Detection of Defaulting Participants of Demand Response Based on Sparse Reconstruction

Shun-Ichi Azuma, *Senior Member, IEEE*, Daisuke Sato, Koichi Kobayashi, *Member, IEEE*, and Nobuyuki Yamaguchi, *Member, IEEE*

***Abstract*—In demand response (DR) based on contracts with consumers, some participants have the potential to default on providing their scheduled negawatt energy due to demand-side fluctuations. Thus, the detection of defaulting participants is an important function of the aggregator. In particular, it is preferable to detect them with limited information and not by real-time continuous metering because of communication costs and social acceptance. This paper addresses the problem of detecting defaulting participants in contract-based DR, provided that the aggregator can inspect the total negawatt energy and the negawatt energies of a limited number of participants via smart meters. By focusing on the property that only a few participants are defaulting under their contracts, we propose a detection method based on *sparse reconstruction*, i.e., reconstructing a sparse vector from a small number of scalar equations. The proposed method is iterative, and each iteration improves the sparse reconstruction by including the inspection data from the previous iteration. It is theoretically guaranteed that the proposed method derives the exact solution under practical conditions. Finally, direct load control is incorporated into the detection method to eliminate defaulting participants.**

***Index Terms*—Demand response, default detection, sparse reconstruction, direct load control.**

Nomenclature

- 0 Zero scalar or zero vector
- 1_n n-dimensional column vector whose elements are all one
- c_{ij} Scheduled negawatt energy (kWh) of participant i at time slot j
- C Collective matrix of c_{ij} ($i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$), called the *scheduled negawatt table*
- $C(p)$ Matrix resulting from the modification of the matrix C so that the p_1-, p_2-, …, p_v-th column vectors are replaced with zero vectors, where $p = \{p_1, p_2, \dots, p_v\}$
- e_i i-th standard basis of the space $\mathbf{R}^{1\times n}$
- m Number of time slots for default detection
- n Number of participants of the demand response
- $\mathbf{P}(t)$ List of participants whose negawatt energy has never been inspected until time t in the proposed algorithm
- $\bar{\mathbf{P}}(t)$ List of the participants whose negawatt energy has been inspected until time t in the proposed algorithm
- $\mathbf{R}$ Real number field
- s_j Total amount (kWh) of negawatt energy generated in the demand response at time slot j
- s Collective vector of s_j ($j = 1, 2, \dots, m$), called the *total negawatt vector*
- $\mathrm{SR}(t)$ Sparse reconstruction problem at time t for estimating the failure rates of participants in the proposed algorithm
- x_i Failure rate of participant i
- x Collective vector of x_i ($i = 1, 2, \dots, n$), called the *failure rate vector*
- x_i^* True value of the failure rate of participant i
- x^* Collective vector of x_i^* ($i = 1, 2, \dots, n$)
- $\hat{x}_i(t)$ Failure rate of participant i at time t after when the proposed algorithm for detection and direct load control starts.

Manuscript received August 31, 2018; revised February 26, 2019 and April 16, 2019; accepted May 17, 2019. Date of publication June 13, 2019; date of current version December 23, 2019. This work was supported by JST CREST, Japan, under Grant JPMJCR15K1. Paper no. TSG-01272-2018. *(Corresponding author: Shun-Ichi Azuma.)*

S.-I. Azuma and D. Sato are with the Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan (e-mail: shunichi.azuma@mae.nagoya-u.ac.jp; sato.daisuke@b.mbox.nagoya-u.ac.jp).

K. Kobayashi is with the Graduate School of information Science and Technology, Hokkaido University, Sapporo 060-0808, Japan (e-mail: k-kobaya@ssi.ist.hokudai.ac.jp).

N. Yamaguchi is with the Department of Electrical Engineering, Faculty of Engineering, Tokyo University of Science, Tokyo 125-8585, Japan (e-mail: n-yama@ee.kagu.tus.ac.jp).

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

Digital Object Identifier 10.1109/TSG.2019.2922435

I. Introduction

Owing to supply-side anomalies such as outages of power plants, fluctuations in wind and solar generation, and fluctuations in the fuel price, electric power providers need to modify the load profiles of their consumers. The demand response (DR), i.e., the changes in electricity usage of consumers in response to incentive payments [1], is expected to be a solution to supply-side anomalies (see, e.g., [2]–[4] and the references therein). In fact, the DR has potential as an alternative energy source at a relatively low cost [5]. Moreover, smart meters have been rapidly deployed worldwide [2], which enables us to realize the DR.

A typical system architecture for the DR is shown in Fig. 1. The service provider of the DR, called the *aggregator*, manages participants to collect negawatt energy for sale on electricity markets or to other electric power providers. In the process of the DR, the aggregator predicts the future demand of negawatt energy and sends DR requests to appropriately selected participants. In response to receiving the requests,

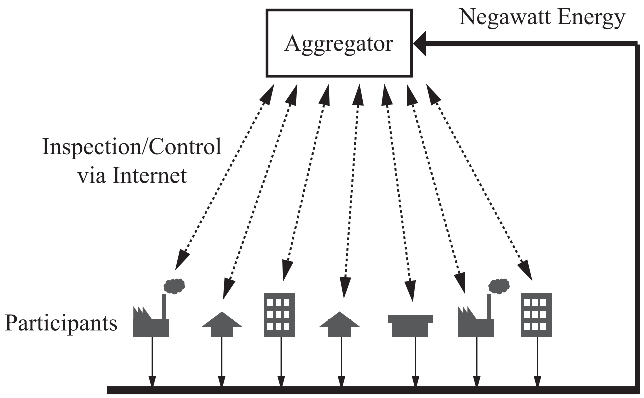

Fig. 1. Demand response.

the participants generate negawatt energy by changing their normal consumption patterns.

The DR takes various forms depending on its design, including price/incentives, prearranged contracts, direct load control, and so on [2]. In contract-based DR, the aggregator contracts with individual consumers for, in addition to their participation, their scheduled amounts of negawatt energy, as illustrated in Table I. Meanwhile, it is inevitable that some of the participants default in providing the scheduled negawatt energy owing to demand-side fluctuations such as instrument faults, schedule changes in production, unexpected visitors, and the uncertainties associated with manual operations. Therefore, the detection of failure sources (i.e., defaulting participants) in an expeditious manner is an important function of the aggregator. Although individual participants may have a little ability to provide negawatt energy in a smart grid, leaving failure sources untreated results in the growth of the number of them over time and may cause major failure soon. On the other hand, if failure sources are detected, the aggregator can subsequently carry out the appropriate procedures for the DR, which eliminates the risk of major failure at early stage. Moreover, as a service to the participants, it is favorable to provide fault information to defaulting ones.

The detection of defaulting participants may be easily performed if the aggregator can continuously meter their real-time consumption via smart meters. However, such metering is difficult in practice from the viewpoint of communication costs (including data traffic). Moreover, real-time continuous metering will be a barrier to social acceptance for the DR. In fact, such metering reveals the participants' highly private information, e.g., the lifestyles of the participants and business and industrial secretes. Consequently, it may result in the rejection of the DR. Thus, it is preferable to detect defaulting participants with more limited information, e.g., by irreversible data compression and intermittent metering.

