Monitoring Massive Appliances by a Minimal Number of Smart Meters

YONGCAI WANG, XIAOHONG HAO, LEI SONG, CHENYE WU, and YUEXUAN WANG, Tsinghua University
CHANGJIAN HU and LU YU, NEC Labs.

This article presents a framework for deploying a minimal number of smart meters to accurately track the ON/OFF states of a massive number of electrical appliances which exploits the sparseness feature of simultaneous ON/OFF switching events of the massive appliances. A theoretical bound on the least number of required smart meters is studied by an entropy-based approach, which qualifies the impact of meter deployment strategies to the state tracking accuracy. It motivates a meter deployment optimization algorithm (MDOP) to minimize the number of meters while satisfying given requirements to state tracking accuracy. To accurately decode the real-time ON/OFF states of appliances by the readings of meters, a fast state decoding (FSD) algorithm based on the hidden Markov model (HMM) is presented to track the state sequence of each appliance for better accuracy. Although traditional HMM needs $O(t2^{2N})$ time complexity to conduct online sequence decoding, FSD improves the complexity to $O(tn^{U+1})$, where $n < N$ and U is an upper bound of the simultaneous switching events. Both MDOP and FSD are verified extensively using simulations and real PowerNet data. The results show that the meter deployment cost can be saved by more than 80% while still getting over 90% state tracking accuracy.

Categories and Subject Descriptors: C.3 [**Computer Systems Organization**]: Special Purpose and Application-Based Systems—*Real-time and embedded systems*

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Energy auditing, smart meter, deployment optimization, state tracking, compressive sensing, energy saving, smart building

ACM Reference Format:
Yongcai Wang, Xiaohong Hao, Lei Song, Chenye Wu, Yuexuan Wang, Changjian Hu, and Lu Yu. 2014. Monitoring massive appliances by a minimal number of smart meters. ACM Trans. Embedd. Comput. Syst. 13, 2s, Article 56 (January 2014), 20 pages.
DOI: http://dx.doi.org/10.1145/2544375.2544376

1. INTRODUCTION

A recent survey shows that in our offices, up to 70% of computers and related equipment are left on all the time [Oxford St. Hughs College Data 2010]. To reduce the energy waste caused by such idle running, the real-time ON/OFF states of the electrical devices are required as necessary information for smart control technologies. The ON/OFF

This work was supported by the National Basic Research Program of China, Grants 2011CBA00300 and 2011CBA00301, and by the National Natural Science Foundation of China, Grants 61202360, 61033001, 61061130540, and 61073174.
Authors' addresses: Y. Wang (corresponding author), X. Hao, L. Song, C. Wu, and Y. Wang, Institute for Theoretical Computer Science, Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing 100084, P. R. China; C. Hu and L. Yu, NEC Labs, Beijing 100084, P. R. China; corresponding author's email: wangyc@tsinghua.edu.cn.

DOI: http://dx.doi.org/10.1145/2544375.2544376

state information is also an important reference for energy auditing, because the *power pattern* of an appliance (i.e., how much energy an appliance consumes when it is on) can be learned offline [Norford and Leeb 1996; Leeb et al. 1995], and therefore its energy consumption can be inferred from its ON/OFF durations.

However, because the electrical appliances are now massive and widely distributed in buildings, tracking their ON/OFF states is a challenging problem. Current state tracking technologies are generally facing a difficult choice between reducing the meter deployment cost and pursuing good state tracking accuracy. Dense smart meter deployment will provide accurate state monitoring but suffers high meter deployment, maintenance, and data collection costs. Deploying small numbers of meters generally suffers state tracking inaccuracy and can monitor only limited numbers of appliances. Real applications highly desire a low-cost, efficient, and scalable method for tracking the states of massive electrical appliances.

To address these challenges, this article proposes a framework for tracking the ON/OFF states of N appliances by deploying $m \ll N$ smart meters on the load tree. The load tree is the tree-structure power distribution network in a building. The root of the load tree is the building power entrance. All the electrical appliances are the leaf nodes, and the intermediate nodes of the tree are the power switches or outlets.

The framework is based on the idea of compressive sensing (CS) in which m low-cost, embedded, smart AC meters are deployed at optimally selected positions on the load tree. Each meter monitors the aggregated power consumption of the electrical appliances in the subtree rooted at itself. The readings of all the meters are collected in real time to a central server. The central server runs a state decoding algorithm to utilize both the tree structure and the temporal state correlation feature to infer the real-time ON/OFF states of N electrical appliances.

To minimize the meter deployment cost while preserving good state tracking accuracy, the critical problems in such a CS framework are as follows.

(1) Where should we deploy the limited number of smart meters and how many smart meters are enough for reaching a required state tracking accuracy?
(2) How do we efficiently and accurately decode the states of N appliances by the continuous readings from only m meters?

1.1. Our Contributions

A novel observation of this work is that the ON/OFF states of an appliance are highly correlated in the time domain. When an appliance is turned on, it generally works a long time before it is turned off, and an appliance must be turned off (on) before it can be turned on (off), So for N appliances, their ON/OFF switching events are quite sparse during a short observation period (e.g., a second), and the ON/OFF events of any appliance must happen in turn. We call these two features *switching sparseness* and *sequence feasibility constraint*. A differential matrix is therefore presented to compare state differences between two adjacent observations which converts the problem of state tracking to a sparse event detection problem. It enables the compressive sensing framework.

In the CS framework, we map the observation matrix construction problem to the meter deployment problem on the load tree. A desired meter deployment scenario should not only minimize the meter deployment cost but also guarantee good state tracking accuracy. We propose an entropy-based approach to qualify the impact of meter deployment to the state decoding accuracy. Based on it, a meter deployment optimization algorithm (MDOP) is proposed to minimize the number of meters and to optimize the deployment positions of the meters. MDOP gives a near-optimal, adjustable deployment strategy for any given requirement to the state tracking accuracy.

To accurately decode the real-time states of N appliances by the readings of m meters, we propose a hidden Markov model (HMM)-based sequence decoding algorithm for better accuracy. Although a traditional Viterbi algorithm needs $O(t2^{2N})$ time complexity to decode the state sequence of HMM, which is impractical when N is large, a fast sequence decoding (FSD) algorithm is proposed. FSD exploits the idea of offline load tree splitting, state vector pre-ordering, and complexity-bounded online forward and backward search. It reduces the online decoding time complexity to be polynomial, that is, $O(n^{U_t+1})$, where $n < N$ and U_t is the upper bound of simultaneous switching events within a sampling slot. FSD enables efficient and accurate online state tracking for massive appliances.

Extensive evaluations based on the simulated data and the real PowerNet data show that the meter deployment cost can be saved by more than 80% while still getting more than 90% state tracking accuracy.

1.2. Related Work

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

(1) *Bottom-Up Monitoring Approach.* The first category focuses on designing smart meter networks for detailed energy monitoring. An early work is the MIT Plug system [Lifton et al. 2007], where the design and development of the smart metering system were reported with a trial deployment of 35 smart meters on a floor of a building. Jiang et al. [2009a] reported the design and development of the Berkeley AC meter network exploiting the idea of Web of Things. The same authors reported utilizing contextual metadata for the high-fidelity monitoring and spatial, functional, and individual decomposition of electric usage in buildings [Jiang et al. 2009b]. A recent work [Dawson-Haggerty et al. 2012] shared insights obtained from a year-long, 455 meter deployment of wireless plug-load electric meters in a large commercial building. Kazandjieva et al. [2009], introduced PowerNet, which was a hybrid sensor network for monitoring the power and utilization of computing systems in a large academic building. Jung and Savvides [2010] proposed energy breakdown research considering minimizing meter deployment cost, but they assume that the appliance's ON/OFF states are sensed by additional RFID sensors. There are also solutions from the industry, such as Tendril [2012], GreenBox [2009], and Energy Hub [2009]. In contrast to existing work, this article studies, for the first time, the impact of the meter deployment strategy on the state decoding accuracy. By the CS framework, the meter deployment costs are saved remarkably while guaranteeing highly accurate state tracking by sequential decoding.

