

# Solar Power Prediction Assisted Intra-task Scheduling for Nonvolatile Sensor Nodes

Daming Zhang, Yongpan Liu, *Senior Member, IEEE*, Jinyang Li, Chun Jason Xue, *Member, IEEE*, Xueqing Li, *Member, IEEE*, Yu Wang, *Senior Member, IEEE*, and Huazhong Yang, *Senior Member, IEEE*

**Abstract**—With the advent of the era of trillion sensors, solar-powered sensor nodes are widely used as they do not require battery charging or replacement. However, the limited and intermittent solar energy supply seriously affects deadline miss rate (DMR) of tasks. Furthermore, traditional solar-powered sensor nodes also suffer from energy loss of battery charging and voltage conversion. Recently, a storage-less and converter-less power supply architecture has been proposed to achieve higher energy efficiency by removing the leaky energy storage and dc voltage conversion. Without energy storages, a node using inter-task scheduling is more sensitive to solar variations, which results in high DMRs. This paper proposes an intra-task scheduling scheme for the storage-less and converter-less solar-powered sensor nodes, whose features include power prediction based on classified solar profiles, a trigger mechanism to select scheduling points, an artificial neural network to calculate task priorities and a fine-grained task selection algorithm. Experimental results show that the proposed algorithm reduces DMR by up to 30% and improves energy utilization efficiency by 20% with trivial energy overheads.

**Index Terms**—Deadline miss rate (DMR), energy utilization efficiency, intra-task scheduling, storage-less and converter-less nonvolatile sensor nodes.

Manuscript received July 3, 2015; revised October 30, 2015; accepted January 11, 2016. Date of publication February 10, 2016; date of current version April 19, 2016. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61271269 and Grant 61373026, in part by the High-Tech Research and Development (863) Program under Contract 2013AA01320, in part by the Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions under Contract YETP0102, in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CityU 9042209, in part by the 973 Project under Grant 2013CB329000, and in part by the Tsinghua University Initiative Scientific Research Program. This paper was recommended by Associate Editor S. Hu, S. Hu, and A. Y. Zomaya. (*Corresponding author: Yongpan Liu.*)

D. Zhang was with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: zdm06@mails.tsinghua.edu.cn). He is now with the China Academy of Launch Vehicle Technology, Beijing 100076, China (e-mail: 15901028892@163.com).

Y. Liu, J. Li, Y. Wang, and H. Yang are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: ypliu@mail.tsinghua.edu.cn; yu-wang@mail.tsinghua.edu.cn; yanghz@mail.tsinghua.edu.cn; lly01@mails.tsinghua.edu.cn).

C. J. Xue is with the Department of Computer Science and Engineering, City University of Hong Kong, Hong Kong 999077 (e-mail: jasonxue@cityu.edu.hk).

X. Li is with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA 16802 USA (e-mail: lixueq@cse.psu.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2016.2527710

## I. INTRODUCTION

WIRELESS sensor nodes are widely used, such as structure health monitoring (SHM) [1], body motion analysis [2], forest watershed deployment [3] and industrial control [4]. In most cases, sensor nodes are powered by batteries, where frequent replacement or recharging is needed. In the coming trillion sensor world [5], maintaining such a great quantity of sensor nodes becomes a prohibitive task both in time and cost. Energy harvesting techniques are promising to reduce the maintenance costs and prolong the operating time of sensor nodes. Among common energy harvesting sources, such as solar, thermal, wireless, and vibration energy [6], solar energy provides the highest power density and hence solar-powered sensor nodes are the most widely used. Since the solar energy is limited and intermittent, it is difficult to guarantee that the deadline of each task can be satisfied when the solar power is used as the only energy supplier. It is no surprise that plenty of energy-driven task scheduling techniques are investigated to minimize deadline miss rate (DMR) in solar-powered systems.

Most of the existing task scheduling techniques [7], [8] perform well for sensor nodes with the “harvest-store-use” [9] supply architecture [see Fig. 1(a)]. This supply system contains dc–dc converters and energy storage components, such as batteries, super capacitors, and so on. The converters regulate and stabilize the supply voltage at a required level and the energy storages work as energy buffers to relieve the impacts of the solar power variations. The load system contains one or multiple processing elements (PEs) to match the load consumptions to the power profiles [10]. However, it is pointed out that such an architecture suffers from low energy efficiency [11], due to the nontrivial energy loss caused by the dc–dc converters and the charging/discharging procedures of capacitors.

Recently, a storage-less and converter-less supply system for the solar-powered sensor nodes [12], [13] is developed, which achieves up to 23% higher energy efficiency than the harvest-store-use architecture. However, since there is no energy storage component in the new architecture, the power supply is not guaranteed to be stable but intermittent. Meanwhile, existing scheduling algorithms are inter-task methods and cannot be interrupted during executions. As a result, these algorithms often cause a high DMR when the power supply fails (even temporarily), leading to more task failures or rescheduling operations.

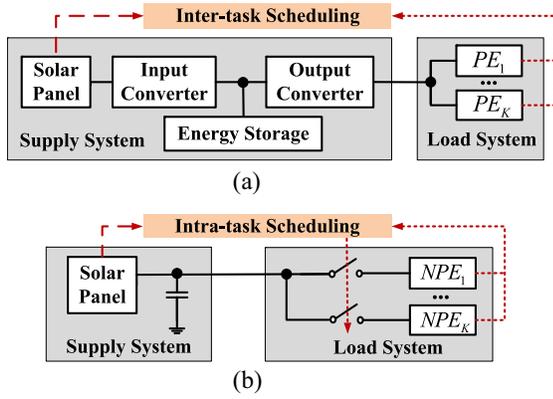


Fig. 1. Architectures of the solar-powered sensor nodes. (a) Harvest-store-use architecture. (b) SCSN architecture.

In this paper, we propose a solar power prediction assisted intra-task scheduling for the storage-less and converter-less solar-powered sensor node (SCSN). Compared with the inter-task scheduling, the intra-task scheduling, which is a form of preemptive scheduling, can adjust tasks at any time and closely follow the solar variations. To the best of our knowledge, this is the first approach with such capabilities. The experimental results show the proposed algorithm reduces the DMR by up to 30% and improves the energy efficiency by 20% on average. The main contributions are listed as follows.

- 1) Formulating the system model of the SCSN architecture as shown in Fig. 1(b), which contains a storage-less and converter-less supply system with several nonvolatile PEs (NPEs) (e.g., FeRAM-based [14] or ReRAM-based [15]–[17] NPEs) for load matching.
- 2) Providing a schedulability analyzing method and an optimal DMR estimation algorithm.
- 3) Proposing an integer nonlinear programming (INLP) solution for the optimal intra-task scheduling.
- 4) Developing an online intra-task scheduling algorithm on the SCSN architecture, consisting of classified solar power prediction, a trigger mechanism, an artificial neural network (ANN) to calculate task priorities, and a task selection method.
- 5) Validating the proposed model and algorithm on a prototype and comparing it with several up-to-date scheduling methods.

The rest of this paper is organized as follows. Section II presents the motivation and challenges of the intra-task scheduling. Section III introduces the system model and Section IV analyzes the schedulability and provides an optimal DMR estimation method, as well as an INLP formulation for optimal scheduling. The online algorithm is illustrated in Section V. A prototype is used to validate the system model and the online algorithm in Section VI. Section VII shows the experimental results via simulations. Section VIII discusses the related work and Section IX concludes this paper.

## II. OVERVIEW

This section first gives a motivational example for intra-task scheduling on the SCSN architecture and then discusses the challenges.

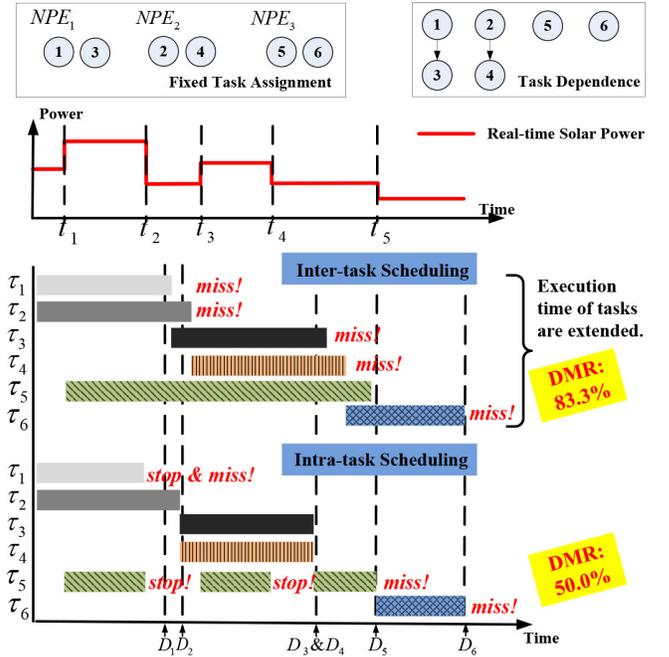


Fig. 2. Motivation for intra-task scheduling on the SCSN architecture.

### A. Motivation

Compared with the inter-task scheduling, the intra-task one adjusts the task in a more timely and fine-grained way. Fig. 2 compares two approaches above on the SCSN architecture. Given a set of periodic tasks ( $\tau_1$ – $\tau_6$ ) with corresponding deadlines ( $D_1$ – $D_6$ ), the tasks assigned on each NPE are presented on the top left corner of Fig. 2, where one task can only be executed on one NPE. The task dependence is given on the top right corner, where data dependency exists if there is an arrow between two tasks.

Since tasks cannot be interrupted in the inter-task scheduling, no new task can be rescheduled unless the current one is completed. When there is a solar power drop, the task execution time increases and five tasks miss their deadlines (DMR: 83.3%), because rescheduling happens too late due to the long delay to finish the previous task. On the other side, the intra-task scheduling adjusts tasks in time once variations of solar power or changes of task status happen. For example, task  $\tau_1$  is stopped during execution at  $t_2$  when solar power decreases. Task  $\tau_2$  can continue to be executed and catch its deadline with the power saved by stopping  $\tau_1$ , since task  $\tau_2$  has a loose deadline than  $\tau_1$ . Moreover, in order to complete task  $\tau_3$  and  $\tau_4$  in time, task  $\tau_5$  is executed intermittently according to solar variations from  $t_1$  to  $t_5$  and is not completed at last. In this way, DMR is reduced to 50.0% using the intra-task scheduling scheme.