The closest topic to the detection of defaulting participants in DR is anomaly detection in energy management systems. Anomaly detection refers to the methodology of detecting abnormal events that do not conform to the usual patterns of energy consumption. Several results have been obtained for building energy management systems [6]–[13]. Similar results have been developed for gas supply systems [14] and for a more general purpose [15]. These results aim at detecting *event time instants* (when an anomaly occurs) in terms of the total energy consumption of buildings etc. In other words, the results are not for detecting *failure sources* (what/who causes an anomaly). Moreover, the above results are not for DR. On the other hand, to the best of our knowledge, there are only two studies [16], [17] on anomaly detection in DR. However, these works do not address the problem of detecting failure sources, and in particular detecting defaulting participants. In fact, the former has presented a method to identify anomalous days, and the latter has presented a DR method incorporating the detection of abnormal usage. Therefore, it is a new challenge to develop a method for detecting failure sources in DR.

TABLE I
EXAMPLE OF THE SCHEDULED NEGAWATT ENERGY OF A PARTICIPANT

Time slot	1	2	$\cdots$	24
Negawatt energy (kWh)	2.2	2.5	$\cdots$	1.8

This paper addresses the problem of detecting defaulting participants. We consider a DR in Fig. 1 where the aggregator collects negawatt energy while metering the total amount of negawatt energy in real time. In the DR, the aggregator contracts with consumers so that each participant provides a scheduled amount of negawatt energy at each time slot, as illustrated in Table I, and the aggregator can inspect the actual negawatt energy of a limited number of participants via smart meters. This implies that the aggregator can use—in addition to the data of the time series of the total amount of negawatt energy—the data of the failure rates of some participants to detect defaulting participants.

To solve the problem, we focus on the fact that the DR is prearranged by contracts. This allows us to assume that only a few participants are defaulting on providing their scheduled negawatt energy. On the basis of this prior knowledge, we apply the technique of *sparse reconstruction*, i.e., reconstructing a sparse vector from a small number of scalar equations (see, e.g., [18]–[20]), to the detection problem.

The proposed method is developed in the following way. We first introduce a vector called the *failure rate vector*, in which each element corresponds to the failure rate of each participant, and formulate the detection problem as a sparse reconstruction problem with respect to the failure rate vector. It is then shown that the exact solution is not always derived by direct application of the standard sparse reconstruction technique to the problem. By observing this result, we develop an iterative method that improves the sparse reconstruction in each iteration by including inspection data from the previous iteration. We give a theoretical guarantee for the proposed method that the exact solution is derived in a finite number of iterations. Moreover, a stopping rule is presented for the method, by which we can derive the exact solution after a practically small number of iterations. Finally, direct load control is incorporated into the detection method to eliminate defaulting participants from the DR.

As a final remark, we note that the proposed method is *not* a direct application of sparse reconstruction. In fact, the idea of introducing individual inspection to sparse reconstruction is originally proposed in this paper. Moreover, the selection rule of a participant to be inspected and the update rule of the sparse reconstruction problem are newly developed by exploiting special properties of the detection problem. In this sense, our results are not straightforward consequences of sparse reconstruction.

II. Default Detection Problem

A. Demand Response

We consider the DR offered by an aggregator to consumers, as shown in Fig. 1, which is supposed to be implemented in a smart grid. The aggregator collects a certain amount of negawatt energy from participants in exchange for incentives and sells it to other electric power providers or in electricity markets. The participants are supposed to be in the residential, commercial, and industrial sectors, such as households, buildings, stores, and industrial plants.

The contract between the aggregator and each consumer contains the following clauses.

- The participant commits a certain amount of negawatt energy to the aggregator at each instance that the participant is signaled for the DR.
- The participant agrees to release its smart meter data, but the aggregator can access the *real-time* data only when an anomaly is detected.
- A violation of the scheduled negawatt energy is subject to a penalty.

As a result of the first clause, the participant provides a scheduled amount of negawatt energy to the aggregator at each time slot. This is essential for estimating the DR capacity of the aggregator and selecting participants to whom the DR is requested. The second clause, which is related to data release, is also a standard clause in DR contracts, while the real-time data access is limited to enhance consumer acceptance. In addition, the limited data access reduces the communication costs of the aggregator. The third clause is to ensure that there are as few defaulting participants as possible.

Three remarks are given for the contract.

First, the facilities for the DR (i.e., to generate negawatt energy) are not specified in the contract because they are supposed to be arbitrarily selected by each participant. However, air conditioners, refrigerators, washing machines, production machinery, etc. may be used.

Second, there are various methods for generating negawatt energy from facilities. Typical methods are switching the operation mode of a facility, shifting the operation time of a facility, and operating a facility on battery power. The first one includes changing the temperature settings of air conditioners and refrigerators and operating a facility in the so-called energy-saving mode. The second one is applied to facilities with the flexibility of operational timing, such as washing machines and production machinery. It is performed by finding an appropriate time by an energy management system to satisfy the scheduled negawatt energy. The final one is for participants with a battery.

Finally, the scheduled negawatt energy is assumed to be prespecified, as illustrated in Table I. This assumption is reasonable. In fact, it is known that the consumption of typical facilities providing DR service can be modeled (estimated) by decomposing the total consumption into the base consumption (unrelated to weather conditions) and consumption under weather conditions (temperature, humidity, etc.) [21]. Furthermore, the DR participants are expected to install an energy management system and in such a case the scheduled negawatt energy will be automatically generated.

Under the contract, the DR program entails the following steps. Before the DR event day, e.g., the day of the tight power supply situation, the aggregator predicts the required negawatt energy and selects participants to obtain a required amount of negawatt energy (possibly with a margin) for that period by using the information of their scheduled negawatt energy. The aggregator sends the DR requests to the selected participants; then, the participants manage their electricity usage to provide the scheduled negawatt energy. In the process, the aggregator meters the total amount of negawatt energy in real time in order to manage the DR.

B. Problem Formulation

In practice, some of the participants may default in providing the scheduled negawatt energy. Individual participants may have a little ability to provide negawatt energy in a smart grid, but the aggregator must detect defaulting participants as soon as possible to change their electricity usage in order to eliminate the risk of major failure caused by the growth of the number of failure sources. Therefore, let us formulate a default detection problem.

Without loss of generality, we assume that participants $1, 2, \dots, n$ are selected to provide a required amount of negawatt energy at time slots $1, 2, \dots, m$. Let $c_{ij} \in [0, \infty)$ be the scheduled negawatt energy (kWh) of participant i at time slot j, and let $x_i \in [0, 1]$ denote the failure rate of participant i. Note that $x_i > 0$ if participant i is defaulting; $x_i = 0$ otherwise. We denote the total amount (kWh) of negawatt energy produced by the DR at time slot j by $s_j \in \mathbf{R}_{0+}$.

From the first clause of the contract in Section II-A, the negawatt energy generated by participant i at time slot j is represented by $c_{ij}(1 - x_i)$. Thus, we have

$$\sum_{i=1}^{n} c_{ij}(1 - x_i) = s_j \tag{1}$$

at time slot j and obtain

$$Cx = C1_n - s \tag{2}$$

for m time slots, where

$$C := \begin{bmatrix} c_{11} & c_{21} & \cdots & c_{n1} \\ c_{12} & c_{22} & \cdots & c_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ c_{1m} & c_{2m} & \cdots & c_{nm} \end{bmatrix} \in [0, \infty)^{m \times n},$$

$x := [x_1 \; x_2 \; \cdots \; x_n]^\top \in [0, 1]^n$, and $s = [s_1 \; s_2 \; \cdots \; s_m]^\top \in [0, \infty)^m$. If $C1_n = s$ (i.e., $x = 0$ in (2)), then no participant is defaulting; otherwise, there exists a defaulting participant.

