Smart Meter Deployment Optimization for Efficient Electrical Appliance State Monitoring

Xiaohong Hao, Yongcai Wang, *Member, IEEE,* Chenye Wu, *Student Member, IEEE,*
Amy Yuexuan Wang, *Member, IEEE,* Lei Song, Changjian Hu, and Lu Yu

***Abstract*—Monitoring the energy consumptions of the massive electrical appliances in buildings has attracted great attentions for smart, green and sustainable living. Traditional approaches generally require large-scale smart sensor/meter networks, and thus suffer from the high deployment, maintenance and data collection costs. In this paper, we propose methodologies and algorithms to optimize the smart meter deployments to track the on/off states of the massive electrical appliances by using the minimal number of smart meters. Particularly, based on the tree structure of the power distribution networks we show the deployment of m meters will decompose the power distribution network into a forest of m mono-meter trees. Each mono-meter tree has depth 1, with one meter at the root and a set of appliances at the leaves. A mono-meter tree is clear if the meter at the root can decode the on/off states of its leaves without error. Based on it, the smart meter deployment optimization problem is to optimize the meter deployment locations, so as to minimize the number of required meters while keeping all the mono-meter trees being clear. We prove this problem is NP-hard, and propose a greedy algorithm to approach it by utilizing the bounds of degree and the maximum power of the load tree. We show the greedy algorithm has at most 2 approximation ratio. Finally,we assess our approach in different structure power networks by simulations. The simulation results suggest a reasonable good performance of our proposed smart meter deployment strategy.**

***Keywords*—Energy monitoring, smart meter, deployment optimization, energy auditing, NP hard, greedy algorithm, approximation ratio, smart building, sensor network**

I. Introduction

In the past decade, electricity consumption has grown at a remarkable rate, almost 2.7 percent annually on average. Recent reports show that residential and commercial buildings account for 72% of the total energy consumption [1] and 30% of energy consumed in the buildings is wasted [2]. To better utilize the energy in such buildings, researches in the field of smart building and smart grid are exploring an efficient energy control strategy.

To achieve the efficient energy control strategy, monitoring the real-time on/off states of the electrical appliances is the primary concern, since the monitoring provides the necessary information for smart device control. Recent advantages of wireless sensor/meter network enable fine-grained appliance state monitoring. However, the massive, broadly distributed electrical appliances generally require deploying many smart meters, which poses high costs to the smart meter deployment, data collection and system maintenance. In practice, the building facility managers generally require the energy auditing system to be low cost, highly accurate and scalable. In other words, they want to minimize the number of deployed smart meters while tracking the states of massive appliances accurately and in real-time.

The first five authors are with Institute for Theoretical Computer Science, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China. E-mails: haoxiaohong.ivy@gmail.com, wangyc@tsinghua.edu.cn, wucy05@mails.thu.edu.cn, amywang@tsinghua.edu.cn, leisong03@gmail.com. The last two authors are with NEC Labs. China. E-mails: hu_changjian@nec.cn, yu_lu@nec.cn

This work was supported by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61061130540, 61073174, 61202360.

A. Related Work

The energy auditing problem has caught tremendous attentions from both academia and industry for the last decade. There are two main bodies of related literature.

The first one focused on the non-intrusive load monitor (NILM) based on/off state disaggregation. In particular, the NILM-based method can efficiently reduce the deployment cost for smart meters, since it only deploys *one* high-frequency smart meter at the root of the power load tree to disambiguate the on/off states of the appliances by transient or static signal processing and pattern recognition. With this approach, Hart proposed a reactive load signature detection in [3]; Norford *et al.* introduced a transient event detection method; Patel *et al.* performed the preconization by electrical noise in [4]; and Farinaccio *et al.* addressed a major end-use detection in [5]. However, the NILM-based approach generally needs high-cost meter for high frequency sampling and is limited in application scale. To tackle this challenge, in this paper, we propose a lightweight approach to deploy several low cost smart meters, which can work in a relatively low frequency. In [6], Wang *et al.* proposed a light weight compressive sensing framework to track the on/off states of massive electrical appliances. However, how to optimize the smart meter deployment in the compressive sensing framework is an unaddressed problem, impacting the energy monitoring cost and the tracking efficiency, which is studied in this paper.

