \Phi$  then
(17)       $d_j --$ ;
(18)      remove  $t_j$  from  $T$  to  $\Omega$ ;
(19)     $t_j = \arg(\min_{\forall t_i \in \Omega} \{\delta_i\})$ ;
(20)     $d_j ++$ ;
(21) return  $\{d_1, \dots, d_n\}$ ;

```

ALGORITHM 3: Stretch L2U heuristic algorithm.

Our system can supply energy to the sensor node subsystem uninterruptedly (Property 2). Based on this situation, the sensor node subsystem can be seen as a uniprocessor system. So any task scheduling for uniprocessor systems can be used in our system. The utilization constraint assumes that preemptive EDF scheduling is used. A different utilization formula can be used with different scheduling algorithms, such as RM and DM.

- (ii) *Energy Constraint.* In order to stably supply power to system tasks, at any time, the available energy in energy storage components is not less than zero. The formulation of this constraint is the same as inequality (8).
- (iii) *Period Constraint.* The assigned period cannot exceed the lower and upper bound of the period:

$$d_i^l \leq d_i \leq d_i^u. \quad (19)$$

**6.2. Efficient Heuristic Algorithms.** The harvesting energy and the computational resource limit the system reward. So to maximize the system reward the frequently executed task should have a large optimized objective  $r_i/d_i$ , a small power requirement  $E_i^t/d_i$ , and a small utilization  $e_i/d_i$ , that is, using lower cost to obtain higher reward. Therefore, for each task  $t_i$ , we define the following metric:

$$\delta_i = \frac{r_i/d_i}{(e_i/d_i) \cdot (E_i^t/d_i)} = \frac{r_i \cdot d_i}{e_i \cdot E_i^t}. \quad (20)$$

The task with the larger metric should be executed more frequently and vice versa. Based on the metric  $\delta$ , we propose two heuristic algorithms (L2U and U2L) to maximize the system reward.

**6.2.1. Stretch (from the Lower Bound to the Upper Bound).** For the first heuristic algorithm L2U (shown in Algorithm 3), we begin with assigning all periods to the lower bound (line (4)). If they can satisfy inequalities (18) and (8) (lines between (5) and (6)), the initial assignment will be the solution. Otherwise, we need to *stretch* the period of the task with the minimum  $\delta$  (lines between (7) and (18)) in the set  $T$ , which contains the tasks with the stretchable period. If the period  $d_j$  is stretched to its upper bound  $d_j^u$ , it will be removed from the set  $T$ , to prevent the period from exceeding its upper bound (lines between (11) and (12)). For the task  $t_j$ , if the set  $\{d_1, \dots, d_j, \dots, d_n\}$  cannot satisfy inequalities (18) and (8) and the set  $\{d_1, \dots, d_j + 1, \dots, d_n\}$  can satisfy the inequalities, then we call the task  $t_j$  as the *critical task*.  $\Omega$  is used to denote the critical task set. During the process of stretching, once the task is removed from the set  $T$  to the set  $\Omega$ , its period cannot be increased until there are no tasks in the set  $T$ . Otherwise, tasks in the set  $T$  can no longer be optimized. Then when the period of the critical task with the minimum  $\delta$  is increased 1 (lines between (19) and (20)), according to the definition of critical task, the current periods can satisfy all inequalities of this problem. Even if the set  $\Omega$  is an empty set, Algorithm 3 can also obtain a solution. Since lines between (16) and (18) are not executed, then all periods are assigned to the upper bound  $d_i^u$ . In this situation, lines between (1) and (3) have

```

Require:  $T, B, x, a, E_i^d (i \in [1, h])$ 
Ensure:  $\{d_1, \dots, d_n\}$ 
(1) for each  $t_i \in T, d_i = d_i^l$ ;
(2) if  $\{d_1, \dots, d_n\}$  satisfies Inequality (8) and (18) then
(3)   return  $\{d_1, \dots, d_n\}$ ;
(4) for each  $t_i \in T, d_i = d_i^u$ ;
(5) if  $\{d_1, \dots, d_n\}$  cannot satisfy Inequality (8) and (18) then
(6)   exit: the input parameters are not feasible;
(7) while  $T \neq \Phi$  do
(8)   repeat
(9)      $t_j = \arg(\max_{\forall i \in T} \{\delta_i\})$ ;
(10)     $d_j - -$ ;
(11)    if  $d_j == d_j^l$  then
(12)       $T \leftarrow T - \{t_j\}$ ;
(13)    until  $\{d_1, \dots, d_n\}$  cannot satisfy Inequality (8) and (18);
(14)     $d_j ++$ ;
(15)     $T \leftarrow T - \{t_j\}$ ;
(16) return  $\{d_1, \dots, d_n\}$ ;

```

ALGORITHM 4: Compression U2L Heuristic Algorithm.