Solar power prediction is another critical factor to reduce DMR in the intra-task scheduling. Previous prediction methods, such as exponentially weighted moving average (EWMA) [18] and weather conditioned moving average (WCMA) [19]), are effective to estimate common solar profiles in average cases. Given a large quantity of real solar data [20], Fig. 3 shows different types of solar profiles

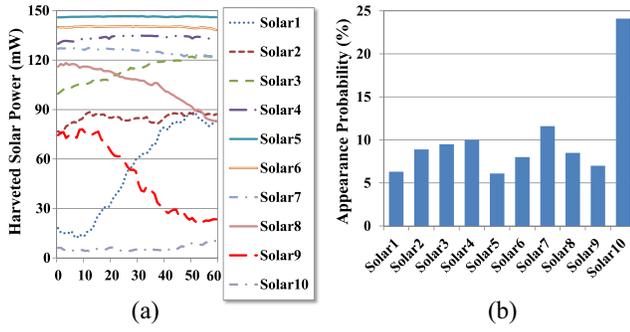


Fig. 3. Types and appearance probabilities of solar profiles in a period. (a) Time in a period (min). (b) Type of solar profiles in a period.

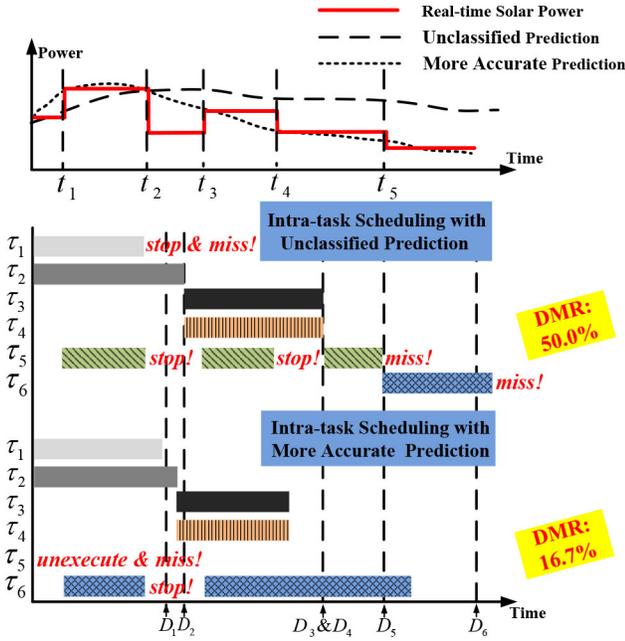


Fig. 4. Intra-task scheduling with different prediction methods.

with different appearance probabilities. Fig. 4 compares the scheduling effects of using classified solar profiles based prediction with that of using average-case prediction. As we can see, task  $\tau_5$  is thoroughly abandoned and  $\tau_6$  is completed, which reduce DMR to 16.7%. Therefore, intra-task scheduling scheme assisted by classified solar profiles based prediction is more promising.

### B. Challenges

Since the intra-task scheduler can execute tasks at any time slots in a period, design space increases exponentially with the number of tasks and time slots. The huge optimization space brings three major challenges.

- 1) When to start a scheduling procedure?
- 2) How to reschedule the tasks during execution?
- 3) How to predict solar profiles more accurately?

To solve the problems, we extract the system model in Section III. We develop a theoretical analysis for offline intra-task scheduling, including the schedulability analysis, the optimal DMR estimation method and an INLP formulation for the optimal scheduling in Section IV. Finally, an

TABLE I  
PARAMETERS AND VARIABLES OF THE MODEL

Parameter	Description	
Task	$V$	Task Set, $V = \{\tau_1, \dots, \tau_i, \dots, \tau_N\}$
	$D_i$	Deadline of $\tau_i$ in Each Period (s)
	$L_i$	Total Execution Time of $\tau_i$ in Each Period (s)
	$P_i$	Maximal Power of $\tau_i$ (mW)
	$E_{i,j}$	Task Dependence from $\tau_i$ to $\tau_j$
System	$ T $	Number of Time Slots in Each Period
	$\Delta t$	Length of Each Time Slot (s)
	$P^s(t)$	Average Solar Power in the $t$ th Time Slot (mW)
	$A_k$	Tasks Executed on the $k$ th NPE
Variable	$x_i(t)$	Scheduling Results for $\tau_i$ in the $t$ th Time Slot
	$p_i(t)$	Average Power of $\tau_i$ in the $t$ th Time Slot (mW)
	$l_i^{rem}(t)$	Remaining Execution Time of $\tau_i$ at the Beginning of the $t$ th Time Slot (s)

efficient online intra-task scheduling algorithm is proposed in Section V.

## III. SYSTEM MODELING

This section describes the system modeling for the intra-task scheduling, including task parameters, system parameters, and scheduling variables.

### A. Task Parameters

Table I presents the parameters and variables of the system model. This paper targets a set of periodic tasks executed on the SCSN architecture. A directed acyclic graph  $G(V, E)$  describes the tasks and their dependencies, where the nodes in  $V$  denote the tasks set and the edges in  $E$  represent the dependencies. In  $V$ ,  $N$  tasks ( $\{\tau_1, \tau_2, \dots, \tau_N\}$ ) are executed by  $K$  NPEs. For each task ( $\tau_i$ ), there are three parameters: 1)  $D_i$  is the periodic deadline of  $\tau_i$  in each period; 2)  $L_i$  is the required execution time; and 3)  $P_i$  is the power consumption of  $\tau_i$ . Note that tasks executed in one period are independent of those in others. In  $E$ ,  $E_{i,j}$  denotes the data dependency from  $\tau_i$  to  $\tau_j$ . That is,  $E_{i,j} = 1$ , if  $\tau_j$  depends on the results of  $\tau_i$ .

### B. System Parameters

The system contains four parameters. A period is denoted as  $T = \{1, 2, \dots, t, \dots, |T|\}$ . It has  $|T|$  time slots and the length of each time slot is  $\Delta t$ . The time slot is the smallest unit for intra-task scheduling and tasks can be adjusted at the beginning of each time slot.  $P^s(t)$  denotes the average solar power in the  $t$ th time slot.  $A_k$  ( $k \in [1, K]$ ) is a task set including the tasks executed on the  $k$ th NPE. Each NPE executes one task at a time.

### C. Scheduling Variables

We define the scheduling variables as follows.  $x_i(t)$  is the independent 0-1 variable, which denotes the scheduling results for  $\tau_i$  in the  $t$ th time slot. That is,  $x_i(t) = 1$ , if  $\tau_i$  is executed.

Based on  $x_i(t)$ , we propose two intermediate variables:  $p_i(t)$  and  $l_i^{rem}(t)$ .  $p_i(t)$  is the average power consumption of  $\tau_i$  in

the  $t$ th time slot which is calculated as follows:

$$p_i(t) = \begin{cases} 0, & \text{if } x_i(t) = 0 \\ P_i, & \text{else if } P^s(t) \geq \sum_{k=1}^N x_k(t) \cdot P_k \\ P^s(t) \cdot \frac{P_i}{\sum_{k=1}^N x_k(t) \cdot P_k}, & \text{otherwise} \end{cases} \quad (1)$$

where  $p_i(t) = 0$ , if  $\tau_i$  is not executed;  $p_i(t) = P_i$ , if the solar power is sufficient for load consumption; and  $p_i(t)$  is a fraction of the solar power ( $P^s(t)$ ) consumed by  $\tau_i$  (less than  $P_i$ ), if the solar power is not sufficient in the  $t$ th time slot.

$l_i^{\text{rem}}(t)$  is the remaining execution time of  $\tau_i$  at the beginning of the  $t$ th time slot, which is calculated via the energy consumption as follows:

$$l_i^{\text{rem}}(t) = L_i - \Delta E_i(t-1)/P_i$$

$$\Delta E_i(t-1) = \sum_{m=1}^{t-1} x_i(m) \cdot p_i(m) \cdot \Delta t \quad (2)$$

where  $\Delta E_i(t-1)$  and  $\Delta E_i(t-1)/P_i$  are total energy consumption and the total execution time of  $\tau_i$  during the last  $t-1$  time slots, respectively. Thus,  $l_i^{\text{rem}}(t) = L_i$ , if  $\tau_i$  has never been executed and  $l_i^{\text{rem}}(t) = 0$ , if  $\tau_i$  is completed.

In addition, we define a function  $\theta()$  to denote whether  $\tau_i$  misses its deadline or not

$$\theta(l_i^{\text{rem}}(\lceil D_i/\Delta t \rceil)) = \begin{cases} 1, & \text{if } l_i^{\text{rem}}(\lceil D_i/\Delta t \rceil) > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$\tau_i$  misses its deadline, if  $l_i^{\text{rem}}(\lceil D_i/\Delta t \rceil) > 0$ ; otherwise,  $\tau_i$  meets its deadline and the tasks will be included in DMR calculation.

#### IV. OFFLINE THEORETICAL ANALYSIS AND INLP FORMULATION

This section first analyzes the schedulability with the changeable energy supply. The optimal DMR estimation for the nonschedulable cases is then presented. After that, an INLP formulation for optimal intra-task scheduling is developed.

##### A. Schedulable Condition for Energy Harvesting Sensor Nodes

The tasks are schedulable if all of them meet deadlines under the given constraints, such as energy, computation resource, etc. Previous work had well studied the schedulability of deterministic scheduling problems [21]–[23]. However, it is absent for energy harvesting sensor nodes, where power profiles are unknown. We propose a schedulable condition in a period for energy harvesting sensor nodes as follows:

$$\forall t \in [1, |T|], \sum_{m=1}^t E_{\max}^s(m) \geq \sum_{i=1}^N E_i^c(t) \quad (4)$$

where  $E_{\max}^s(m)$  is the maximum energy supply at the  $m$ th time slot and  $E_i^c(t)$  is the energy requirement of task  $\tau_i$  at the  $t$ th time slot. It means that the accumulated energy supply should always satisfy the total energy requirement of tasks, whose deadlines are approaching, at any time slot  $t$ .  $E_{\max}^s(m)$  ( $m \in [1, |T|]$ ) is calculated as follows:

$$E_{\max}^s(m) = \min(P^s(m), P_{\text{const}}^{\text{NPE}}, P_{\max}^c(m)) \cdot \Delta t \quad (5)$$

where  $P^s(m)$  is the harvested solar power,  $P_{\text{const}}^{\text{NPE}}$  is the total peak power of all NPEs, and  $P_{\max}^c(m)$  is the maximum power to execute the tasks which can be started at the present time slot.  $E_{\max}^s(m)$  is limited by above three parameters due to the following reasons. First,  $E_{\max}^s(m)$  is always smaller than the harvested solar energy, as there is no other energy source. Second, it cannot be larger than  $P_{\text{const}}^{\text{NPE}}$ , since  $P_{\text{const}}^{\text{NPE}}$  is the maximum power which the node can draw. Third, the schedulability is also constrained by the task dependence, which is characterized by  $P_{\max}^c(m)$ . It implies that the schedulability cannot be met due to the computing resource and task dependence constraints, even if the harvested solar power is large enough to supply all tasks when the deadlines are approaching.