The other body of literature is related to the fidelity energy auditing systems. For example, in [7], Jiang *et al.* utilized contextual metadata for the design and deployment of a wireless sensor network for the high-fidelity monitoring of the electrical usage in buildings. In [8], Kazandjieva *et al.* introduced PowerNet, which was a hybrid sensor network for monitoring the power and utilization of computing systems in a large academic building. In [9], Dawson-Haggerty *et al.* shared their insights obtained from a year-long, 455 meter deployment of wireless plug-load electric meters in a large commercial building. Different from [7]–[9], instead of assuming the on/off states of the appliances are sensed by RFID sensors and treating the amount of energy consumptions as unknown, to the best of our knowledge, we are the first to

978-1-4673-0911-0/12/$31.00 ©2012 IEEE

present the relationship between the smart meter deployment cost and the state tracking accuracy.

B. Our Contribution

The electricity network in a building can be modeled by a load tree, where the leaves represent the electricity appliances and the non-leaf nodes are switches or outlets. We assume the smart meters can be placed at any node on the tree to monitor the aggregated real-time energy consumptions of all the appliances in the subtree rooted at the monitored node. The on/off states of the appliances are the key *unknown* variables in the energy monitoring process, which provide the fundamental state information for smart control technologies. In practice, n appliances generally have 2^n combinatorial on/off states. As such, to track the real-time states of the appliances, it generally requires a large amount of smart meters and high computation costs.

However, we note that, among the 2^n states, many states are easy to be distinguished from the others. Based on this observation, we find that there is a chance to reduce the deployment cost of smart meters while guaranteeing the state tracking accuracy. The key challenges is to understand how the deployment of a smart meter will impact the state tracking accuracy, and how to deploy the minimal number of smart meters while guaranteeing the accurate state tracking. These challenges inspire us to present the following contributions:

- *Load Tree Decoding:* Any deployment scenario splits the load tree into a set of mono-meter trees (a sub-tree with only one meter at the root). An efficient decoding method is proposed for the mono-meter tree to decode the states of leaf-appliances by the smart meter at the root. Based on the decoding scheme, we propose a necessary condition for the high-accuracy decoding.
- *Hardness Proof and Algorithm Design:* We prove the problem of minimizing the smart meter deployment cost while satisfying the necessary condition for state decoding is *NP-hard*. Thus, we present a 2-approximation parametric algorithm based on the fact that in practice, load trees have only bounded node degree and appliance have bounded power consumption.
- *Simulation Assessment:* We assess our decoding algorithm and smart meter deployment scheme with field data. Then, we show our proposed approximation algorithm works reasonably well and can reduce the number of smart meters by more than 70% in most power load networks by simulation.

The rest of this paper is organized as follows. We introduce the system model in Section II. We present the state vector decoding method in Section III. Then, in Section IV, we propose our approximation algorithm and prove its approximation ratio. Section V performs simulation studies for our proposed approaches for decoding and smart meter deployment. Finally, concluding remarks are presented in Section VI.

II. System Model

The energy distribution network in a building has a typical tree-like structure, which can be modeled as hierarchical load tree [7]. The root of the load tree is the power entrance of the building. Each non-leaf node is a power break or an outlet while the leaves stand for the electrical appliances. As mentioned in Section I, the smart meters can be deployed at any node in the load tree. We only require the smart meters to measure the total power consumptions of the appliances in the subtree rooted at the monitored node. Next, we will introduce the power pattern and the observation model respectively.

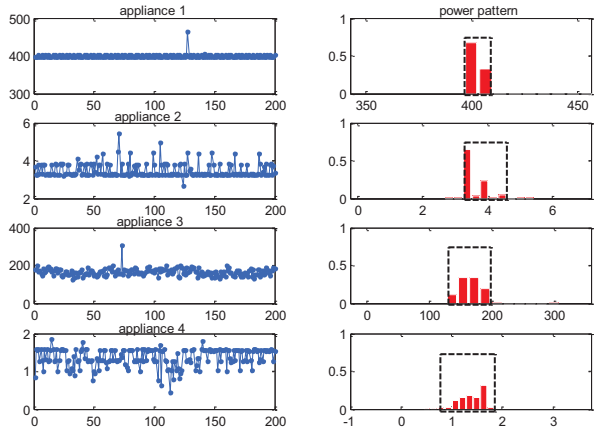

Fig. 1. Power Pattern for different electrical appliances.

A. Power Pattern

Power pattern indicates the energy consumption value when an appliance is on. For most appliances, its power pattern can be learned off-line from the nominal power data sheet or by an off-line learning process. Fig. 1 shows the real-time power patterns of four different types of appliances in a 200-second period. We represent the statistical power pattern of each appliance i by a tuple (p_i, θ_i), where p_i is the expected power consumption, and θ_i is the power derivation. The real time power consumption can be bounded by $[(1-\theta_i)p_i, (1+\theta_i)p_i]$ with high probability by various training methods.