excluded the possibility that all upper bounds cannot satisfy inequality constraints. So the set  $\{d_1^u, \dots, d_n^u\}$  is the solution, when the set  $\Omega$  is empty.

The number of iterations of **repeat until** loop is  $O(y)$ , where  $y = \sum_{i=1}^n (d_i^u - d_i^l)$ . In the worst case, the complexity of line (9) and line (15) is  $O(n)$  and  $O(h)$ , respectively. In each iteration of **while**, at least one task is removed from the set  $T$ , so the complexity of **while** loop is  $O(n)$ . Therefore, the complexity of Algorithm 3 is  $O(y \cdot n \cdot (h + n))$ .

**6.2.2. Compression (from the Upper Bound to the Lower Bound).** Instead of assigning all periods to the lower bound  $d_i^l$  at the beginning, following an opposite direction, the second heuristic algorithm begins with assigning the upper bound  $d_i^u$  (line (4)). If they cannot satisfy inequalities (8) and (18), the input parameters are not feasible (lines between (4) and (6)). Otherwise, we need to *compress* the period of the task with the maximum  $\delta$ , until they cannot satisfy inequalities (8) and (18) (lines between (8) and (13)). The last  $d_j$  that satisfied the inequalities is assigned to the task  $t_j$  (line (14)), and  $t_j$  is removed from the set  $T$  (line (15)), which contains the tasks with the compressible period. Although the period of the task with the maximum  $\delta$  cannot be compressed again, the other tasks in the set  $T$  may be able to be compressed. So the ending condition of Algorithm 4 is that the set  $T$  is empty (line (7)).

The steps in Algorithm 4 are similar to those in Algorithm 3. Therefore, the complexity of Algorithm 4 is the same as Algorithm 3.

## 7. Experimental Results

We proposed a battery-supercapacitor hybrid architecture and several algorithms to jointly manage the energy harvesting, storage, and usage. In this section, we compare our

architecture 3SC and charging/discharging scheduling algorithm CDSA with the state-of-the-art research [14] and discuss the results of the capacity optimization algorithm COA and the system reward maximization algorithms (L2U and U2L) based on the historical data of solar radiation of 365 days, from July 1, 2012, to June 30, 2013, provided by the Measurement and Instrumentation Data Center (MIDC) of National Renewable Energy Laboratory (NREL).

**7.1. Capacity Optimization.** We evaluate our proposed capacity optimization algorithm COA based on the historical data over the past year (i.e.,  $h = 365$ ), which is shown in Figure 3(a). From the harvesting energy profile, we can see that there is a great difference between the harvested energies in different seasons. In summer, the harvested energy is larger than the task requirement. The excess energy is stored in the battery and the execution of tasks is independent of the stored energy. But, in winter, if the harvested energy is less than the task requirement, the stored energy must be supplied to power tasks. Moreover, if there is no energy in the battery, to guarantee the task requirement, the area of the solar panel should be enlarged to harvest more energy. Therefore, the initial energy of the battery and the season, in which the sensor node starts running, will impact system components' capacity optimization.

There are 12 batteries as shown in Table 1. We set  $w_{sp} = 1$  and  $w_{ba} = 100$ , since the area  $a$  and the rate  $C_i$  differ by two orders of magnitude. *Win* represents that the starting season is winter and the initial energy is equal to  $Win \cdot E_i^b$ . Similarly, *Sum* denotes that the starting season is summer. Figure 2 shows the optimized capacities of system components under different task requirements. From the figure, we can see the following.

- (i) When  $Win \leq 0.23$ , with the same amount of battery and under the same task requirement, the more

TABLE I: Batteries parameters.