(2) *Top-Down Disaggregation Approach.* The second category focuses on the nonintrusive load monitoring (NILM) for ON/OFF state disaggregation. In particular, the NILM-based method can efficiently reduce the deployment cost of smart metering, 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. The first NILM approach was proposed by Hart [1992], which used real and reactive power measurements to detect the specific load signatures of individual appliances. Norford and Leeb [1996] and Leeb et al. [1995] proposed transient event detection methods to analyze the specific patterns in the spectral domains. Patel et al. [2007] tried to recognize the electrical noise on the residential power lines to detect the ON/OFF switching events. Farinaccio and Zmeureanu [1999] proposed a method to disaggregate the total electricity consumption into the major end-uses by pattern recognition. However, the

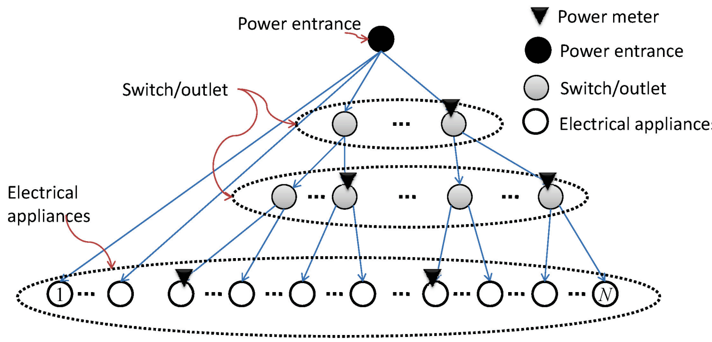

Fig. 1. Sparse meter deployment scenario on the load tree.

NILM-based approach generally needs the high frequency meter to capture the transient patterns, and the state monitoring accuracy decreases quickly with the number of monitored appliances. The reason being n appliances having 2^n combinatorial states. Ambiguity (i.e., states having similar power patterns) will become serious when n becomes large. To tackle this challenge, the CS-frame work in this article monitors massive appliances via smart meter deployment optimization and sequential decoding.

(3) *On/Off Detection by Additional Sensors.* Instead of utilizing smart meters, the final body addressed the ON/OFF state monitoring with other types of sensors. For example, Kim et al. [2009] developed the ViridiScope system, which detected the ambient signals emitted from appliances to infer the power consumption of appliances. Gupta et al. [2010] proposed ElectriSense, which sensed EMI (electromagnetic interference) by a single point sensing for electrical event detection and classification in the home. Rowe et al. [2010] used contactless sensing to monitor the variations in electromagnetic fields. Taysi et al. [2010] proposed Tinyears to utilize audio sensor nodes. The method in this article uses low-cost smart meters.

The rest of this article is organized as follows. We introduce the system model in Section 2. The HMM-based efficient sequential decoding method is introduced in Section 3. A deployment optimization algorithm is presented in Section 4. Evaluation results will be presented in Section 5, and the article is concluded with discussion of future work in Section 6.

2. SYSTEM MODEL

The energy distribution network in a building has a typical tree-like structure [Jiang et al. 2009b]. The root of the load tree is the main power entrance of the building; each node in the middle tier is a power break or an outlet, and the leaf nodes on the tree are the electrical appliances. In the load tree, the power consumption at a node equals the sum of the power consumptions of the appliances in the subtree rooted at the node. Smart meters can be deployed at any node on the tree.

2.1. Observation Model

m meters are deployed in the load tree to track the ON/OFF states of N appliances. We can imagine the meter deployment scenario as shown in Figure 1. Each meter measures the real-time aggregated power of the appliances in the subtree rooted at itself. We assume all the meters are synchronized. At time t, the observation model of a meter i can be formulated by

$$\mathbf{Z}_{t,i} = \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j}, \tag{1}$$

where $S(i)$ is the subtree rooted at meter i, $\mathbf{X}_{t,j} \in [0, 1]$ is the state of appliance j at time t, and $\mathbf{P}_{t,j}$ is the real-time power consumption of appliance j at time t. The goal of the state decoding at meter i is to find the state vector of the appliances $\{\mathbf{X}_{t,j}, j \in S(i)\}$ to minimize the following expected square error:

$$\underset{\mathbf{X}}{\textbf{minimize:}}\ \mathbb{E}\left\{\left(\mathbf{Z}_{t,i} - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathbf{P}_{t,j}\right)^2\right\}. \tag{2}$$

By assuming $\mathbf{P}_{t,j}$ is a random variable with mean $\mathscr{P}_j$ and variance $\delta_j = \alpha_i \mathscr{P}_j$, we prove in the Appendix that the solution of Problem (2) is approximately the same as the following:

$$\underset{\mathbf{X}}{\textbf{minimize:}}\ \left(\mathbf{Z}_{t,i} - \alpha/2 - \sum_{j \in S(i)} \mathbf{X}_{t,j} \mathscr{P}_j\right)^2, \tag{3}$$

where α is defined as the expectation of the variance/mean ratio of the appliances' power consumption patterns, that is, $\alpha = \mathbb{E}\{\delta_j / \mathscr{P}_j\}$.

2.2. Compressive Sensing Model

Based on Eq. (3), we denote $\mathbf{Y}_{t,i} = \mathbf{Z}_{t,i} - \alpha/2$ and consider the problem of decoding the states of N appliances based on the readings of m meters during periods 1 to t.

Let matrix $\mathbf{Y} \in \mathbb{R}^{m \times t} = \{\mathbf{Y}_{i,j}, i = 1, 2 \ldots, m, j = 1, 2, \ldots, t\}$ be the readings of m meters from period 1 to t. Let matrix $\mathbf{X} \in (0, 1)^{N \times t} = \{\mathbf{X}_{i,j}, i = 1, 2, \ldots, N, j = 1, 2, \ldots, t\}$ be the states of N appliances from period 1 to t. Let $\mathbf{P} \in \mathbb{R}^{m \times N} = \{\mathbf{P}_{i,j}, i = 1, 2, \ldots, m, j = 1, 2, \ldots, N\}$ be the *pattern matrix*. The pattern matrix can be constructed from a given meter deployment scenario on the load tree, where item $\mathbf{P}_{i,j}$ is determined by judging whether appliance j is in the subtree of a meter i as follows.

(1) $\mathbf{P}_{i,j} = \mathscr{P}_j$, if appliance d_j is in the subtree of meter i; otherwise $\mathbf{P}_{i,j} = 0$.
(2) The ith row of $\mathbf{P}$ indicates the ith meter.

Based on these notations, we can build a model to describe the relationship between the states of N appliances and the readings of m meters from period 1 to t.

$$\mathbf{Y} = \mathbf{P}\mathbf{X}. \tag{4}$$

Note that the appliances switch their states infrequently, which means that the state switching events of an appliance from period 1 to t are sparse. This suggests that the state differences could be sparsely represented if we consider the difference between two adjacent sample values. This motivates us to design the following difference matrix $D \in \mathbb{R}^{t \times t}$:

$$D = \begin{bmatrix} 1 & -1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & -1 & \ldots & 0 & 0 \\ 0 & 0 & 1 & \ldots & 0 & 0 \\ \vdots & \ldots & \ldots & \ldots & \ldots & \vdots \\ 0 & 0 & 0 & \ldots & 1 & -1 \\ 0 & 0 & 0 & \ldots & 0 & r \end{bmatrix}, \tag{5}$$

where the last element $0 < r \leq 1$ ensures that D is invertible.

Let $\mathbf{\Delta}$ be the projection of $\mathbf{X}$ on D, that is, $\mathbf{\Delta} = \mathbf{X}D$. Then, $\mathbf{\Delta} \in (0, 1, -1)^{N\times t}$ is a matrix indicating the state switching events of N appliances from period 1 to t, which is a sparse matrix containing many zero entries. Particularly, $\mathbf{\Delta}_{i,j} = 0$ means that appliance i does not change state at time j; $\mathbf{\Delta}_{i,j} = 1$, and $\mathbf{\Delta}_{i,j} = -1$ means that the ith appliance is switched on and off at time j, respectively.