B. Observation Model

In our model, we assume there are m meters deployed on the load tree to monitor the on/off states of n appliances. Thus, we can formulate the observation model between appliance states and the smart meter readings as $\mathbf{aP} \approx \mathbf{z}$, where:

- $\mathbf{a} \in \{0,1\}^n$ is the *state vector*, indicating the on/off states of the n appliances.
- $\mathbf{z} = (z_i | i = 1, \cdots, m) \in \mathbb{N}^m$ is the *observation vector*, and each z_i indicates the power measurements of smart meter i.
- $\mathbf{P} \in \mathbb{N}^{n \times m}$ is the *pattern matrix* of the load tree. Note that, $\mathbf{P}$ is determined by the smart meter deployment settings and the expected powers of n appliances. More specifically, the j^{th} column of $\mathbf{P}$ is determined by smart meter j. And $P_{i,j} = p_i$, only if appliance i is in the subtree of meter j. Otherwise, $P_{i,j} = 0$.

Fig. 2 shows an example of the pattern matrix. Two smart meters S_1 and S_2 are deployed on a load tree. Since d_1, d_2, d_3 are in the subtree of meter S_1, the 1^{st} column of $\mathbf{P}$ is $[p_1, p_2, p_3, 0, 0]^T$. The 2^{nd} column of $\mathbf{P}$ is $[0, 0, 0, p_4, p_5]^T$ because only d_4, d_5 are in the subtree of S_2.

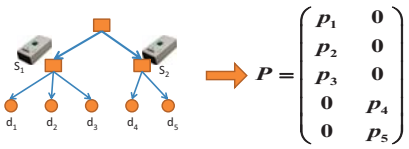

Fig. 2. The mapping from the meter deployment to the observation matrix.

Remember, our goal of appliance state monitoring is to recover the on/off states of the n appliances by the readings of m deployed smart meters. The difficulty is that there are n unknown states while only m equations, and usually, we have $m \ll n$. To eliminate the potential ambiguities in decoding $\mathbf{a}$, our task is to design a good pattern matrix by understanding the relationship between decoding accuracy and the smart meter deployment.

III. State Vector Decoding

In this section, we try to explore the relationship between the decoding accuracy and the smart meter deployment. At each time slot, we need to find the state vector $\mathbf{a}$ on a given pattern matrix $\mathbf{P}$ such that

$$\begin{aligned} \textbf{minimize} \quad & \|\mathbf{aP} - \mathbf{z}\|_2 \\ \textbf{subject to} \quad & \mathbf{a_i} \in \{0,1\}, \forall i = 1, \cdots, n. \end{aligned} \tag{1}$$

This problem can be approached by Least Square Estimation (LSE) and can be solved by various algorithms, *e.g.,* the Tibshirani algorithm [10]. However, considering the short time interval and the limited computational power of the metering system, the previous approaches are still not acceptable. Thus, we need to further utilize the properties of the load tree to reduce the decoding complexity.

A. Mono-meter Tree Decomposition

Note that the decoding problem can be decomposed and executed in parallel in a set of sub-trees. That is, in the decoding phase, the power load tree can be transformed into a forest of *mono-meter* trees. For example, for a node v in the load tree, if it is monitored by a smart meter, the power consumptions of the appliances in its subtree $ST(v)$ will be continuously monitored by the smart meter. That is, all the parents of v can know the power consumptions of its subtree. Based on this observation, we can perform the load tree decomposition, *e.g.,* for a load tree $T = (V, E)$ with m meters deployed on it, it can be split into a forest of m mono-meter trees, where each mono-meter tree has only the root equipped by a smart meter. Fig.3 shows a load tree split example. Fig. 3(a) shows the smart meter deployment in a load tree. We denote the nodes equipped with smart meter as black nodes and the other nodes as white ones. Fig. 3(b) shows the forest of mono-meter trees transformed from Fig. 3(a).

Assuming the on/off states of the appliances are independent from each other, the decoding task could be distributed to each mono-meter tree and solved in parallel. For the rest of the paper, we only discuss the decoding phrase on mono-meter trees due to the space limitation. However, the technique is quite similar for the general case.

