

PAPER

Lifetime-Aware Battery Allocation for Wireless Sensor Network under Cost Constraints*

Yongpan LIU^{†a)}, *Member*, Yiqun WANG[†], Hengyu LONG[†], and Huazhong YANG[†], *Nonmembers*

SUMMARY Battery-powered wireless sensor networks are prone to premature failures because some nodes deplete their batteries more rapidly than others due to workload variations, the many-to-one traffic pattern, and heterogeneous hardware. Most previous sensor network lifetime enhancement techniques focused on balancing the power distribution, assuming the usage of the identical battery. This paper proposes a novel fine-grained cost-constrained lifetime-aware battery allocation solution for sensor networks with arbitrary topologies and heterogeneous power distributions. Based on an energy-cost battery pack model and optimal node partitioning algorithm, a rapid battery pack selection heuristic is developed and its deviation from optimality is quantified. Furthermore, we investigate the impacts of the power variations on the lifetime extension by battery allocation. We prove a theorem to show that power variations of nodes are more likely to reduce the lifetime than to increase it. Experimental results indicate that the proposed technique achieves network lifetime improvements ranging from 4–13× over the uniform battery allocation, with no more than 10 battery pack levels and 2-5 orders of magnitudes speedup compared with a standard integer nonlinear program solver (INLP).

key words: wireless sensor network, battery allocation, lifetime-aware

1. Introduction

Wireless Sensor Networks (WSN) are distributed data acquisition systems consisting of numerous wireless sensor nodes. They have the potential to allow sensing in applications and environments where it was previously impossible or prohibitively expensive. For example, WSNs may be used in weather monitoring, security, tactical surveillance, disaster management, and intelligent traffic control applications [2]. Infrastructure-free operation is one of their primary advantages. However, this beneficial attribute introduces a penalty. Distributed infrastructure-free operations in the remote locations make replacing batteries expensive. Energy constraints are therefore extremely tight.

Due to the limited energy capacity enforced by the low cost requirement of WSNs, the lifetime of a WSN to execute continuous monitoring tasks is critical. It is desirable that all nodes in WSN should cooperate with each other to sense and transmit information and run out of energy together. Otherwise, nodes in some areas will be unable to transmit their sensing data to the collective node because other battery depleted nodes will break the transmission routes to those areas. However, the many-to-one traffic pattern in WSN naturally leads to an imbalance power distribution. Previous

work had shown that such an imbalance will seriously shorten the lifetime of a WSN.

Recently, research work attempted to balance the energy consumption of the network by constructing the lifetime-aware WSN with heterogeneous nodes, since moving tasks among the nodes to balance power is not energy efficient. By providing the nodes in the second tier stronger processing ability and larger battery capacity, Hou et al. [3] presented a two-tier lifetime-aware infrastructure. Wu et al. [4] presented a non-uniform node deployment to exploit more relay nodes to deal with the power peak in the traffic-heavy area to prolong the lifetime. Several other researchers [5], [6] illustrated a mobile sink or multiple sinks approach to adjust the traffic flow and thus the energy distribution. Others proposed a battery allocation technique [1], [7] to equip power-hungry sensor nodes with different battery packs to relieve the imbalance. Among the above lifetime-aware techniques, the heterogeneous battery allocation received more and more attentions due to its less overheads on the original WSN.

Sichitiu et al. [7] were the first to report that equipping sensor nodes with different battery capacities can prolong the WSN lifespan. However, their formulation has two major drawbacks: First, their battery allocation object is the coarse-grained network tier instead of the fine-grained node, i.e. the sensor nodes in each tier are equipped with the same battery and it can not deal with the power imbalance within each network tier; Second, they assumed a monolithic power distribution in a circular WSN from the leaf nodes to the sink node, i.e. it is inapplicable to the WSNs with multiple distributed power peaks, which is common in the WSNs that adopt clustering or multiple sinks or other energy efficient routing techniques; Those limitations prevent the method to be used in the real WSNs. Instead of those drawbacks, this paper proposed a novel fine-grained node-level battery allocation for the battery-powered networked embedded systems. Our work makes the following contributions:

1. We presented a novel fine-grained node-level battery allocation method under cost constraints. It is formulated as an integer nonlinear programming problem and a quite efficient heuristic is built based on the proved optimal node partitioning theorem and the energy-cost model for the battery packs.
2. We discussed the impacts of power variations on the lifetime extensions using the battery allocation method. Theoretical analysis shows that the power variations

Manuscript received October 19, 2010.

Manuscript revised October 25, 2011.

[†]The authors are with the Electronic Engineering Department, Tsinghua University, 100084, Beijing, P.R. China.

*This paper was presented at ICCAD2009 [1].

a) E-mail: ypliu@tsinghua.edu.cn

DOI: 10.1587/transcom.E95.B.1651

of the nodes are more likely to reduce the lifetime of WSNs than to increase it. It implies that a larger pack level over a threshold is no need considering the power variations.

- Experiments indicate that the proposed method can prolong the lifetime by 4–13× with no more than 10 battery pack levels compared with the uniform battery allocation approach. Furthermore, the heuristic achieves 2–5 orders of magnitudes speedup and longer time improvements compared with a standard integer nonlinear program (INLP) solver.

After we discuss the motivation of this work in the following Sect. 2, the node-level battery allocation problem is formulated in Sect. 3. The battery allocation problem is solved in Sect. 4 and the power variations' impacts on it are analyzed in Sect. 5. Section 6 presents the experimental results. We conclude the challenges and future work in Sect. 7.

2. Motivation

This section describes the motivation of the lifetime-aware battery allocation method for WSNs with various topologies, node configurations, and power distributions.

First of all, we define the lifetime-aware design flow for WSNs. Figure 1 shows the lifetime-aware flow to extend the working time of WSN to meet the requirements. As we can see, the major difference between the traditional performance-driven flow and the lifetime-aware one, is the design objective changing to the lifetime under certain constraints, such as costs and performance. The power analysis stage will estimate the power consumption of each node based on the initial WSN deployment and network protocols. In the second stage, the power balancing techniques are adopted to relieve the power imbalance and extend the lifetime. However, the power imbalance is very difficult to be eliminated due to the essential many-to-one communication pattern in WSNs. Therefore, the battery allocation method is used in the third stage.