In the following, we calculate  $P_{\text{const}}^{\text{NPE}}$  and  $P_{\max}^c(m)$ , respectively.  $P_{\text{const}}^{\text{NPE}}$  is a constant value given by

$$P_{\text{const}}^{\text{NPE}} = \sum_{k=1}^K \max_{i \in A_k} (P_i) \quad (6)$$

where  $\max(P_i)$  is the maximum power consumption of the task executed on the  $k$ th NPE.  $P_{\max}^c(m)$  is the maximum power requirement of all the tasks in the  $m$ th time slot. It is calculated as follows:

$$P_{\max}^c(m) = \sum_{i=1}^N \text{exe}_i(m) \cdot P_i$$

$$\text{exe}_i(m) = \begin{cases} 1, & \text{if } \lceil R_i/\Delta t \rceil \leq m \leq \lceil D_i/\Delta t \rceil \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where  $\text{exe}_i(m)$  denotes whether a task can be executed in the  $m$ th time slot.  $\text{exe}_i(m) = 1$ , means task  $\tau_i$  can be executed during the period from its start time to the deadline; at other times,  $\text{exe}_i(m) = 0$ . In (7),  $R_i$  ( $i \in [1, N]$ ) is the start time of task  $\tau_i$ .  $R_i = 0$ , if  $\tau_i$  does not depend on any tasks, which means  $\tau_i$  can start to be executed at the beginning of the period; otherwise, we estimate  $R_i$  as follows:

$$R_i = \max(L_k \cdot E_{k,i}) \quad (8)$$

where  $k \in [1, N]$ .  $R_i$  is equal to the longest execution time of the tasks ( $\tau_k$ ), on which  $\tau_i$  depends. This value is equal to the actual start time of  $\tau_i$ , if all of these tasks have no data dependencies; otherwise,  $R_i$  is earlier than the actual start time.

The energy requirement  $E_i^c(t)$  ( $t \in [1, |T|]$ ) is calculated as follows:

$$E_i^c(t) = \left\lfloor 1 + \frac{t \cdot \Delta t - D_i}{|T| \cdot \Delta t} \right\rfloor \cdot (L_i \cdot P_i) \quad (9)$$

where  $E_i^c(t) = L_i \cdot P_i$ , if the deadline of task  $\tau_i$  arrives ( $t \cdot \Delta t \geq D_i$ ) and the energy requirement needs to be satisfied; otherwise,  $E_i^c(t) = 0$ .

According to (4), we can draw the curve of  $\sum_{m=1}^t E_{\max}^s(m)$  and  $\sum_{i=1}^N E_i^c(t)$  in Fig. 5. In Fig. 5(a), the tasks are schedulable since the schedulable condition is satisfied. In this situation, one or more scheduling methods to meet deadline constraints exist and the DMR is zero. However, the tasks are not schedulable in Fig. 5(b), since the energy violation happens.

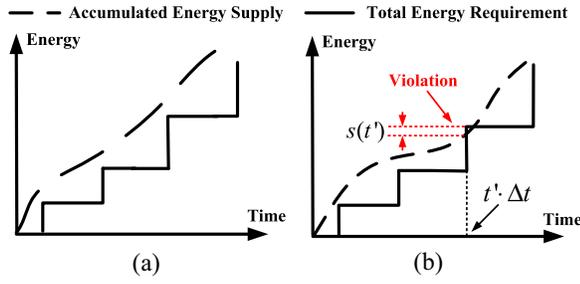


Fig. 5. Relationship between energy supply and requirement. (a) Schedulable case. (b) Nonschedulable case.

### B. Analytical DMR Estimation

When the tasks are not schedulable, we can estimate the optimal (lowest) DMR by omitting some tasks to eliminate energy violations. We use a set  $T' = \{t' | \text{Energy violation happens}\}$  to store the time slots, where an energy violation happens. Thus, the energy violation  $s(t')$  in the  $t'$ th time slot can be described as follows:

$$s(t') = \sum_{i=1}^N E_i^c(t') - \sum_{m=1}^{t'} E_{\max}^s(m) \quad (10)$$

where  $s(t')$  is equal to the extra energy requirement in the  $t'$ th time slot, if the energy supply is less than the requirement; otherwise,  $s(t') = 0$ . The violation reduction ( $\Delta s_i$ ) by removing a task  $\tau_i$  ( $i \in [1, N]$ ) is calculated as follows:

$$\Delta s_i = \sum_{t' \in T'} \min(L_i \cdot P_i, s(t')) \cdot \sigma(D_i - t' \cdot \Delta t) \quad (11)$$

$$\sigma(D_i - t' \cdot \Delta t) = \begin{cases} 1, & \text{if } D_i \leq t' \cdot \Delta t \\ 0, & \text{otherwise} \end{cases}$$

where  $\Delta s_i$  is equal to the sum of the minimum value between its energy requirement ( $L_i \cdot P_i$ ) and the energy violation ( $s(t')$ ) in the  $t'$ th time slot only if its deadline  $D_i$  is no later than the end of the slot ( $t' \cdot \Delta t$ ). Based on the definition of  $s(t')$  and  $\Delta s_i$ , we propose a theorem to estimate the optimal DMR.

**Theorem 1:** The lower bound of the optimal DMR is  $|V'|/|V|$ , where  $|V'|$  and  $|V|$  are the numbers of the tasks in  $V'$  and  $V$ , respectively.  $V$  represent the original task set to be executed and  $V'$  is the set containing the tasks removed from  $V$  with the following method. Remove the task with the maximum  $\Delta s_i$  without any dependent tasks in  $V$  and add it into  $V'$ . Randomly select one task when there is a tie. Update  $s(t')$  and  $\Delta s_i$  and repeat the step iteratively until no energy violations exist for the remaining tasks.

**Proof:** Without losing generality, suppose that a smaller DMR ( $(|V'| - 1)/|V|$ ) can be achieved if we remove a different task  $\tau_j$  to eliminate all energy violations in the last second iteration, while the task selected by Theorem 1 is  $\tau_i$ . It is guaranteed that  $\Delta s_i \geq \Delta s_j$  by Theorem 1, because  $\tau_i$  is the task to achieve the maximal value of violation reduction. Since we still need one more task to be removed from  $V$  after  $\tau_i$ , we have  $\Delta s_i < \sum_{t' \in T'} s(t')$ . However, we have  $\Delta s_j = \sum_{t' \in T'} s(t')$ , since the remaining tasks in  $V$  are schedulable after removing  $\tau_j$ . Therefore, it leads to the conclusion

$\Delta s_i < \Delta s_j$ , which is inconsistent with  $\Delta s_i \geq \Delta s_j$ . Hence we prove that the assumption does not hold.

Theorem 1 enables us to estimate the optimal DMR when the original tasks are not schedulable. When there is no task dependence, the theorem provides the accurate optimal DMR estimation. When the task dependence exists, it provides a theoretical lower bound of the optimal DMR, because the start time of the tasks estimated by (8) may be smaller than the actual values. The complexity of the DMR estimation is linear  $O(N)$ , where  $N$  is the number of tasks. Thus, it is suitable for fast optimal DMR estimation for design space exploration in large-scale cases. Although the theorem provides an efficient way to estimate the optimal DMR, we still need algorithms to obtain the scheduling solution, which is presented in Section IV-C.

### C. INLP Formulation for Intra-task Scheduling

We develop an offline formulation for the optimal intra-task scheduling results. Based on the parameters and variables defined in Section III, the formulation can be developed as an INLP model as follows. Without loss of generality, the objective is to find the optimal scheduling results for all the tasks in a period ( $\{x_i(m)\}$ ), which minimizes the DMR from any time slot  $t$  ( $t \in [1, |T|]$ ) to the end of a period. The inputs consist of the given solar power ( $\{P^s(m)\}$ ,  $m \in [t, |T|]$ ) and the current task status at the beginning of the  $t$ th time slot ( $\{l_i^{\text{rem}}(t)\}$ ). The outputs are the optimal scheduling results ( $\{x_i(m)\}$ ,  $m \in [t, |T|]$ )

$$\text{objective: } \min \sum_{i=1}^N \theta(l_i^{\text{rem}}(D_i))/N \quad (12)$$

subject to:

1) task dependence constraint

$$l_j^{\text{rem}}(m) = L_j, \quad \text{if } \sum_{i=1}^N E_{i,j} \cdot l_i^{\text{rem}}(m) > 0 \quad (13)$$

2) solar power constraint

$$\sum_{i=1}^N x_i(m) \cdot p_i(m) \leq P^s(m) \quad (14)$$

3) task energy constraint

$$\sum_{m=t}^{|T|} p_i(m) \cdot \Delta t \leq P_i \cdot l_i^{\text{rem}}(t) \quad (15)$$

4) NPE resource constraint

$$\sum_{\forall i \in A_k} x_i(m) \leq 1 \quad (16)$$

where  $i, j \in [1, N]$ ,  $m \in [t, |T|]$  and  $k \in [1, K]$ . There are four constraints in the formulation. The task dependence constraint (13) means that  $\tau_j$  starts only if all its depending tasks are completed. The solar power constraint (14) means that the load power is no more than the solar power. The task energy constraint (15) means that the remaining energy consumption of  $\tau_i$  cannot be violated. The NPE resource constraint (16) means that an NPE can only run one task at the same time.

The complexity of the formulation is  $O(2^{N \cdot (|T| - t + 1)})$ , where  $N$  is the number of tasks and  $|T|$  denotes the number of time

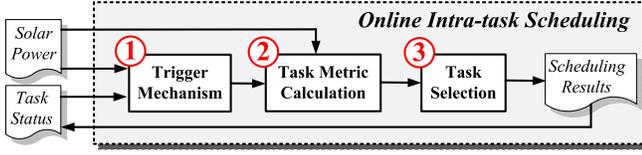


Fig. 6. Diagram of the online intra-task scheduling algorithm.

slots in a period. It can be solved by a nonlinear programming solver such as LINGO. In addition, the optimal DMR serves as the exact lower bound in the experiment. However, the optimal results are obtained offline with the known solar power and it cannot be directly applied for online scheduling. Thus, we develop an online intra-task scheduling algorithm in Section V.

## V. ONLINE SCHEDULING ALGORITHM

This section presents the diagram of the online intra-task scheduling on the SCSN architecture and then introduces its key components.

### A. Online Intra-task Scheduling Algorithm

To solve the challenges in Section II-B, we design the online intra-task scheduling algorithm based on the following ideas.

- 1) Considering the scheduling points, we propose a trigger mechanism to choose the suitable scheduling points based on the current solar power and task status.
- 2) Considering the scheduling strategy, we calculate the real-time task priorities for online task selection.
- 3) Considering solar power prediction, we classify the period-level solar profiles into several types based on their basic shapes and develop a classified solar power prediction method.