The observation model of the mono-meter tree $ST(v)$ with n_v appliances can be formulated by $\mathbf{a}^v\mathbf{p}^v \approx z^v$, where $\mathbf{a}^v \in \{0,1\}^{n_v}$ is the state vector of $ST(v)$, indicating the on/off states of the appliances in $ST(v)$, $\mathbf{p}^v = (p_1^v, p_2^v, ..., p_{n_v}^v)^T$ is the pattern vector of $ST(v)$[1], and $z^v \in \mathbb{N}$ is observation value of $ST(v)$.

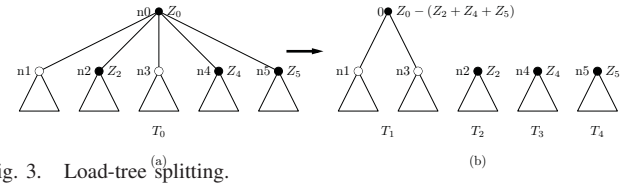

Fig. 3. Load-tree splitting.

We can directly apply the algorithms for LSE for decoding in each mono-meter tree. Since mono-meters trees could be decoded in parallel, and the number of nodes in a mono-meter tree is $n_v \ll n$, the running time could be reduced sharply. Moreover, since there is only one observation value on a mono-meter tree, we can store all the possible observation values and their corresponding state vectors in a mapping table. The mapping table could be built by an off-line pre-process, where we enumerate all the state vectors and sort them according to their power consumption in time $O(n_v 2^{n_v})$. Thus, the running time for online decoding at each time interval is $O(n_v)$.

B. State Decoding Accuracy and Meter Deployment

In energy auditing applications, users generally desire to minimize the number of smart meters in order to reduce the deployment costs and the maintenance costs. That is, we want each mono-meter tree contain as much leaves as possible. However, in the meanwhile, each mono-meter tree must can disambiguate the on/off state of its appliances correctly. Mathematically, we note that, for a mono-meter tree $ST(v)$, the expected observation value of $\mathbf{a}^v$ is $\widehat{z^v_{\mathbf{a}^v}} = \mathbf{a}^v\mathbf{p}^v$. And the observation derivation from $\mathbf{a}^v$ is $\theta_{\mathbf{a}^v} = (\Sigma_{i=1}^{n_v} a_i^v p_i^v \theta_i^v)/(\Sigma_{i=1}^{n_v} a_i^v p_i^v)$. In an ideal situation, we assume that there is no noise and measurement error. Thus, the observation value $z^v_{\mathbf{a}^v}$ should fall into $R(\mathbf{a}^v) = [(1-\theta_{\mathbf{a}^v})\widehat{z^v_{\mathbf{a}^v}}, (1+\theta_{\mathbf{a}^v})\widehat{z^v_{\mathbf{a}^v}}]$. In the preprocess, the possible power consumption of an appliance is modeled by an interval $[p_i - \theta_i, p_i + \theta_i]$. Therefore, the possible total power consumption of a set of appliances in different on/off states may locate in a set of observation ranges, as shown in Fig. 4. However, if some observation ranges overlap with each other, it is impossible to guarantee the overlapped value is mapped to the right state vector. In such situation, we call the mono-meter tree is *blurry*. We now formally define a *blurry mono-meter tree* as follows:

Definition 1: (Blurry Mono-meter Tree) For a mono-meter tree $ST(v)$ rooted on node v, $ST(v)$ is blurry *if and only if* there exist two different state vector $\mathbf{a}_1^v$, $\mathbf{a}_2^v$ such that $R(\mathbf{a}_1^v) \cap R(\mathbf{a}_2^v) \neq \emptyset$. Otherwise, the mono-meter tree is clear.

For example, in Fig. 4, there are two observation ranges $[42, 48]$ and $[40, 44]$. They overlap on $[42, 44]$. When the observation value is in $[42, 44]$, the decoding algorithm cannot guarantee the right output state. Therefore, to ensure a correct decoding, we have Theorem 1:

[1]Note that we denote the power patten of appliance i in $ST(v)$ by (p_i^v, θ_i^v).

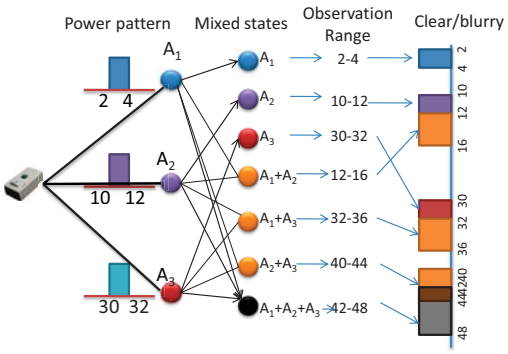

Fig. 4. An example to explain the blurry condition.