The matrix C is called the *scheduled negawatt table*, and the vectors x and s are called the *failure rate vector* and *total negawatt vector*, respectively.

Our problem is formulated as follows.

Problem 1: Consider the above DR and assume that

(i) participants $1, 2, \ldots, n$ are selected to provide a required amount of negawatt energy at time slots $1, 2, \ldots, m$ and

(ii) the scheduled amounts c_{ij} $(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, m)$ and (past) total amounts s_j $(j = 1, 2, \ldots, m)$ of negawatt energy are known to the aggregator.

If $C1_n \neq s$, i.e., a defaulting participant exists, estimate the failure rates x_i $(i = 1, 2, \ldots, n)$ of the participants (estimate the failure rate vector x).

By the first clause of the contract in Section II-A, each participant will provide negawatt energy to satisfy its scheduled amount. In this sense, the participants are supposed to independently generate negawatt energy in the DR. On the other hand, Problem 1 corresponds to finding defaulting participants from the data of the total negawatt energy, i.e., the sum of the negawatt energy of all the participants. Thus the solution will utilize the relation among the participants.

III. Sparse Reconstruction and Its Direct Application to Default Detection

A. Sparse Reconstruction

We review the framework of *sparse reconstruction*, i.e., reconstructing a sparse vector from a small number of scalar equations (see, e.g., [18]–[20] for more detail).

In this subsection, we focus our attention on a general class of linear equations apart from Problem 1 in Section II-B.

Consider the following linear equation composed of m scalar equations:

$$Ax = b, \tag{3}$$

where $x \in \mathbf{R}^n$ is the unknown vector and $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ are a constant matrix and constant vector, respectively, which are exactly known.

As is well-known, if $m \geq n$, there exists either no solution or a unique solution subject to $\mathrm{rank}(A) = n$. Thus, we can completely solve the linear equation in this case. In contrast, if $m < n$, the linear equation has infinitely many solutions, by which the solution x cannot be uniquely determined.

The sparse reconstruction is a solution to the latter case with the prior knowledge that the unknown vector x is *sparse*, i.e., x has only a few nonzero elements. It is formulated as the optimization problem

$$\min_{x\in\mathbf{R}^n} \|x\|_0 \quad \text{s.t.} \quad Ax = b \tag{4}$$

where $\|x\|_0$ is the ℓ_0-norm of the vector x, which corresponds to the number of nonzero elements in x. By definition, the minimization of $\|x\|_0$ reduces the number of nonzero elements, which renders the solution x sparse. In particular, the sparsest solution to (3) is provided from (4).

However, the problem in (4) is a combinatorial problem that is NP-hard [18]. It is therefore reasonable to relax the problem and substitute the solution of the relaxed problem for the solution to (4). The closest convex relaxation is to use the ℓ_1-norm as

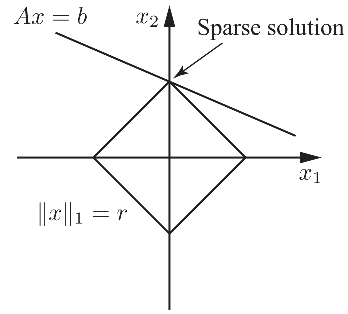

Fig. 2. Principle of the sparse reconstruction with ℓ_1-norm relaxation.

$$\min_{x\in\mathbf{R}^n} \|x\|_1 \quad \text{s.t.} \quad Ax = b, \tag{5}$$

which is known to be equivalently transformed into a linear programming problem. Thus, the relaxed problem is easily solved by standard optimization techniques. Moreover, under mild conditions on the matrix A and the level of the sparsity of the solution x, the relaxed problem has a sparse solution that is equal to the original sparse solution to (4) [18]–[20]. Fig. 2 illustrates the equality constraint and a ball in the ℓ_1-norm (which is a square), where the constraint and the ball of a certain radius intersect at the corner of the ball, and the corner corresponds to a sparse vector x. Thus, the minimization of $\|x\|_1$ results in a sparse solution. In this way, an unknown sparse vector satisfying a linear equation can be reconstructed (with a few exceptions) by solving the corresponding ℓ_1-optimization problem.

There are a number of applications for sparse reconstruction. One application is polynomial curve fitting (regression), which is fairly basic in machine learning. The problem is to find a polynomial of order $n-1, f(t) := c_0+c_1t+c_2t^2+\cdots+c_{n-1}t^{n-1}$, such that $y_i = f(t_i)$ $(i = 1, 2, \ldots, m)$ for a given dataset $\{(t_1, y_1), (t_2, y_2), \ldots, (t_m, y_m)\}$. It is clear that the problem is reduced to the linear equation

$$Tc = y$$

for $y := [y_1\ y_2\ \cdots\ y_m]^\top$, $c := [c_0\ c_1\ \cdots\ c_{n-1}]^\top$, and

$$T := \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^{n-1} \\ 1 & t_2 & t_2^2 & \cdots & t_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & t_m & t_m^2 & \cdots & t_m^{n-1} \end{bmatrix}.$$

If $m < n$ and we have the prior knowledge that c is sparse, this problem is reduced to (5) for $A := T$, $b := y$, and $x := c$.

B. Default Detection by Standard Sparse Reconstruction

Now, let us return our attention to Problem 1 in Section II.

As stated before, the participants contract with the aggregator so that each participant provides a scheduled amount of negawatt energy and violation is subject to a penalty. Moreover, defaulting is more likely to happen by an instrument fault when the consumption is automatically controlled

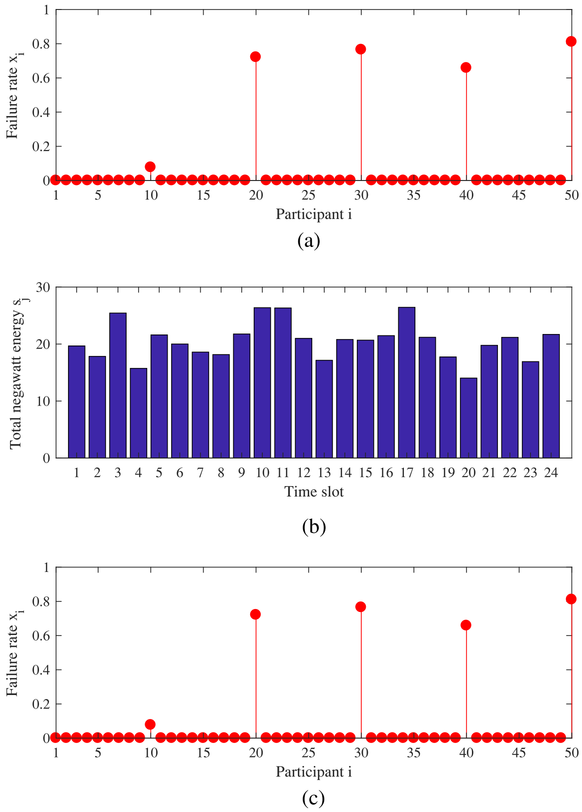

Fig. 3. Results estimated by standard sparse reconstruction ($n = 50$). (a) Failure rates x_i. (b) Total amounts of negawatt energy s_j. (c) Estimation of x_i.