Battery	1	2	3	4	5	6	7	8	9	10	11	12
$C_i$ (mAh)	530	680	710	900	1200	1300	1600	2200	2400	2600	2800	3000
$E_i^b$ (W)	4942	6340	6620	8392	11189	13521	14918	20513	22378	24242	26107	27972
$v_i^{\min}$ (V)	3.0											

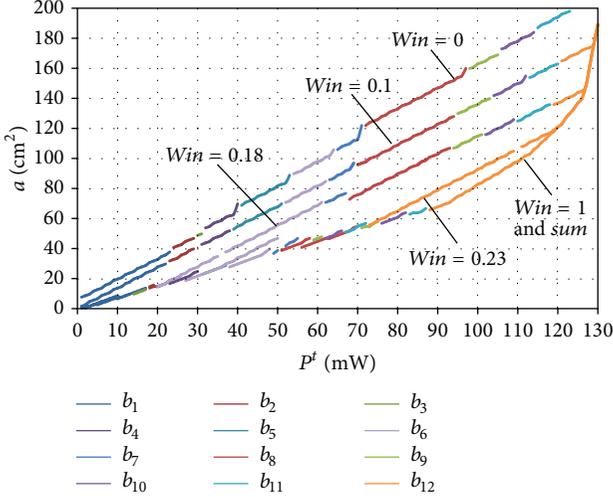
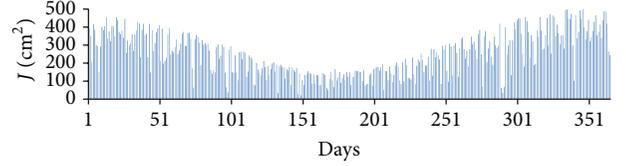


FIGURE 2: The results of COA with different starting seasons and initial energy.

the initial energy  $Win$ , the less the area  $a$ . It is because the initial energy compensates the task requirement for the decrease of the harvested energy.

- (ii) When  $Win > 0.23$ , the result is nearly the same as  $Sum$ . The reason is that the initial energy is enough to compensate for the lacking of the solar energy in winter. Therefore, if the initial energy of the battery is large enough, the starting season almost has no impact on components' capacity selection.
- (iii) When the starting season is summer, no matter what the initial energy is, the results of COA are the same (shown as  $Sum$ ). The system components, which can guarantee tasks' execution in winter, will harvest more energy than tasks required in summer. The remaining energy is stored in the battery. So in summer the battery is charged fully, no matter what the initial energy stored in the battery is.

We developed a simulator in C++ to trace the changes of the energy stored in the battery. Figure 3 shows the comparison of the stored energy profile under different capacities of system components. First, in Figure 3(b),  $\{x = 8, a = 41\}$  is the COA's optimized result. Recall that 10% of energy is reserved (see Section 5) and 20% of the capacity of the battery is lost (see Section 4). So the range of  $y$ -axis is 10% to 80%. Then, in Figure 3(c), we set the area  $a$  to 39 without changing other parameters, and, in Figure 3(d), the battery  $b_7$  is chosen instead of  $b_8$ . From the comparison, we can see that slightly decreasing system components' capacity leads to



(a) Harvesting energy profile

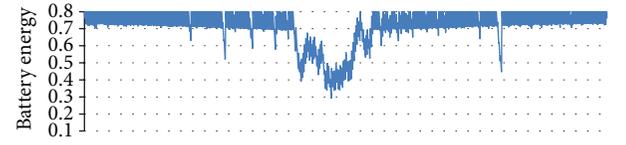
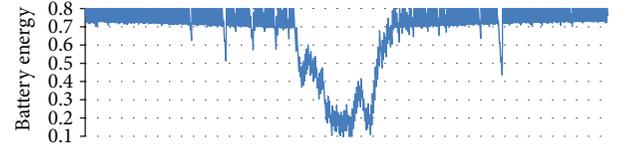
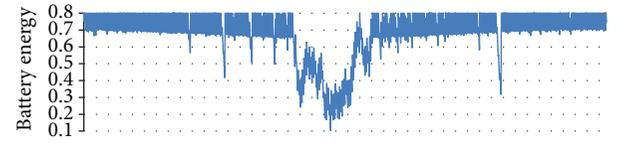
(b) The algorithm COA's optimized result  $x = 8, a = 41$ (c)  $x = 8, a = 39$ (d)  $x = 7, a = 41$ 

FIGURE 3: The energy profile of the battery under different system components' capacities.

energy shortage in some days. Therefore, the lower bound of available energy (Theorem 2) is fairly tight in practice.