Theorem 1 (Necessary condition for zero-error monitoring): For a given load tree T, if an algorithm can decode the state vector from an observation based on a smart meter deployment D without error, then there is no blurry mono-meter tree T under deployment D.

IV. Smart Meter Deployment

We want to find a scheme that reduce the deployment cost yet ensuring the monitoring accuracy. Based on the necessary condition for the error-free monitoring in Theorem 1, we formally define the smart meter deployment problem as:

Smart Meter Deployment Problem (SMDP):

- **Input**: A tree $T = (V, E)$ with N nodes. We denote $L \subseteq V, |L| = n$ as the set of leaves in T. Each leaf $i \in L$ has a 2-tuple power pattern (p_i, θ_i) with it.
- **Output**: The minimal set of deployed smart meters while ensuring there is no blurry mono-meter tree.

A. Hardness of SMDP

We can show that SMDP is NP-hard [11], thus finding an efficient algorithm that output the optimal solution may not be feasible. Yet in practice, the power load tree has some particular properties that may be helpful to design good approximation algorithms. For example, the degree of the power load tree are usually small, and the largest power consumption of an appliance in the tree is bounded. Thus we can design an efficient approximation algorithm based on the following assumptions:

Assumption 1: We assume the maximum degree of the node in T is upper bounded by a constant d, and $p_i \leq B, \forall i$, where B is a constant.

B. Greedy Algorithm for SMDP

Recalling Theorem 1, a feasible error-free monitoring deployment needs all the mono-meter trees clear. To reduce the number of deployed meters, clearly, we need to ensure that all the meters are useful, *i.e.,* every smart meter must contribute to the clearness of some subtree in T. Thus we have the following lemma:

Lemma 2: In the best strategy, a meter could be place on the node v only if $ST(father(v))$ is Blurry, where $father(v)$ is the father of v.

Based on Lemma 2, we propose a greedy strategy for S-MDP. We search each node from leaves up to the root.

Algorithm 1 Greedy Algorithm for SMDP

INPUT : Tree $T = (V, E)$ with integers on the leaves
OUTPUT: Set $D \subseteq V$ to deploy smart meters
$D = \emptyset$
for all $i \leftarrow 1$ to n **do**
  let v be the i^{th} node when searching from bottom to the root
  if $ST(v)$ is "Blurry" **then**
    $C_{\min} = Children(v)$
    for all $C \subseteq Children(v)$ **do**
      $ST'(v) = ST(v) \setminus (\bigcup_{u \in C} ST(u))$
      if $|C_{\min}| > |C|$ and $ST'(v)$ is "Clear" **then**
        $C_{\min} \leftarrow C$
      end if
    end for
    $D \leftarrow D \cup C_{\min}$
    $T \leftarrow T \setminus ST(u), \forall u \in C_{min}$
  end if
end for
return D

Let $v_1, v_2, ..., v_N$ be the nodes in the visiting order of the algorithm. In each iteration, we may cut off some subtrees from the original tree to ensure that all the mono-meter trees generated from visited nodes are clear. We denote T_i as the *remaining tree* after visiting node v_i, for $i = 1, \cdots, N$. As such, when the algorithm visits v_i,

- *If $ST_i(v)$ is blurry,* we need to deploy some meters on the children of node v to make $ST_i(v)$ clear. Suppose $Children(v)$ is the set of all the children of node v. There are at most 2^d subsets of $Children(v)$. And it takes time $O(1)$ to enumerate all of them. Then we could find the smallest subset $C_{\min} \subseteq Children(v)$ such that the subtree $ST_i(v)$ is clear by removing all the subtrees $ST_i(u), u \in C_{\min}$. The smart meter should be deployed on node u for all $u \in C_{\min}$. And the new remaining tree T_{i+1} is generated from T_i by removing all the subtrees $ST_i(u), u \in C_{\min}$;
- *If $ST_i(v)$ is clear,* Lemma 2 implies that no smart meter is needed to be deployed on any child of v.

The algorithm visits all the nodes one by one until reaching the root. Note that a smart meter is always needed on the root. Algorithm 1 shows the pseudo code for our proposed greedy algorithm. Note that we only need T_{i-1} in the ith iteration, thus we don't need to store every V_i.