Figure 2 shows a typical power distribution of a WSN using power-balanced compression and clustering techniques [8]. Obviously, the difference between sensor nodes are significant. Assuming that the total energy is denoted as E_{tot} and a uniform battery allocation is used, each sensor node is equipped with a battery containing E_{tot}/n energy. The first sensor node failure time T_{life} can be decided by $T_{life} = E_{tot}/(n \cdot p_{max})$, where p_{max} is the maximum power consumption in all sensor nodes. The dotted line indicates the power level at which the maximum-power (p_{max}) sensor node depletes its battery. The batteries of the sensor nodes located in the diagonal line region still have energy but they cannot be used due to the failure of other nodes. Therefore, a heterogeneous battery allocation in Fig. 1 will help to extend the lifetime.

The above example motivates the primary research questions addressed in this paper: Given a budget of energy cost for a WSN with arbitrary topology, node configu-

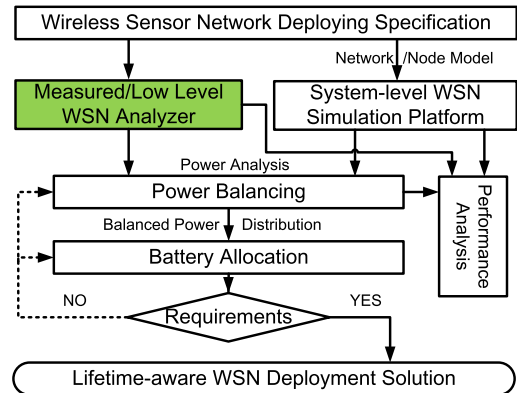


Fig. 1 Lifetime-aware deploying flow of WSN.

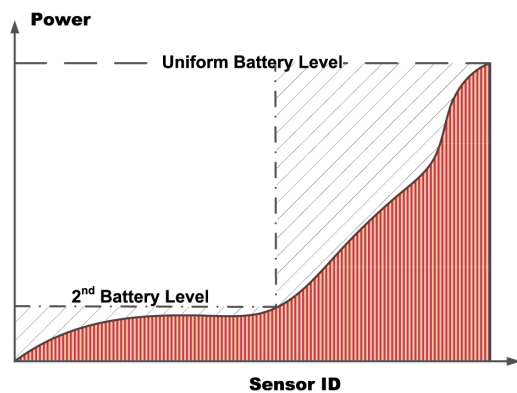


Fig. 2 Uniform vs. 2-level battery allocation.

ration, and power distribution, how should battery energy be assigned to sensor nodes to maximize the network lifetime under constraints on cost and the number of available battery pack levels? The solution should include the lifetime-aware node partitioning to enable battery pack capacities and battery allocations to be determined. Furthermore, to make the battery allocation technique more practical, we need also evaluate the impact of the power variations on the technique to prolong the lifetime of the WSNs.

3. Problem Formulation

This section first formulates a battery pack assignment problem and then analyzes the complexity of the searching space to show the necessity to develop a heuristic algorithm in the next section.

3.1 Problem Formulation

Before illustrating the lifetime-aware battery allocation algorithm, we first define the following terms. Let $P = (p_1, p_2, \dots, p_n)$ be the power distribution of the n -node network. There are m types of batteries with energy capacities E_1, E_2, \dots, E_m and costs C_1, C_2, \dots, C_m . The relationship between energy and cost is represented as follows: $C_i = f(E_i), i = 1, \dots, m$. By combining battery units into packs,

M types of battery pack levels $E_{pk\{1\}}, E_{pk\{2\}}, \dots, E_{pk\{M\}}$ are achieved. If $\omega(i, k)$ denotes the number of battery units k assigned to battery pack i , then $E_{pk\{i\}} = \sum_{k=1}^m \omega(i, k) \cdot E_k$. Each node is equipped with one battery pack. The sensor nodes are divided into M sets: L_1, L_2, \dots, L_M . Each L_i has N_i nodes, and each node in L_i is assigned a battery pack $E_{pk\{i\}}, i = 1, 2, \dots, M$.

By defining the network lifetime as the *first node failure time*, the working time of node set L_i is

$$T_i = \frac{E_{pk\{i\}}}{g_i} = \frac{\sum_{k=1}^m \omega(i, k) \cdot E_k}{g_i}, \quad i = 1, \dots, M \quad (1)$$

where g_i is the maximum power in L_i . The lifetime of the system is therefore

$$T = \min_{i=1}^M T_i \quad (2)$$

The battery allocation problem is formulated as follows: given a cost constraint $C_{total} \leq C_{cons}$ and the number of battery pack M , determine the number of the battery units in the each level to maximize T . The total cost can be represented as

$$C_{total} = \sum_{i=1}^M \left(N_i \cdot \sum_{k=1}^m \omega(i, k) \cdot f(E_k) \right) \quad (3)$$

The optimization objective is now formulated as follows:

$$T = \min_{i=1}^M \left(\frac{\sum_{k=1}^m \omega(i, k) \cdot E_k}{g_i} \right) \rightarrow \max \quad (4)$$

subject to:

1. $\sum_{i=1}^M (N_i \cdot \sum_{k=1}^m \omega(i, k) \cdot f(E_k)) \leq C_{cons}$
2. $\omega(i, k)$ is a nonnegative integer, $i = 1, \dots, M$ and $k = 1, \dots, m$

3.2 Complexity Analysis

Given a capacity-price function $C_i = f(E_i), i = 1, \dots, m$, the optimization problem is formulated as an INLP that can be solved by a standard INLP solver, such as LINGO. However, experimental results will demonstrate that the running time of a general INLP solver is too long to be tolerated. The complexity of such a problem is illustrated as below. Because any node can be equipped with several batteries with different capacities, the search space is extremely large. Assuming the case with n nodes and m battery types, the time complexity is $O((Dim + 1)^{m \times n})$, where Dim is a constant specifying the maximum number of each battery assigned to each pack. When $m = 5, M = 10$ and $n = 100$, the computation complexity reach 10^{2300} . Though, smart algorithms can be adopted in the commercial solvers to reduce such a huge search space, our experimental results in Sect. 6 showed that it failed to solve the problem in 24 hours when the network size is larger than 64. Therefore, a faster heuristic algorithm

is necessary.

