On the Balance of Meter Deployment Cost and NILM Accuracy

Xiaohong Hao[*]

Tsinghua University

Beijing, China

haoxiaohong.ivy@gmail.com

Bangsheng Tang[†]

Hulu LLC.

Beijing, China

bangsheng.tang@gmail.com

Yongcai Wang[*]

Tsinghua University

Beijing, China

wangyc@tsinghua.edu.cn

Abstract

Non-Intrusive Load Monitoring (NILM) uses one smart meter at the power feed to disaggregate the states of a set of appliances. Multiple NILM meters are deployed to achieve high monitoring accuracy in large-scale power systems. Our work studies the tradeoff between monitoring accuracy and meter deployment, in a quantitative and extensible way. In particular, we introduce a *clearness function* as an abstract indicator of expected monitoring accuracy given any NILM method, and then showcase two concrete constructions. With the notation of a clearness function, we propose solutions to the *smart meter deployment problem (SMDP)*, that is, the problem of finding a deployment scheme with minimum number of meters while attaining a required monitoring accuracy. Theoretically, SMDP is shown NP-hard and a polynomial-time approximation scheme (PTAS) is proposed in this paper. For evaluation, we show that our proposed scheme is efficient and effective in terms of approximation ratio and running time. On real and simulated datasets, our proposed framework achieves a higher monitoring accuracy at a much lower cost, outperforming common baseline algorithms.

1 Introduction

Real-time monitoring on/off states of electrical appliances in buildings is a crucial component for smart control systems. Deploying smart meters on every appliance guarantees *high-fidelity appliance state monitoring* [Jiang *et al.*, 2009], but the cost is unbearably high especially in the case of modern commercial buildings with a huge number of appliances.

An alternative approach is *Non-Intrusive Load Monitoring (NILM)* [Hart, 1992]: at the electrical feed of the appliances, a meter is deployed to disaggregate their on/off states by pattern analysis in temporal or frequency domain. Early works

on NILM took advantage of steady-state power changes [Sultanem, 1991; Marceau and Zmeureanu, 2000]. The key idea is to detect jumps in active/reactive power patterns specific to individual appliances, therefore it is more suitable for steady, finite-state appliances. Harmonic analysis was used to identify continuously variable appliances [Laughman *et al.*, 2003; Lee *et al.*, 2005; Berges *et al.*, 2009]. Latest machine learning techniques like support vector machine [Patel *et al.*, 2007], neural networks [Roos *et al.*, 1994], sparse coding [Wang *et al.*, 2014], pattern recognition [Farinaccio and Zmeureanu, 1999] and a number of unsupervised learning methods [Shao *et al.*, 2013; Kolter and Jaakkola, 2012; Parson *et al.*, 2012] all found applications in the field of NILM.

While most aforementioned papers strive for better NILM algorithms for accuracy and efficiency, scalability remains a major limitation of NILM. In a single-meter NILM system, monitoring error increases as the power system scales up. Multiple NILM meters are required in a complex power system. Considering the cost of the meters including deployment and maintenance, it is desirable to use as few meters as possible while attaining a satisfactory monitoring accuracy. In contrast to the extensively investigated NILM algorithms, the problem of balancing deployment cost and monitoring accuracy lies greatly unexplored. Only few prior works studied this problem (cf. [Jung and Savvides, 2010; Hao *et al.*, 2012; Bellala *et al.*, 2012]). In this paper, we investigate the quantitative tradeoff between the meter deployment cost and the NILM state monitoring accuracy. In particular, we propose:

- **Clearness function** For any power load tree with a meter located at the root, a clearness function takes values in $[0, 1]$, where higher value indicates higher monitoring accuracy. This notation induces a quantitative criteria for monitoring accuracy.

- **Smart meter deployment problem (SMDP)** That is, finding a meter deployment satisfying the quantitative monitoring accuracy requirement with minimum number of meters. We show that SMDP is NP-hard and then present a PTAS to approximately solve SMDP, which is highly efficient for practice use.

- **A framework for tradeoff** The tradeoff problem could be decomposed into two subproblems: 1) design a clearness function for the underlying power model; 2) solve the SMDP given the clearness function. This framework

is generic in that it can easily incorporate with various NILM methods, and algorithmic results and quantitative relations will carry over naturally.

2 A Framework for Accuracy-Cost Tradeoff

This section devotes to the high-level idea of our framework for tradeoff. We assume that the power network structure is a priori knowledge, which is for example obtainable from the blueprints [Jung and Savvides, 2010; Shao *et al.*, 2013]. Following [Parson *et al.*, 2012], we assume access to some basic prior information of an appliance's power pattern. However, the on/off correlation between the appliances is unknown since it is closely related to user behaviors.

2.1 Power Load Tree

Power network in a commercial building typically has a tree-like structure, which is called *power load tree* (PLT) [Jung and Savvides, 2010] and denoted by $\mathcal{T}(V, E)$. The root of a PLT is the power entrance of a building. Each internal node corresponds to a power break or an outlet, and each leaf node corresponds to an electrical appliance. There is an edge $e_{i,j} \in E$, if power flows through an internal node $v_i \in V$ to another node $v_j \in V$. Let n_e be the total number of leaves (namely, electrical appliances) and $n = |V|$ be the number of nodes in $\mathcal{T}(V, E)$. State monitoring is to track the on/off states of the n appliances in real-time, i.e., a vector $\mathbf{s}^v \in \{0, 1\}^{n_e}$. NILM methods have been employed in the task of tracking $\mathbf{s}^v$ accurately with multiple meters placed on nodes in PLT.