C. Clear/Blurry Judgement

We now discuss how to judge whether a tree is clear or not. By Theorem 1, if a mono-meter tree is blurry, there must exist some overlapped observation ranges. On the mono-meter tree $ST(v)$, for any pair of different state vectors $(\mathbf{a}_1^v, \mathbf{a}_2^v)$, let their expected observation values be $\widehat{z_{\mathbf{a}_1^v}^v}$, $\widehat{z_{\mathbf{a}_2^v}^v}$, and observation derivations be $\theta_{\mathbf{a}_1^v}$ and $\theta_{\mathbf{a}_2^v}$, respectively. The observation range $R(\mathbf{a}_1^v)$ and $R(\mathbf{a}_2^v)$ overlap if $|\widehat{z_{\mathbf{a}_1^v}^v} - \widehat{z_{\mathbf{a}_2^v}^v}| \leq (\theta_{\mathbf{a}_1^v} \widehat{z_{\mathbf{a}_1^v}^v} + \theta_{\mathbf{a}_2^v} \widehat{z_{\mathbf{a}_2^v}^v})$. So we only need to check whether there exists such a pair in the mono-meter tree or not. The basic idea is derived from

dynamic programming. The process of judging a mono-tree is blurry or not is composed of two steps.

1) Enumerate all possible expect observation values: For the given mono-meter tree $ST(v)$ with n_v appliances, $p_1^v, p_2^v, ...p_{n_v}^v$ are the expected power consumption for appliances in $ST(v)$. The set of state vectors for $ST(v)$,

$$\mathcal{S}^j = \{\mathbf{a}^v | \mathbf{a}^v \in \{0,1\}^{n_v}, \text{ and } a_i^v = 0, \forall i > j\}. \quad (2)$$

Let $U_v[0, \cdots, n_v B, i] \in \{0,1\}$ be an boolean array for mono-meter tree $ST(v)$. $U_v[\hat{z}, i]$ is 1 *if and only if* there exists a vector $\mathbf{a}^v \in \mathcal{S}^i$ with expected observation value $\hat{z}$. $\theta_v[\hat{z}, i]$ is the largest observation derivation among all the state vectors $\mathbf{a}^v \in \mathcal{S}^i$ with expected observation value $\hat{z}$. Enumerating i from 1 to n_v, the value of U_v and θ_v can be calculated by dynamic programming:

$$U_v[\hat{z}, i] = \begin{cases} 1, & U_v[\hat{z} - p_i^v, i-1] = 1 \text{ or }, \\ & U_v[\hat{z}, i-1] = 1, \text{ or } (\hat{z}=0) \wedge (i=0), \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

$$\theta_v[\hat{z}, i] = \max\{\theta_v[\hat{z}, i], \frac{\theta_v[\hat{z}](\hat{z} - p_i^v) + \theta_i^v p_i^v}{\hat{z}} U_v[\hat{z} - p_i^v, i-1]\}. \quad (4)$$

Finally, we can obtain the value of arrays $U_v[\hat{z}, i]$ and $\theta_v[\hat{z}, i]$. There exists a possible expected observation value $\hat{z}$ *if and only if* $U[\hat{z}][n_v]$ is 1. Notice that if $U_v[\hat{z}, i-1]$ and $U_v[\hat{z} - p_i^v, i-1]$ are both 1, there are two different ways to make $U_v[\hat{z}, i]$ be 1. It implies that there are two different state vector with the same observation $\hat{z}$ and the two vector are different on the i^{th} position. Therefore, the mono-meter tree is already blurry and we do not need to go to the next step. In the coding phrase, since we only need $U_v[\hat{z}, i-1]$ to compute $U_v[\hat{z}, i]$, we can reuse the memory of $U_v[\hat{z}, j], \forall j < i$. It is the same for $\theta_v[\hat{z}, i]$.

2) Check the existence of overlapped observation segments: After the first step, we obtain all the possible expected observation values in and ascending order. Then we need check if there is an overlap between each two successive observation ranges. Algorithm 2 indicates the clear/blurry judging process.