4. Battery Allocation Algorithm

Figure 3 shows the flowchart of the proposed method. The first step is shown in the dotted block, in which the sensor nodes are partitioned into M sets to allow the lifetime of wireless sensor network to be optimized in the second step. Based on the partitioning, the second step in the upper-left block presents a CLPS (Cost Limited Pack Select) heuristic algorithm to choose proper battery pack configurations for each set. Finally, the node set partition, the corresponding battery pack configuration, and the lifetime of wireless sensor network are produced.

4.1 Optimal Partition to Maximize the Lifetime

This section describes a technique to divide the sensor nodes into M sets. As the later section has pointed out, the cost constraint C_{cons} could be transformed as the total energy constraint E_{cons} , we do the node partition based on the energy constraint. The relationship between C_{cons} and E_{cons} will be stated in Sect. 4.2.1. Given the power distribution of sensor nodes in a WSN and the set number, we need the sensor node partition achieving the maximum network lifetime. First, we present a theorem on the lifetime as below:

Theorem 1: Given an energy constraint E_{cons} and the power distribution p_1, p_2, \dots, p_n , the network lifetime under any node partition of M sets will at most be

$$T = \frac{E_{cons}}{\sum_{i=1}^M (g_i \cdot N_i)} \quad (5)$$

where g_i is the maximum power consumption in set L_i , and N_i is the number of nodes in L_i .

Theorem 1 is proven in Appendix A. It shows the maximum lifetime of a given node partition. The lifetime is related to g_i and N_i and varies under different node partitions.

Next, we will show how to achieve the optimal partition to maximize T . Based on Eq. (5), achieving the maximum T is simplified as the minimum value problem of $\sum_{k=1}^M (g_k \cdot N_k)$

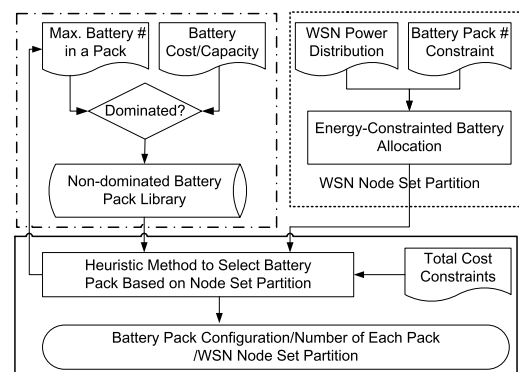


Fig. 3 Proposed method flow: Optimal partition and CLPS algorithm.

when E_{cons} is fixed.

To divide the nodes into M sets, we first sort the power consumption of nodes p_1, p_2, \dots, p_n in an ascending order as $q_1 \leq q_2 \leq \dots \leq q_n$. We define $M + 1$ boundary points to indicate the partition, where x_i and x_{i-1} denote the index of two boundary nodes in set L_{i-1} . Thus, the number of nodes in set L_{i-1} is $N_{i-1} = x_i - x_{i-1}$. As x_i is the largest index in set L_{i-1} , $g_{i-1} = q_{x_i}$. Therefore, the optimization problem is represented as:

$$V_{disc} = \sum_{k=1}^M (g_k \cdot N_k) = \sum_{i=2}^{M+1} (q_{x_i} \times (x_i - x_{i-1})) \rightarrow \min \quad (6)$$

where x_i is the optimizing variables, $i = 1, 2, \dots, M + 1$. They stand for the node number in the sorted power distribution q_1, q_2, \dots, q_n . The constraint conditions are listed as the following:

1. x_i is a nonnegative integer, $i = 1, \dots, M + 1$.
2. $x_1 = 0, x_{M+1} = n$.
3. $x_i > x_{i-1}, i = 2, \dots, M + 1$.

This problem differs from the traditional nonlinear programming formulation because the variables are the subscripts of a discrete mapping. As the objective function (Eq. (6)) could not be expressed as an elementary function, there's no direct method to solve it. However, since the power consumption sequence q_1, q_2, \dots, q_n is monotonically increasing, we can transform the original optimizing function into a piecewise continuous function $\mathbf{q}(x)$ assuming x is a continuous variable. In this way, the problem can be solved by a standard INLP algorithm. The regressive function $\mathbf{q}(x)$ is defined as:

$$\mathbf{q}(x) = q_{\lfloor x \rfloor} + (q_{\lfloor x \rfloor + 1} - q_{\lfloor x \rfloor})(x - \lfloor x \rfloor) \quad (7)$$

where $\lfloor x \rfloor$ is the lower-round of x . The optimization objective is

$$V_{cont} = \sum_{i=2}^{M+1} (\mathbf{q}(x_i) \times (x_i - x_{i-1})) \rightarrow \min \quad (8)$$

subject to:

1. $x_1 = 0, x_{M+1} = n$.
2. $x_i - x_{i-1} \geq 1, i = 2, \dots, M + 1$.

After obtaining the optimal x_i of the continuous objective function, the near-optimal discrete solution is given by rounding each x_i to $\lfloor x_i \rfloor$. The rounded solution is defined as:

$$V_{round} = \sum_{i=2}^{M+1} (\mathbf{q}(\lfloor x_i \rfloor) \times (\lfloor x_i \rfloor - \lfloor x_{i-1} \rfloor)) \rightarrow \min \quad (9)$$

where x_i is the solution of the continuous problem (Eq. (8)), while $\lfloor x_i \rfloor$ is the rounded value of x_i . We use the solution of Eq. (9) to approximate that of Eq. (6). Section 6 demonstrated that those approximates cause ignorable deviations from the optimal solution. Therefore, it provides an approximate method to obtain the node partition achieving the

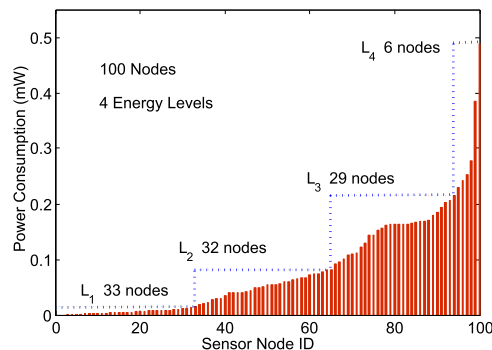


Fig. 4 Optimal 4-level energy allocation for a 100-node network.

maximal lifetime. An example of the partition method is shown in Fig. 4, with a 100 node network and 4 energy levels. The above node partitioning can be solved very fast and efficiently in Matlab.