Fig. 6 presents the diagram of the proposed scheduling algorithm, which contains three parts: 1) trigger mechanism; 2) task priority calculation; and 3) task selection. First, a trigger mechanism launches the scheduling algorithm by detecting both solar variation and task status at the beginning of each time slot. After that, the current task priorities are obtained with classified solar power prediction by two ANNs. Finally, a task selection method determines the execution status of the tasks based on their priorities. The three parts of the algorithm are presented in details in the rest of this section.

### B. Trigger Mechanism for Intra-task Scheduling

As tasks can be interrupted at the beginning of any time slot, we first need to choose scheduling points for online intra-task scheduling, as scheduling at the beginning of all the time slots is costly and inefficient. Observing the variation of solar power and task status on the SCSN architecture, we find that intra-task scheduling should be done in the following situations, which may affect the DMR of the periods.

- 1) The executed tasks are completed or the new tasks start.
- 2) The executed tasks miss their deadlines.
- 3) The solar variation happens.

Thus, we develop a trigger mechanism, including all the situations as triggers.

Let  $over_i$  denote the completion trigger of  $\tau_i$ . A scheduling point is triggered, if  $over_i = 1$ . It means  $\tau_i$  is completed. Let  $start_i$  describe whether  $\tau_i$  can be executed or not. That is,  $start_i = 1$ , if all its depending tasks are completed and it can be executed. In the worst case, the number of the scheduling points triggered by  $\{over_i\}$  and  $\{start_i\}$  is  $N$ , which is the number of tasks.

Let  $miss_i$  denote the deadline missing trigger of  $\tau_i$ . It is triggered, if  $miss_i = 1$ . It means  $\tau_i$  misses its deadline ( $l_i^{rem}(D_i) > 0$ ) and it does not have dependent tasks ( $\sum_{j=1}^N E_{i,j} = 0$ ). In the worst case, the number of the scheduling points is  $N$ .

Let  $sv(t)$  denote the trigger of the solar variation at the beginning of the  $t$ th time slot and it is calculated as follows:

$$sv(t) = \begin{cases} 1, & \text{if } |P^s(t) - P^s(t')| > \delta_1 \& (t - t') \cdot \Delta t > \delta_2 \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

where  $t'$  is the time slot of the last scheduling point.  $\delta_1$  and  $\delta_2$  are the thresholds of solar variation quantity and time, respectively. They are defined by users. The scheduling point is triggered, if  $sv(t) = 1$ . It means the solar variation quantity is larger than the threshold ( $|P^s(t) - P^s(t')| > \delta_1$ ) and it is not a short-time shock ( $(t - t') \cdot \Delta t > \delta_2$ ). As  $sv(t)$  is related to the thresholds ( $\delta_1$  and  $\delta_2$ ), smaller thresholds mean larger numbers of scheduling points triggered by  $sv(t)$ . In the worst case, the number is  $|T|$ , which is the total number of the time slots in a period.

### C. Task Priority Calculation

After solving the problem of choosing the scheduling points by the trigger mechanism, we need to decide which tasks to be executed at each scheduling point? The optimal DMR is obtained by offline formulation in Section IV-C. However, we cannot use the optimal scheduling results ( $\{x_i(m)\}, m \in [t, |T|]$ ) for online scheduling directly. It is because the solar power obtained in real applications is different from that used in the formulation. A parameter to characterize the long term trend of task priority is more preferred, as they are more stable under small solar variations. Based on the scheduling results ( $\{x_i(m)\}, m \in [t, |T|]$ ), we define the task priorities ( $\{\lambda_i\}, i \in [1, N]$ ) as follows:

$$\lambda_i = \frac{\sum_{m=t}^{|T|} x_i(m) \cdot (|T| - m + 1) \cdot \Delta t}{l_i^{rem}(t)} \quad (18)$$

which is the weighted execution status of task  $\tau_i$ . Larger values mean higher priorities. That is, we prefer to assign higher priorities to the tasks, which are executed earlier based on the optimal scheduling results. According to the priorities, we can select tasks to execute on the NPEs.

As the inputs (solar power in a period and task status) and outputs (types of solar profiles and task priorities) have quite complex nonlinear relationships, they cannot be directly described by the traditional data fitting models (e.g., linear, polynomial, exponential models, etc.). ANN models have much better ability of nonlinear data fitting and self-learning with new training samples, which is more suitable to be trained for online solar profile prediction and task priority calculation on the sensor nodes. Zhang *et al.* [24] developed a shallow

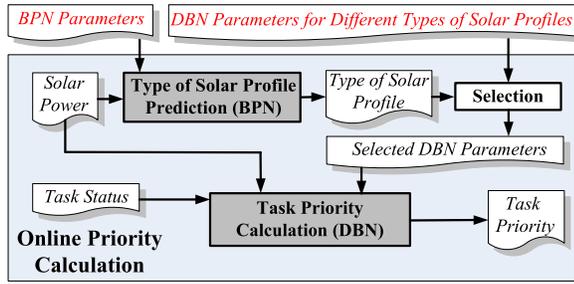


Fig. 7. Proposed diagram of task priority calculation.

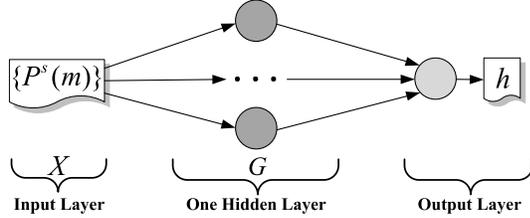


Fig. 8. Architecture of the BPN.

learning model [back propagation network (BPN)] with fixed parameters, which are trained offline with the priorities based on the unclassified solar power. Thus, it leads to inaccurate prediction when the solar profiles are inconsistent with the average case. Besides, the BPN is too simple to calculate the suitable priorities effectively based on the complex inputs of the solar power and task status.

Thus, we first predict the type of solar profile in the current period by a BPN and select the corresponding parameters for a deep learning model [deep belief networks (DBNs)] based on the predicted type. After that, we calculate the task priorities by the DBN with the selected parameters (see Fig. 7). The reason to select them is that the BPN and DBN are the most suitable models for one and multidimensional general data (e.g., solar power and task status) among the frequently-used ANN models, respectively. The BPN is used to predict the type of the solar profile with a small complexity while the DBN has the ability of data feature extraction and calculates the priorities more effectively.

1) *BPN-Based Solar Profile Prediction*: The architecture of the BPN is presented in Fig. 8. It contains three parts: 1) the input layer; 2) the hidden layer; and 3) the output layer. The inputs are the solar power ( $\{P^s(m)\}$ ,  $m \in [t-|T|+1, t]$ ) during the last  $|T|$  time slots and a vector  $X$  is used to store them.  $G$  is the set of neurons in the hidden layer and it contains  $|G|$  neurons. The output layer contains one neuron and the predicted type of solar profile  $h$  ( $h \in [1, H]$ ), where  $H$  is the number of types.

We get the predicted type ( $h$ ) by online BPN calculation as follows:

$$\begin{aligned} G &= \text{sigmoid}(B^1 \cdot X + I^1) \\ h &= \text{sigmoid}(B^2 \cdot G + I^2) \\ \text{sigmoid}(x) &= (1 + e^{-x})^{-1} \end{aligned} \quad (19)$$

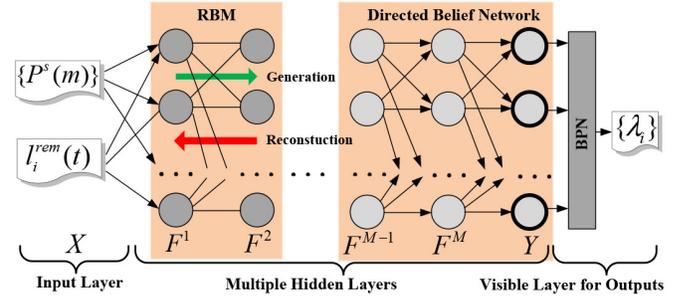


Fig. 9. Architecture of the DBN.

where  $\text{sigmoid}()$  is the nonlinear activation function.  $B^1$  is the weight coefficient matrix and  $I^1$  is the threshold vector from the input layer to the hidden layer.  $B^2$  is the weight coefficient vector and  $I^2$  is the threshold value from the hidden layer to the output layer.  $\{B^1, B^2, I^1, I^2\}$  are obtained as follows. We first classify all the period-level solar profiles ( $\{P^s(m)\}$ ,  $m \in [t, t + |T| - 1]$ ) into  $H$  types based on their basic shapes with the  $k$ -means algorithm. The solar power before classification and the corresponding type ( $h$ ) are used as the training samples. After that, the parameters of the BPN  $\{B^1, B^2, I^1, I^2\}$  are trained offline with these samples. The complexity of online BPN calculation is  $O(|G| + 1)$ , which is equal to the total number of the neurons in the model.

2) *DBN-Based Task Priority Calculation*: The architecture of the DBN is shown in Fig. 9. The DBN contains three parts: 1) the input layer; 2) the multiple hidden layers; and 3) the visible layer for outputs. The inputs are the solar power ( $\{P^s(m)\}$ ,  $m \in [t-|T|+1, t]$ ) during the last  $|T|$  time slots and the task status at the beginning of the  $t$ th time slot ( $\{I_i^{\text{rem}}(t)\}$ ). A vector  $X$  is used to store the inputs. The multiple hidden layers extract the features of the inputs by unsupervised learning on two types of networks: 1) restricted Boltzmann machine (RBM) and 2) directed belief network. The visible layer calculates the outputs by a BPN with  $|G|$  neurons. The outputs are the optimal task priorities ( $\{\lambda_i\}$ ).

The optimal task priorities ( $\{\lambda_i\}$ ) are obtained by online DBN calculation as follows. First, a joint probability distribution function is calculated

$$P(X, F^1, \dots, F^M, Y) = P(F^1, X) \dots P(Y|F^M) \quad (20)$$

where  $\{F^1, \dots, F^M\}$  denotes the sets of neurons and  $Y$  is the visible vector in  $M + 1$  hidden layers. The conditional probability functions in (20) are independent of each other and they are calculated as follows:

$$\begin{aligned} P(F^1, X) &= \left( 1 + \exp\left(-\sum_j (F^1)^T \cdot B^1 \cdot X\right) \right)^{-1} \\ P(Y|F^M) &= \prod_i P(y_i|F^M) \\ P(y_i = 1|F^M) &= \left( 1 + \exp\left(-\sum_j b_{i,j}^M \cdot f_j^M\right) \right)^{-1}, \\ & y_i \in Y, b_{i,j}^M \in B^M, f_j^M \in F^M \end{aligned} \quad (21)$$

**Algorithm 1:** Task Selection Method

---

**input** :  $\{P_i\}, P^s(t), \{\lambda_i\}$ ;  
**output**: online scheduling results  $\{\tilde{x}_i(t)\}$ .

- 1 Initialization: Set  $\{\tilde{x}_i(t)\}$  to zeros and set  $n = 1$ ;
- 2       Set all the NPEs in the available states.
- 3 Sort  $\{\lambda_i\}$  in a descending order.
- 4 **while**  $\sum_{k=1}^n \tilde{x}_k(t) \cdot P_k \leq P^s(t)$  &  $n \leq N$  **do**
- 5     Get  $\tau_n$  with the  $n$ th priority in  $\{\lambda_i\}$ .
- 6     **if**  $\tau_n$ 's corresponding NPE is available &
- 7        $over_n = 0$  &  $start_n = 1$  &  $miss_n = 0$  **then**
- 8         Set  $\tilde{x}_n(t) = 1$  and execute  $\tau_n$  on the NPE.
- 9     **else**
- 10        $\tau_n$  remains in the idle state.
- 11     **end**
- 12      $n = n + 1$ .
- 13 **end**
- 14 **return**  $\{\tilde{x}_i(t)\}$ .