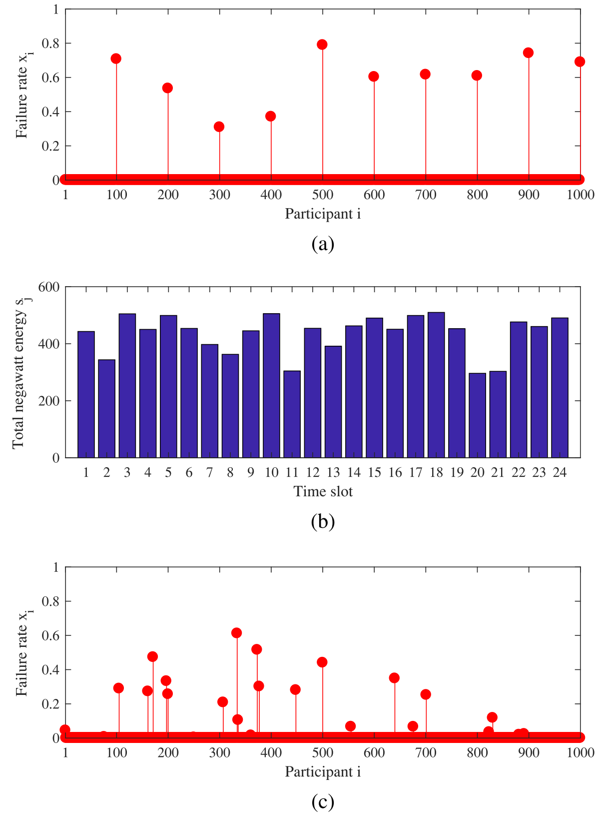

Fig. 4. Results estimated by standard sparse reconstruction ($n = 1000$). (a) Failure rates x_i. (b) Total amounts of negawatt energy s_j. (c) Estimation of x_i.

by an energy management system. Thus, it is reasonable to assume that only a few participants are defaulting. In other words, we have the prior knowledge that the failure rate vector x is sparse as a mathematical model of the third clause of the contract in Section II-A. On the other hand, the linear equation in (2) holds for the vector x. Therefore, Problem 1 is reduced to the problem of finding a sparse vector x satisfying the linear equation in (2), and a solution may be presented by solving the sparse reconstruction problem in (5) by regarding C and $C1_n - s$ as A and b. Note again that the minimization of $\|x\|_1$ plays a role in finding the sparse vector x.

Let us show an example. Consider the DR with $n = 50$ and $m = 24$. The scheduled amounts of negawatt energy c_{ij} ($i = 1, 2, \ldots, 50$, $j = 1, 2, \ldots, 24$) are randomly generated from the uniform distribution on the interval $(0, 1]$. The (true) failure rates x_i ($i = 1, 2, \ldots, 50$) are given in Fig. 3 (a), where participants $10, 20, \ldots, 50$ are defaulting. Note that the choice of defaulting participants does not affect the following result because the defaults of one participant and another are independent, i.e., the default of one participant does not affect the default of another. The total amounts of negawatt energy s_j ($j = 1, 2, \ldots, 24$) are shown in Fig. 3 (b). Fig. 3 (c) shows the results estimated by sparse reconstruction, i.e., the ℓ_1-optimization problem in (5) with $A := C$ and $b := C1_n - s$, where the command *linprog* of MATLAB is used to solve the ℓ_1-optimization problem. We see that the sparse reconstruction technique exactly estimates the failure rates x_i ($i = 1, 2, \ldots, 50$). In this case, the estimation error

$$E := \frac{\|\tilde{x} - x^*\|_2}{\|x^*\|_2} \tag{6}$$

is nearly equal to 3.617×10^{-15} (which is almost zero), where $x^* \in [0, 1]^n$ is the collective vector of the true failure rates and $\tilde{x} \in [0, 1]^n$ is the estimation result.

Next, consider the case with $n = 1000$ and $m = 24$. The scheduled amounts c_{ij} ($i = 1, 2, \ldots, 1000$, $j = 1, 2, \ldots, 24$) are randomly generated in a similar manner as the former case. The failure rates x_i ($i = 1, 2, \ldots, 1000$) are given as Fig. 4 (a), where participants $100, 200, \ldots, 1000$ are defaulting. The total amounts s_j ($j = 1, 2, \ldots, 24$) are shown in Fig. 4 (b). The results estimated by sparse reconstruction are shown in Fig. 4 (c). In this case, the solution x is sparse but different from the true failure rate vector x in Fig. 4 (a). In fact, the estimation error defined as (6) is nearly equal to 1.108, which is much larger than the value of the former case.

Except for the singular case for the scheduled negawatt table C, e.g., $\mathrm{rank}(C) < m$, the reason for such an incorrect result is that $m \ll n$, i.e., there are too few time slots for the number of participants. Even if the sparsity of x is assumed, the number of equations contained in (2) is not sufficient to determine x, which results in an incorrect solution. Meanwhile, it is typical that $m \ll n$ in DR. This motivates us to develop a new solution to Problem 1.

IV. Default Detection by Sparse Reconstruction With Individual Inspection

A. Default Detection Method and Its Performance

As shown in Section III-B, the exact solution is not always obtained by the straightforward application of sparse reconstruction to Problem 1 owing to the ratio of unknowns to equations (i.e., $m \ll n$). On the other hand, the aggregator can inspect the actual negawatt energy of an arbitrarily selected participant via its smart meter, as stated in Section II-A. Thus, by the inspection of participant i, we can directly obtain the information of the failure rate x_i, i.e., part of the solution x. This improves the ratio of unknowns to equations, which brings us closer to the exact solution. On the basis of this idea, we propose a method incorporating the inspection of actual negawatt energy into sparse reconstruction.

Here, we denote the true values of the failure rate of participant i and the collective vector by x_i^* and x^*, respectively, i.e., $x^* = [x_1^* \ \ x_2^* \ \cdots \ x_n^*]^\top$. Furthermore, let t denote the discrete time corresponding to the repetition time of the main routine for estimation. Note that t is not the index of time slots and is the time on a relatively faster time scale. Its time granularity is supposed to be subseconds or seconds in the real world.

The idea of the proposed method is to iterate the following two operations for each time t:

(a) Estimate the failure rates of all participants by solving a sparse reconstruction problem formulated in the form in (5).

(b) Inspect the actual negawatt energy of the most suspicious participant indicated by the result of (a), and reflect the inspection result in the sparse reconstruction problem to be solved at the next time.

This is formalized as Algorithm 1.

In this algorithm, $\mathrm{SR}(t)$ is the sparse reconstruction problem to be solved at time t, and $\mathbf{P}(t)$ represents the list of participants whose actual negawatt energy has never been inspected until time t. Step 1 corresponds to the initialization of $\mathrm{SR}(t)$ and $\mathbf{P}(t)$. Step 2 is the main part corresponding to the aforementioned two operations (a) and (b): Step 2-(1) is given for (a) and Steps 2-(2) and 2-(3) are for (b). In particular, the most suspicious participant is selected in Step 2-(2) and is inspected in Step 2-(3), which is agreed to in the second clause of the contract in Section II-A. Moreover, Step 2-(4) updates the sparse reconstruction problem $\mathrm{SR}(t)$ and the list $\mathbf{P}(t)$ for the next time.

It should be remarked that participant $k(t)$ is not certified as a defaulting participant in Step 2-(2), and it is merely selected for inspection by this algorithm. Moreover, the following point should be noted. In Step 2-(3), the negawatt energy is calculated by subtracting the actual consumption from the *baseline*, which is the usual consumption without the DR. The information of the actual consumption is obtained via a smart meter, while the baseline is computed by recently developed methods, e.g., those in [22], [23].