2.2 Multi-meter NILM

Since NILM meters measure power profiles, like current and voltage, which are linearly additive. By assuming the meters are reliable and accurate, a PLT with a (partial) deployment $D \subseteq V$ could be decomposed into several *mono-meter trees*. Each mono-meter tree contains exactly one meter deployed at its root. Also each appliance in the original PLT is assigned to a mono-meter tree rooted at a closest meter-monitored node. Let $\mathcal{T}(u) = (V(u), E(u))$ be the subtree of $\mathcal{T}$ rooted at u. For a meter deployed at $u \in D$, the mono-meter tree generated from this meter is $\mathcal{T}_m(u) = (V_m(u), E_m(u))$, where $V_m(u) = V(u) \setminus \{w \mid w \in V(v), v \in D \cap V(u) \setminus \{u\}\}$, $E_m(u) = \{(v, w) \mid (v, w) \in E, v \in V_m(u), w \in V_m(u)\}$. Fig 1 gives an example of mono-meter tree decomposition.

Since the correlation between appliances is unknown, the on/off appliance states on mono-meter trees are decoded independently. By such a decomposition, the original multi-meter on/off decoding process is separated into several independent single-meter on/off decoding process. Apparently, the multi-meter decoding is acceptable only if the decoding accuracy on each mono-meter tree is good enough.

2.3 Tradeoff Framework

The tradeoff between monitoring cost and monitoring accuracy can be easily seen from the mono-meter tree decomposition. Deploying more meters at appropriate positions will reduce the size of each meter's appliance set. It also reduces the probability of state ambiguity, which helps improve state
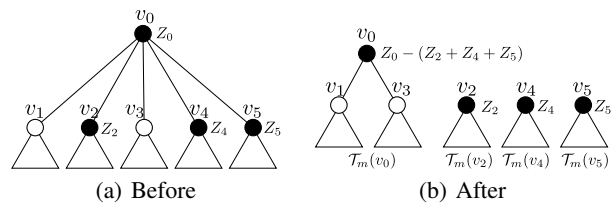

(a) Before (b) After

Figure 1: Mono-meter tree decomposition: a PLT rooted at v_0 is decomposed into mono-meter trees $\mathcal{T}_m(v_0)$, $\mathcal{T}_m(v_2)$, $\mathcal{T}_m(v_4)$, $\mathcal{T}_m(v_5)$ by deploying meters at v_0, v_2, v_4, v_5. The *mono-meter reading* of a meter is its own reading subtracted by the sum of the readings of meters on its descendants. e.g. the mono-meter reading on v_0 is $Z_0 - (Z_2 + Z_4 + Z_5)$.
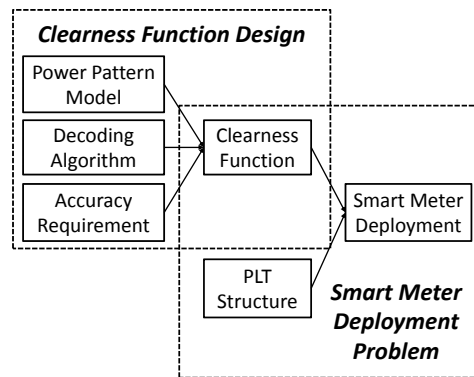

Figure 2: Accuracy-Cost Tradeoff Framework

decoding accuracy. To balance accuracy and cost, we propose a tradeoff framework consisting of two parts, as in Fig 2 – clearness function design and meter deployment.

Clearness Function Design *Clearness function* is a quantitative indicator of decoding quality of mono-meter trees. It depends on the underlying NILM model, including power patterns of the appliances, decoding algorithm and accuracy requirements (Fig 2). For a given mono-meter tree, the clearness function takes values in $[0, 1]$, and a higher value indicates a higher chance of correct decoding. For any deployment scheme, monitoring quality could be measured by computing the clearness function for all mono-meter trees.

Meter Deployment Given the clearness function as a black box, accuracy requirement can be stated as clearness function on every mono-meter tree evaluates to a large enough value. For any given accuracy requirement, *Smart Meter Deployment Problem(SMDP)* is the problem of finding a deployment scheme with minimum number of meters while satisfying the requirement. Later in this paper, we formalize this problem and propose an efficient and effective solution.

3 Clearness Function Design

Construction of clearness function varies in different NILM models. Many previous approaches on NILM [Leeb *et al.*,

1995; Parson *et al.*, 2012] assumed specific power patterns and proposed decoding algorithms accordingly. In this case the clearness function for each mono-meter tree could be calculated by off-line simulations. However it is more common that the NILM model, specially the decoding algorithm, is not well-defined beforehand, whereby the design of clearness function would be radically different.

For a mono-meter tree $\mathcal{T}_m(v)$, denote the set of appliances in it as $\mathcal{A}(v)$. Let $\mathbf{s}^v = \{0,1\}^{|\mathcal{A}(v)|}$ be the appliance state vector of $\mathcal{T}_m(v)$. For a fixed state $\mathbf{s}^v$, $\mathcal{R}^v(\mathbf{s}^v)$ is the set of all possible readings on a single meter. Mathematically, we say a reading R^v is *ambiguous* if there exist some $\mathbf{s}_1^v \neq \mathbf{s}_2^v$, s.t. $R^v \in \mathcal{R}^v(\mathbf{s_1}^v) \cap \mathcal{R}^v(\mathbf{s_2}^v)$, i.e., more than one state maps to a same reading. Otherwise, if R^v has a unique pre-image s^v, we say R^v is *clear*.