4.2 Heuristic Method to Select Battery Pack

As Fig. 3 shows, the first stage provided an algorithm to obtain the optimal node partition. This section presents a CLPS (Cost Limited Pack Selection) procedure to transform the ideal energy based solution to a real battery pack allocation under cost constraints. We organize the procedure as follows: First, an energy-cost model for battery pack is built based on the real battery data. Second, we present a battery assignment method for each pack according to the previous node partition. Finally, the synthetic CLPS algorithm is built based on above two steps.

4.2.1 Energy-Cost Model for Battery Pack

Compared with a customer-specified battery, the battery pack is a much less expensive way to acquire batteries with various volumes under a cost constraint. This is due to the fact that many kinds of alkaline or NiMH batteries are commercially available and inexpensive. By packing standard batteries, various battery packs with different capacities and supply voltages can be obtained. In order to build an energy-cost model for battery packs, we adopted a real capacity-price model for NiMH AAA battery from the website of PowerStream [9].

Next, we propose an algorithm to build the energy-cost model. Algorithm 1 is designed to find all non-dominated battery combinations for all possible battery packs. We define a battery combination as non-dominated if no other battery combinations have a lower or equal price with a larger or equal capacity. The input Dim denotes the maximum number of each battery in one pack. If $Dim = 3$, the number of each battery in a pack ranges from 0–3. Line 3–12 show the process to enumerate all possible combinations given a Dim . After achieving all possible combinational levels, the dominated battery combinations would be removed (Line 14). The Pareto curve of the energy-cost relationship for all battery combinations with 6 battery types and $Dim = 3$

Algorithm 1 BatComb

Input: $Dim, BatUni, CostUni$
Output: $PackLev, Cost, Comb$

- 1: $PackLev = null$ (empty set)
- 2: $Cost = null$
- 3: **for** $i_1 = 1$ to Dim **do**
- 4: **for** $i_2 = 1$ to Dim **do**
- 5: ...
- 6: **for** $i_m = 1$ to Dim **do**
- 7: Pick (i_1, i_2, \dots, i_m) batteries of each unit type from $BatUni$, calculate its cost.
- 8: Add (i_1, i_2, \dots, i_m) into $Comb$, along with its energy into $PackLev$ and its cost into $Cost$.
- 9: **end for**
- 10: ...
- 11: **end for**
- 12: **end for**
- 13: ascending sort $PackLev$ and $Cost$
- 14: remove those dominated combinations

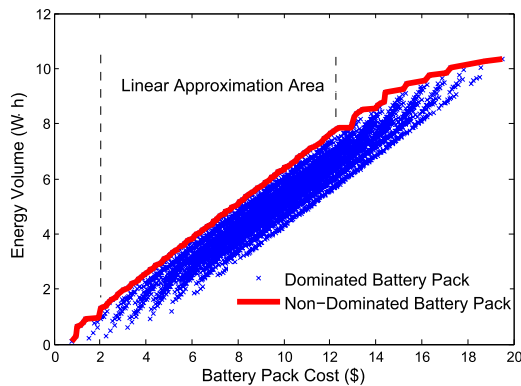


Fig. 5 Non-dominated battery pack selection from the battery unit combinational usage.

is shown in Fig. 5.

As Fig. 5 has shown, the Pareto energy–cost relationship for battery packs is near-linear in most ranges. This can be explained by the following facts: First, using a single type of battery to construct packs would lead to an exactly linear energy–cost relationship. In case of several battery types, the battery with the lowest price per unit capacity would be used as much as possible, while other batteries will be seldom used when there is a discontinuity in capacity. As reference [9] has shown, the price per capacity for different battery does not vary greatly. Therefore, the energy–cost Pareto curve can be approximated by a linear function $C = a + b \cdot E$. The fitting error is analyzed and evaluated in Sect. 6, which validated the accuracy of this approach. This property greatly simplifies the optimization, which will transform the cost constrained problem into an energy constrained one.

4.2.2 Energy Assignment and Quantization

Given the maximum battery pack and an optimal node partitioning, the energy assignment procedure allocates proper battery packs with various capacities to each set of nodes.

Algorithm 2 PackAssign

Input: $Ref, PackLev, Cost, G$
Output: $T, TotCost, Alloc$

- 1: assign $PackLev(Ref)$ to each node in set L_M
- 2: **for** each node set $L_i \neq L_M$ **do**
- 3: assign $\lceil \frac{G(i)}{G(M)} \cdot PackLev(Ref) \rceil$ to each node in L_i
- 4: **end for**
- 5: calculate T and $TotCost$, record the allocation $Alloc$

Algorithm 3 CLPS

- 1: $Dim \leftarrow 1$
- 2: **while** 1 **do**
- 3: $(PackLev, Cost, Comb) \leftarrow$
 BatComb ($Dim, BatUni, CostUni$)
- 4: $Ref \leftarrow$ index of the Maximum $PackLev$
- 5: $(T, TotCost, Alloc) \leftarrow$
 PackAssign ($Ref, G, PackLev, Cost$)
- 6: **if** $TotCost > C_{cons}$ **then**
- 7: break
- 8: **else**
- 9: $Dim \leftarrow Dim + 1$
- 10: **end if**
- 11: **end while**
- 12: **while** $TotCost > C_{cons}$ **do**
- 13: $Ref \leftarrow Ref - 1$
- 14: $(T, TotCost, Alloc) \leftarrow$
 PackAssign ($Ref, G, PackLev, Cost$)
- 15: **end while**
- 16: output $Alloc$ and $Comb$

The sensor nodes in each set are equipped with battery packs with the same capacity. It is straightforward to obtain a node-level battery allocation by assigning just one node to each set. The input of Algorithm 2 contains the combinational energy capacity vector $PackLev$, the corresponding price vector $Cost$ from Algorithm 1, and the power vector G , where each $G(i)$ refers to the maximum power consumption in separate node sets L_i , $i = 1, 2, \dots, M$. In Line 1, the pack with the maximum capacity $PackLev(Ref)$ is assigned to the most power consuming node set L_M . For other node sets, their capacities are determined by multiplying the maximum capacity with the energy ratio $\frac{G(i)}{G(M)}$ given by the optimal node partition (Appendix A). To map those energy capacities to real battery packs, the pack with the nearest capacity from $PackLev$ is chosen. The algorithm outputs the network lifetime T , the total battery price $TotCost$, and the battery allocation result $Alloc$.