Now, the performance of Algorithm 1 is disclosed. For a subset $p = \{p_1, p_2, \ldots, p_v\}$ of $\{1, 2, \ldots, n\}$, let $C(p) \in \mathbf{R}$ be the matrix resulting from the modification of the matrix C so that the p_1-, p_2-, …, p_v-th column vectors are replaced with zero vectors. For instance, $C(p) = [C_1 \ \ 0 \ \ 0 \ \ C_4]$ for $C = [C_1 \ C_2 \ C_3 \ C_4]$, and $p := \{2, 3\}$, where C_i is the i-th column vector of C.

Algorithm 1 Detection of Defaulting Participants

(Step 1) Let $\mathrm{SR}(0)$ be the ℓ_1-optimization problem in (5) for $A := C$ and $b := C1_n - s$, and let $\mathbf{P}(0) := \{1, 2, \ldots, n\}$.

(Step 2) For each time $t = 0, 1, \ldots, n-1$ (corresponding to each iteration), execute the following operations:

(1) Solve $\mathrm{SR}(t)$ and let $x(t)$ be a solution. If there exist multiple solutions to $\mathrm{SR}(t)$, an arbitrarily selected solution is set to $x(t)$. Let $x_i(t)$ be the i-th element of the vector $x(t)$.

(2) Let $k(t)$ be the index $i \in \mathbf{P}(t)$ of the participant with the largest element of $x(t)$ in the group $\mathbf{P}(t)$, i.e., $x_{k(t)}(t) \ge x_i(t)$ for every $i \in \mathbf{P}(t)$. If there exist multiple participants with the largest element, an arbitrarily selected participant is set to $k(t)$.

(3) Inspect the negawatt energy of participant $k(t)$ via its smart meter, and obtain the information of its true failure rate $x_{k(t)}^*$.

(4) Let $\mathrm{SR}(t+1)$ be the optimization problem resulting from the modification of $\mathrm{SR}(t)$ so that the equality constraint $x_{k(t)} = x_{k(t)}^*$ is embedded as an additional constraint, where $x_{k(t)}$ is the $k(t)$-th element of the variable x of $\mathrm{SR}(t+1)$ and $x_{k(t)}^*$ is a constant number given in (3). Let $\mathbf{P}(t+1) := \mathbf{P}(t) \setminus \{k(t)\}$.

Theorem 1: Consider Problem 1. The following statements hold for Algorithm 1.

(i) Let $\mathbf{C}_{n-m}$ be the set of $(n-m)$ combinations of $\{1, 2, \ldots, n\}$ and assume that

(A1) $\mathrm{rank}(C(p)) = m$ holds for any $p \in \mathbf{C}_{n-m}$.

Then $x(n-m) = x^*$.

(ii) Assume that

(A2) C is a positive matrix (i.e., all elements are positive).

If $x_{k(t)} = 0$ in Step 2-(2) for a time t, then, $x(t) = x^*$.

Proof: See Section IV-C. ■

This theorem guarantees that Algorithm 1 provides the exact solution after a certain number of iterations under the conditions for the scheduled negawatt table C. In particular, (i) gives an upper bound on the number of iterations subject to (A1), and (ii) presents a stopping rule of the algorithm under (A2), which enables the algorithm to be stopped before reaching the upper bound. Therefore, the exact solution is obtained after at most $n - m$ iterations but possibly after a smaller number of iterations.

Four remarks are given for the algorithm.

First, (A1) and (A2) are satisfied in the practice of the DR with rare exceptions. In fact, C is composed of the scheduled negawatt energy of each participant, which implies that c_{ij} $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, m)$ are arbitrarily given positive values and therefore usually nonuniform and unstructured.

Second, one may consider that Theorem 1 does not hold for the case where $\mathrm{SR}(t)$ may have multiple solutions (e.g., in the situation where $Ax = b$ is parallel with one side of the tilted square in Fig. 2); however, this is not the case because the theorem is proven only under (A1) and (A2), as shown in Section IV-C. Meanwhile, if $\mathrm{SR}(t)$ has multiple solutions, $x(t)$ intermediately generated in the algorithm may not be sparse. Thus, when the true value x^* is sparse, the stopping rule is

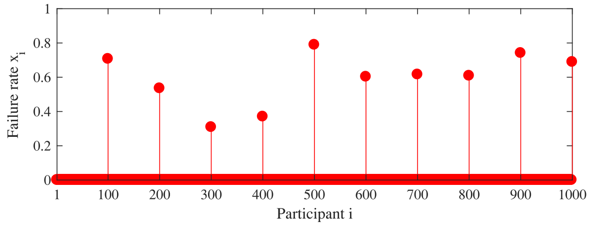

Fig. 5. Results estimated by the proposed method ($n = 1000$).

unlikely to be satisfied. Consequently, the number of iterations will be large.

Third, Theorem 1 makes the proposed method preferable for both the aggregator and participants. In fact, the theorem guarantees that the number of actual inspections is at most $n - m$ and often lower (actually, significantly lower as shown later). This property avoids a large amount of data traffic, which will be an advantage for the aggregator. Moreover, it reduces the chance of inspection for each participant, which must alleviate participants' feeling of being continuously monitored by the aggregator.

Finally, the defaulting participants are determined by extracting the indices with a nonzero value from the resulting $x(t)$ and the normal participants are its complement.

B. Simulations

The proposed method is demonstrated by simulations. As stated in Section I, there exists no method for detecting defaulting participants in DR except for our method. Thus we show here that the proposed method exactly estimates defaulting participants and their failure rates with a small number of inspections.

1) The Case Where $n = 1000$: Consider the DR with $n = 1000$ in Section III-B. As shown before, the failure rates x_i ($i = 1, 2, \ldots, 1000$) and the total amounts of negawatt energy s_j ($j = 1, 2, \ldots, 24$) are given in Figs. 4 (a) and (b). Moreover, the scheduled negawatt table C satisfies (A1) and (A2).

Fig. 5 shows the results estimated by Algorithm 1. In this case, $n - m = 976$, which is the worst-case number of iterations from Theorem 1. However, the condition $x_{k(t)} = 0$ in Theorem 1 holds for $t = 41$, which stops Algorithm 1 at $t = 41$ and gives the solution $x(41)$ in Fig. 5. By comparing this result with Fig. 4 (a), it turns out that the proposed method exactly estimates the failure rates x_i^* ($i = 1, 2, \ldots, 1000$) with a small number (4.1% of the number of participants) of inspections. In this case, the estimation error defined as (6) is equal to exactly zero because all the defaulting participants are inspected until $t = 41$.

In this example, we used *linprog* of MATLAB to solve the sparse reconstruction problem SR(t). The computation time was within several seconds by a laptop computer with an Intel Core i7-7500U and 16GB of memory.