In a mono-meter tree $\mathcal{T}_m(v)$, only clear readings could be decoded correctly. Clearness function $f_c(\mathcal{T}_m(v))$ is defined in a way such that its value is proportional to the probability of seeing clear readings. i.e.

$$f_c(\mathcal{T}_m(v)) \propto \Pr[\text{clear readings}] \qquad (1)$$

Next, we showcase constructions of clearness function for two basic power patterns without specifying NILM decoding algorithm. And we will evaluate the performance of our tradeoff framework for these two models.

Example 1: Interval Power Pattern Model
In this model, power pattern is characterized as an interval containing the majority of the probability mass. Specifically, power pattern of the ith appliance is represented by a tuple (p_i, θ_i), where p_i is its expected power consumption, and θ_i is the deviation. θ_i is minimized such that the real-time power consumption takes values in $[(1-\theta_i)p_i, (1+\theta_i)p_i]$ with overwhelming probability, e.g. 95%. For electrical appliances like lights, computers and televisions, real-time power consumptions are concentrated on their rated values. This model precisely describes the appliance with concentrated power consumption. However, for those appliances with constantly changing power consumption, the deviation is large and it is harder to decode their on/off states from an aggregated power measurement.

For a mono-meter tree $\mathcal{T}_m(v)$ with n_v appliances, the expected observation value and deviation for state vector $\mathbf{s}^v$ are

$$E(\mathbf{s}^v) = \mathbf{s}^v \mathbf{p}^{vT} \quad , \quad \theta(\mathbf{s}^v) = \frac{\sum_{i=1}^{n_v} s_i^v p_i^v \theta_i^v}{\sum_{i=1}^{n_v} s_i^v p_i^v}$$

Ideally, all real-time power consumptions should fall into their corresponding intervals, and the observed value for state vector $\mathbf{s}^v$ should fall into the *observation interval* $\mathsf{Int}(\mathbf{s}^v) = [(1 - \theta(\mathbf{s}^v))E(\mathbf{s}^v), (1 + \theta(\mathbf{s}^v))E(\mathbf{s}^v)]$. Therefore, the total power consumption for $\mathcal{T}_m(v)$ must fall into one of the observation intervals. However, if some interval overlaps with another, there is no guarantee that every value can be correctly decoded to a state vector – ambiguity happens. Let $\Pr[R^v = x]$ be the probability of reading x, estimated by exploiting prior knowledge. The clearness function for interval power pattern model is defined as:

$$f_c(\mathcal{T}_m(v)) := 1 - \int_{x \text{ is ambiguity}} \Pr[R^v = x]\mathrm{d}x \qquad (2)$$

In the case that $\Pr[R^v = x]$ is not given in closed form, the clearness function is

$$f_c(\mathcal{T}_m(v)) := 1 - \frac{\int_{x \in \mathsf{Int}(\mathbf{s}_1^v) \cap \mathsf{Int}(\mathbf{s}_2^v), \forall \text{state vectors } \mathbf{s}_1^v \neq \mathbf{s}_2^v} 1 \cdot \mathrm{d}x}{\int_{x \in \mathsf{Int}(\mathbf{s}^v), \forall \text{state vector } \mathbf{s}^v} 1 \cdot \mathrm{d}x} \qquad (3)$$

The naïve way to calculate $f_c(\mathcal{T}_m(v))$ is to enumerate all possible range segments and scan each of them to get the lengths of ambiguous part and clear part. This will not scale well since for each $\mathcal{T}_m(v)$ with n_v appliances, there are 2^{n_v} possible segments. The exhaustive enumeration can be circumvented by dynamic programming, resembling the algorithm for knapsack. The key idea is to map the observation intervals to integers in $[0, B]$, preserving order and B being the maximum meter range; and then calculate how many times each integer is mapped from those observation intervals. This algorithm runs in $O(B \cdot n_v + B^2)$ time and $O(B)$ space.

Example 2: Distributional Power Pattern Model
In reality, many appliance have multiple power consumption states and each of the state have a certain probability to be on. Appliance's power consumption may follow a certain distribution on those states. Here we introduce the *distributional power pattern model* where each appliance's consumption distributes over a set of discrete values. Given a mono-meter tree $\mathcal{T}_m(v)$ with n_v appliances, let $p_1^v, \ldots, p_{n_v}^v$ be the power patterns of appliances indexed by $1, \ldots, n_v$. For appliance i in $\mathcal{T}_m(v)$, $p_i^v := \{\langle w_{i,0}^v, q_{i,0}^v \rangle, \langle w_{i,1}^v, q_{i,2}^v \rangle, ..., \langle w_{i,m_i^v}^v, q_{i,m_i^v}^v \rangle\}$ where $w_{i,j}$ is a power consumption value for appliance i and $q_{i,j}$ is the corresponding probability. $w_{i,0} = 0$ is the power consumption when the appliance is off.

To construct clearness function for distributional power pattern model, we use the notation of *entropy* from information theory, which is the average amount of information for a random event. Summation of power consumptions can be viewed as compression of the binary state vector $\mathbf{s}_i^v$. The more information available from the readings, the easier it is to achieve high monitoring accuracy. In light of the intuition, the clearness function can be defined as the ratio of monitoring entropy and on/off state entropy.