Algorithm 2 Clear/Blurry Judgement for State Vector

INPUT : A tree $ST(v)$
RETURN: "Clear" if $ST(v)$ is clear;"Blurry" otherwise.
for all $\hat{z} \leftarrow 1$ to $n_v B$ **do**
  $U_v[\hat{z}] = 0$, $\theta_v[\hat{z}] = 0$
end for
$U_v[0] = 1$
for all $p_i, i = 1...n_v$ **do**
  for all $\hat{z} \leftarrow n_v B$ downto 1 **do**
    if $\hat{z} - p_i^v \geq 0$ and $U_v[\hat{z} - p_i^v] = 1$ **then**
      if $U_v[\hat{z}] = 1$ **then**
        return "Blurry"
      else
        $U_v[\hat{z}] = 1$
        $\theta_v[\hat{z}] = \max\{\theta_v[\hat{z}], \frac{\theta_v[\hat{z}](\hat{z} - p_i^v) + \theta_i^v p_i^v}{\hat{z}}\}$
      end if
    end if
  end for
end for
$\widehat{z_1} \leftarrow 0$
for all $\widehat{z_2} \leftarrow \widehat{z_1} + 1$ to $n_v B$ **do**
  if $U_v[\widehat{z_2}] = 1$ **then**
    if $\widehat{z_2} - \widehat{z_1} \leq \theta_v[\widehat{z_1}]\widehat{z_1} + \theta_v[\widehat{z_2}]\widehat{z_2}$ **then**
      return "Blurry"
    else
      $\widehat{z_1} = \widehat{z_2}$
    end if
  end if
end for
return "Clear"

D. Algorithm Analysis

Note that when visiting a node v, there are at most 2^d candidates subset of its children. A clear/blurry judgement with running time $O(Bn_v{}^2)$ is needed for each subset. Since $n_v < N$, the total running time is $O(2^d BN^3)$, which is polynomial to N when d and B are constant. We can further prove that:

Theorem 3: The meter deployment $G(T) \subseteq V$ obtained from our greedy algorithm satisfies $|G(T)| \leq 2|\text{OPT}(T)|$, where $\text{OPT}(T)$ denotes the optimal deployment.
The proof of Theorem 3 can be found in [12].

V. Simulation Results

In addition to the theoretical analysis, numerical evaluations are preformed to evaluate the performances of the proposed algorithm. In our simulations, we generate the load trees with different scales, different node degrees and different appliance power distributions at random. We want to investigate our greedy algorithm's performance on cost saving, and compare it with the optimal smart meter deployment.

A. Deployment Cost Saving Performance

Fig. 5 shows the cost saving performance of the algorithm under error-free state vector decoding constraint on the load trees under different settings. In our simulation, we evaluate the performance of our approach under three different kinds of power distributions, such as the uniform distribution, the normal distribution, and the exponential distribution. The mean of each distribution is set to be 500. The *cost ratio* is defined as the number of deployed meters divided by the number of appliances in the load tree. From the result, we can see that for the load trees with different sizes and different structures, our algorithm can always save more than 70% meter deployment cost when comparing with the naive monitoring method, where we deploy a smart meter at each node. The cost ratio under exponential and uniform power distribution are similar. While the cost ratio under the normal power distribution is higher than the other two. Intuitively, the power levels of appliances are more concentrated in the normal distribution, which implies more smart meters are required for disambiguating the states of similar-power appliances.

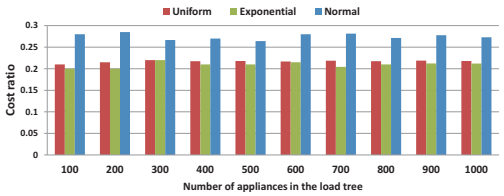

Fig. 5. Cost ratio for error-free monitoring on the load trees under different settings.

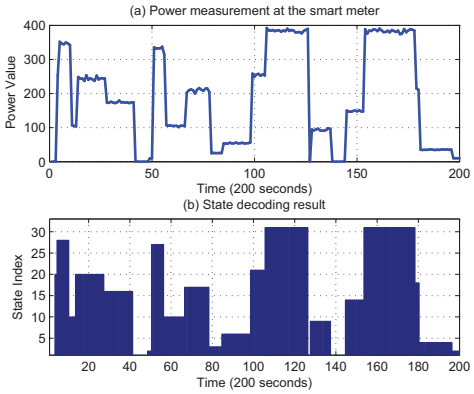

Fig. 6. State recovery result of decoding algorithm for SMDP.

B. Monitoring Performance

The result in each scenario is an average over 10 random generated trees with different maximum degrees. In theory, our deployment algorithm provides a error-free monitoring guarantee. However, in practice, the power pattern cannot always be within $p_e(1-\theta, 1+\theta)$, where p_e is the expected value of an appliance. We use the normal distribution $\mathcal{N}(p_e, p_e\theta/2)$ to simulate the real time power consumption, which implies that more than 95% of the real time appliance power consumption is within $p_e(1-\theta, 1+\theta)$. Fig. 6 shows an example trace of the on/off state decoding process in a load tree of 5 appliances, over 200 seconds. The y-axis in Fig. 6(b) shows the index of the 2^5 different decoded states, where the state indices are sorted according to their expected observation values. The overall decoding accuracy is higher than 90%. Thus, we can conclude that our monitoring performance in this case is still pretty well with the limited smart meters deployed by our greedy approach.