4.2.3 Cost Limited Pack Selection

Based on the energy–cost model and the energy assignment procedure, CLPS Algorithm 3 completes the entire battery allocation problem under the total cost C_{cons} . The first phase of Algorithm 3 is to determine the maximum number of each battery in one pack Dim (Line 1–11). The algorithm begins its search from $Dim = 1$. By calling Algorithm 1, the capacity vector $PackLev$ and the price vector $Cost$ are obtained. Those vectors are given to Algorithm 2, which generates a total battery pack allocation. If the total

cost of such an allocation is less than C_{cons} , we will increase Dim until the budget running out. The second phase determines the final allocation (Line 12–15). By reducing Ref incrementally, a total cost just below and close to the constraint C_{cons} is achieved. Therefore, the battery assignment $Alloc$ and the combination of each pack $Comb$ are given out.

5. Impacts of Power Variations on Lifetime

In real deployments, the power analysis in the earlier design stage may suffer from its accuracy due to the varying workloads, wireless link quality and sampling ratios. This section would discuss and bound the impacts of power variations on the lifetime.

5.1 Definition of Variations

We denote the power consumption for node j as p_j and the power variation Δp_j for node j is a random variable. Thus, the accurate power of the node j can be expressed as $p_j + \Delta p_j$. Suppose the allocation procedure assigns the node j a pack with energy $E_{node\{j\}}$, its real lifetime is

$$t_{j,real} = \frac{E_{node\{j\}}}{p_j + \Delta p_j} \quad (10)$$

According to Eqs. (1) and (2), the network lifetime becomes $T_{real} = \min_{j=1}^n t_{j,real}$

5.2 Lifetime Bounds under Variations

Next, we would bound the network lifetime under the power variations. Without the power variations, we denote the node j with maximal power consumption in set L_i as g_j . Considering the power variations, the node k is the most power consuming in set L_i , which is defined as $\max_{k \in L_i} (p_k + \Delta p_k)$. The index k may not always be equal to j . We define the power variation constant ρ of a node j as $\rho_j = \frac{\Delta p_j}{p_j}$. Based on Eqs. (1) and (2), the lower and upper bound of the network lifetime can be separately expressed as:

$$T_{real,lowbound} = \frac{T_{cal}}{1 + \max_j \{\rho_j\}} \quad (11)$$

$$T_{real,upbound} = T_{cal} \{1 + \max_j \{\rho_j\}\} \quad (12)$$

where T_{cal} represents the estimated network lifetime without considering the power variations. The lower bound of the network lifetime is reached when the largest positive power variation ρ happens in the most power consuming node. The upper bound is reached when the the largest negative power variation happens in the most power consuming node when its power consumption is still the largest one under power variations.

5.3 Variations Tend to Reduce Lifetime

We will prove a theorem to illustrate that the lifetime decreasing probability is usually larger than the increasing one

when the battery pack level M is rather large. We denote the increasing power consumption probability of the node j as α_j , while the decreasing and unchanging probability is $\beta_j = (1 - \alpha_j)$. Assume the most power consuming node in each partitioned set is denoted as $\{x_i\}, i = 1, 2, \dots, M$. We have the following theorem.

Theorem 2: Given α_j and β_j of each node j , the decreasing probability of the network lifetime is $p_{dec,real}$ while the increasing and unchanging probability is $p_{inc,real}$. They obey the following equations:

$$p_{dec,real} \geq 1 - \prod_{i=1}^M (1 - \alpha_{x_i}) = 1 - \prod_{i=1}^M \beta_{x_i} \quad (13)$$

$$p_{inc,real} = 1 - p_{dec} \leq \prod_{i=1}^M \beta_{x_i} \quad (14)$$

When the battery pack level M is large enough, the following relationship holds:

$$p_{inc,real} \leq p_{dec,real} \quad (15)$$

Theorem 2 is proven in Appendix B. When variable α and β are equal to 50%, $p_{dec,real} \approx 1 - (0.5)^M, p_{inc,real} \approx (0.5)^M$. Assuming $M = 10, p_{dec,real} \approx 0.9990$. Therefore, the lifetime will decrease in most cases. Experimental results in Sect. 6 will further validate those analysis through exhaustive simulations. It implies that the lifetime improvement may not be obtained when the battery pack level increases above a certain threshold considering the power variations. This will limit the effectiveness of a fair large battery pack number in practise.

6. Evaluation

This section would evaluate the proposed battery allocation by experiments. It first describes the experimental setup and then compares the CLPS approach with the uniform traditional one to show its advantages on lifetime. The CLPS heuristic is further compared with a standard INLP solver to show its performance and solution quality. Finally, we describe the impacts of power variations on lifetime.

6.1 Experimental Setup

To evaluate the typical power distribution in WSNs, we adopted the real μ AMPS-1 node [10] to extract the power profiles. The number of sensor nodes ranges from 10 to 900. The average node-to-node distance d_0 is 20 m and the transmission parameters are extracted from real measurements. We use a distance and density based clustering protocol from Reference [8]. It can compress the data based on the spatial correlation and lighten the workload of cluster heads. It can reduce the total communication power and balance the intra-cluster power consumption. Though the power balance technology is applied in this protocol, the variance between nodes cannot be eliminated due to the many-to-one network topology. For a WSN adopting above protocol containing 100 nodes, the difference between the maximum and minimum node power consumption can reach 2–3 orders of

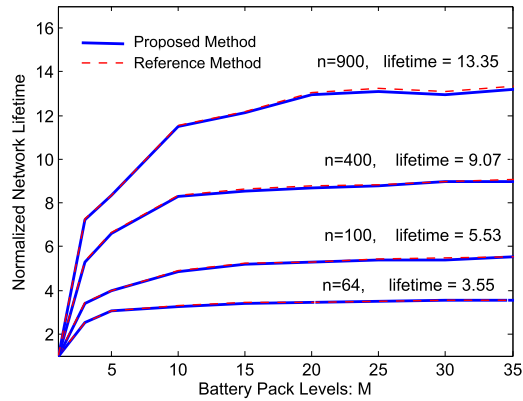


Fig. 6 Normalized lifetime for different battery pack levels and network sizes.

magnitudes in Fig. 4. The energy–cost model for the battery pack is built based on the real data [9]. Six kinds of batteries are considered in our experiments. The proposed lifetime-aware deployment framework, including power analysis and battery allocation procedure, is implemented in MATLAB running on a PC with a 2.67 GHz Intel processor with 2 GB RAM.