For $\mathcal{T}_m(v)$, the total entropy of appliance states is:

$$H_s(v) := - \sum_{i \in \mathcal{T}(v)} \sum_{j=1}^{m_i^v} q_{i,j}^v \log q_{i,j}^v \qquad (4)$$

Let $\{r_1^v, r_2^v, \ldots, r_{M^v}^v\}$ be the set of distinct meter readings, each corresponds to at least one combination of appliance states in $\mathcal{T}_m(v)$. $\forall i \in \{1, 2, ..., M^v\}$, let $\mathcal{K}(i)$ be the set of tuples $(k_1, k_2, \ldots, k_{n_v})$ such that $\sum_{j=1}^{n_v} w_{j,k_j} = r_i^v$. Since all the appliances take readings i.i.d. , it holds that

$$\Pr[R^v = r_i^v] = \sum_{(k_1,\ldots,k_{n_v}) \in \mathcal{K}(i)} \prod_{j=1}^{n_v} q_{j,k_j}^v \qquad (5)$$

where R^v is the reading at root v and n_v is the number of appliances in $\mathcal{T}_m(v)$. The entropy of the smart meter is then defined as:

$$H_d(v) := - \sum_{i=1}^{M^v} \Pr[R^v = r_i^v] \log \Pr[R^v = r_i^v] \qquad (6)$$

Note that $H_d(v) < H_s(v)$ implies that there are two different states with the same aggregate power consumption. This ambiguity prevents extracting exact states. Therefore, error-free monitoring can be viewed as lossless compression. Namely, $f_c(\mathcal{T}_m(v)) = 1$ if and only if $H_d(v) = H_s(v)$

However, requiring error-free monitoring is impractical because of inevitable noise and measurement errors. Here we relax the condition from error-free to low-error, and we define the clearness function as

$$f_c(\mathcal{T}_m(v)) := \frac{H_d(v)}{H_s(v)} \qquad (7)$$

In network science, entropy usually plays the role of a metric in deployment schemes for data gathering system (cf. [Guestrin *et al.*, 2005; Bellala *et al.*, 2012]). Instead of maximizing monitoring entropy, we seek to minimize the number of deployed meters subject to the accuracy criteria that the monitoring entropy remaining larger than a given threshold.

3.1 Clear Deployment Condition

In our setting, clearness indicates high accuracy monitoring for each mono-meter tree. Because of the presence of noise in practice, instead of requiring perfect state decoding, we set a threshold $\tau \in [0, 1]$ for decoding error, which we call *clear ratio*. Quantitatively, for a mono-meter tree $\mathcal{T}_m(v)$, if $f_c(\mathcal{T}_n(v))$ is larger than a predefined τ, we say that $\mathcal{T}_m(v)$ is *clear*; otherwise, it is *ambiguous*.

Similar to each mono-meter tree, the clearness of the PLT indicates all appliances are monitored as accurately as required. Based on the mono-meter decomposition, we quantify monitoring accuracy by the following,

Definition 1 (Clear Deployment Condition). *Given a PLT $\mathcal{T}(V, E)$ and a clear ratio τ,* clear deployment condition *for a meter deployment scheme $D = \{v_1, v_2, \dots, v_m\}$ is that $\forall v \in D, f_c(\mathcal{T}_m(v)) \geq \tau$.*

We would require deployments to satisfy clear deployment condition, and to verify clear deployment condition it suffices to evaluate $f_c(\cdot)$ on each mono-meter tree. When $f_c(\cdot)$ is given, finding a minimum-cost deployment is independent of the underlying power pattern model. Thus, our framework can be extended to various NILM models.

4 Smart Meter Deployment

The clearness function quantifies state monitoring accuracy of each mono-meter tree. Armed with this, our objective is to find a deployment scheme with minimum number of meters, such that all the mono-meter trees satisfy clear deployment condition with respect to the clearness function. Formally, we solve the following:

Problem 1. *(Smart Meter Deployment Problem (SMDP))*

INPUT *A PLT $\mathcal{T}(V, E)$, a clearness function $f_c(\cdot)$, a clear ratio τ and a set of power patterns to compute clearness function.*

OUTPUT *A set $D \subseteq V$, referred to as a* deployment scheme*, indicating where meters are deployed, such that $|D|$ is minimized while satisfying clear deployment condition.*

Algorithm 1 A PTAS for SMDP: $\mathcal{A}_{\mathrm{SMDP}}$

1: **INPUT :** PLT $\mathcal{T}(V, E)$ with power patterns, clearness function $f_c(\cdot)$ and monitoring accuracy threshold τ
2: **OUTPUT:** A smart meter deployment $D \subseteq V$
3: Let $v_1, \dots, v_n$ be the vertices in V in postorder
4: $\mathcal{T}^0 \leftarrow \mathcal{T}, D \leftarrow \emptyset, \mathcal{I} \leftarrow \emptyset$
5: **for** $i \leftarrow 1$ **to** $n - 1$ **do**
6: $\quad$ opt$(\mathcal{T}^{i-1}(v_i)) \leftarrow$ OPTDEPLOYMENT$(\mathcal{T}^{i-1}(v_i))$
7: $\quad$ **if** opt$(\mathcal{T}^{i-1}(v_i)) \geq 1 + \lceil 1/\epsilon \rceil$ **then**
8: $\quad\quad$ $\mathcal{T}^i \leftarrow \mathcal{T}^{i-1} \setminus \mathcal{T}^{i-1}(v_i)$
9: $\quad\quad$ $D \leftarrow D \cup$ opt$(\mathcal{T}^{i-1}(v_i))$
10: $\quad\quad$ $\mathcal{I} \leftarrow \mathcal{I} \cup \{i\}$
11: $\quad$ **else**
12: $\quad\quad$ $\mathcal{T}^i \leftarrow \mathcal{T}^{i-1}$
13: $\quad$ **end if**
14: **end for**
15: $D(v_n) \leftarrow$ OPTDEPLOYMENT$(\mathcal{T}^{n-1})$
16: $D \leftarrow D \cup$ opt$(\mathcal{T}^{i-1}(v_i))$
17: **return** D