Let $\mathbf{\Gamma} \in \mathbb{R}^{m\times t}$ be the projection of $\mathbf{Y}$ on D, $\mathbf{\Gamma} = \mathbf{Y}D$. Then $\mathbf{\Gamma}_{i,j} = \mathbf{Y}_{i,j} - \mathbf{Y}_{i,j-1}$ is the measurement difference of meter i at time j. Thus, the model in Eq. (4) can be converted to a new model of sparse switching event tracking:

$$\mathbf{\Gamma} = \mathbf{P}\mathbf{\Delta}. \tag{6}$$

Let $\mathbf{\Delta}_t$ be the vector to indicate the state switching events of N appliances at period t. Then, the sparseness of $\mathbf{\Delta}_t$ is related to the length of the sampling interval and is also related to time t, because the switching events occur more often in the morning and evening but seldom at night. Therefore, we assume that by choosing a short observation interval at different time, the sparseness of $\mathbf{\Delta}_t$ is bounded by U_t, that is, $\|\mathbf{\Delta}_t\|_1 \leq U_t$, where U_t is a small integer and $U_t \ll N$.

Then the problem of decoding $\mathbf{\Delta}$ by the meter's differential measurements $\mathbf{\Gamma}$ can be solved by a least-squares estimation (LSE) method with an L1-Norm constraint, which is also known as the constraint type LASSO (Least Absolute Selection and Shrinkage Operator) problem [Tibshirani 1994].

$$\begin{aligned} \textbf{minimize:} \quad & \|\mathbf{P}\mathbf{\Delta} - \mathbf{\Gamma}\|_2\,, \\ \textbf{subject to:} \quad & \mathbf{1}.\ \forall t, \|\mathbf{\Delta}_t\|_1 < U_t, \\ & \mathbf{2}.\ \forall i, \forall t, \mathbf{\Delta}_{i,t} \in [0, 1, -1]. \end{aligned} \tag{7}$$

Note that $\mathbf{\Delta} \in (0, 1, -1)^{N\times t}$ and $\mathbf{\Gamma} \in \mathbb{R}^{m\times t}$ contain all the state switching events and meters' measurement difference vectors from time 1 to t. The decoding complexity is high when N is large. There are two key challenges in the CS framework at Eq. (7):

(1) *How to construct the pattern matrix* $\boldsymbol{P}$. Because $\mathbf{P}$ is determined by the meter deployment scenario, the construction of $\mathbf{P}$ is mapped to the smart meter deployment optimization problem, that is, how and where to deploy the minimal number of smart meters while guaranteeing good state tracking accuracy. This problem is addressed by an entropy-based approach in Section 3, which gives the necessary conditions for meter deployment and a meter deployment optimization algorithm.

(2) *How to efficiently decode* $\mathbf{\Delta}$ *from Eq. (7).* Since we care only about $\mathbf{\Delta}_t$ at time t, a trick is to decode $\mathbf{\Delta}_t$ sequentially. Because the state switching events have Markov property, we can utilize all the historical observations from period 1 to t and the temporal sparseness of the switching events to jointly infer the state changing events of N appliances at time t, which is called *sequential decoding*. A fast sequential decoding algorithm based on the hidden Markov model is presented in Section 4, which is a polynomial-time algorithm.

3. METER DEPLOYMENT OPTIMIZATION

As previously mentioned, one of the most important tasks in Problem (7) is the observation matrix construction, which is mapped to the power meter deployment optimization problem. It not only dominates the system deployment, maintenance, and data collection costs, but also affects the state decoding complexity and accuracy. Users generally hope to place the least number of meters to get enough information for decoding the ON/OFF states of the appliances.

3.1. Problem Definition

We formally define the Meter Deployment Optimization Problem (MDOP) as follows.

Problem 1 (*MDOP Problem*). Given a load tree $T = (V, E)$ with N nodes, let $L \subseteq V$ be the set of leaves in T. Each leaf $l_i \in L$ has a power pattern $\mathscr{P}_i$ on it. A subtree $ST(v) = (V(v), E(v))$ denotes the subtree of T with node $v \in V$ as its root. $V(v)$ and $E(v)$ denotes the set of nodes and edges in the subtree $ST(v)$, respectively. A binary $\mathbf{X}_{i,t} \in \{0, 1\}$ is assigned to each leaf $l_i \in L$, indicating the ON/OFF state of the appliance at time t. If a smart meter is deployed at node v, it can measure the total power consumed by its subtree $ST(v)$, that is, measure $\sum_{i \in ST(v)} \mathbf{X}_{i,t} \mathscr{P}_i$. The goal of smart meter deployment optimization is to minimize the number of deployed meters while still getting enough information to know the value of each $\mathbf{X}_{i,t}$.

The MDOP problem is investigated from three aspects. (i) Entropy-based analysis is proposed to investigate how the meter deployment scenario affects the state tracking accuracy. (ii) A concept, *clear ratio*, that is, $r(D, T)$ is presented to qualify the goodness of a deployment scenario for the state decoding accuracy. (iii) By utilizing the degree bound and total power consumption bound of the load tree, an approximation algorithm is presented to optimize the meter deployment while satisfying a given clear ratio requirement.

3.2. The Impact of Meter Deployment to State Decoding Accuracy

Understanding how the deployment scenario will affect the state tracking accuracy is a fundamental problem in the proposed compressive state tracking framework. We exploit information entropy to answer this problem. In information theory, entropy (or Shannon's entropy) is a measurement of the uncertainty associated with a random variable. In our problem, the sum function of the energy consumptions could be treated as a way of compressing the binary state information $\mathbf{X}_{i,t}$. The states of the appliances can be correctly decoded only if the entropy (information) provided by the m meters is not less than the entropy of N appliances' state changing events. Therefore, we compare the entropy obtained from the meters with the entropy of the appliances' states.

3.2.1. Transform Load Tree to Monometer Tree Forest. An important fact for simplifying the entropy calculation is that the load tree T can be split into m monometer trees when m meters are deployed on T. A monometer tree is a tree with depth one, which has only the root equipped by a meter while all the leaves are not.

Given T and D, 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 so that all the parents of v can know the power consumptions of its subtree. Thus, the subtree $ST(v)$ can be split from the full tree T. By running a breadth-first-search-based tree splitting algorithm [Wang et al. 2012], the load tree T can be transformed into forest F containing m monometer trees.

Figure 2(a) shows an example of a load tree monitored by four meters. The leaf nodes stand for appliances, while the others stand for outlets. The nodes equipped with a smart meter are represented in black, and the nodes without smart meters are in white. Figure 2(b) shows the forest of four monometer trees which are transformed from Figure 2(a).