---

where  $\{B^1, \dots, B^{M+1}\}$  are the weight coefficient matrix from one hidden layer to the next. The outputs are then obtained by calculating the BPN based on the visible vector ( $Y$ ).  $\{B^1, \dots, B^M\}$  are obtained as follows. We first calculate the optimal task priorities based on the INLP formulation (see Section IV) and (18) with the solar power classified in each type. The solar power ( $\{P^s(m)\}$ ,  $m \in [t - |T| + 1, t]$ ), the task status ( $\{I_i^{rem}(t)\}$ ) and the corresponding task priorities ( $\{\lambda_i\}$ ) are used as training samples. After that, the parameters of the DBN  $\{B^1, \dots, B^M\}$  are trained offline separately with the samples related to each type. In addition, the parameters in each RBM ( $B^{k+1}$ ,  $k \in [0, M]$ ) are trained iteratively as follows (see Fig. 9). Based on the initial  $B^{k+1}$  obtained randomly, the current  $F^{k+1}$  are generated by the original  $F^k$ . After that,  $F^k$  are reconstructed by the generated  $F^{k+1}$ . Thus, the corresponding  $B^{k+1}$  are modified based on the difference between the original and reconstructed  $F^k$ . After several iterations, the final  $B^{k+1}$  is obtained. The complexity of online DBN calculation is  $O(N_{DBN} \cdot (M + 1) + |G| + N)$ , where  $N_{DBN}$  is the number of the neurons in each hidden layer. The complexity is determined by the total number of the neurons in the model.

**D. Task Selection**

With the calculated task priorities ( $\{\lambda_i\}$ ), the task selection method is illustrated in Algorithm 1 and we illustrate it as follows. The inputs are the maximum power consumption of the tasks ( $\{P_i\}$ ), the current solar power  $P^s(t)$  and the calculated task priorities  $\{\lambda_i\}$ . The outputs are the scheduling results for the tasks in the  $t$ th time slot ( $\tilde{x}_i(t)$ ). In the algorithm, initialization is executed (lines 1 and 2) and the task priorities  $\{\lambda_i\}$  are sorted (line 3). Tasks are then chosen to be executed or not based on their priorities (lines 4–12). A task is executed (line 8), if the load power consumption is no more than the solar power supply (line 4) and the task is available to choose (lines 6 and 7); otherwise, it remains in the idle state (line 10). Finally, we get the scheduling results  $\{\tilde{x}_i(t)\}$  (line 14).

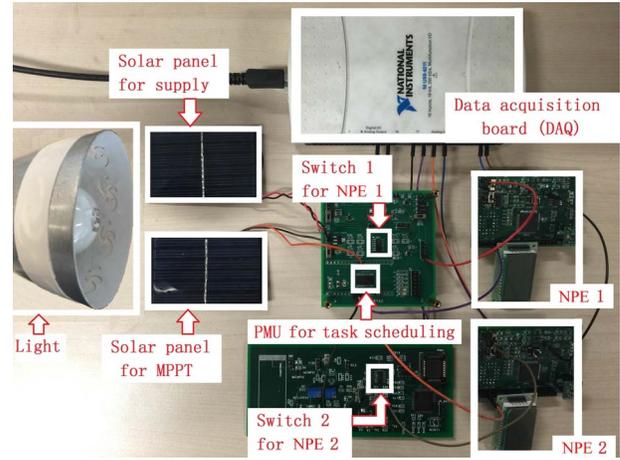


Fig. 10. Prototype of solar-powered nonvolatile sensor node.

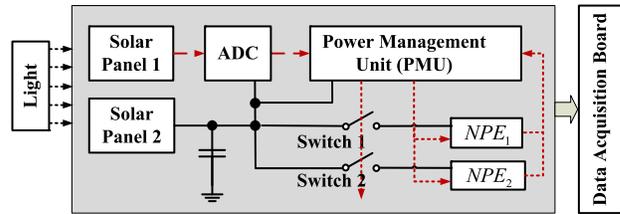


Fig. 11. Architecture diagram of solar-powered nonvolatile sensor node.

**VI. PROTOTYPE VALIDATION**

This section introduces the prototype of storage-less and converter-less solar-powered nonvolatile sensor node and verifies the system model on the prototype.

**A. Prototype Description**

The prototype and its block diagram are presented in Figs. 10 and 11. A light is used to simulate solar power, where solar variation is realized by adjusting intensity. Solar panel 1 monitors the real-time solar power by detecting the open-circuit voltage, while solar panel 2 harvests energy for workload. In practice, a much smaller solar panel can detect the open-circuit voltage by multiplying a certain coefficient. Both solar panels are 24.75 cm<sup>2</sup> and the average converting efficiency is 6%. The open-circuit voltage is digitalized by an analog-to-digital converter and given to the power management unit, which predicts the solar power profiles based on the scheduling algorithm. Four tasks are executed in two NPEs powered by the solar panel via a switch. Execution time and power consumption are measured by an oscilloscope and a data acquisition (DAQ) board. Though power management unit in this prototype is implemented by a microcontroller (MSP430), it can be replaced by a specific chip with negligible power consumption in real applications. As the wakeup/sleep energy consumption of FeRAM-based NPEs are 2–3 orders of magnitude lower than traditional Flash-based processors, switching overheads are quite small (less than 0.7% [12]).

**B. System Model Validation**

We validate the system model on the prototype under different kinds of solar power profiles. Table II compares the results

TABLE II  
COMPARISON OF SYSTEM MODELING AND HARDWARE MEASUREMENT

Solar Power	Execution Time (s)			Energy Consumption (J)			DMR (%)		
	Model	Measure	Diff.	Model	Measure	Diff.	Model	Measure	Diff.
Solar1	33.4	33.2	0.60%	3.73	3.82	2.41%	25	25	0%
Solar2	25.1	23.5	6.37%	2.52	2.67	5.95%	50	50	0%
Solar3	23.1	22.9	0.87%	2.83	2.88	1.76%	50	50	0%
Solar4	21.6	21.1	2.31%	2.13	2.25	5.63%	75	75	0%

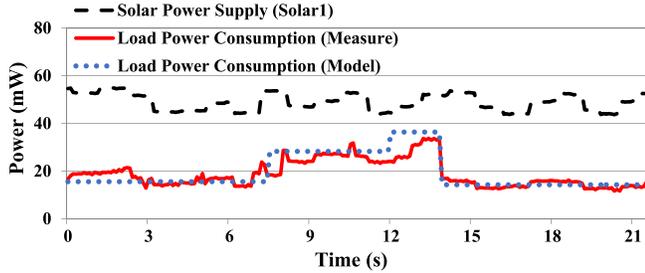


Fig. 12. Detailed comparison of power consumption under Solar1.

both from models and measured results. The total execution time of the models are larger while the energy consumption are smaller than the measured ones. It is because the modeling results are calculated in the ideal situation, where the power fluctuations of workload is not considered. However, DMRs of them are matched well, as small power fluctuations do not affect the scheduling results. The maximum difference (“Diff.”) is 6.37%, which validate that the system model is reasonable for online intra-task scheduling. Finally, the power consumption of workload under modeling and measured results are given in Fig. 12.

## VII. EXPERIMENTAL EVALUATION

As the prototype in Section VI provides us insights to the modeling approximations in simulators, this section investigates intra-task scheduling via simulation-based exploration. It first explains experimental configurations and verifies the theorem of optimal DMR estimation. After that, the proposed intra-task scheduling is evaluated in different benchmarks. Finally, algorithm sensitivity is discussed.

### A. Experimental Setup

In the experiment, seven benchmarks are used. Three of them are synthetic benchmarks (from R1 to R3). The number of tasks ranges from 4 to 8 and the number of NPEs ranges from 2 to 6. In addition, Fig. 13 presents four real benchmarks: 1) wild animal monitoring (WAM); 2) SHM; 3) electrocardiogram (ECG) applications; and 4) media processing (media). The execution time and average power consumption of each task are obtained by running the benchmarks on the NPEs. Given a real solar database [20], power profiles in five individual days (from Day1 to Day5, see Fig. 14) are selected for daily test (short term) and power profiles in two months are used for monthly test (long term). In addition, a period is divided into 60 time slots ( $|T|$ ) and each time slot lasts for 60 s ( $\Delta t$ ). The trigger thresholds of solar variation quantity

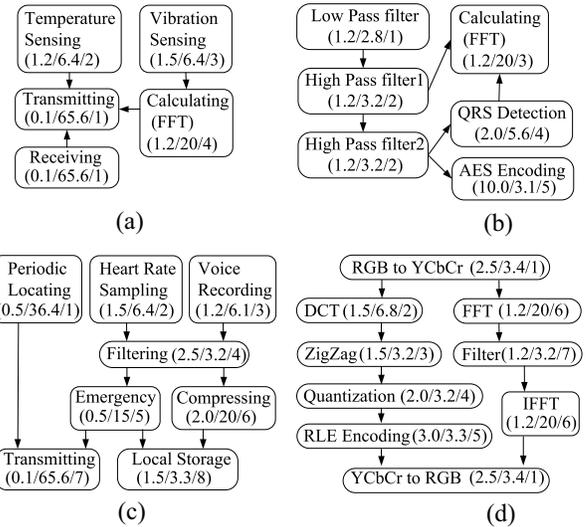


Fig. 13. Configurations of four real benchmarks [execution time (min)/power (mW)/NPE index]. (a) SHM. (b) Electrocardiograph. (c) WAM. (d) Media application.

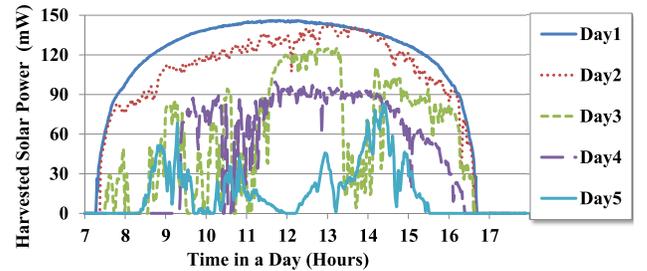


Fig. 14. Harvested solar power (7:00–17:00) in the five individual days.