In practice, PLTs usually have special structures that can be exploited to design good deployment algorithms. In particular, we make a mild assumption that each node in the PLT has degree upper bounded by a constant d. Consider Multi-Cut, a variant of the unweighted multicut problem on trees [Călinescu *et al.*, 2003], that is, given an unweighted tree, and some forbidden sets of leaves, find a set of edges of minimum size, whose removal separates at least one node pair from each forbidden set. It can be shown that each instance of MultiCut can be efficiently translated to an SMDP instance with a clearness function exactly characterizing the forbidden sets. This enables us to show the NP-hardness of SMDP, since MultiCut can be reduced from Exact-3SAT: let n be the number of variables, and m the number of clauses; for each variable x, construct a tree with leaves corresponding to x and $\bar{x}$, and a forbidden set containing the two leaves; for each clause, construct a tree with three leaves corresponding to its literals, and also three forbidden sets each containing two of the leaves; all these trees are connected at the bottom layer of a full binary tree with at least $n + m$ leaves. It can be proved that the Exact-3SAT instance is satisfiable if and only if the constructed MultiCut instance has optimal value $n + 4m$.

Suppose there exists an algorithm to compute $f_c(\cdot)$ in $O(1)$ time[1]. Next we turn to present $\mathcal{A}_{\mathrm{SMDP}}$, a polynomial-time approximation scheme (PTAS). Let opt$(\mathcal{T})$ be an optimal solution for SMDP on tree $\mathcal{T} = (V, E)$. The algorithm iterates over the nodes of $\mathcal{T}$ in postorder. Let v_i be the node visited in i^{th} iteration and $\mathcal{T}^i$ be the PLT after i^{th} iteration. When visiting v_i, if $|\mathsf{opt}(\mathcal{T}^{i-1}(v_i))| \geq 1 + \lceil 1/\epsilon \rceil$, we deploy meters on opt$(\mathcal{T}^{i-1}(v_i)) \cup \{v_i\}$ and cut subtree $\mathcal{T}^{i-1}(v_i)$ from $\mathcal{T}^{i-1}$. Otherwise directly visit next node v_{i+1}. Postorder traversal ensures that all the descendants of v_i have been visited before v_i, so $|\mathsf{opt}(\mathcal{T}^{i-1}(u))| < 1 + \lceil 1/\epsilon \rceil$, $\forall u \in C(v_i)$, where $C(v_i)$ is the set of v_i's children. OPTDEPLOYMENT$(\cdot)$ returns an optimal deployment for a given tree.

---

[1]This is just for convenience sake, as long as it can be computed in polynomial time, the later results will carry through.

Since $\bigcup_{u \in C(v_i)} \mathsf{opt}(\mathcal{T}^{i-1}(u)) \cup \{v_i\}$ is feasible for $\mathcal{T}^{i-1}(v_i)$, it holds that $|\mathsf{opt}(\mathcal{T}^{i-1})(v_i)| \leq \sum_{u \in C(v_i)} |\mathsf{opt}(\mathcal{T}^{i-1}(u))| + 1 \leq d(1 + \lceil 1/\epsilon \rceil) + 1$. Thus OPTDEPLOYMENT($\mathcal{T}^{i-1}(v_i)$) takes the following steps,

STEP 1 Enumerate all possible meter deployment schemes of size no larger than $d\lceil 1/\epsilon \rceil + d + 1$.

STEP 2 Filter out the schemes not satisfying clear deployment condition.

STEP 3 Choose the scheme with minimum size.

Based on the discussion above, we summarize $\mathcal{A}_{\mathrm{SMDP}}$ in Algorithm 1. We also establish a series of lemmas to show correctness and efficiency of $\mathcal{A}_{\mathrm{SMDP}}$.

Lemma 1. *D is a feasible solution to SMDP for $\mathcal{T}$.*

Proof. Let S_i be the set of mono-meter trees generated from $\mathsf{opt}(\mathcal{T}^{i-1}(v_i))$ on subtree $\mathcal{T}^{i-1}(v_i)$. All mono-meter trees in S_i are clear. The mono-meter trees generated from D on PLT $\mathcal{T}$ is the union of S_i, $\forall i \in \mathcal{I}$. Therefore, clear deployment condition is satisfied, which implies D is a feasible solution. $\square$

Lemma 2. *$|D| \leq (1 + \epsilon)|\mathsf{opt}(\mathcal{T})|$*

Proof. Observe that for any subtree $\mathcal{T}(v)$, $(\mathsf{opt}(\mathcal{T}) \cap \mathcal{T}(v)) \cup \{v\}$ is a feasible solution for $\mathcal{T}(v)$, namely $|\mathsf{opt}(\mathcal{T}) \cap \mathcal{T}(v)| + 1 \geq |\mathsf{opt}(\mathcal{T}(v))|$. Then,

$$
\begin{aligned}
|D| &= \sum_{i \in \mathcal{I}} |\mathsf{opt}(\mathcal{T}^{i-1}(v_i))| \\
&\quad (\text{Recall } |\mathsf{opt}(\mathcal{T}^{i-1}(v_i))| \geq 1 + \lceil 1/\epsilon \rceil) \\
&\leq \sum_{i \in \mathcal{I}} (1 + \epsilon)(|\mathsf{opt}(\mathcal{T}^{i-1}(v_i))| - 1) \\
&\leq (1 + \epsilon) \sum_{i \in \mathcal{I}} |\mathsf{opt}(\mathcal{T}) \cap \mathcal{T}^{i-1}(v_i)| \\
&= (1 + \epsilon)|\mathsf{opt}(\mathcal{T})|
\end{aligned}
$$