2) The Case Where $n = 10000$: Next, let us consider the case with $n = 10000$, i.e., ten times as many participants, and $m = 24$. The scheduled amounts of negawatt energy c_{ij} ($i = 1, 2, \ldots, 10000$, $j = 1, 2, \ldots, 24$) are given in a similar manner as the case where $n = 1000$ so that (A1) and

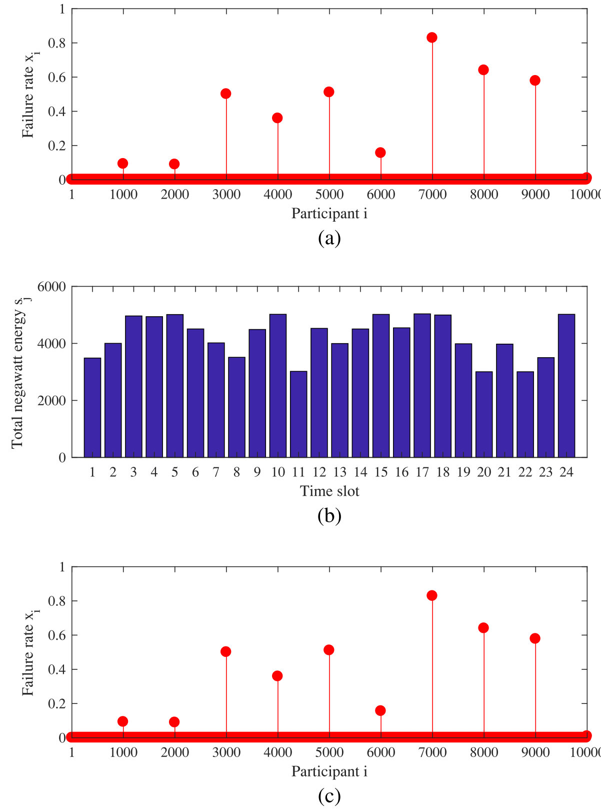

Fig. 6. Results estimated by the proposed method ($n = 10000$). (a) Failure rates x_i. (b) Total amounts of negawatt energy s_j. (c) Estimation of x_i.

(A2) hold. The failure rates x_i ($i = 1, 2, \ldots, 10000$) are given in Fig. 6 (a), where participants $1000, 2000, \ldots, 10000$ are defaulting (the failure rate of participant 10000 is small). The total amounts of negawatt energy s_j ($j = 1, 2, \ldots, 24$) are given in Fig. 6 (b).

Fig. 6 (c) shows the results estimated by Algorithm 1, where $x_{k(t)} = 0$ holds for $t = 264$ and the solution $x(264)$ is depicted. Also in this case, the estimation error is equal to exactly zero for the same reason as above. For this large-scale DR, we see that our method exactly estimates the failure rates x_i^* ($i = 1, 2, \ldots, 10000$) with a small number (2.64% of n) of inspections. The computation time was within 3 min by the same computer.

3) Performance Evaluation by Monte Carlo Simulation: Finally, the performance of the proposed method is evaluated by Monte Carlo simulation.

Fig. 7 shows the relation between the number of inspections in Algorithm 1 and the number of participants n. The figure shows a box plot based on 100 trials for each $n \in \{200, 400, \ldots, 1000\}$, where $m = 24$; the *default rate*, i.e., the ratio of the number of defaulting participants to n, is 10%; the scheduled amounts of negawatt energy c_{ij} ($i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, 24$) are randomly generated from the uniform distribution on $(0, 1]$; and the (true) failure rates x_i ($i = 1, 2, \ldots, n$) are given so that $x_1, x_2, \ldots, x_{0.1n}$ are independently generated from the uniform distribution on $(0, 1]$ and the others are equal to zero (for which the default rate is 10%). Note here that we have no prior information for the

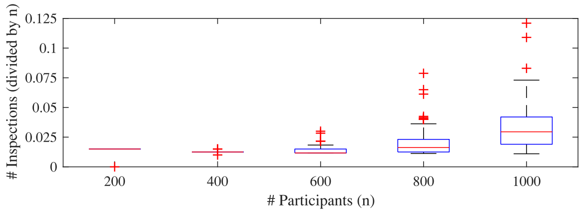

Fig. 7. Relation between the number of inspections and the number of participants (default rate 10%).

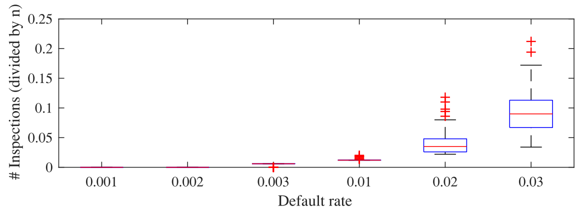

Fig. 8. Relation between the number of inspections and the default rate ($n = 500$).

scheduled negawatt energy at present and thus we use the uniform distribution aiming at evenly examining all possibilities of the scheduled amounts of negawatt energy. We see from the figure that the number of inspections increases with the number of participants. This result suggests that it may be difficult to use the proposed algorithm for very large n. In this case, it is practical to break the problem down into several subproblems with small groups and apply the proposed method to each subproblem.

Fig. 8, on the other hand, depicts the relation between the number of inspections and the default rate, where $n = 500$, $m = 24$, and c_{ij} ($i = 1, 2, \ldots, 500$, $j = 1, 2, \ldots, 24$) and x_i ($i = 1, 2, \ldots, 500$) are given in the same manner as before. The box plot is based on 100 trials for each default rate. It turns out that the number of inspections rapidly increases with the default rate. This is because the proposed method utilizes the sparsity of the (true) failure rate vector x, and the level of sparsity decreases as the default rate increases. Thus, in our framework, the aggregator must design the DR program (e.g., penalties and incentives) so that the default rate is sufficiently small.

We conducted the same experiments (based on 100 trials for each parameter) 10 times with different seeds for the random number generator. All the results qualitatively agreed with the results in Figs. 7 and 8, in terms of the range of the vertical axis and the monotone increasing property with respect to the parameters. Therefore, it is concluded that the above results exhibit the performance of the proposed method.

C. Proof of Theorem 1

1) Preliminary: First, a preliminary result is given.

Lemma 1: Consider Algorithm 1. For any $t \in \{0, 1, \ldots, n-1\}$, the true solution x^* is feasible for the optimization problem SR(t).

Proof: This is proven by using mathematical induction.

The constraint of SR(0) is given by (2). Moreover, x^* satisfies (2) by definition. Hence, x^* is feasible for SR(0).

We next show that x^* is feasible for SR($t+1$) if x^* is feasible for SR(t). By noting Step 2-(4) in Algorithm 1, the constraint of SR($t+1$) is composed of the constraint of SR(t) and $x_{k(t)} = x^*_{k(t)}$, which is the $k(t)$-th equation of the vector equation $x = x^*$. Thus, x^* is feasible for SR($t+1$). ■

2) Statement (i): By the definition of SR(0) in Step 1 and the operation in Step 2-(4), the constraint of SR($n-m$) is composed of (2) and $x_{k(t)} = x^*_{k(t)}$ ($t = 0, 1, \ldots, n-m-1$). That is, the constraint of SR($n-m$) is expressed as

$$\begin{bmatrix} C \\ e_{k(0)} \\ e_{k(1)} \\ \vdots \\ e_{k(n-m-1)} \end{bmatrix} x(k) = \begin{bmatrix} C1_n - s \\ x^*_{k(0)} \\ x^*_{k(1)} \\ \vdots \\ x^*_{k(n-m-1)} \end{bmatrix}, \tag{7}$$

where $e_{k(t)} \in \mathbf{R}^{1\times n}$ is the $k(t)$-th standard basis in the space $\mathbf{R}^{1\times n}$. Note here that the numbers $k(0), k(1), \ldots, k(n-m-1)$ are different from each other because $k(t) \in \mathbf{P}(t)$ from Step 2-(2) and $k(t) \notin \mathbf{P}(t+1)$ from Step 2-(4).