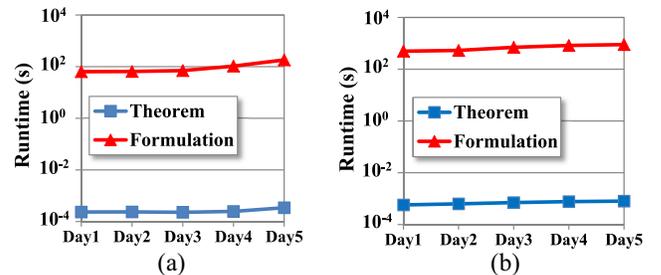


Fig. 15. Runtime of the theorem and the INLP formulation. (a) R1 case. (b) ECG case.

and time ( $\delta_1$  and  $\delta_2$ ) are 5 mW and 240 s, respectively. We classify solar profiles into eight types. The BPN has one layer with 30 neurons ( $|g|$ ). The DBN has five hidden layers ( $|f|$ ) and each layer contains 40 neurons ( $N_{DBN}$ ).

### B. Theorem Validation

We first validate Theorem 1 on R1 and ECG benchmarks in five individual days. Table III compares DMRs estimated by the theorem and the optimal values obtained from INLP formulation. Since no task dependence exists in R1 case, the theorem can obtain the exact optimal DMRs. However, since

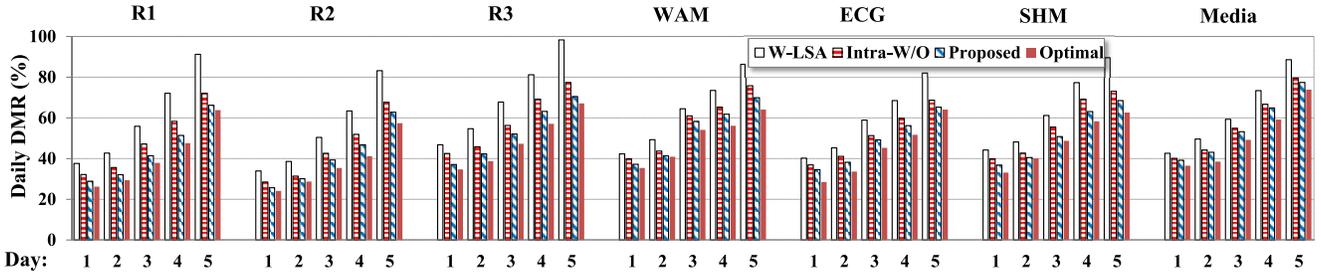


Fig. 16. Comparison of DMR in five individual days.

TABLE III  
DMR OF THEOREM AND INLP FORMULATION

Solar Power	R1 Case Without Task Dependence			ECG Case With Task Dependence		
	Theorem	Formulation	Difference	Theorem	Formulation	Difference
Day1	26.2	26.2	0%	26.4	28.5	7.37%
Day2	29.4	29.4	0%	30.3	33.6	9.82%
Day3	37.9	37.9	0%	40.5	45.3	10.6%
Day4	47.6	47.6	0%	45.2	51.7	12.6%
Day5	63.8	63.8	0%	58.2	64.1	9.20%

there are several task dependences in ECG benchmark, DMR obtained by the theorem is smaller than the optimal value. It is because theoretical DMRs are obtained based on the start time of tasks estimated by (8), which are smaller than actual values. However, the average difference between them is 9.91%, which provides a reasonable lower bound for DMR estimation. Furthermore, Fig. 15 compares the runtime of two approaches. Compared with the INLP approach, the theorem is  $10^5 \times$  much faster, which is less than 0.01 s. Therefore, the theorem is effective for fast optimal DMR estimation, especially for scheduling problems in large scales.

C. Inter-task Versus Intra-task Scheduling Under Different Benchmarks

We compare our proposed algorithm with traditional inter-task scheduling (W-LSA [7]) and intra-task scheduling without classified solar power prediction [24] (intra-W/O). In addition, optimal DMRs obtained from INLP formulation are used as lower bounds. DMRs and energy utilization efficiency of these algorithms are compared under both short- and long-term conditions.

Fig. 16 compares different algorithms on seven benchmarks under short-term condition. Compared with [7], the proposed algorithm reduces DMR by up to 28.8%, where the reduction becomes more significant when solar power becomes more limited. The proposed algorithm reduces DMR by up to 7.90% compared to intra-task scheduling [24], as classified solar power prediction generates more accurate task priority. Moreover, the difference between the proposed algorithm and the optimal INLP is only 3.65%.

Table IV shows the improvements of total and usable energy utilization efficiency compared to inter-task scheduling [7]. The total energy utilization efficiency equals to the ratio between the energy to execute all tasks and the harvested energy, while the usable energy utilization efficiency denotes

TABLE IV  
IMPROVEMENTS OF ENERGY UTILIZATION EFFICIENCY (%) IN FIVE INDIVIDUAL DAYS

Benchmark		Day1	Day2	Day3	Day4	Day5	Average
R1	Total	8.32	11.6	14.7	17.9	19.2	14.3
	Usable	14.8	17.7	22.6	24.2	27.1	21.3
R2	Total	7.71	10.6	12.3	15.5	17.1	12.6
	Usable	12.3	14.8	18.2	21.5	21.7	17.7
R3	Total	9.23	12.7	14.5	16.9	19.8	14.6
	Usable	14.9	18.9	22.6	23.4	25.2	21.0
WAM	Total	7.97	11.6	16.7	20.5	24.4	16.2
	Usable	13.3	16.7	22.6	25.2	27.1	21.0
ECG	Total	11.4	14.7	16.1	20.0	21.2	16.7
	Usable	14.4	18.6	19.3	25.2	29.5	21.4
SHM	Total	14.8	16.7	18.6	21.4	22.2	18.7
	Usable	15.1	18.2	19.5	24.9	26.1	20.8
Media	Total	12.2	14.1	16.7	18.9	20.1	16.4
	Usable	15.6	17.1	18.6	21.2	23.0	19.1

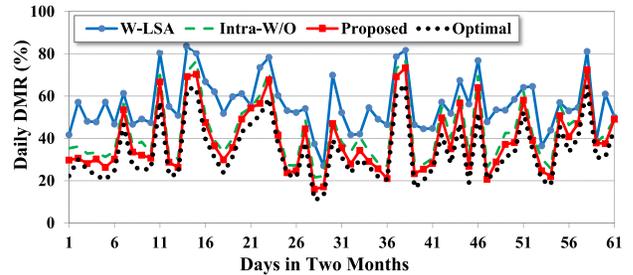


Fig. 17. Comparison of DMR in two months.

the ratio between the energy to execute the tasks meeting their deadline and the harvested energy. The usable energy utilization efficiency reflects the ability to execute tasks satisfying quality of service. As we can see, the usable energy utilization efficiency is larger (up to 8.30%) than the total energy utilization, which means that the inter-task algorithm [7] tends to execute tasks missing their deadline and wastes harvested energy. Moreover, the proposed algorithm performs better (up to 29.5%) when the solar power budget becomes tighter.

In long-term cases, we compare DMR and energy utilization efficiency in two months (from September to October) for R1 case. Compared with [7] and [24], the proposed algorithm reduces DMR by up to 29.4% and 6.32% shown in Fig. 17. Fig. 18(a) and (b) presents the total and usable energy utilization efficiency under different approaches. The average differences are 11.6% and 20.3%.

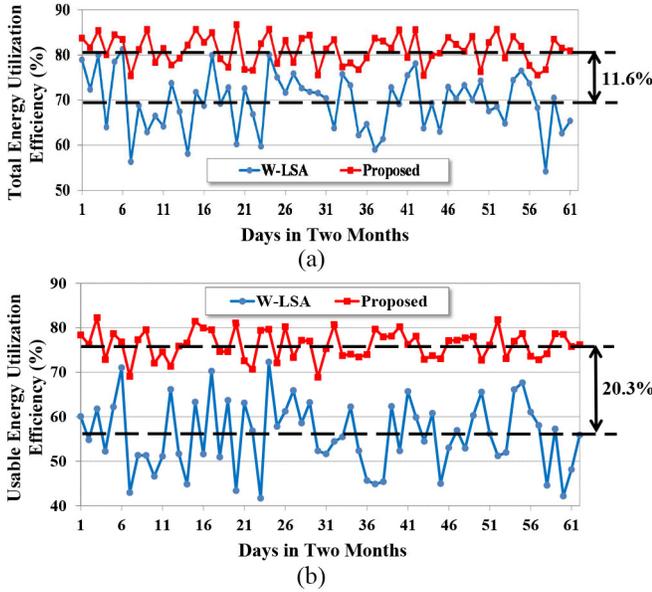


Fig. 18. Comparison of energy utilization efficiency in two months. (a) Total energy utilization efficiencies of two scheduling algorithms. (b) Usable energy utilization efficiencies of two scheduling algorithms.

TABLE V  
COMPLEXITY COMPARISON AMONG DIFFERENT ALGORITHMS

	Inter-task	Intra-task Without Classification	This Work
Complexity per Execution	$O(N)$	$O(N +  g )$	$O(N_{DBN} \cdot (M + 1) + 2 \cdot  g  + N + 1)$
Cycles per Execution	14,100	387,000	1,155,000
Number of Scheduling Points in A Day	52	158	109
Proportion of Energy Consumption in A Day	0.005%	0.354%	0.889%

Finally, we compare algorithm complexities. The MSP430 runs scheduling algorithms at 1 MHz with an oscilloscope and DAQ board to get the execution cycles and energy consumption. In Table V, the proposed algorithm increases both scheduling points and the algorithm complexity compared with previous ones, however, energy overheads are less than 1%.

#### D. Sensitivity Analysis of Algorithm

In this section, we analyze algorithm sensitivity under four factors: 1) the number of trigger points; 2) the type number of solar profiles; 3) the parameters of neural network; and 4) the task features.

1) *Sensitivity Analysis of Trigger Points*: Fig. 19 presents the relationship between DMR and the number of trigger points for R2 benchmark in Day4. Situations with and without solar variation time threshold are considered. As more trigger points are used, DMR becomes lower, which means more fine-grained scheduling is adopted. However, DMR becomes stable when the number of trigger points reaches a certain value, since small power variations has less impact on DMR.

2) *Sensitivity Analysis of Solar Power Type*: Fig. 20 presents the relationship between DMR and the type number of solar profiles for R1. DMR decreases from 39.6% to

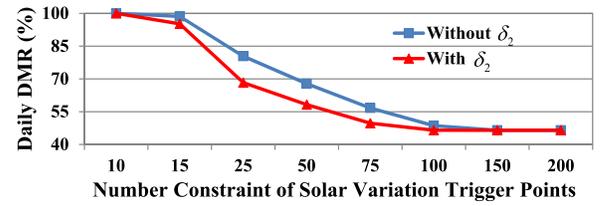


Fig. 19. Sensitivity analysis of solar variation trigger points.