$\square$

Lemma 3. *$\forall i \in \mathcal{I}$, OPTDEPLOYMENT($\mathcal{T}^{i-1}(v_i)$) returns in polynomial time.*

Proof. There are $O(n^{d\lceil 1/\epsilon \rceil + d + 1})$ possible deployment schemes. For each scheme, there are at most $d\lceil 1/\epsilon \rceil + d + 1$ corresponding mono-meter trees, so it takes $O(d\lceil 1/\epsilon \rceil + d + 1)$ time to test clear deployment condition for each of them. In total, OPTDEPLOYMENT($\mathcal{T}^{i-1}(v_i)$) returns within time $O((d\lceil 1/\epsilon \rceil + d + 1)n^{d\lceil 1/\epsilon \rceil + d + 1})$. $\square$

Lemma 4. *$\mathcal{A}_{\mathrm{SMDP}}$ is a polynomial-time algorithm.*

Proof. Running time of each iteration is dominated by OPT-DEPLOYMENT($\cdot$). Combining Lemma 3, $\mathcal{A}_{\mathrm{SMDP}}$ runs in $O((d\lceil 1/\epsilon \rceil + d + 1)n^{d\lceil 1/\epsilon \rceil + d + 2})$ time, which is polynomial in n for $\epsilon > 0$. $\square$

Theorem 5. *For any fixed $\epsilon > 0$, $\mathcal{A}_{\mathrm{SMDP}}$ is a polynomial-time algorithm that outputs an $(1 + \epsilon)$-optimal solution.*

Proof. Correctness, approximation ratio and efficiency follow from Lemma 1, Lemma 2 and Lemma 4 respectively. $\square$

The output of $\mathcal{A}_{\mathrm{SMDP}}$ goes arbitrarily close to optimal by shrinking ϵ, but the running time goes up rapidly at the same time. In our later evaluation, we set $\epsilon = 1$, a balance in approximation ratio and running time.

5 Evaluation

We carry out experiments using simulated and real data to evaluate the effectiveness of clearness function and $\mathcal{A}_{\mathrm{SMDP}}$'s performance, through comparison with three baselines:

Dense deployment deploy meters at all appliances

Random deployment deploy meters uniformly randomly

Following the electrons (FE) a greedy algorithm proposed in [Bellala *et al.*, 2012] to maximize information acquired with a given number of meters.

5.1 Experiments on PowerNet Dataset

The first experiments are conducted on real-world data from PowerNet [2]. PowerNet provides per-device energy and usage statistics of an office building in Stanford University. More specifically, we use data collected from 126 different appliances in Sept. 2011. Sampling rate is basically 1Hz. To neutralize noise, each appliance's consumption is smoothened by taking the average consumption of intervals of 30 seconds as data points. Fig. 3 shows the real-time power measurements of three appliances on Sept. 25, 2011.

Power Pattern Analysis

We first investigate the *concentration property* of each appliance's power consumption. An appliance is said having power consumption *concentrated* on p_m at level θ, if there exists a p_m, such that 95% of the readings of the appliance's consumption are in the interval $[p_m(1 - \theta), p_m(1 + \theta)]$. In our data set, 73%, 77% and 89% of all the appliances have concentrated power consumption at levels 3%, 5%, and 10% accordingly. This justifies the *interval power pattern* model. Further, Fig. 4 shows the histogram of the concentrated powers of all the appliances at level 5%. One may observe that the majority of the appliances have concentrated power consumption less than 50W; the distribution is fairly flat in the interval between 50W and 250W. 2 appliances with power consumption far more than 500W are omitted in the figure.

Cost-saving Performance

Since PowerNet does not provide the PLT explicitly, we simulate the power network with randomly generated PLTs and put 126 appliances at leave nodes. We evaluate the cost-saving performance of $\mathcal{A}_{\mathrm{SMDP}}$ on random PLTs with average degrees of 2, 4, 6 and 8. We define *cost ratio* as the number of deployed meters divided by the number of appliances. Intuitively, a smaller cost ratio is preferable. As an extreme example, in a dense deployment where every appliance is monitored by a meter, the cost ratio is always 1.

Cost ratio for each generated PLT is calculated and depicted in Fig. 5. It can be observed that the number of meters

---

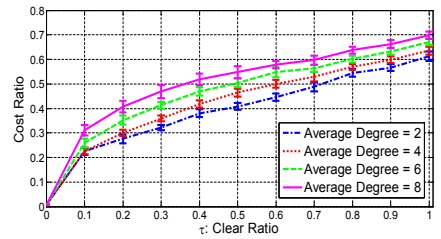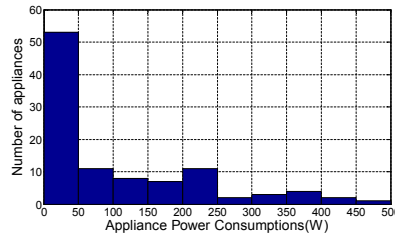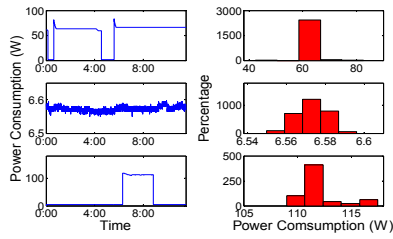[2]PowerNet: http://powernet.stanford.edu/