**7.2. Architecture Comparison.** We compare our architecture 3SC to the state-of-the-art research HY [14], which focuses on real-world applications. In this experiment, we adopt the algorithm COA's optimized result  $\{x = 12, a = 110\}$  and assume that the two supercapacitors of the architecture HY are big enough. Figure 4 shows the ratio of the harvested energy of HY to that of our architecture 3SC over one year.

$\alpha C$  is the charging current. Note that as long as inequality (1) is satisfied (i.e., when  $x = 12$  and  $a = 110$ ,  $\alpha$  is greater than or equal to 0.19), the harvested energy of our architecture is independent of  $\alpha$ . From Figure 4, we can see that the less the charging current, the less the harvested energy of the architecture HY. The reason is that when two supercapacitors are assigned to the embedded system and the battery, the architecture HY cannot harvest energy until the energy in one of the two supercapacitors is charged to the battery.

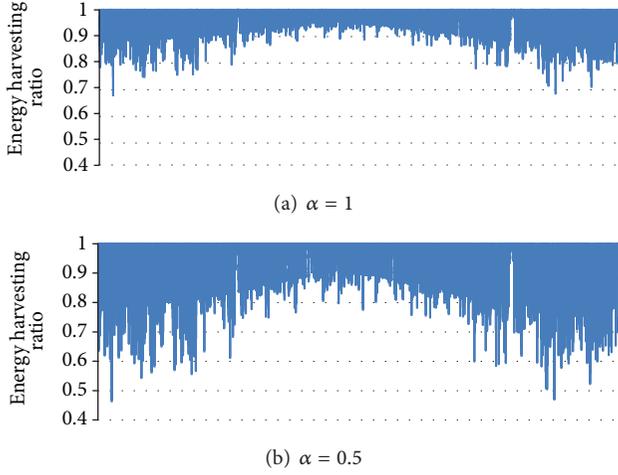


FIGURE 4: The ratio of the harvested energy of HY to that of our architecture 3SC over one year.

So the charging time will impact the energy harvesting. The charging current  $\alpha C$  determines the charging time and further impacts the energy harvesting of the architecture HY. This comparison proves that the charging-discharging conflict reduces the harvested energy and our conflict-free architecture can harvest more energy.

**7.3. System Reward Maximization.** In this subsection, the ILOG CPLEX solver is exploited to obtain the optimal solution from the MIQCP formulation ((17), (18), and (8)) compared to which we evaluate the performance of the proposed system reward maximization algorithms (L2U and U2L). In addition, we implement two variant algorithms: G-L2U and G-U2L. They assign periods according to the reward instead of the metric equality (20). In order to make it solvable by the ILOG CPLEX, the parameters are set as follows. We uniformly sample sixteen  $E_i^d$ s from the data over the past year. According to the optimal solution obtained from the algorithm COA, we set  $a = 20$ ,  $x = 6$ , and  $P^t = 24$ . The task set is generated randomly, where  $r_i \in [1, 100]$ ,  $e_i \in [10, 2000]$  (ms),  $d_i^l = 1$  (s), and  $d_i^u = 20$  (s). The power of each task is generated according to the *Unifast* algorithm [35] such that we can create the solvable task set. The power of each task multiplied by  $d_i^l$  is equal to  $E_i^t$ , and the execution time  $e_i$  is monotonically nondecreasing with increasing  $E_i^t$ .

Figure 5 shows the normalized system rewards of CPLEX and the four algorithms for different number of tasks. Each measurement point is the average value of 200 task sets. The results of U2L and L2U are at least 93% and 85% of the optimal results, respectively. From Figure 5, we observe the following.

- (i) G-U2L and G-L2U have a lower system reward, since they do not take  $e_i$  and  $E_i^t$  into consideration; that is, they ignore the trade-off between the task schedulability and the system reward.