Since only the $k(t)$-th element is nonzero in the vector $e_{k(t)}$, the matrix on the left-hand side of (7) can be transformed into

$$\begin{bmatrix} C(p) \\ e_{k(0)} \\ e_{k(1)} \\ \vdots \\ e_{k(n-m-1)} \end{bmatrix}$$

by elementary row operations, where $p := (k(0), k(1), \ldots, k(n-m-1))$ and the matrix $C(p)$ has zero vectors at the $k(0)$-, $k(1)$-, ..., $k(n-m-1)$-th columns. The rank of this matrix is equal to n subject to (A1); thus, the rank of the original matrix in (7) is equal to n under (A1). Therefore, it follows that there exists a unique solution $x(k)$ to (7). This fact and Lemma 1 complete the proof.

3) Statement (ii): Consider the time t satisfying $x_{k(t)} = 0$ in Step 2-(2) of Algorithm 1. Let $C_i \in \mathbf{R}^m$ be the i-th column vector of C and $\bar{\mathbf{P}}(t) := \{1, 2, \ldots, n\} \setminus \mathbf{P}(t)$. Note that $\bar{\mathbf{P}}(t)$ is the list of the participants inspected in Step 2-(3) until time t, and $\mathbf{P}(t)$ is that of the participants who have never been inspected until time t.

Statement (ii) is a straightforward consequence of $\mathbf{P}(t) \cup \bar{\mathbf{P}}(t) = \{1, 2, \ldots, n\}$ and the following two facts:

(a) $x_i(t) = x_i^*$ for every $i \in \mathbf{P}(t)$.

(b) $x_i(t) = x_i^*$ for every $i \in \bar{\mathbf{P}}(t)$.

Fact (b) is trivial by the definition of SR(t) and the fact that $\bar{\mathbf{P}}(t)$ is the list of inspected participants.

On the other hand, (a) is proven as follows. By the definitions of x^* and $x(t)$, we have

$$\sum_{i\in\mathbf{P}(t)} C_i x_i^* + \sum_{i\in\bar{\mathbf{P}}(t)} C_i x_i^* = C1_n - s, \tag{8}$$

$$\sum_{i\in\mathbf{P}(t)} C_i x_i(t) + \sum_{i\in\bar{\mathbf{P}}(t)} C_i x_i(t) = C1_n - s. \tag{9}$$

Then $x_i(t) = 0$ for every $i \in \mathbf{P}(t)$ because $x_{k(t)} = 0$ and participant $k(t)$ has the largest element of $x(t)$ in the group

$\mathbf{P}(t)$. Applying this fact and (b) to (9) provides

$$\sum_{i\in\bar{\mathbf{P}}(t)} C_i x_i^* = C1_n - s. \tag{10}$$

Furthermore, (8) and (10) give

$$\begin{aligned}\sum_{i\in\mathbf{P}(t)} C_i x_i^* + \sum_{i\in\bar{\mathbf{P}}(t)} C_i x_i^* &= \left(\sum_{i\in\mathbf{P}(t)} C_i x_i^*\right) + C1_n - s\\ &= C1_n - s;\end{aligned}$$

therefore,

$$\sum_{i\in\mathbf{P}(t)} C_i x_i^* = 0. \tag{11}$$

Equation (11) and (A2) imply (a).

V. Incorporation of Direct Load Control

We next incorporate direct load control for defaulting participants into the default detection method in Section IV.

A. Direct Load Control for Defaulting Participants

The scenario considered here is outlined as follows. In the same manner as Section IV, the aggregator executes the default detection algorithm to estimate the defaulting participants as well as their failure rates. If a defaulting participant is found in the inspection step (Step 2-(3)), the aggregator directly operates the facilities of the participants to reduce the failure rates to zero. This control is assumed to be agreed to in the contracts.

Let t denote the time immediately after time slot m. Then, $t = 0$ corresponds to the start time of control. Let $\hat{x}_i^*(t)$ be the failure rate of participant i at time $t \geq 0$, and its initial value $\hat{x}_i^*(0)$ is given as

$$\hat{x}_i^*(0) = x_i^* \tag{12}$$

for the (true) failure rate x_i^* before control. We assume that, at time $t \geq 0$, the aggregator can control the consumption of any participant i to be $\hat{x}_i^*(t+1) = 0$ by operating the facilities of participant i. This is modeled as

$$\hat{x}_i^*(t+1) = \begin{cases} 0 & \text{if the aggregator controls the} \\ & \text{consumption of participant } i \\ \hat{x}_i^*(t) & \text{otherwise.} \end{cases} \tag{13}$$

Then, the algorithm incorporating direct load control, called Algorithm 2, is given by modifying Algorithm 1 so that the following operation is inserted between (3) and (4) in Step 2.

(3') If $x_{k(t)}^* \neq 0$, operate the facilities of participant $k(t)$ to reduce the failure rate $\hat{x}_{k(t)}^*(t+1)$ to zero.

By considering that the condition $x_{k(t)}^* \neq 0$ implies that participant $k(t)$ is defaulting, the above operation is direct load control of a defaulting participant. Note that Algorithm 2 is executed with (12) and (13). Note also that $\hat{x}_i^*(0)$ is unknown to the aggregator before executing the algorithm because x_i^* is unknown.

We obtain the following result for Algorithm 2.

Theorem 2: Consider Problem 1. Let $\hat{x}^*(t) := [\hat{x}_1^*(t)\ \hat{x}_2^*(t)\ \cdots\ \hat{x}_n^*(t)]^\top$. The following statements hold for Algorithm 2 with (12) and (13):

(i) $\hat{x}^*(n) = 0$.

(ii) Assume (A2) in Theorem 1. If $x_{k(t)} = 0$ in Step 2-(2) for a time t, then $\hat{x}^*(t) = 0$.

Proof: (i) In Algorithm 2, $\mathbf{P}(0) = \{1, 2, \ldots, n\}$ in Step 1, $k(t)$ is picked from $\mathbf{P}(t)$ in Step 2-(2), and $\mathbf{P}(t+1) = \mathbf{P}(t) \setminus \{k(t)\}$ in Step 2-(4). Thus we have $\{k(0), k(1), \ldots, k(n-1)\} = \{1, 2, \ldots, n\}$, which implies that Step 2-(3') is executed for all participants until $t = n - 1$. This proves (i).

(ii) From Theorem 1 (ii) (in which (A2) is assumed), it is clear that Steps 2-(3) and 2-(3') are executed for all defaulting participants until $x_{k(t)} = 0$. This fact and the operation of Step 2-(3') mean that $\hat{x}^*(t) = 0$ when $x_{k(t)} = 0$. ∎

Note that $\hat{x}^*(t) = 0$ implies that no defaulting participant exists at time t. This theorem states that Algorithm 2 reduces the number of defaulting participants to zero after at most n inspections. In particular, (ii) presents the same stopping rule as given for Algorithm 1, which guarantees that the number defaulting participants becomes zero when $x_{k(t)} = 0$.

B. Simulation

Consider the DR with $n = 1000$ in Sections III-B and IV-B1. Similar to the results in Section IV-B1, the condition $x_{k(t)} = 0$ in Theorem 2 holds for $t = 41$.

Figs. 9 and 10 show the results of Algorithm 2. The former depicts several snapshots of the controlled failure rates $x_i^*(t)$ $(i = 1, 2, \ldots, 1000)$, and the latter illustrates the number of (actually) defaulting participants for each time t. It turns out that the defaulting participants decrease with the increase in t.