3.2.2. The Entropy Captured by the Meter in a Monometer Tree. By splitting the load tree into m monometer trees, the entropy provided by a meter in one monometer tree can be evaluated explicitly. Let n be the number of appliances in a monometer tree. There will be 2^n combinatorial states, which may project to $w \leq 2^n$ distinct aggregated power values $[a_1, a_2 \ldots a_w]$ to be measured by the meter. Note that some states may

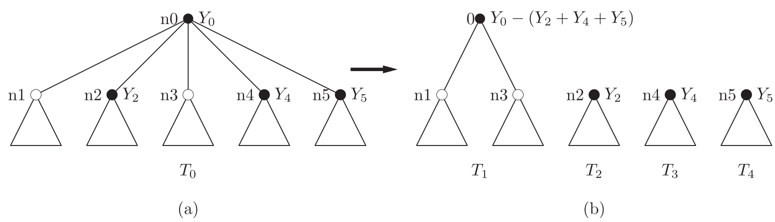

Fig. 2. Load-tree splitting example.

project to the same aggregated power value at the root, which are called ambiguous states. Consider the ith distinct aggregated power value a_i measured at the root: it may correspond to m_i number of ambiguous states of the leaves. Let set S_i be any subset of appliances such that $\sum_{j\in S_i} \mathscr{P}_j = a_i$. Let d_v be a measurement at the smart meter, since all the appliances take measurements in i.i.d. (independent and identically distributed), the probability that the meter measures a value a_i is

$$Prob(d_v = a_i) = \sum_{S_i} \prod_{j\in S_i, k\in S\setminus S_i} (1 - q_j) q_k. \tag{8}$$

Thus, the entropy of the power measurements at the smart meter is

$$H_d(v) = -\sum_{i=1}^{w} Prob(d_v = a_i) \log(Prob(d_v = a_i)). \tag{9}$$

3.2.3. The Entropy of the Appliance States in a Monometer Tree. Correspondingly, let q_i denote the probability of appliance $i \in ST(v)$ staying in state 0 at time t, that is, $q_i = Prob(\mathbf{X}_{i,t} = 0)$, the entropy of the appliance states in a monometer tree $ST(v)$ can be evaluated as follows.

$$H_s(v) = -\sum_{i\in ST(v)} q_i \log q_i. \tag{10}$$

If $H_d(v) = H_s(v)$, the meter provides lossless measurements, which means error-free decoding is possible in the monometer tree. By checking this condition in all the monometer trees, we can judge whether a deployment D can provide lossless monitoring.

Definition 3.1 (*Lossless Monitoring*). A meter deployment scenario D splits the load tree T into a monometer tree forest F. D is called lossless monitoring if for each monometer tree $ST(v) \in F$, the entropy measured by the meter ($H_d(v)$) is equal to the entropy of the appliance states ($H_s(v)$).

3.3. Evaluate the Goodness of Deployment by Clear Ratio

It is generally very expensive to guarantee that all the monometer trees be lossless, because the lossless monitoring condition requires all the 2^n states to be distinctive. As a result, each monometer tree has only a small number of appliances; the number of monometer trees, that is, the number of meters m therefore becomes large.

In state decoding, for the sparseness of state-switching events, even if there are ambiguous states in a monometer tree, the real-time state can be inferred correctly by exploiting the temporal correlation feature. Therefore, the combinatorial states in a monometer tree are not necessary to be all distinctive.

We call the monometer tree $ST(v)$ *blurry* if $H_d(v) < H_s(v)$. Otherwise, if all the states are distinctive, we call $ST(v)$ *clear*. From information theory, we know that the less information we get, the harder it is to recover the original state vector. Therefore, we define the clear ratio to evaluate the goodness of the meter deployment scenario D.

Definition 3.2 (*Clear Ratio*). We define the clear ratio $r(D, T)$ for a deployment D on load tree T as the minimal $\frac{H_d(v)}{H_s(v)}$ ratio among all the subtrees rooted at $v \in D$.

$$r(D, T) = \min_{v \in D} \frac{H_d(v)}{H_s(v)}. \tag{11}$$

The clear ratio proposes a measurement to the meter deployment scenario. The state combinations of appliances can be disaggregated without error when $r(D, T) = 1$ and decoding ambiguities increase as $r(D, T)$ decreases. Given a constant factor $r(D, T)$ as the threshold of the clear ratio to guarantee the decoding performance, the deployment optimization problem is to find an optimal deployment D over T to maximize $r(D, T)$ while minimizing the number of deployed meters.

3.4. MDOP Algorithm for the Bounded Trees with Given Clear Ratio

We have proved in a previous work that the problem of finding an optimal deployment D over T while satisfying clear ratio $r(D, T)$ is NP-complete by a polynomial time reduction from the 3-SAT problem [Hao et al. 2012a]. Therefore, finding an efficient algorithm that outputs the optimal solution is hard. But in practice, the degrees of the power load tree are usually small, and the total power consumptions of the tree are also bounded. Thus we can still design an efficient MDOP algorithm to solve the practical problems. We make two following assumptions to bound the degree and the maximum power consumption of the tree.

(1) The maximum degree of the node in T is upper bounded by a constant d.
(2) $\sum_{i \in V} \mathscr{P}_i \leq P_{max}$, where P_{max} is a constant.

Then we will introduce a polynomial algorithm to minimize the number of meter deployments when a clear ratio $r(D, T)$ is given. The flowchart shown in Figure 3 describes the algorithm. It outputs the set $D \in V$, which indicates the nodes that should be deployed by smart meters.

The algorithm searches node v from the bottom of tree T up to the root. Thus, during investigating node v, all the nodes in subtree $ST(v)$ (except for v) must have been visited before v. In each iteration, the algorithm find a set of children C in $ST(v)$ to deploy smart meters, that is, add C to D, and cut all the subtrees rooted at nodes deployed by meters, that is, $T = T \setminus \bigcup_{u \in C} ST(u)$. The algorithm repeats until T is empty.

Suppose in the ith iteration, we visit node v. Then we decide meter deployment C in $ST(v)$ and cut $\bigcup_{u \in C} ST(u)$ accordingly by checking the clear ratio of the subtree $ST(v)$.

(1) If the clear ratio of $ST(v)$ is less than $r(D, T)$, more meters need to be deployed in $ST(v)$. Suppose $Children(v)$ is the set of all the children of node v. There are at most 2^d subsets of $Children(v)$, where d is the maximum degree. It takes time $O(1)$ to enumerate all of these subsets. Then we could find the smallest subset $C_{best} \subseteq Children(v)$ such that the clear ratio of the remaining subtree in $ST(v)$ is not less than $r(D, T)$ by removing all the subtrees $ST(u), u \in C_{best}$. Then meters are deployed on node u for all $u \in C_{best}$, and the remaining tree T is updated by removing all the subtrees $ST(u), u \in C_{best}$.
(2) If the clear ratio of $ST(v)$ is larger or equal than $r(D, T)$, we need not deploy more meters in $ST(v)$. We just connect all leaves of $ST(v)$ directly to node v.

Note that a meter is needed to be deployed on the root if the remaining tree is not empty. We have proved that the algorithm outputs a deployment strategy D which is at most two times the size of an optimal solution for any given T and the clear ratio

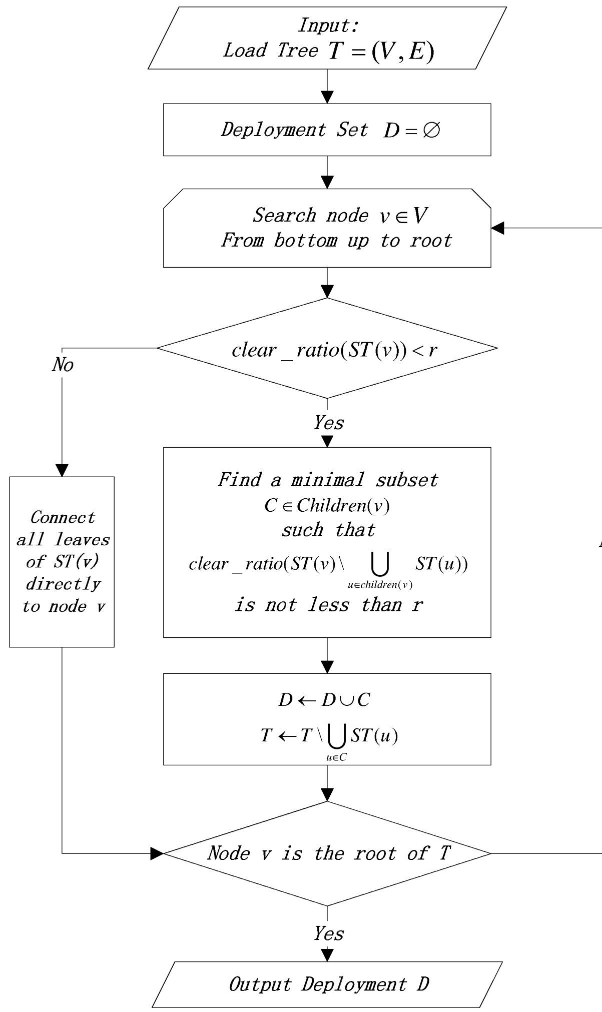

Fig. 3. MDOP algorithm for the bounded trees.

requirement r [Hao et al. 2012b]. The total running time is $O(2^d \mathscr{P}_{max} n^2)$, which is polynomial to n when d and $\mathscr{P}_{max}$ are constant.