Figure 3: Real-time Power Consumption   Figure 4: Appliance power distribution   Figure 5: Cost-saving on PowerNet data

Table 1: Monitoring Accuracy and Clear Ratio

Clear Ratio	1	0.9	0.8	0.7	0.6
Accuracy(%)	99.5	97.1	96.5	92.4	87.4
Clear Ratio	0.5	0.4	0.3	0.2	0.1
Accuracy(%)	77.6	74.8	52.7	31.8	10.4

needed and the topology of the underlying PLT are closely related. Larger average degrees generally imply more meters. Moreover, the number of required meters increases with the clear ratio τ. For each value of τ, we run $\mathcal{A}_{\mathrm{SMDP}}$ on 20 randomly generated trees. The error bar is the standard deviation of 20 runs. As shown in Fig. 5, even in the worst case, the cost ratio of $\mathcal{A}_{\mathrm{SMDP}}$ is about 0.7, outperforming dense deployment (cost ration = 1) by 30%.

Monitoring Accuracy Performance

To evaluate state monitoring accuracy, we run Viterbi-based state decoding in each meter's subtree based on [Wang *et al.*, 2012]. State decoding experiments are conducted on randomly generated PLTs with average degrees 4 and 6. Table 1 presents the relation between state decoding accuracy and clear ratio, which are, as expected, positively correlated. This result justifies clear ratio as a reasonable metric for state decoding during meter deployment.

5.2 Simulations

There are only limited types of appliance power patterns in PowerNet. For comprehensiveness, we evaluate $\mathcal{A}_{\mathrm{SMDP}}$'s performance on simulated data with diverse power patterns.

Impact of Power Consumption Distribution

We generate datasets from three different power patterns: uniform distribution $\mathcal{U}(0, 1000)$, normal distribution $\mathcal{N}(500, 167^2)$ and exponential distribution with $\lambda = 0.002$. Here we assume all appliances have concentrated power with $\theta = 0$. Fig. 6 shows the cost-saving performance for clear ratio $\tau = 1$. For different sizes and structures of PLTs, $\mathcal{A}_{\mathrm{SMDP}}$ always achieves cost ratios less than 0.3. Cost ratios for exponential and uniform power distributions are close. In contrast, the cost ratio for normal distribution is significantly higher. The intuition behind is that, the power levels of appliances are more concentrated in the normal distribution, thus more smart meters are required to disambiguate the states of those appliances with similar power patterns.
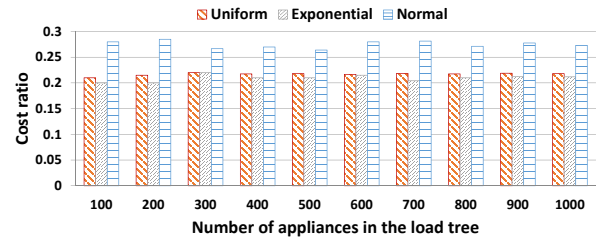

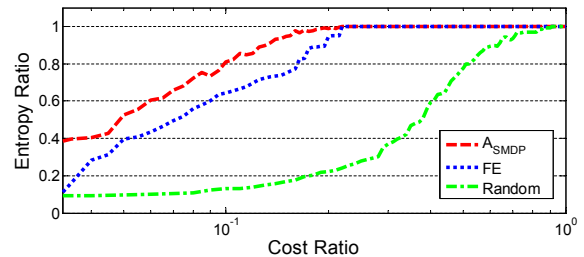Figure 6: Cost-saving in different power distributions

Figure 7: Comparison of $\mathcal{A}_{\mathrm{SMDP}}$ and random deployment

Comparison of Accuracy

We further compare $\mathcal{A}_{\mathrm{SMDP}}$ with random meter deployment and FE in terms of accuracy aspect. Distinct from previous evaluations, here distributional power pattern model is used. *Entropy ratio* is defined as $r_e = \min_{v \in D} f_c(v)$, which works as a metric for monitoring accuracy. As shown in Table 1, a larger r_e indicates a higher monitoring accuracy at the worst monitored appliance, and hence a better monitoring performance. We compare entropy ratios of different deployments at the same deployment cost.

The simulation is conducted on a PLT with 200 nodes, including about 120 appliances. Power consumption of each appliance is uniformly randomly sampled from a distribution shaped resembling the histogram in Fig. 4 with mean 50 watts. The results are summarized in Fig. 7. The entropy ratio increases as the number of meters increases for all three algorithms. Both $\mathcal{A}_{\mathrm{SMDP}}$ and FE beat random deployment easily. $\mathcal{A}_{\mathrm{SMDP}}$ always outperforms FE when cost ratio is < 0.2. For example, at entropy ratio $r_e = 0.8$, $\mathcal{A}_{\mathrm{SMDP}}$ costs 40% less than FE. As cost ratio approaches 0.2, both $\mathcal{A}_{\mathrm{SMDP}}$ and FE already achieves entropy ratio 1, which is the theoretical limit. This is also achievable by a random or dense deployment, but at a much higher cost.