6.2 Battery Allocation for Lifetime Maximization

This section first compares the battery allocation method with the uniform approach and then we demonstrate the reduction of the lifetime difference of each node by battery allocation. Finally, we show that the battery allocation method can significantly prolong the lifetime with less costs.

6.2.1 Lifetime Improvement via Battery Allocation

Figure 6 illustrates the impacts of the battery pack level (1–35) on the lifetime compared with the traditional uniform battery allocation. We scan the network size from 64 to 900. As we can see, given a cost budget, the network lifetime (normalized by that of the uniform-level battery case) becomes up to 13× longer when the number of the battery pack level M increases. The lifetime expansion becomes larger when the network size increases. The normalized lifetime expansion for a WSN with 64 nodes is 3.55, while it is up to 13.35 for a WSN with 900 nodes. It is due to the fact that larger network causes much higher power imbalance due to the many-to-one traffic pattern.

The differences between the proposed method and reference in Fig. 6 are small, which implies the discrete level approximation leads to slight errors. The reference line indicates the unrealistic but largest lifetime when the ideal battery allocation is adopted, i.e., there is a battery pack level for every node. Increasing the number of the battery pack level M has an impressive effect on lifetime extension at the beginning. However, as the battery pack level M becomes larger, the lifetime of WSN increases slowly. On the contrary, more battery pack levels lead to higher battery manufacturing and deployment costs and more sensitivities to the

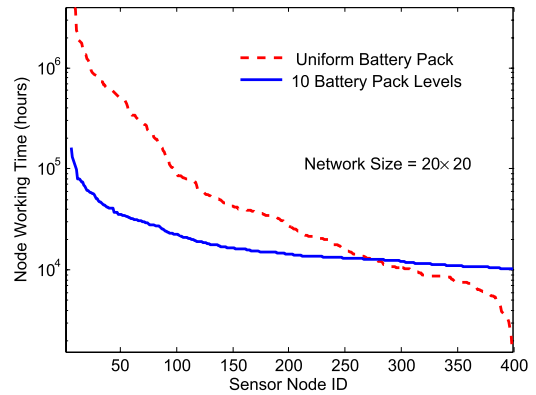


Fig. 7 Lifetime difference in different nodes of uniform and 10 battery pack levels in a 400-node network.

power variations. In our experiments, 10 battery pack levels were always sufficient to improve the lifetime to 3–11×. It implies that a reasonable number of battery pack level would be enough for the real sensor deployment.

6.2.2 Lifetime Difference among Nodes

Theorem 1 has pointed out that the ideal battery allocation would lead to the equal lifetime for each node. Figure 7 shows the lifetime difference of each sensor node in a 20×20 network with 10 battery pack levels. The lifetime differs severely (nearly 7 orders of magnitude) from node to node if only a single battery pack levels is used. The proposed method with 10 battery pack levels balances the working time distribution. However, the lifetime of some nodes can not be restricted to the average value by the battery assignment because the power consumption of some nodes is extremely small. Even if the battery pack with the minimum volume are assigned to those nodes, they still have much longer life than others. Except for those deviations in certain nodes, the proposed method effectively reduces the standard variance of lifetime from 4.06×10^5 hours to 8.70×10^2 hours among nodes.

6.2.3 Lifetime Extensions vs Battery Costs

Figure 8 provides a lifetime curve of a 100-node WSN given different battery costs. The battery budget ranges from 100–400\$ and the scanning step is 20\$. Since the cheapest battery is larger than 1\$ in the experiment, the minimal budget should be larger than 100\$. The battery pack level 1, 5 and 10 are considered. As we can see, the multiple battery pack level can extend the WSN lifetime linearly when the battery cost increases. The slope ratio becomes larger when more battery pack levels are allowed, which means more lifetime increase with extra costs. Given a 400\$ battery budget, the solution with 5 battery pack levels would give almost 4 × lifetime extensions than the one with 1 battery pack level. Furthermore, the curve provides the minimum battery budget to reach a given lifetime requirement. Obviously, multiple battery pack level allocation method needs much less

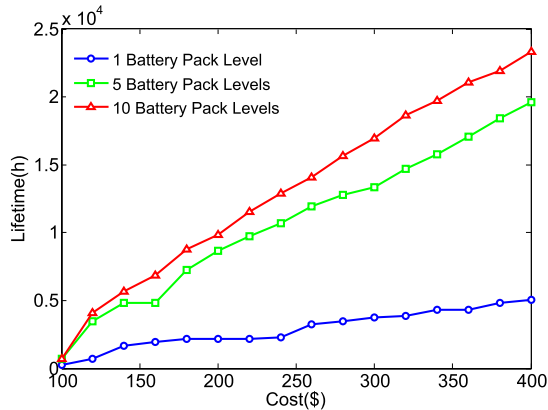


Fig. 8 Lifetime curve of a 100-node WSN under different battery budgets.

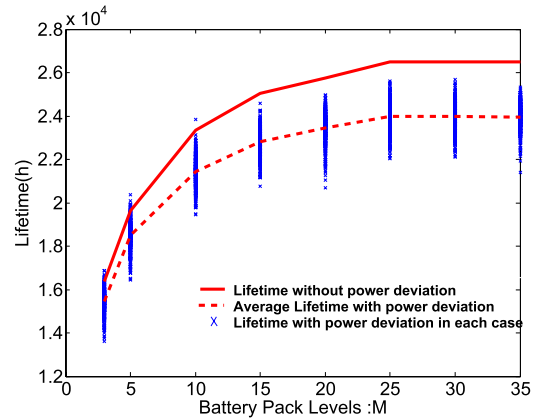


Fig. 9 WSN lifetime under power variations compared with the one without power variations for different battery pack levels.

Table 1 Time & performance comparisons between CLPS and LINGO.

Network Size	Normalized Lifetime			Execution Time (s)	
	CLPS	LINGO	Ref	CLPS	LINGO
36	3.22	1.93	3.26	0.004	125
49	3.31	1.99	3.38	0.004	405
64	3.59	3.13	3.70	0.015	1.06×10^5
100	5.63	N/A	5.80	0.010	N/A
400	9.58	N/A	10.43	0.014	N/A
900	15.11	N/A	15.59	0.067	N/A

energy budget than the uniform battery approach under the same lifetime requirement.