C. Comparison of Greedy Algorithm for SMDP and Optimal Solution

Table I compares the deployment costs of the greedy algorithm for SMDP and the optimal solution. We define the *approximation ratio* as the number of meters computed by greedy algorithm divided by the number of meters needed in the optimal solution. Remind that, in the algorithm analysis, we prove that the approximation ratio is upper bounded by 2. In the simulation, we use a brute force algorithm to find the optimal solution on the random generated power load trees with 8 to 28 appliances. The result in each scenario is an average over 10 random generated trees with different maximum degrees. Because of the NP-hardness of SMDP, the running time for computing the optimal solution is exponential

TABLE I
PERFORMANCE OF OUR PROPOSED GREEDY ALGORITHM FOR SMDP.

♯ Appliances in load tree	8	12	16	20	24	28
♯ Meters (Greedy Algorithm)	2.2	2.3	3.2	4.2	4.8	6
♯ Meters (Optimal Solution)	2.2	2.2	3.2	4.1	4.6	5.5
Average approximation ratio	1	1.05	1	1.02	1.04	1.09
Largest approximation ratio	1	1.33	1	1.25	1.2	1.2

to the input size. As such, the scenario with 28 appliances is currently the best we can do in the simulation. As shown in the table, the average performance is, in fact, quite close to 1, which implies the greedy algorithm could find a solution very close to the optimal. And the simulation results fit into the theoretical bound.

VI. Conclusion and Future Work

This paper studied the optimal smart meter deployment problem to use the minimal number of smart meters to track the on/off states of the massive electrical appliances. We provided a necessary deployment condition for error-free state decoding. Then we proved finding an optimal meter deployment strategy to satisfy the necessary condition is NP-hard, and presented a 2-approximation parametric algorithm to solve the problem. Simulation results show that our algorithm can significantly reduce the number of smart meters in various structure power load networks.

Our work can be extended in several ways. For example, we would like to study the optimal deployment problem for monitoring appliances with multi-mode and dynamic patterns. Furthermore, by utilizing the time dependence for state sequence decoding, we may further reduce the scale of the smart meter deployment.

References

[1] U.S. DOE., "Buildings energy data book," 2010. [Online]. Available: http: //buildingsdatabook.eere.energy.gov
[2] ——, "Commercial buildings energy consumption survey," 2003. [Online]. Available: http://www.eia.doe.gov/emeu/cbecs
[3] G. W. Hart, "Nonintrusive appliance load monitoring," in *Proceedings of the IEEE*, December 1992, pp. 1870–1891.
[4] S. Patel, T. Robertson, J. Kientz, et al, "At the flick of a switch : Detecting and classifying unique electrical events on the residential power line," *Work*, vol. 4717, pp. 271–288, 2007.
[5] L. Farinaccio and R. Zmeureanu, "Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses," *Energy and Buildings*, vol. 30, no. 3, pp. 245–259, 1999.
[6] Y. Wang, X. Hao, L. Song, C. Wu, Y. Wang, C. Hu, and L. Yu, "Tracking states of massive electrical appliances by lightweight metering and sequence decoding," in *Proceedings of SensorKDD'12*. ACM, 2012, pp. 34–42.
[7] X. Jiang, M. Van Ly, J. Taneja et al., "Experiences with a high-fidelity wireless building energy auditing network," in *Proceedings of ACM SenSys*, Berkeley, CA, 2009.
[8] M. Kazandjieva, O. Gnawali B. Heller et al., "Identifying energy waste through dense power sensing and utilization monitoring," *Stanford PowerNet technical report*, 2010.
[9] S. Dawson-Haggerty, S. Lanzisera, J. Taneja et al., "@scale: insights from a Large, long-lived appliance energy WSN," in *Proceedings of ACM IPSN*, Beijing, China, 2012.
[10] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.
[11] X. Hao, Y. Wang, C. Wu, A. Y. Wang, and L. Song, "Hardness proof," http://wcy.name/papers/proof1.pdf.
[12] ——, "Proof of theorem 3," http://wcy.name/papers/proof2.pdf.