- (ii) When the number of tasks is small, the system reward of U2L is higher than that of L2U. It is because U2L aims at decreasing the period of the task with high

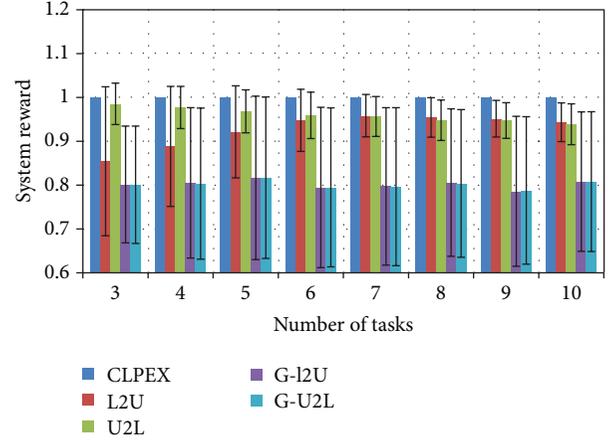


FIGURE 5: The system reward comparison.

reward and this objective is the same as the objective of system reward maximization. However, L2U stretches the period from the lower bound to upper bound. The system reward is inversely proportional to periods. So increasing small periods will bring more impact on the system reward than decreasing large periods; that is, L2U skips more solutions. Therefore, when the number of task is less than 6, U2L is more effective and stable than L2U. When the number of tasks is greater than 5, the solution space of the large task set contains more solutions. So L2U can also find a nearly optimal solution.

## 8. Conclusion

In this paper, we study the joint management of energy harvesting, storage, and usage for green wireless sensor networks. First, we propose a novel system architecture to eliminate the conflicts on charging and discharging between energy components. Based on the system architecture, a charging/discharging scheduling algorithm is designed to effectively harvest, store, and use the solar energy. Then we prove the lower bound of available solar energy. Based on the lower bound, one system components' capacity optimization algorithm, which is proved to be able to find the optimal result, and two system reward maximization algorithms are proposed. We evaluate our architecture and algorithms based on real-life data traces. For our architecture and the charging/discharging scheduling algorithm, the comparison experiment proves that the conflict-free architecture can harvest more energy than the architecture HY. For the capacity optimization algorithm, experimental results show the impacts of realistic data traces and the initial energy on the capacity optimization. In addition, the result of the simulation for the battery's energy shows that the lower bound of available solar energy is fairly tight in practice. For the system reward maximization algorithms, their results are nearly optimal.

## Notations

$i, j, g, k, q$ :	Variables
$H_i, s, y$ :	Intermediate variables
SP:	Solar panel
$SC_i, E_{SC_i}$ :	The $i$ th supercapacitor and energy stored in $SC_i$
BA:	Battery
SN:	Sensor node subsystem
$\xi_{SP}$ :	The supercapacitor that is charged by the solar panel
$\xi_{BA}$ :	The supercapacitor that charges the battery
$\xi_{SN}$ :	The supercapacitor or the battery that supplies energy to the sensor node subsystem
$E_{th}$ :	The minimum energy to start a sensor node subsystem
$a$ :	The area of the solar panel
$P_{cm}$ :	The power density ( $mW/cm^2$ ) of the solar panel
$P_{cm}^{max}$ :	The maximum $P_{cm}$
$P_{SP}$ :	The output power of the solar panel
$v_{BA}$ :	The voltage of the battery
$v_{BA}^{min}$ :	The cut-off voltage
$C$ :	The battery charging and discharging rate
$\alpha$ :	The coefficient of the rate $C$ , $0 < \alpha < 1$
$n$ :	The number of tasks in a task set
$T, t_i$ :	The task set and the $i$ th task ( $i \in [1, n]$ )
$e_i$ :	The worst-case execution time of the task $t_i$
$E_i^t$ :	The energy required by the task $t_i$
$r_i$ :	The reward of the task $t_i$
$d_i$ :	The period of the task $t_i$
$d_i^u, d_i^l$ :	The upper and lower bound of the period $d_i$
$P^t$ :	The power required by a sensor node subsystem
$m$ :	The number of selectable batteries
$B, b_i$ :	The set of selectable batteries and the $i$ th battery ( $i \in [1, m]$ )
$C_i$ :	The charging/discharging rate of the battery $b_i$
$v_i^{min}$ :	The cut-off voltage of the battery $b_i$
$E_i^b$ :	The maximum available energy of the battery $b_i$
$x$ :	The $x$ th battery $b_x$ chosen to use in the energy harvesting subsystem, $x \in [1, m]$
$E_i^d$ :	The generated energy per unit area of the solar panel at the $i$ th day
$h$ :	The number of days of the historic values
$w_{SP}, w_{BA}$ :	The given weight coefficient of the solar panel and the battery, respectively
$A$ :	The maximum acceptable area of the solar panel
$T_{day}$ :	The number of seconds in a day
$F_i^d$ :	The lower bound of available energy in the battery at the end of the $i$ th day
$F_q$ :	The available energy at the time $q$

$\delta_i$ :	The task's metric used to optimize the system reward
$\Omega$ :	The critical task set
$Win$ :	The starting season is winter and the initial energy is equal to $Win \cdot E_i^b$
$Sum$ :	The starting season is summer.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

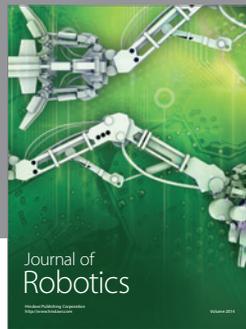
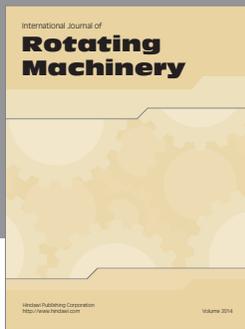
## Acknowledgments

This work was partially supported by the National High Technology Research and Development Program of China (863 Program no. 2011AA040103), the Important National Science and Technology Specific Project no. 2013ZX03005004, and the Natural Science Foundation of China no. 61100023.