Note that the meter deployment cost decreases with the decreasing of $r(D, T)$, while the probability of ambiguous decoding increases with the decreasing of $r(D, T)$. To accurately decode the appliance states when $r(D, T) < 1$, the switching sparseness and the sequence feasibility constraint should be exploited to resolve the ambiguities by the sequential dependence of states. In the next section, a hidden Markov model-based state tracking method is presented.

4. HMM-BASED STATE SEQUENCE TRACKING ALGORITHM

The state transition of electrical appliances has a Markovian property, that is, an appliance's state at period t is only related to its state at period $t-1$. The states cannot be directly observed but are inferred by measuring the aggregated power using the smart meter. Thus the state decoding problem in Eq. (7) can be modeled by a hidden

Markov model $\lambda = (\mathbf{X}_0, \mathbf{A_t}, \mathbf{B})$, where $\mathbf{X}_0$ is the initial state distribution; $\mathbf{A_t}$ is the state transition matrix and $\mathbf{B}$ is the observation matrix.

Let's consider $\mathbf{A}_t$ is time variant. An entry of $\mathbf{A}_t$ is $\mathbf{a}_{i,j,t} = P\left(\mathbf{X}_t = S_j | \mathbf{X}_{t-1} = S_i\right)$, $i, j \in [1, \ldots, 2^N]$, which indicates the state transition probability from S_i at $t-1$ to S_j at t. An atom of $\mathbf{B}$ is $b_{i,j} = P(\mathbf{Y}_t = v_i | \mathbf{X}_t = S_j)$, which indicates the likelihood of state S_j when the observation is v_i, where $v_i \in V = \{v_1, v_2, \ldots, v_M\}$ and V is the alphabet indicating that all the distinct observations may be measured by the power meters. According to such an HMM model, the state sequence decoding problem can be stated as follows.

Problem 2. Given the sequence of power measurements by m meters from time 1 to t: $\mathbf{Y} = \{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_t\}$, and the HMM model λ, we want to find the state sequence of N appliances, $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t\}$ that maximizes the following conditional probability.

$$\begin{aligned} &\underset{\mathbf{X}_1,\ldots,\mathbf{X}_t}{\textbf{maximize}} \; P\left(\mathbf{X}_1, \ldots, \mathbf{X}_t, \mathbf{Y}_1, \ldots, \mathbf{Y}_t | \lambda\right) \\ &\textbf{subject to:} \;\; \mathbf{1}.\ \forall \tau \in [1, t],\ \|\mathbf{X}_\tau \mathbf{P} - \mathbf{Y}_\tau\|_2 < \varepsilon, \\ &\qquad\qquad\qquad \mathbf{2}.\ \forall t \geq 1,\ \|\mathbf{X}_t - \mathbf{X}_{t-1}\|_1 < U_t, \end{aligned} \tag{12}$$

where in the first constraint, ε is the tolerable measurement error of the smart meters; the second constraint indicates the switching sparseness constraint in Eq. (7).

4.1. Fast Sequence Decoding (FSD) Algorithm

The HMM model in Eq. (12) has 2^N states and the sequence length is t. Traditional Viterbi decoding algorithms need $O(t2^{2N})$ complexity to decode the most likely hidden state sequence, which is impractical to calculate in real time when N is large. To enable online state decoding, the online decoding complexity must be reduced. By exploiting the tree structure of the power network and the sparseness of ON/OFF switching events, a fast sequence decoding algorithm (FSD) is presented here. The algorithm runs in parallel in the monometer trees, which provides polynomial complexity $O(n^{U_t+1})$, where $n < N$ is the number of appliances in a subtree. The procedure of the FSD is as follows.

4.1.1. Offline State Sorting in Each Monometer Tree. In the monometer tree forest, because the meters are independent, the state decoding can be run in parallel in each monometer tree. For a monometer tree with n appliances, it has 2^n possible states. Finding the feasible states that most likely generate the smart meter's observation at time t need $O(2^n)$ comparisons. Offline state sorting is proposed to speed up the online searching. We offline sort the 2^n states according to their energy values to prepare an ordered state vector for online binary search. Although the sorting operation has complexity $O(2^n log(2^n)) = O(n2^n)$, its needs only to be executed once offline or infrequently in the case of monometer tree structure changes.

4.1.2. Sequence Decoding Model in HMM Graph. After the offline state sorting, an HMM-based online state decoding algorithm is designed in each monometer tree. Consider the HMM model, as shown in Figure 4. The HMM graph contains t *layers* (time intervals), and each layer contains 2^n *vertices* (states). Let $S^0_{x_t}$ be the vertex reward, which is the likelihood that the observation y_t is generated by state x_t. Let $S_{x_t,x_{t-1}}$ be the edge reward, which indicates the state transition probability. A *path* is a sequence of vertices $x_1, x_2, \ldots, x_t$ crossing t layers, whose *reward* is evaluated by the product of edge rewards and vertex rewards associated with the path.

$$w(x_1, x_2, \ldots, x_t) = \alpha_t \prod_{u=1}^{t} S^0_{x_u} \prod_{u=2}^{t} S^1_{x_u, x_{u-1}}. \tag{13}$$

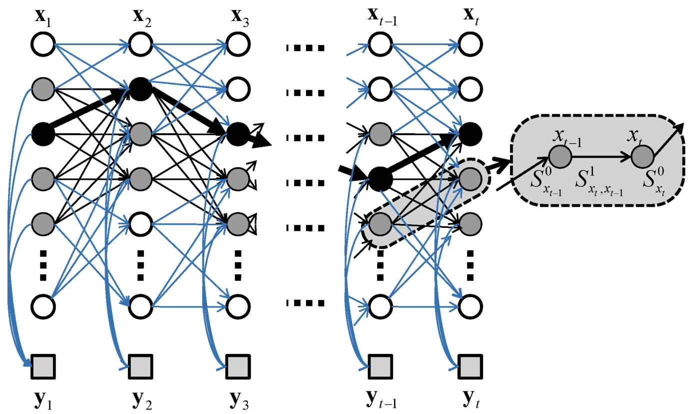

Fig. 4. Online fast sequence decoding algorithm.

In Eq. (13),

$$\alpha_t = \frac{1}{\sum_{x_1,x_2,\ldots,x_t \in \mathbb{P}_t} \prod_{u=1}^{t} S^0_{x_u} \prod_{u=2}^{t} S^1_{x_u,x_{u-1}}} \tag{14}$$

is a normalizer that keeps the total rewards of all possible paths at time t equal to 1. We define $\gamma(x_t)$ as the maximal reward associated to any path across t layers.

$$\gamma(x_t) = \max_{x_1,x_2,\ldots,x_{t-1}} w(x_1, x_2, \ldots, x_{t-1}, x_t). \tag{15}$$

Thus, we can calculate the reward of the best path $\gamma(x_t)$ via

$$\gamma(x_t) = \begin{cases} \alpha_1 S^0_{x_1}, & \text{if } t = 1, \\ \max\limits_{x_{t-1}} \left(\alpha_t \cdot \gamma(x_{t-1}) \cdot S^1_{x_t,x_{t-1}} S^0_{x_t}\right), & \text{otherwise.} \end{cases} \tag{16}$$

The sequence decoding algorithm is for finding the best reward path.

$$\gamma(x_t)^* = \max_{x_t} \gamma(x_t). \tag{17}$$

A traditional Viterbi algorithm *opens* 2^n states at time t to evaluate $S^0_{x_t}$ and backtracks $t-1$ steps for calculating Eq. (16). In each backtrack step, up to 2^n predecessors are opened. We use *open* to mean an operation of reward calculation, so Viterbi has $O(t2^{2n})$ complexity. In this article, a polynomial-time online decoding algorithm is proposed. The main idea is to open only necessary states in the forward and backward steps.