6.3 Performance Comparisons with INLP Solver

As Sect. 3.1 has stated, the running time of a general-purpose INLP solver was excessive in the proposed battery allocation procedure. We now compare the proposed algorithm with LINGO, a popular solver for linear and nonlinear programming problems. The best-case continuous node-level references are also provided to illustrate the deviation from the optimality. Since LINGO could not handle the node partitioning, we do not limit the number of the battery pack types in the comparison. A branch-and-bound solver and a default iteration number are used in LINGO. The results are listed in Table 1.

For the settings in Table 1, the proposed CLPS method gains a speedup of up to 2–5 orders of magnitudes over LINGO. Furthermore, our approach can solve the battery energy allocation problem for a WSN with 400 nodes in less than 0.02 seconds while LINGO fails to find the local optimal solutions within 24 hours. In the small cases, our approach gives even better solutions than the locally optimal solutions by LINGO; LINGO does not necessarily provide the globally optimal solutions. Our approach considers the characteristics of solution to reduce the search space. It may be possible to provide LINGO other configurations to get a better result using more execution time. However, this comparison provides evidence that the straightforward use of a general-purpose INLP solver is inappropriate for the WSN

Table 2 Number of nodes whose lifetime are under the original 100-node network lifetime.

Battery Pack Level	3	5	10	15	20	25	30	35
Number of Nodes	7	7	12	17	21	23	25	23

battery energy allocation problem, and that the proposed solution rapidly produces high-quality results.

6.4 The Impact of Power Variations on Lifetime

This experiment illustrates the impact of the node's power variations on the lifetime. In this case, the node number of the WSN is 100 and the battery budget is 400\$. The battery pack level ranges from 3 to 35. We assume each node's power variation q_i satisfies a Gauss distribution $N(q_i, 0.06q_i)$, in which a maximal 20% power deviation is observed among 500 samples[†]. Given a battery pack level, the WSN lifetime are evaluated under 500 random power profiles and it is determined by the node with the shortest lifetime in the network.

Figure 9 showed the lifetime variations under 500 samples using each battery pack level. As we can see, the average lifetime is usually smaller than the original one without power variations. It validates Theorem 2 experimentally. When the battery pack level is small, the lifetime presents a better tolerance to the power variations. It is due to the fact that fewer battery pack levels lead to more energy redundancy for more nodes. In order to show the effects of power variations on each node, Table 2 gave the number of nodes whose lifetime become smaller after considering power variations. As we can see, 7 nodes in a 100-node WSN with 3 battery pack levels becoming shorter, while the

[†]The power variations of each node come from many factors, such as different protocols, process variations, voltage and temperature variations. Reference [11] analyzed the power variations of a sensor node with a general configuration. It showed that the power variations approximately follow a normal distribution with the standard deviation 6% of the average value. Though the results only hold for their configuration. It should represent a typical trend for many real sensor nodes.

number of such nodes reach up to 23 with 35 battery pack levels. Those phenomena indicated that a larger battery pack level is not guaranteed to acquire a longer lifetime due to the node's power variations in reality. A not very large battery pack level should provide both lifetime extension and enough tolerance to the power variations.

7. Conclusions and Future Works

Low-cost battery-powered wireless sensor nodes have quite tight power budgets. Unbalanced power distributions due to the intrinsic many-to-one traffic in WSN results in uneven battery depletion and short lifetimes. This paper proposed a fine-grained node-level battery allocation technique. It formulates the cost-constrained heterogeneous WSN battery allocation problem as an INLP and provides a fast heuristic that produces near-optimal solutions. Experimental results show that the proposed techniques can provide 4–13× lifetime improvement with no more than 10 battery pack levels compared with the uniform approach. Furthermore, the proposed heuristic method gains a speedup of 2–5 times and better results over a popular INLP solver. The impacts of power variations on lifetime are also discussed and bounded in theory. Our future work includes evaluating the methods in a physical sensor network system and exploiting this methodology in other battery-powered ad-hoc networks to extend their lifetime.

Acknowledgement

The authors would like to thank Prof. Robert Dick for his helpful suggestions. This work was supported in part by the NSFC under grant 60976032, National Science and Technology Major Project under contract 2010ZX03006-003-01 and High-Tech Research, Development 863 Program under contract 2009AA01Z130 and in part by 2011 national special fund for IoT development.

References

- [1] H. Long, Y. Liu, Y. Wang, R.P. Dick, and H. Yang, "Battery allocation for wireless sensor network lifetime maximization under cost constraints," Proc. Int. Conf. Computer-Aided Design, pp.705–712, 2009.
- [2] C.Y. Chong and S.P. Kumar, "Sensor networks: Evolution, opportunities, and challenges," Proc. IEEE, vol.91, pp.1247–1256, 2003.
- [3] Y. Hou, Y. Shi, H. Sherali, and S. Midkiff, "Prolonging sensor network lifetime with energy provisioning and relay node placement," Proc. Sensor and Ad Hoc Communications and Networks, pp.295–304, 2005.
- [4] X. Wu, G. Chen, and S.K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," IEEE Trans. Parallel Distrib. Syst., vol.19, no.5, pp.710–720, 2008.
- [5] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy efficient schemes for wireless sensor networks with multiple mobile base stations," Proc. Global Telecommunications Conf, pp.377–381, 2003.
- [6] E. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks," Proc. International Communications Conf., pp.3663–3667, 2004.
- [7] M. Sichitiu and R. Dutta, "Benefits of multiple battery levels for the lifetime of large wireless sensor networks," Proc. Networking, pp.1440–1444, 2005.
- [8] H. Long, Y. Liu, X. Fan, R. Dick, and H. Yang, "Energy-efficient spatially-adaptive clustering and routing in wireless sensor networks," Proc. Design, Automation & Test in Europe Conf, pp.1267–1272, 2009.
- [9] "Guidelines for battery packs," <http://www.powerstream.com/BPD.htm>
- [10] R. Min and A. Chandrakasan, "A framework for energy-scalable communication in high-density wireless networks," Proc. Int. Symp. Low Power Electronics & Design, pp.36–41, 2002.
- [11] T. Matsuda, T. Takeuchi, H. Yoshino, M. Ichien, S. Mikami, H. Kawaguchi, C. Ohta, and M. Yoshimoto, "A power-variation model for sensor node and the impact against life time of wireless sensor networks," Proc. Communications and Electronics, pp.106–111, 2006.