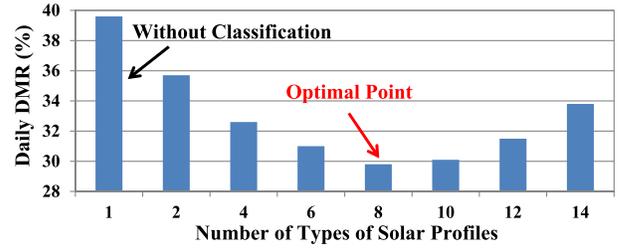


Fig. 20. Sensitivity analysis of types of solar profiles.

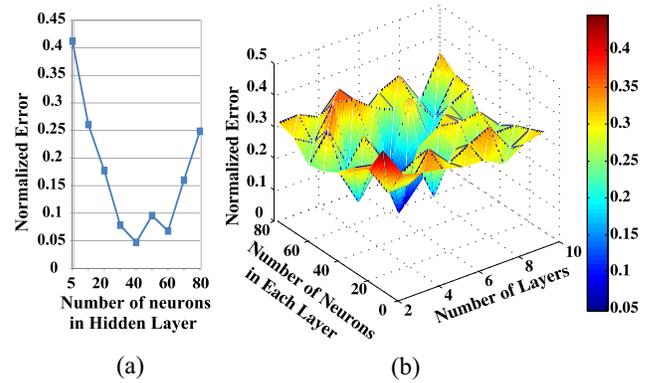


Fig. 21. Architecture optimization for the BP and DBN. (a) Architecture optimization for the BP model. (b) Architecture optimization for the DBN model.

29.8% when the type number becomes larger (from 1 to 8), as more types make the solar power prediction more accurate. However, there is a breakeven value for the type number. When the type number is larger than a certain threshold (eight types in this case), DMR becomes even worse (from 29.8% to 33.8%). This can be analyzed as follows. First, as the type number becomes larger, the training samples in each type is smaller given a certain amount of solar data. Thus, the prediction accuracy degrades in an ANN model (e.g., BPN). Even if we can increase training samples, the ANN model with a fixed scale may lead to over-fitting and also decrease the prediction accuracy. It is because the ANN model with the fixed scale is not large enough to describe the complex relationship of the increased training samples. Thus, the small-scale model with too many samples causes over-fitting, which hurts the prediction accuracy badly.

3) *Sensitivity Analysis of Neuron Network*: We explore the ANN model from two aspects: the architecture and the inputs of solar data. To optimize the architecture, we investigate the number of layers and neurons in Fig. 21. The minimum normalized error (4.70%) is achieved when BPN has 40 neurons. The minimum daily error (9.36%) is achieved when DBN has

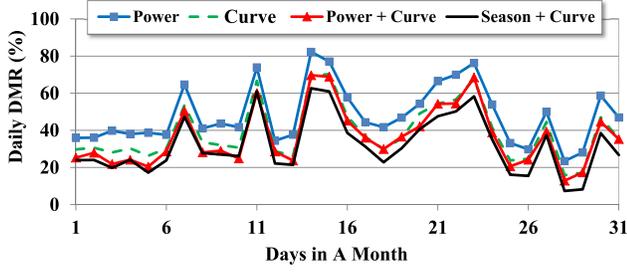


Fig. 22. Inputs optimization for the DBN.

five layers and 50 neurons in each layer. More layers and neurons may cause under-fitting while less of them may cause over-fitting.

In addition, we can achieve better prediction accuracy by training the DBN with better inputs of solar data. Fig. 22 compares DMRs under different solar data in a month. The original data is historical solar power. The statistical data represents the mean value and variation of historical solar power in each period. The season data denotes the season information corresponding to historical solar power in each day. Compared with the original data, higher distinguishability of solar data means better prediction, which contributes to DMR reduction (up to 21.5%).

4) *Sensitivity Analysis of Task Features*: DMR is also affected by several task features, such as power consumption and task dependence. We define a metric ( $PR_{avg}$ ) to describe the power requirement of a task set as follows:

$$PR_{avg} = \sum_{i=1}^N P_i \cdot L_i / (D_i - R_i) \quad (22)$$

where  $R_i$  is the start time of  $\tau_i$ .  $PR_{avg}$  is equal to the average power requirement of all tasks in a period. The task dependence rate (TDR) is defined as follows:

$$TDR = \left( \sum_{i=1}^N \sum_{j=1}^N E_{i,j} \right) / N \quad (23)$$

where TDR is the ratio of edge number to task number. Larger TDR means more connections among tasks.

Fig. 23(a) presents the relationship between DMR and power requirement. Compared with inter-task scheduling algorithm (W-LSA, [7]), the proposed algorithm achieves larger DMR reductions (from 3.5% to 8.7%) when  $PR_{avg}$  becomes smaller (from 136.2 to 63 mW). It is because smaller  $PR_{avg}$  means looser power requirement and larger scheduling space for the scheduler. In addition, Fig. 23(b) presents the relationship between DMR and TDR.

## VIII. RELATED WORK

This section introduces the related work of energy-driven task scheduling, which contains the scheduling strategies and the solar power prediction methods.

### A. Offline Scheduling Algorithms

There are lots of offline scheduling algorithms developed for embedded systems under certain constraints [25], [26].

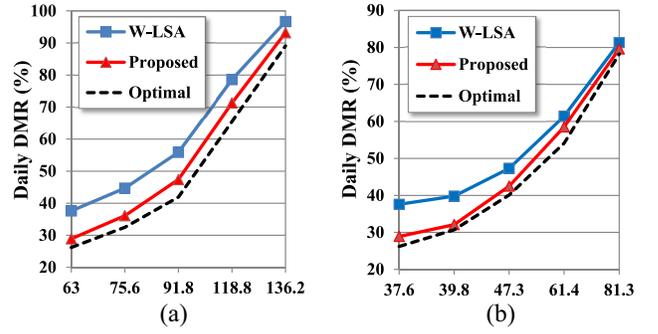


Fig. 23. Sensitivity analysis of energy requirement and task dependence. (a) Power requirement (mW). (b) TDR (%).

For energy harvesting systems, the offline scheduling algorithms make use of the historical energy profiles and task timing parameters. Kansal *et al.* [18] proposed a scheduling method to tune duty-cycling based on historical solar power for periodic tasks. Audet *et al.* [27] developed a static scheduler to assign execution time to the tasks, whose priorities are higher if they consume more energy. Though these scheduling algorithms perform well in the offline situations, they are less effective as the online power profiles are usually inconsistent with the offline ones.

### B. Online Scheduling Algorithms

To match the online workload variations, previous work has proposed many online algorithms for real-time systems [28]. Solar-powered sensor nodes need specific scheduling algorithms to match the power supply and workload. Lazy scheduling algorithms (LSAs) [29], [30] are adopted to determine the start time of each task for maximal energy utilization. Recas *et al.* [7] improved the LSA by adding predictions for real-time solar variations. Furthermore, task decomposition and combination were proposed by Zhu *et al.* [31] to realize more accurate and flexible load matching. To utilize task slacks, dynamic voltage and frequency scaling (DVFS) was integrated into load matching algorithms [32], [33] to achieve better DMR. Researchers further improved this approach by considering super capacitor charging efficiency [8] and multicore systems [34]. Moreover, Wang *et al.* [10] proposed an integer linear programming (ILP) formulation for the online DVFS on a multicore system. Although these algorithms are effective for online scheduling, the energy loss caused by energy storage and dc-dc convertor cannot be avoided in the traditional harvest-store-use architecture. Recently, Wang *et al.* [12], [13] developed an energy-efficient SCSN, which achieves up to 23% higher energy efficiency than the traditional architecture. Developing appropriate scheduling algorithms for such an architecture is still an open problem, since it is more sensitive to the power variations due to lacking of energy buffer.

### C. Solar Power Prediction Methods for Scheduling

Besides scheduling algorithms, the solar power prediction methods play another important role, as they have great effects on the scheduling results. In EWMA method [18],

harvested energy at a particular time is calculated as weighted average energy at the same time over a set of previous days. Later, researchers developed a prediction method of WCMA [7], [19], which adapts to seasonal changes as well as sudden weather changes. Furthermore, more advanced models, such as support vector machine [35], back propagation network (BPN) [36] and recurrent neural networks (RNN) [37], were proposed based on wind speeds, temperatures, dew points, sky cover, precipitation potential, etc. Though they are good for the average case, high DMR in solar-powered sensor nodes still occurs when solar profiles are quite different from the average case. Therefore, this paper developed a solar power prediction method based on the classified solar profiles.

## IX. CONCLUSION

Solar-powered nonvolatile sensor nodes without energy storage and dc voltage conversion achieve high energy efficiency. This paper proposes intra-task scheduling with flexible and timely task adjustment. We first build a system model, analyze the schedulability with optimal DMR estimation in theory and develop an INLP formulation for optimal scheduling. An online intra-task scheduling algorithm is then proposed. We validate the proposed algorithm with a hardware prototype and analyze it with several benchmarks. The experimental results show that the algorithm reduces the DMR by up to 30% and improves the energy utilization efficiency by 20% on average. The algorithm incurs less than 1% overhead of the total energy consumption. Furthermore, future studies will explore the scheduling algorithms on the sensor nodes with other (or hybrid) energy sources such as thermal, wireless, and vibration.