6 Conclusion and Future Work

This paper presented an extensible framework to investigate the tradeoff between meter deployment cost and appliance state monitoring accuracy. We answered two questions: 1) how to evaluate the quality of a deployment; and 2) how to find a good deployment. In particular, we proposed clear deployment condition to characterize when a deployment satisfies the monitoring accuracy requirements. Based on it, an efficient and effective deployment optimization algorithm was developed. The validity and effectiveness of our approach was then verified both theoretically and experimentally. There are multiple ways that our work can be extended. For example, our framework could be applied in monitoring appliances with multi-mode, dynamic power patterns. Another possible direction to further reduce the number of smart meters is by identifying and tracking only major energy consumers.

References

[Bellala *et al.*, 2012] Gowtham Bellala, Manish Marwah, Martin Arlitt, Geoff Lyon, and Cullen Bash. Following the electrons: methods for power management in commercial buildings. In *SIGKDD'12*, pages 994–1002. ACM, 2012.

[Berges *et al.*, 2009] Mario Berges, Ethan Goldman, H Scott Matthews, and Lucio Soibelman. Learning systems for electric consumption of buildings. In *ASCI international workshop on computing in civil engineering*, 2009.

[Călinescu *et al.*, 2003] Gruia Călinescu, Cristina G Fernandes, and Bruce Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. *Journal of Algorithms*, 48(2):333–359, 2003.

[Farinaccio and Zmeureanu, 1999] Linda Farinaccio and Radu Zmeureanu. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy and Buildings*, 30(3):245–259, 1999.

[Guestrin *et al.*, 2005] Carlos Guestrin, Andreas Krause, and Ajit Paul Singh. Near-optimal sensor placements in gaussian processes. In *ICML '05*, pages 265–272. ACM, 2005.

[Hao *et al.*, 2012] Xiaohong Hao, Yongcai Wang, Chenye Wu, Amy Yuexuan Wang, Lei Song, Changjian Hu, and Lu Yu. Smart meter deployment optimization for efficient electrical appliance state monitoring. In *SmartGridComm'12*, pages 25–30. IEEE, 2012.

[Hart, 1992] George William Hart. Nonintrusive appliance load monitoring. *Proceedings of the IEEE*, 80(12):1870–1891, 1992.

[Jiang *et al.*, 2009] Xiaofan Jiang, Minh Van Ly, Jay Taneja, Prabal Dutta, and David Culler. Experiences with a high-fidelity wireless building energy auditing network. In *SenSys'09*, pages 113–126. ACM, 2009.

[Jung and Savvides, 2010] Deokwoo Jung and Andreas Savvides. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *Sensys'10*, pages 225–238. ACM, 2010.

[Kolter and Jaakkola, 2012] J. Zico Kolter and Tommi Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *International Conference on Artificial Intelligence and Statistics*, pages 1472–1482, 2012.

[Laughman *et al.*, 2003] Christopher Laughman, Kwangduk Lee, Robert Cox, Steven Shaw, Steven Leeb, Les Norford, and Peter Armstrong. Power signature analysis. *Power and Energy Magazine, IEEE*, 1(2):56–63, 2003.

[Lee *et al.*, 2005] Kwangduk D. Lee, Steven B. Leeb, Leslie K. Norford, Peter R. Armstrong, Jack Holloway, and Steven R. Shaw. Estimation of variable-speed-drive power consumption from harmonic content. *Energy Conversion, IEEE Transactions on*, 20(3):566–574, 2005.

[Leeb *et al.*, 1995] Steven B. Leeb, Steven R. Shaw, and James L. Kirtley Jr. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *Power Delivery, IEEE Transactions on*, 10(3):1200–1210, 1995.

[Marceau and Zmeureanu, 2000] Medgar Louis Marceau and R. Zmeureanu. Nonintrusive load disaggregation computer program to estimate the energy consumption of major end uses in residential buildings. *Energy Conversion and Management*, 41(13):1389–1403, 2000.

[Parson *et al.*, 2012] Oliver Parson, Siddhartha Ghosh, Mark Weal, and Alex Rogers. Non-intrusive load monitoring using prior models of general appliance types. In *AAAI '12*, 2012.

[Patel *et al.*, 2007] Shwetak N. Patel, Thomas Robertson, Julie A. Kientz, Matthew S Reynolds, and Gregory D. Abowd. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *UbiComp'07*, pages 271–288. Springer, 2007.

[Roos *et al.*, 1994] J. G. Roos, I. E. Lane, E. C. Botha, and G. P. Hancke. Using neural networks for non-intrusive monitoring of industrial electrical loads. In *IMTC'94*, pages 1115–1118. IEEE, 1994.

[Shao *et al.*, 2013] Huijuan Shao, Manish Marwah, and Naren Ramakrishnan. A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings. In *AAAI '13*, 2013.

[Sultanem, 1991] F Sultanem. Using appliance signatures for monitoring residential loads at meter panel level. *Power Delivery, IEEE Transactions on*, 6(4):1380–1385, 1991.

[Wang *et al.*, 2012] Yongcai Wang, Xiaohong Hao, Lei Song, Chenye Wu, Yuexuan Wang, Changjian Hu, and Lu Yu. Tracking states of massive electrical appliances by lightweight metering and sequence decoding. In *SensorKDD '12*, pages 34–42. ACM, 2012.

[Wang *et al.*, 2014] Yongcai Wang, Xiaohong Hao, Lei Song, Chenye Wu, Yuexuan Wang, Changjian Hu, and Lu Yu. Monitoring massive appliances by a minimal number of smart meters. *ACM Transactions on Embedded Computing Systems (TECS)*, 13(2s):56, 2014.