Appendix A

Proof 1: For a given node partition L , there is an energy partition E which divides E_{cons} into $E_{set\{1\}}, E_{set\{2\}}, \dots, E_{set\{M\}}$, each corresponds to a node set L_i , $i = 1, 2, \dots, M$. The energy partition E ensures the lifetime longest by satisfying

$$\frac{E_{set\{i\}}}{E_{cons}} = \frac{g_i \cdot N_i}{\sum_{k=1}^M (g_k \cdot N_k)}, \quad i = 1, 2, \dots, M \quad (A.1)$$

The following paragraphs will prove this is true.

The energy allocation E satisfies Eq. (A.1) has the following property: In each L_i , the lifetime T_i is:

$$T_i = \frac{E_{set\{i\}}}{g_i \cdot N_i} = \frac{E_{cons} \cdot g_i \cdot N_i}{g_i \cdot N_i \cdot \sum_{k=1}^M (g_k \cdot N_k)} = \frac{E_{cons}}{\sum_{k=1}^M (g_k \cdot N_k)} \quad (A.2)$$

T_i is unrelated to i , that means the lifetime of each node set L_i is equal. Furthermore, any energy partition E which obeys this property satisfies Eq. (A.1), because for any s and t , we get:

$$\frac{E_{set\{s\}}}{g_s \cdot N_s} = \frac{E_{set\{t\}}}{g_t \cdot N_t} = \frac{E_{cons}}{\sum_{k=1}^M (g_k \cdot N_k)} \quad (A.3)$$

Therefore, Eq. (A.2) is the necessary and sufficient condition of Eq. (A.1).

Suppose that there is another energy partition E' , which doesn't assure the equal lifetime of each L'_i property. There necessarily exist two sets L'_s and L'_t such as $T'_s < T'_t \leq T'_k$, ($k \neq s, t$). That means T'_s is the minimum lifetime of all the L'_i . From Eq. (1), we get:

$$\frac{E'_{set\{s\}}}{g'_s \cdot N'_s} < \frac{E'_{set\{t\}}}{g'_t \cdot N'_t} \quad (A.4)$$

If a tiny adjustment is made on $E'_{set\{s\}}$ and $E'_{set\{t\}}$ by δ , the following inequality is still satisfied:

$$\frac{E'_{set\{s\}} + \delta}{g'_s \cdot N'_s} < \frac{E'_{set\{t\}} - \delta}{g'_t \cdot N'_t} \quad (A.5)$$

So, the minimum lifetime of all the L'_i turns a little higher. That means the energy partition E' is not the best one.

By now, we have proved that for any node partition L , the best energy partition makes the lifetime in each L_i equivalent. The longest network lifetime is as Eq. (5) describes. \square

Appendix B

Proof 2: Assume only the most power consuming node without considering power variations in each partitioned set $\{x_i\}$, $i = 1, 2, \dots, M$ can affect the network lifetime, where M stands for the number of battery levels. We denote the decreasing probability of lifetime as p_{dec} and the increasing or unchanging probability as p_{inc} . In real cases, we need remove the assumption to obtain the real decreasing probability of lifetime $p_{dec,real}$ and the increasing or unchanging probability $p_{inc,real}$. Since any nodes may decrease the network lifetime and the boundary-node case is a subset, the following equations hold:

$$p_{dec,real} \geq p_{dec} \quad (\text{A} \cdot 6)$$

$$p_{inc,real} = 1 - p_{dec,real} \leq p_{inc} \quad (\text{A} \cdot 7)$$

Without losing generality, we assume all $\beta_j \leq 1 - \delta$, $\delta > 0$ in Eqs. (13) and (14), the following equations hold when $M > \log_{1-\delta}(0.5)$,

$$p_{inc} = \prod_{i=1}^M \beta_{x_i} < (1 - \delta)^{\log_{1-\delta}(0.5)} = 0.5 \quad (\text{A} \cdot 8)$$

$$p_{dec} = 1 - p_{inc} > 0.5 > p_{inc} \quad (\text{A} \cdot 9)$$

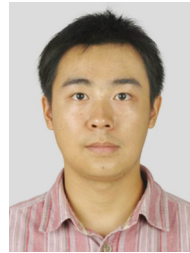
Therefore, when M is larger enough, we have:

$$p_{inc,real} \leq p_{inc} < p_{dec} \leq p_{dec,real} \quad (\text{A} \cdot 10)$$

Assuming an unbalance distribution with $\delta = 0.1$, $\beta = 0.9$, $\alpha = 0.1$, $p_{dec} > p_{inc}$ when $M > 7$. Given a normal distribution, we can observe $p_{dec} \gg p_{inc}$ when $M \geq 10$,



Yiqun Wang was born in Hubei Province. He received his B.S. from Electronic Engineering Department, Tsinghua University in 2009 and now is a Ph.D. student in the same department. His main research interests are low power VLSI designs and non-volatile memory and circuits. He is now working in the project of wireless sensor network SOC design.



Hengyu Long was born in Guizhou Province, P.R. China. He was a master student in the Department of Electronic Engineering, Tsinghua University. His main research interests are power-efficient protocols for wireless sensor network.



Huazhong Yang was born in Sichuan Province, P.R. China, on Aug. 18, 1967. He received B.S., M.S., and Ph.D. Degrees in Electronic Engineering from Tsinghua University, Beijing, in 1989, 1993, and 1998, respectively. Now, he is a Professor and Head of the Institute of Circuits and Systems in Electronic Engineering Department, Tsinghua University, Beijing. His research interests include CMOS radio-frequency integrated circuits, VLSI system structure for digital communications and media processing, wireless sensor network, low-voltage and low-power circuits, and computer-aided design methodologies for system integration. He has authored and co-authored over 30 patents, 7 books, and over 200 journal and conference papers. He was granted National Palmry Young Researcher Fund of China.



Yongpan Liu was born in Henan Province, P.R. China. He received his B.S., M.S. and Ph.D. degrees from Electronic Engineering Department, Tsinghua University in 1999, 2002, and 2007. He worked as a research fellow in Tsinghua University from 2002 to 2004. Since 2007, he became an assistant professor in Tsinghua University. He has published over 40 peer-reviewed conference and journal papers, supported by NSFC, 863, 973 Program. His main research interests include embedded systems,

nonvolatile computing, power-aware architecture and VLSI design and electronic design automation. He is an IEEE member and served as a reviewer of several IEEE conferences and TVLSI.