4.2. Fast Forward Search Strategy

In the forward step at time t, only the states that satisfy Constraint (18) will be open. We call them the *feasible states*. $\mathbf{p} \in \mathbb{R}^{n\times 1}$ denotes the *pattern matrix* of the monometer tree.

$$\|\mathbf{x}_t\mathbf{p} - y_t\|_2 \le \varepsilon. \tag{18}$$

Since, in the offline phase, the 2^n states are ordered according to their energy consumption values using $y_t - \varepsilon$ and $y_t + \varepsilon$ as the searching targets, we can conduct twice the number of binary searches on the 2^n states, which will find all the feasible states that satisfy Eq. (18). ε is the foreknowledge about the metering error bound. Such a binary search step on 2^n states has complexity $O(log(2^n)) = O(n)$, which makes the forward search very quick.

Suppose the noises of meters follow normal distribution $N(0, \sigma^2)$, then the vertex reward can be calculated by

$$S^0_{x_t} = \beta_t \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(|x_t \cdot p - y_t|)^2}{2\sigma^2}}, \tag{19}$$

where σ can be set to ε/k and $k > 3$ for guaranteeing that most of the metering errors are less than ε. Let $\mathbf{F}_t$ denote the feasible states at time t. β_t normalizes the total likelihoods of all feasible states at time t equal to one.

4.3. Fast Backward Search Strategy

After getting a feasible state x_t by the forward search, a backtrack algorithm is needed to calculate the best path reward $\gamma(x_t)$ by Eq. (16). By the state transition model $\mathbf{A}_t$, the link reward from x_t to a predecessor x_{t-1} can be calculated by

$$S^1_{x_t, x_{t-1}} = p^d(1-p)^{n-d}, \tag{20}$$

where $d = ||x_t - x_{t-1}||_1$ is the number of different states between x_t and x_{t-1}.

Since the ON/OFF switching events from $t-1$ to t is sparse, which is upper bounded by U_t, it is not necessary to open all the predecessor states. At most $\sum_{i=1}^{U_t}\binom{n}{i}$ predecessors may be open, which has complexity $O(n^{U_t})$. Further, the infeasible predecessors should not appear in the state sequence, which is needless to open. Thus, $\forall x_t \in \mathbf{F}_t$, at most $\min\{\sum_{i=1}^{U_t}\binom{n}{i}, |\mathbf{F}_{t-1}|\}$ predecessors need to be visited for calculating $\gamma(x_t)$, in which $|\mathbf{F}_{t-1}|$ is generally a very small value, guaranteeing the calculation to be very efficient. Since $\gamma(x_t)$ can be fully determined by the feasible predecessors by backtracking only one step without the need to backtrack t steps, the backward search algorithm has the worst complexity $O(n^{U_t})$, which is polynomial to n. Then $\gamma(x_t)^*$ can be calculated by Eq. (17). For reliable decoding against the ambiguities, FSD keeps the top-K possible paths without assertively choosing one top path. The storage cost is linear to t, which is very small.

4.4. Lightweight HMM Model Training

Training the HMM model is an important step before using HMM to decode the state sequence, which generally requires considerable training efforts. In this article, we propose using the lightweight offline knowledge to build the HMM model.

(1) We propose setting up the transition matrix $\mathbf{A_t}$ with the knowledge of U_t. Assume the ON/OFF transition probabilities are equal and i.i.d. Denote p as the occupance probability of flipping one state. Then p can be calculated by solving the equation $\sum_{i=1}^{U_t} p^k(1-p)^{N-k}\{^N_k\} = 1$. Let d be the number of different states between S_i and S_j, then the transmission probability from S_j to S_i can be modeled by $\mathbf{a}_{i,j,t} = p^d(1-p)^{N-d}$.

(2) Modeling the observation matrix is generally difficult, because the distinct measurements measured by m meters are numerous. We propose using a least-squares-estimation-based online searching scheme (Eq. (18)) to replace the explicit observation matrix model. It eliminates the efforts of training the observation matrix and efficiently speeds up the online state decoding process.

(3) The initial state of the HMM can be set by the offline knowledge of the appliance states. Accurate knowledge of the initial states would improve the tracking accuracy. For example, the tracking algorithm can be started at midnight so that most appliances are in an off state, which can be set as the initial states.

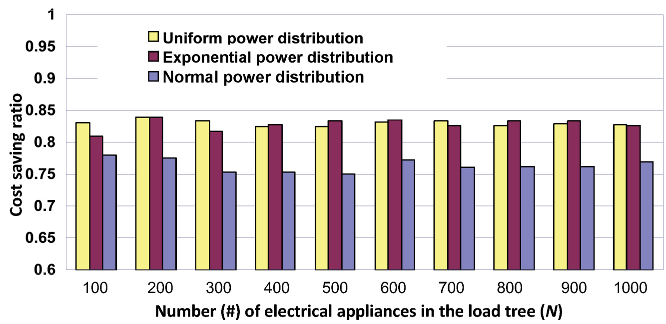

Fig. 5. Cost saving ratio of MDOP for different scale load trees and different power pattern distributions.

Therefore, the HMM model can be set up with very limited offline knowledge, making it practical for online decoding of states of massive appliances. The overall online algorithm has time complexity $O(n^{U_t+1})$ to calculate the top-K paths at period t. It enables efficient online state tracking for massive electrical appliances. The normalizers $\{\alpha_t\}$ and $\{\beta_t\}$ can be calculated in realtime, thus avoiding the difficulties of assigning reward weights in traditional Viterbi [Forney 1973].

5. NUMERICAL EVALUATIONS

We conduct extensive experiments to evaluate the proposed MDOP algorithm and the FSD algorithm using both the simulated data and the real data from the Powernet dataset. In simulations, load trees containing N leaf nodes with maximum D degree were generated randomly, simulating the arbitrary power distribution networks. The power patterns of the electrical appliances (leaf nodes) were generated by uniform, normal, or exponential distributions, in which the uniform power distribution simulates the case when appliances' power levels are almost even, normal distribution simulates the general case, and exponential distribution simulates the case when appliances' powers are very concentrated.

5.1. Performance of MDOP

For evaluating MDOP, we evaluate (1) its performance of saving the meter deployment cost; (2) the subtree character after meter deployment by MDOP; and (3) the impact of the clear ratio to the meter deployment cost.

5.1.1. Cost Saving Ratio. MDOP is applied to load trees with 100–1,000 electrical appliances to evaluate the *cost saving ratio*. The cost saving ratio is defined by the number of deployed smart meters given by MDOP divided by the number of appliances in the load tree. It compares the performance of MDOP with the naive fidelity energy auditing, in which each appliance is monitored by one appliance [Kazandjieva et al. 2009]. In these experiments, the clear ratio in the MDOP algorithm is set to 1; the normal, uniform, and exponential power distributions are evaluated with equal mean = 100. The cost saving results for different load trees are plotted in Figure 5. From the results, we see that for load trees with different sizes and different power distributions, the MDOP algorithm can reduce the meter deployment cost by more than 75% if compared with the one-to-one monitoring method. The cost saving ratios in exponential and uniform power distributions are similar. When the power levels of appliances are more concentrated (in the normal distribution), more smart meters are required for disambiguating the states of similar-power appliances, which accords with our general intuition.