## References

- [1] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, September 2002.
- [2] P. Martin, Z. Charbiwala, and M. Srivastava, "DoubleDip: leveraging thermoelectric harvesting for low power monitoring of sporadic water use," in *Proceedings of the 10th ACM Conference on Embedded Networked Sensor Systems (SenSys '12)*, pp. 225–238, November 2012.
- [3] M. Bocca, J. Toivola, L. M. Eriksson, J. Hollmén, and H. Koivo, "Structural health monitoring in wireless sensor networks by the embedded goertzel algorithm," in *Proceedings of the IEEE/ACM 2nd International Conference on Cyber-Physical Systems (ICCPS '11)*, pp. 206–214, Chicago, Ill, USA, April 2011.
- [4] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 457–462, April 2005.
- [5] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 3, pp. 443–461, 2011.
- [6] J. Li, M. Zhao, and Y. Yang, "OWER-MDG: a novel energy replenishment and data gathering mechanism in wireless rechargeable sensor networks," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 5350–5355, December 2012.
- [7] E. Dong, Q. Yan, S. Zhang, and J. Yang, "Self-powered system of articulated bionic robotic fish using wave energy harvesting," *Robot*, vol. 31, no. 6, pp. 501–512, 2009.
- [8] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time scheduling for energy harvesting sensor nodes," *Real-Time Systems*, vol. 37, no. 3, pp. 233–260, 2007.
- [9] M. Chetto and A. Queudet, "A note on edf scheduling for real-time energy harvesting systems," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 1037–1040, 2013.

- [10] X. Jiang, J. Polastre, and D. Culler, "Perpetual environmentally powered sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, pp. 463–468, April 2005.
- [11] C. Park and P. H. Chou, "AmbiMax: autonomous energy harvesting platform for multi-supply wireless sensor nodes," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks (SECON '06)*, pp. 168–177, Reston, Va, USA, September 2006.
- [12] F. Ongaro, S. Saggini, and P. Mattavelli, "Li-Ion battery-supercapacitor hybrid storage system for a long lifetime, photovoltaic-based wireless sensor network," *IEEE Transactions on Power Electronics*, vol. 27, no. 9, pp. 3944–3952, 2012.
- [13] Z. Mao, C. E. Koksal, and N. B. Shroff, "Near optimal power and rate control of multi-hop sensor networks with energy replenishment: basic limitations with finite energy and data storage," *IEEE Transactions on Automatic Control*, vol. 57, no. 4, pp. 815–829, 2012.
- [14] Y. Xiang and S. Pasricha, "Harvesting-aware energy management for multicore platforms with hybrid energy storage," in *Proceedings of the 23rd ACM International Conference of the Great Lakes Symposium on VLSI (GLSVLSI '13)*, pp. 25–30, May 2013.
- [15] Z. Vale, P. Faria, H. Morais, H. Khodr, M. Silva, and P. Kadar, "Scheduling distributed energy resources in an isolated grid—an artificial neural network approach," in *Proceedings of the Power and Energy Society General Meeting*, pp. 1–7, July 2010.