## REFERENCES

- [1] W. Liu *et al.*, "Application specific sensor node architecture optimization—Experiences from field deployments," in *Proc. Asia South Pac. Design Autom. Conf. (ASPDAC)*, Sydney, NSW, Australia, 2012, pp. 389–394.
- [2] Y.-C. Kan and C.-K. Chen, "A wearable inertial sensor node for body motion analysis," *IEEE Sensors J.*, vol. 12, no. 3, pp. 651–657, Mar. 2012.
- [3] J. Taneja, J. Jeong, and D. Culler, "Design, modeling, and capacity planning for micro-solar power sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, St. Louis, MO, USA, 2008, pp. 407–418.
- [4] M. Paselli *et al.*, "A high-performance wireless sensor node for industrial control applications," in *Proc. Int. Conf. Syst. (ICONS)*, Cancún, Mexico, 2008, pp. 235–240.
- [5] J. Bryzek, "Roadmap for the trillion sensor universe (fairchild semiconductor)," in *Proc. iNEMI Spring Member Meeting Webinar*, Berkeley, CA, USA, 2013, pp. 443–461.
- [6] R. J. M. Vullers, R. van Schaijk, I. Doms, C. V. Hoof, and R. Mertens, "Micropower energy harvesting," *Solid-State Electron.*, vol. 53, no. 7, 2009, pp. 684–693.
- [7] J. R. Piorno, C. Bergonzini, D. A. Atienza, and T. S. Rosing, "HOLLOWS: A power-aware task scheduler for energy harvesting sensor nodes," *J. Intell. Mater. Syst. Struct.*, vol. 21, no. 12, pp. 1317–1335, 2010.
- [8] 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 *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Beijing, China, 2013, pp. 70–75.
- [9] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 3, pp. 443–461, Jul. 2011.
- [10] Y. Wang, R. Chen, Z. Shao, and T. Li, "SolarTune: Real-time scheduling with load tuning for solar energy powered multicore systems," in *Proc. IEEE Int. Conf. Embedded Real-Time Comput. Syst. Appl. (RTCSA)*, Taipei, Taiwan, 2013, pp. 101–110.
- [11] Y. Choi, N. Chang, and T. Kim, "DC–DC converter-aware power management for low-power embedded systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 8, pp. 1367–1381, Aug. 2007.
- [12] C. Wang *et al.*, "Storage-less and converter-less maximum power point tracking of photovoltaic cells for a nonvolatile microprocessor," in *Proc. Asia South Pac. Design Autom. Conf. (ASPDAC)*, Singapore, 2014, pp. 379–384.
- [13] Y. Wang *et al.*, "Storage-less and converter-less photovoltaic energy harvesting with maximum power point tracking for Internet of things," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 2, pp. 173–186, Feb. 2016.
- [14] Y. Wang *et al.*, "PaCC: A parallel compare and compress codec for area reduction in nonvolatile processors," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 7, pp. 1491–1505, Jul. 2014.
- [15] M.-F. Chang *et al.*, "A high-speed 7.2-ns read-write random access 4-mb embedded resistive ram (ReRAM) macro using process-variation-tolerant current-mode read schemes," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 878–891, Mar. 2013.
- [16] M.-F. Chang *et al.*, "A low-voltage bulk-drain-driven read scheme for sub-0.5v 4Mb 65nm logic-process compatible embedded resistive RAM (ReRAM) macro," *IEEE J. Solid-State Circuits*, vol. 48, no. 9, pp. 2250–2259, Sep. 2013.
- [17] M.-F. Chang *et al.*, "Challenges and circuit techniques for energy-efficient on-chip nonvolatile memory using memristive devices," *IEEE J. Emerg. Sel. Topic Circuits Syst.*, vol. 5, no. 2, pp. 183–193, Jun. 2015.
- [18] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, pp. 1–35, Sep. 2007.
- [19] J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," in *Proc. Int. Conf. Wireless Commun. Veh. Technol. Inf. Theory Aerosp. Electron. Syst. Technol. (VITAE)*, Aalborg, Denmark, pp. 6–10, 2009.
- [20] Daily Plots and Raw Data Files. (Jan. 1, 2012). *Measurement and Instrumentation Data Center (MIDC)*. [Online]. Available: <http://www.nrel.gov/midc/>
- [21] J. W. Layland and C. L. Liu, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.
- [22] I. Ripoll, A. Crespo, and A. K. Mok, "Improvement in feasibility testing for real-time tasks," *Real Time Syst.*, vol. 11, no. 1, pp. 19–39, Jul. 1996.
- [23] K. Jeffay, D. F. Stanat, and C. U. Martel, "On non-preemptive scheduling of period and sporadic tasks," in *Proc. Real Time Syst. Symp.*, San Antonio, TX, USA, 1991, pp. 129–139.
- [24] D. Zhang *et al.*, "Intra-task scheduling for storage-less and converter-less solar-powered nonvolatile sensor nodes," in *Proc. IEEE Int. Conf. Comput. Design (ICCD)*, Seoul, Korea, 2014, pp. 149–152.
- [25] H. Salamy and J. Ramanujam, "An effective solution to task scheduling and memory partitioning for multiprocessor system-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 31, no. 5, pp. 717–725, May 2012.
- [26] Y. Wang, Z. Shao, H. C. B. Chan, D. Liu, and Y. Guan, "Memory-aware task scheduling with communication overhead minimization for streaming applications on bus-based multiprocessor system-on-chips," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1797–1807, Jul. 2014.
- [27] D. Audet, L. C. de Oliveira, N. MacMillan, D. Marinakis, and K. Wu, "Scheduling recurring tasks in energy harvesting sensors," in *Proc. IEEE Commun. Commun. Workshops (INFOCOM WKSHPS)*, Shanghai, China, 2011, pp. 277–282.
- [28] K. Kanoun, N. Mastrorade, D. Atienza, and M. van der Schaar, "Online energy-efficient task-graph scheduling for multicore platforms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 8, pp. 1194–1207, Aug. 2014.
- [29] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Lazy scheduling for energy harvesting sensor nodes," in *Proc. Work. Conf. Distrib. Parallel Embedded Syst. (DIPES)*, Braga, Portugal, 2006, pp. 125–134.
- [30] C. Moser, J.-J. Chen, and L. Thiele, "Power management in energy harvesting embedded systems with discrete service levels," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, San Francisco, CA, USA, 2009, pp. 413–418.
- [31] T. Zhu, A. Mohaisen, P. Yi, and D. Towsley, "DEOS: Dynamic energy-oriented scheduling for sustainable wireless sensor networks," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 2363–2371.

- [32] S. Liu, Q. Qiu, and Q. Wu, "Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting," in *Proc. Conf. Design Autom. Test Europe (DATE)*, Munich, Germany, 2008, pp. 236–241.
- [33] S. Liu, J. Lu, Q. Wu, and Q. Qiu, "Load-matching adaptive task scheduling for energy efficiency in energy harvesting real-time embedded systems," in *Proc. ACM/IEEE Int. Symp. Low Power Electron. Design (ISLPED)*, Austin, TX, USA, 2010, pp. 325–330.
- [34] Y. Xiang and S. Pasricha, "Fault-aware application scheduling in low-power embedded systems with energy harvesting," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES)*, New Delhi, India, 2014, pp. 1–10.
- [35] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Brussels, Belgium, 2011, pp. 528–533.
- [36] C. Renner, "Solar harvest prediction supported by cloud cover forecasts," in *Proc. Int. Workshop Energy Neutral Sens. Syst. (ENSSys)*, Rome, Italy, 2013, pp. 1–6.
- [37] N. Zhang, P. K. Behera, and C. Williams, "Solar radiation prediction based on particle swarm optimization and evolutionary algorithm using recurrent neural networks," in *Proc. Int. Syst. Conf. (SysCon)*, Orlando, FL, USA, 2013, pp. 280–286.



**Daming Zhang** received the B.S. and Ph.D. degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2010 and 2016, respectively.

He is currently an Engineer with the China Academy of Launch Vehicle Technology, Beijing. He had researched in the project of a self-powered wireless sensor node sensor platform design. His current research interests include architecture optimization and task scheduling of embedded systems.



**Yongpan Liu** (M'07–SM'15) received the B.S., M.S., and Ph.D. degrees from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 1999, 2002, and 2007, respectively.

He was a Visiting Scholar with Pennsylvania State University, University Park, PA, USA, in 2014. He is a Key Member of the Tsinghua—Rohm Research Center, Beijing, China, and the Research Center of Future ICs, Beijing. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. His current research interests include nonvolatile computation, low power very large-scale integration (VLSI) design, emerging circuits, and systems and design automation. He has published over 60 peer-reviewed conference and journal papers and led over six chip design projects for sensing applications, including the first nonvolatile processor (THU1010N).

Prof. Liu was a recipient of the Design Contest Awards from ISLPED 2012 and 2013, and the Best Paper Award from High-Performance Computer Architecture 2015. He served on several conference technical program committees, such as Design Automation Conference, Asia and South Pacific Design Automation Conference, ISLPED, Asian Solid-State Circuits Conference, International Conference on Computer Design, and VLSI-D. He is an ACM and an IEICE Member.



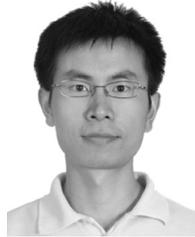
**Jinyang Li** received the B.S. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2015, where he is currently pursuing the master's degree with the Department of Electronic Engineering.

He is currently researching the project of personal ultraviolet monitoring system design with wearable sensors. His current research interests include nonvolatile processor applications and wearable devices in wireless healthcare field.



**Chun Jason Xue** (M'07) received the B.S. degree in computer science and engineering from the University of Texas at Arlington, Arlington, TX, USA, in 1997, and the M.S. and Ph.D. degrees in computer science from the University of Texas at Dallas, Richardson, TX, USA, in 2002 and 2007, respectively.

He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong. His current research interests include nonvolatile memories and embedded systems.



**Xueqing Li** (M'14) received the B.S. and Ph.D. degrees in electronics engineering from Tsinghua University, Beijing, China, in 2007 and 2013, respectively.

Since 2013, he has been a Post-Doctoral Researcher with the Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA. His current research interests include wideband data converters, nonvolatile processors, memory architectures, energy-harvesting systems, and other circuit designs

in CMOS and emerging devices, such as III-V heterojunction tunnel FET, negative capacitance ferroelectric FET, SymFET, Hyper-FET, VO<sub>2</sub>-based oscillators, and single-electron transistors.



**Yu Wang** (S'05–M'07–SM'14) received the B.S. and Ph.D. degrees (Hons.) from Tsinghua University, Beijing, China, in 2002 and 2007, respectively.

He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. He has authored and co-authored over 140 papers in refereed journals and conferences. His current research interests include parallel circuit analysis, application specific hardware computing (especially on the brain related problems), and power/reliability aware system design methodology.

Dr. Wang was a recipient of the IBM X10 Faculty Award in 2010, the Best Paper Award in IEEE Computer Society Annual Symposium on VLSI (ISVLSI) 2012, the Best Poster Award in International Workshop on Highly-Efficient Accelerators and Reconfigurable Technologies 2012, and six Best Paper Nominations in Asia and South Pacific Design Automation Conference (ASP-DAC)/CODES/ISLPED. He serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS and the *Journal of Circuits, Systems, and Computers*. He is the TPC Co-Chair of International Conference on Field-Programmable Technology (ICFPT) 2011, the Finance Chair of ISLPED 2012–2015, and serves as a TPC Member in several important conferences, such as Design Automation Conference, FPGA, DATE, ASP-DAC, ISLPED, ISQED, ICFPT, ISVLSI.



**Huazhong Yang** (SM'13) received the B.S., M.S., and Ph.D. degrees from Tsinghua University, Beijing, China, in 1989, 1993, and 1998, respectively.

He joined the Department of Electronic Engineering, Tsinghua University, in 1993, where he has been a Full Professor since 1998. He is a specially-appointed Professor of the Cheung Kong Scholars Program. He has authored and co-authored over 300 technical papers and 70 granted patents.

His current research interests include wireless sensor networks, data converters, parallel circuit simulation algorithms, nonvolatile processors, and energy-harvesting circuits.