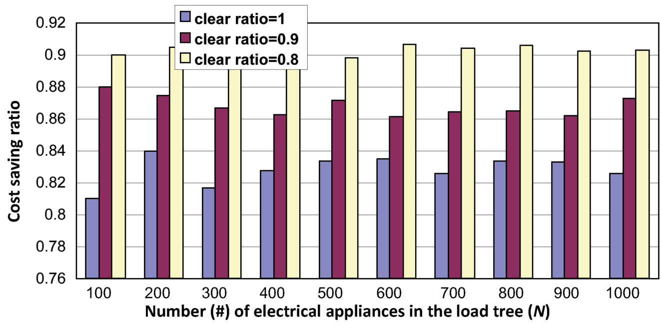

Fig. 6. Cost saving ratio vs. clear ratio.

5.1.2. Cost Saving Ratio vs. Clear Ratio. Figure 6 further shows how the clear ratio in the MDOP algorithm affects the cost saving ratio. The results are shown for load trees with normally distributed power patterns. It can be seen that the cost saving ratio decreases with the clear ratio. When the clear ratio is set to 0.8, almost 90% deployment cost can be saved. The results hold for different power distributions. In next section, we will show the FSD algorithm provides accurate state tracking even when the clear ratio is not high.

5.2. Performance of FSD

The monometer trees are generally not large in size, which reduces the online decoding complexity remarkably, that is, the state space is reduced from 2^N to 2^n, where $n < N$. The FSD algorithm is run in different monometer trees in parallel. Therefore, in evaluating FSD, we focus more on the decoding accuracy than the decoding efficiency. We evaluate the accuracy of FSD over the whole load tree by considering the average accuracy of all the monometer trees.

$$e = \frac{1}{T \cdot N} \sum_{t=1}^{T} \|\mathbf{X}(t) - \hat{\mathbf{X}}(\mathbf{t})\|_1. \tag{21}$$

In Eq. (21), $\mathbf{X}(t)$ indicates the ground truth of N appliances' states at time t. $\mathbf{X}(t)$ is the estimated states. e evaluates the average state tracking error over N appliances and over time T. The state tracking error at time t is evaluated by

$$e(t) = \frac{1}{N} \|\mathbf{X}(t) - \hat{\mathbf{X}}(\mathbf{t})\|_1. \tag{22}$$

5.2.1. Metering Noise vs. Tracking Accuracy. We first evaluate how the metering noises of the smart meters affect the state tracking accuracy. In simulations, we set the metering noise to $N(0, \sigma^2)$. In the FSD algorithm, we set $\varepsilon = 5\sigma$ accordingly to guarantee that most of the metering errors are less than the tolerable error $|\varepsilon|$. In such a setting, for a load tree of 500 nodes with power distribution $N(200, (200/3)^2)$, the real-time tracking errors versus metering noises σ are plotted in Figure 7. In the experiments, meters are deployed by MDOP with clear ratio $r = 1$; each point is the average result of ten experiments. The blue curves show the tracking error of FSD, and the red curves show the tracking error given by a traditional Viterbi algorithm. In Viterbi, the vertex reward is assigned a weight to make it comparable to the link reward. The results show that (1) the FSD algorithm generally has better tracking accuracy than Viterbi, and (2) when the metering error is small, the tracking error of FSD is very small, showing its effectiveness in disambiguating the mixed states. The FSD algorithm performs

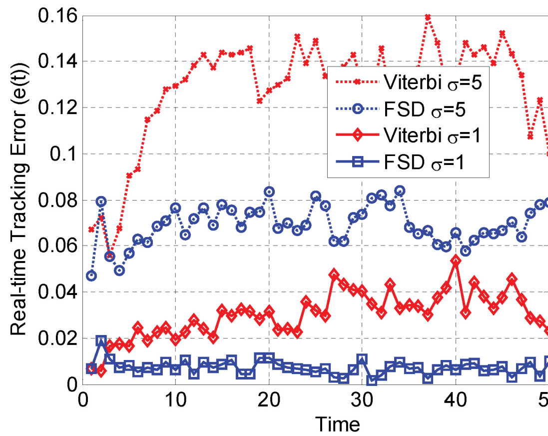

Fig. 7. Metering noise vs. tracking accuracy.

Fig. 8. Tracking accuracy vs. clear ratio vs. cost saving ratio.

better than Viterbi for keeping the top 500 feasible pathes instead of only the top path. Another reason is that FSD uses product-type reward functions, which don't suffer the weight assignment error for balancing link rewards and vertex rewards.

5.2.2. Clear Ratio vs. Tracking Accuracy vs. Cost Saving Ratio. Then we investigate how the clear ratio of the MDOP algorithm affects the state tracking accuracy. For focusing on the effects of clear ratios, the metering noise σ is set to zero so that all the errors are caused by the decoding ambiguities. For the same load tree settings in Figure 7, the tracking accuracy versus clear ratio and corresponding cost saving ratio are plotted in Figure 8. The results show interesting features of this CS-based state monitoring problem. The cost saving ratio increases slowly with the reduction of the clear ratio and reaches a saturated status when the clear ratio is less than 0.6. The tracking errors increase very quickly with the clear ratio. The different trends of the curves indicate some good region for choosing the clear ratio, in which the tracking error is small and most deployment costs can be saved, as the region 0.8 to 1 in the figure.

5.3. Experiments on the Powernet Dataset

The preceding experiments assume appliances have static power patterns. We conduct further experiments using Powernet data [Kazandjieva et al. 2009] to relax this

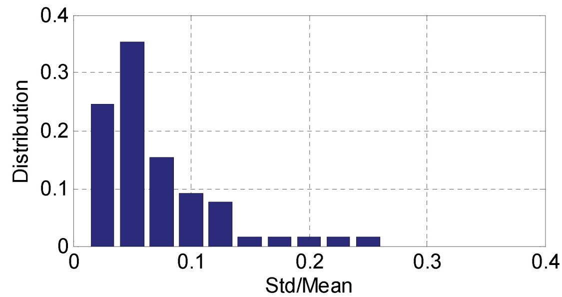

Fig. 9. Power Std/Mean for 65 appliances in Powernet.

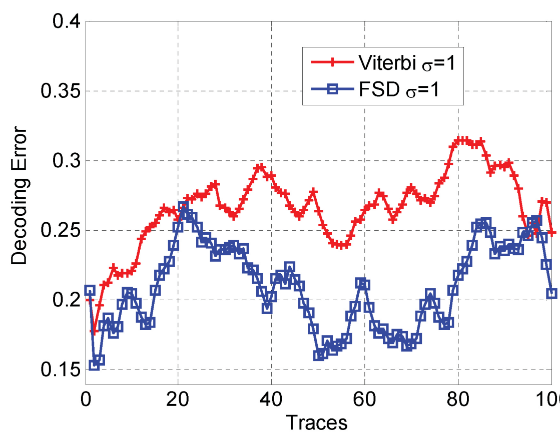

Fig. 10. Tracking accuracy using the PowerNet dataset.

assumption. Powernet was developed by Sing group of Stanford University. They put online the real-time power consumptions of 134 electrical appliances in a computer science lab. We use the data of September 30, 2011, which contains feasible data of 65 appliances. Some appliances were not open on that day, and some appliances had very small power, which are not used. The sampling frequency is 1Hz.

We first build the power patterns of these appliances. By conducting statistical analysis to 500-second power samples of each appliance, the Std/Mean for each appliance is evaluated, which is plotted in Figure 9. It can be seen that more than 75% of the appliances have an Std/Mean less than 0.1, indicating that the power consumption of appliances in real applications is not highly dynamic.

Since the dataset doesn't provide the load tree structure, we offline train each appliance's power pattern by the average energy consumption over ten minutes and randomly generate a load tree to assign 65 appliances randomly to the leaves. The MDOP algorithm was run to optimize the smart meter deployment optimization on the generated load tree with clear ratio requirement $r = 1$. Then, tree splitting and state sorting are conducted offline.

In the online phase, when an appliance is on, its energy consumption is not static but follows its energy consumption trace in the dataset. The meters measure the past 30-second moving average of the mixed real-time energy consumption of its subtree to decode the states of appliances. The state tracking performances by FSD and Viterbi are plotted in Figure 10. The decoding error is generally less than 20% for FSD, which shows the potential of using the proposed framework in tracking the states of the dynamic power appliances, considering its large cost saving ratio. We checked the errors and found the main reason to be that the variation range of some large power alliances in the same monometer tree covers the ON/OFF events of the small power appliances. Such problems can be further resolved by improving the meter deployment scheme to consider both the mean and the variance of different electrical appliances, which will be studied in future work.