- [16] J.-M. Kwon, K.-T. Kim, and B.-H. Kwon, "Comments on three-arm AC automatic voltage regulator," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 1, pp. 646–649, 2012.
- [17] A. Mirhoseini and F. Koushanfar, "HypoEnergy: hybrid supercapacitor-battery power-supply optimization for energy efficiency," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE '11)*, pp. 1–4, Grenoble, France, March 2011.
- [18] Y. Li, H. Yu, B. Su, and Y. Shang, "Hybrid micropower source for wireless sensor network," *IEEE Sensors Journal*, vol. 8, no. 6, pp. 678–681, 2008.
- [19] X. Lin, Y. Wang, S. Yue, N. Chang, and M. Pedram, "A framework of concurrent task scheduling and dynamic voltage and frequency scaling in real-time embedded systems with energy harvesting," in *Proceedings of the IEEE International Symposium on Low Power Electronics and Design*, pp. 70–75, 2013.
- [20] T. Zhu, Y. Gu, T. He, and Z.-L. Zhang, "EShare: a capacitor-driven energy storage and sharing network for long-term operation," in *Proceedings of the 8th ACM International Conference on Embedded Networked Sensor Systems (SenSys '10)*, pp. 239–252, November 2010.
- [21] M. Pedram, N. Chang, Y. Kim, and Y. Wang, "Hybrid electrical energy storage systems," in *Proceedings of the 16th ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED '10)*, pp. 363–368, August 2010.
- [22] T. Voigt, H. Ritter, and J. Schiller, "Utilizing solar power in wireless sensor networks," in *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks*, pp. 416–422, 2003.
- [23] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 210–216, 2014.
- [24] N. Michelusi, L. Badia, R. Carli, L. Corradini, and M. Zorzi, "Energy management policies for harvesting-based wireless sensor devices with battery degradation," *IEEE Transactions on Communications*, vol. 61, no. 12, pp. 4934–4947, 2013.
- [25] S. Kim and P. H. Chou, "Size and topology optimization for supercapacitor-based sub-watt energy harvesters," *IEEE Transactions on Power Electronics*, vol. 28, no. 4, pp. 2068–2080, 2013.
- [26] K. Zhou, J. Ferreira, and D. Haan, "Optimal energy management strategy and system sizing method for stand-alone photovoltaic-hydrogen systems," *International Journal of Hydrogen Energy*, vol. 33, no. 2, pp. 477–489, 2008.
- [27] E. Schaltz, A. Khaligh, and P. O. Rasmussen, "Influence of battery/ultracapacitor energy-storage sizing on battery lifetime in a fuel cell hybrid electric vehicle," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 8, pp. 3882–3891, 2009.
- [28] J. W. Kimball, B. T. Kuhn, and R. S. Balog, "A system design approach for unattended solar energy harvesting supply," *IEEE Transactions on Power Electronics*, vol. 24, no. 4, pp. 952–962, 2009.
- [29] C. Rusu, R. Melhem, and D. Mossé, "Multiversion scheduling inrechargeable energy-aware real-time systems," in *Proceedings of the 15th IEEE Euromicro Conference on Real-Time Systems*, pp. 95–104, 2003.
- [30] C. Moser, J. Chen, and L. Thiele, "Reward maximization for embedded systems with renewable energies," in *Proceedings of the 14th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA '08)*, pp. 247–256, August 2008.
- [31] C. Moser, J. Chen, and L. Thiele, "Power management in energy harvesting embedded systems with discrete service levels," in *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED '09)*, pp. 413–418, August 2009.
- [32] C. Moser, J.-J. Chen, and L. Thiele, "An energy management framework for energy harvesting embedded systems," *ACM Journal on Emerging Technologies in Computing Systems*, vol. 6, no. 2, article 7, 2010.
- [33] T. Zhu, A. Mohaisen, Y. Ping, and D. Towsley, "DEOS: Dynamic energy-oriented scheduling for sustainable wireless sensor networks," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 2363–2371, Orlando, Fla, USA, March 2012.
- [34] Wikipedia, "Rechargeable battery—wikipedia, the free encyclopedia," <http://en.wikipedia.org/wiki/Rechargeable>, July 2013.
- [35] E. Bini and G. C. Buttazzo, "Measuring the performance of schedulability tests," *Real-Time Systems*, vol. 30, no. 1-2, pp. 129–154, 2005.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