6. CONCLUSION AND FUTURE WORK

This article presents a lightweight metering and sequence decoding framework for tracking the ON/OFF states of massive electrical appliances. The rationale is that the power patterns of appliances can be learned offline and the switching events of the electrical appliances in a short interval are sparse. By entropy-based analysis, a clear ratio is proposed to bridge the deployment cost and the tracking accuracy, which can be seen as a parameter for describing the decoding accuracy requirement. Based on it, MDOP, a polynomial-time deployment algorithm, is proposed to deploy the minimal number of smart meters for a given requirement of the clear ratio. A fast state sequence decoding algorithm FSD is proposed to facilitate a polynomial-time decoding algorithm which overcomes the complexities of disambiguating 2^N states. The experimental results show the effectiveness and good performances of the proposed methods.

This work contains some basic assumptions, such as power patterns are static, and state transition probabilities are i.i.d. In future work, more complex power patterns and robust deployment algorithms will be studied. The state transitions could be further modeled by hidden semi-Markov models to consider the work duration distributions of the appliances. Detection of transient signals could help to extract the occasion of the switching events, and the group dependence of ON/OFF switching events could further increase the decoding accuracy.

APPENDIX

PROOF FOR THE OBJECTIVE FUNCTION EQUIVALENCE

In this section, we prove that the objective function in Eq. (2) can be transformed approximately to the objective function in Eq. (3). Mathematically,

$$\begin{aligned}
&\mathbb{E}\left\{\left(\mathbf{Z}_{t,i} - \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathbf{P}_{t,j}\right)^2\right\} \\
&= \mathbf{Z}_{t,i}^2 - 2\mathbf{Z}_{t,j} \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j} + \mathbb{E}\left\{\sum_{j\in S(i)} \mathbf{X}_{t,j}\mathbf{P}_{t,j}\right\}^2 \\
&= \left(\mathbf{Z}_{t,i} - \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j}\right)^2 + \Delta,
\end{aligned} \tag{23}$$

where

$$\begin{aligned}
\Delta &= \mathbb{E}\left(\sum_{j\in S(i)} \mathbf{X}_{t,j}\mathbf{P}_{t,j}\right)^2 - \left(\sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j}\right)^2 \\
&= \sum_{j\in S(i)} \mathbf{X}_{t,j}^2 \delta_j.
\end{aligned} \tag{24}$$

Note that $\mathbf{X}_{t,j}^2 = \mathbf{X}_{t,j}$, and $\delta_j \approx \alpha \mathcal{P}_{t,j}$, thus we have

$$\begin{aligned} \mathbb{E}\left\{\left(\mathbf{Z}_{t,i} - \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathbf{P}_{t,j}\right)^2\right\} &= \mathbf{Z}_{t,i}^2 - \left(2\mathbf{Z}_{t,j} - \alpha\right) \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j} + \left(\sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j}\right)^2 \\ &= \left(\mathbf{Z}_{t,i} - \alpha/2 - \sum_{j\in S(i)} \mathbf{X}_{t,j}\mathcal{P}_{t,j}\right)^2 + \Lambda, \end{aligned} \tag{25}$$

where $\Lambda = \alpha \mathbf{Z}_{t,i} - \alpha^2/4$ is a constant.

REFERENCES

S. Dawson-Haggerty, S. Lanzisera, J. Taneja, R. Brown, and D. Culler. 2012. @scale: Insights from a large, long-lived appliance energy WSN. In *Proceedings of the ACM IPSN*.

Energy hub. 2009. Energy hub. http://www.energyhub.net/.

L. Farinaccio and R. Zmeureanu. 1999. Using a pattern recognition approach to disaggregate the total electricity consumption in a house into the major end-uses. *Energy Build.* 30, 3, 245–259.

G. D. Forney, Jr. 1973. The Viterbi algorithm. *Proc. IEEE* 61, 3, 268–278.

GreenBox. 2009. Green box. http://www.getgreenbox.com/.

S. Gupta, M. S. Reynolds, and S. N. Patel. 2010. Electrisense: Single-point sensing using EMI for electrical event detection and classification in the home. In *Proceedings of Ubicomp'10*. ACM, New York, NY, 139–148.

X. Hao, Y. Wang, C. Wu, L. Song, and Y. Wang. 2012a. Smart meter deployment for efficient appliance state monitoring. In *Proceedings of the 3rd IEEE International Conference on Smart Grid Communications*. 25–30.

X. Hao, Y. Wang, C. Wu, A. Y. Wang, and L. Song. 2012b. Proof of approximation ratio and complexity of MDOP algorithm. http://wcy.name/papers/proof2.pdf.

G. W. Hart. December 1992. Nonintrusive appliance load monitoring. *Proc. IEEE* 80, 12, 1870–1891.

X. Jiang, S. Dawson-Haggerty, P. Dutta, and D. Culler. 2009a. Design and implementation of a high-fidelity AC metering network. In *Proceedings of ACM IPSN'09*. IEEE Computer Society, 253–264.

X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. 2009b. Experiences with a high-fidelity wireless building energy auditing network. In *Proceedings of SenSys'09*. ACM, New York, NY, 113–126.

D. Jung and A. Savvides. 2010. Estimating building consumption breakdowns using on/off state sensing and incremental sub-meter deployment. In *Proceedings of SenSys'10*. ACM, New York, NY, 225–238.

M. A. Kazandjieva, B. Heller, P. Levis, and C. Kozyrakis. 2009. Energy dumpster diving. In *Proceedings of the 2nd Workshop on Power Aware Computing (HotPower'09)*.

Y. Kim, T. Schmid, Z. M. Charbiwala, and M. B. Srivastava. 2009. ViridiScope: Design and implementation of a fine grained power monitoring system for homes. In *Proceedings of Ubicomp'09*. ACM, New York, NY, 245–254.

S. Leeb, S. Shaw, and J. L. Kirtley, Jr. 1995. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Trans. Power Delivery* 10, 3, 1200–1210.

J. Lifton, M. Feldmeier, Y. Ono, C. Lewis, and J. A. Paradiso. 2007. A platform for ubiquitous sensor deployment in occupational and domestic environments. In *Proceedings of IPSN'07*. ACM, New York, NY, 119–127.

L. K. Norford and S. B. Leeb. 1996. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms. *Energy Build.* 24, 1, 51–64.

Oxford St. Hughs College Data. 2010. Energy and water conservation. http://www.st-hughs.ox.ac.uk/welfare-and-facilities/estates/energy-and-water-conservation.

S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. 2007. At the flick of a switch: Detecting and classifying unique electrical events on the residential power line. In *Proceedings of Ubicomp'07*. 271–288.

A. Rowe, M. Berges, and R. Rajkumar. 2010. Contactless sensing of appliance state transitions through variations in electromagnetic fields. In *Proceedings of BuildSys'10*. ACM, New York, NY, 19–24.

Z. C. Taysi, M. A. Guvensan, and T. Melodia. 2010. Tinyears: Spying on house appliances with audio sensor nodes. In *Proceedings of BuildSys'10*. ACM, New York, NY, 31–36.

Tendril. 2012. The Tendril residential energy ecosystem. http://www.tendrilinc.com/.

R. Tibshirani. 1994. Regression shrinkage and selection via the lasso. *J. Royal Stat. Soci. Series B* 58, 267–288.

Y. Wang, X. Hao, L. Song, C. Wu, Y. Wang, C. Hu, and L. Yu. 2012. Tracking states of massive electrical appliances by lightweight metering and sequence decoding. In *Proceedings of the 6th International Workshop on SensorKDD'12*.

Received July 2012; revised November 2012; accepted April 2013