VI. Conclusion

A default detection problem for the DR has been discussed. By imposing the assumption that a few participants are defaulting in contract-based DR, we have established a detection method based on sparse reconstruction. The method is iterative, and each iteration improves the sparse reconstruction by including the inspection data from the previous iteration. We have proven that, under mild conditions, the method derives the exact solution in a finite number of inspections. Moreover, a stopping rule has been presented, which enables us to solve the problem with a small number of inspections. Finally, direct load control is incorporated into the detection method, which is useful for eliminating the defaulting participants in the DR.

In this paper, we have assumed that the failure rates of the participants are time-invariant for a certain period. Such an assumption is reasonable if defaulting occurs by some instrument faults. On the other hand, in practice, there is a possibility that the failure rates are time-varying, i.e., defaulting occurs intermittently. Our framework will be extended to such a case in the future. Moreover, our method should be improved to reduce the number of inspections by incorporating additional prior knowledge of the participants. It is also

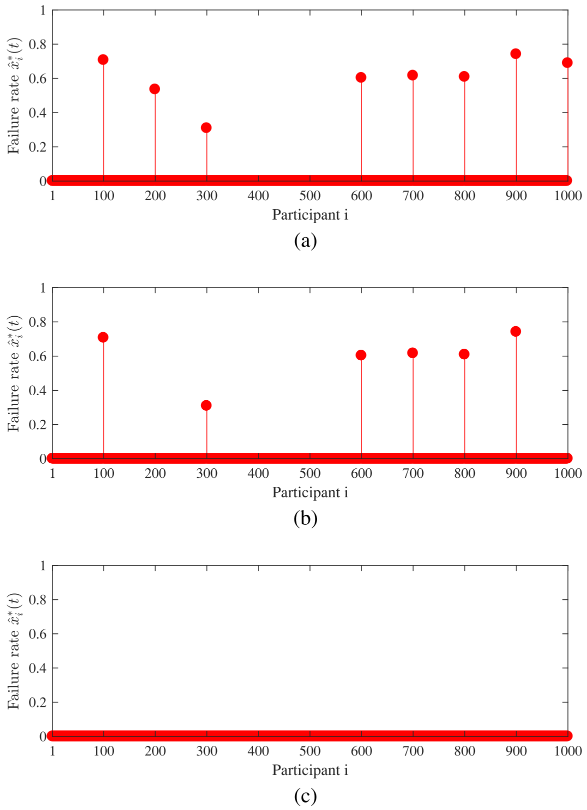

Fig. 9. Control results obtained by the proposed method. (a) $t = 15$. (b) $t = 30$. (c) $t = 41$.

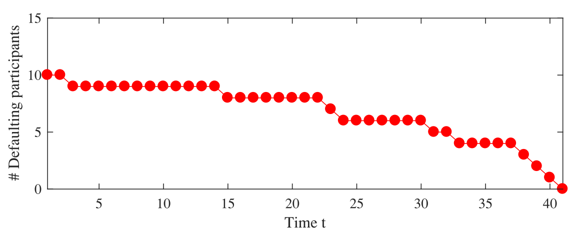

Fig. 10. Number of defaulting participants.

interesting to address default detection for bidding-based DR, e.g., in [24].

REFERENCES

[1] "2012 assessment of demand response and advanced metering," Federal Energy Regulatory Commission, Washington, DC, USA, Staff Rep., 2012. [Online]. Available: https://www.ferc.gov/legal/staff-reports/12-20-12-demand-response.pdf

[2] T. Samad, E. Koch, and P. Stluka, "Automated demand response for smart buildings and microgrids: The state of the practice and research challenges," *Proc. IEEE*, vol. 104, no. 4, pp. 726–744, Apr. 2016.

[3] J. Hansen, J. Knudsen, and A. M. Annaswamy, "Demand response in smart grids: Participants, challenges, and a taxonomy," in *Proc. IEEE Conf. Decis. Control*, Los Angeles, CA, USA, 2014, pp. 4045–4052.

[4] S. Althaher, P. Mancarella, and J. Mutale, "Automated demand response from home energy management system under dynamic pricing and power and comfort constraints," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1874–1883, Jul. 2015.

[5] M. H. Albadi and E. F. El-Saadany, "A summary of demand response in electricity markets," *Elect. Power Syst. Res.*, vol. 78, no. 11, pp. 1989–1996, 2008.

[6] F. Liu, H. Jiang, Y. M. Lee, J. Snowdon, and M. Bobker, "Statistical modeling for anomaly detection, forecasting and root cause analysis of energy consumption for a portfolio of buildings," in *Proc. 12th Int. Conf. Int. Build. Perform. Simulat. Assoc.*, 2011, pp. 2530–2537.

[7] M. Wrinch, T. H. M. El-Fouly, and S. Wong, "Anomaly detection of building systems using energy demand frequency domain analysis," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, San Diego, CA, USA, 2012, pp. 1–6.

[8] J. Ploennigs, B. Chen, A. Schumann, and N. Brady, "Exploiting generalized additive models for diagnosing abnormal energy use in buildings," in *Proc. 5th ACM Workshop Embedded Syst. Energy Efficient Build.*, Rome, Italy, 2013, pp. 1–8.

[9] R. Fontugne *et al.*, "Strip, bind, and search: A method for identifying abnormal energy consumption in buildings," in *Proc. 12th Int. Conf. Inf. Process. Sensor Netw.*, Philadelphia, PA, USA, 2013, pp. 129–140.

[10] J.-S. Chou and A. S. Telaga, "Real-time detection of anomalous power consumption," *Renew. Sustain. Energy Rev.*, vol. 33, pp. 400–411, May 2014.

[11] H. Janetzko, F. Stoffel, S. Mittelstädt, and D. A. Keim, "Anomaly detection for visual analytics of power consumption data," *Comput. Graph.* vol. 38, pp. 27–37, Feb. 2014.

[12] P. Arjunan, H. D. Khadilkar, T. Ganu, Z. M. Charbiwala, A. Singh, and P. Singh, "Multi-user energy consumption monitoring and anomaly detection with partial context information," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy Efficient Built Environ.*, Seoul, South Korea, 2015, pp. 35–44.

[13] D. B. Araya, K. Grolinger, H. F. ElYamany, M. A. M. Capretz, and G. T. Bitsuamlak, "An ensemble learning framework for anomaly detection in building energy consumption," *Energy Build.*, vol. 144, no. 1, pp. 191–206, Jun. 2017.

[14] M. De Nadai and M. van Someren, "Short-term anomaly detection in gas consumption through ARIMA and artificial neural network forecast," in *Proc. IEEE Workshop Environ. Energy Struct. Monitor. Syst.*, Trento, Italy, 2015, pp. 250–255.

[15] X. Liu, N. Iftikhar, P. S. Nielsen, and A. Heller, "Online anomaly energy consumption detection using Lambda architecture," in *Proc. Int. Conf. Big Data Anal. Knowl. Disc.*, Porto, Portugal, 2016, pp. 193–209.

[16] Y. Zhang, W. Chen, and J. Black, "Anomaly detection in premise energy consumption data," in *Proc. Power Energy Soc. Gen. Meeting*, San Diego, CA, USA, 2011, pp. 1–8.

[17] P. Mahya, H. Tahayori, and A. Sadeghian, "An online demand response EMS with anomaly usage detection," in *Proc. IEEE Int. Conf. Smart Energy Grid Eng.*, Oshawa, ON, Canada, 2017, pp. 271–275.

[18] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.

[19] K. Hayashi, M. Nagahara, and T. Tanaka, "A user's guide to compressed sensing for communications systems," *IEICE Trans. Commun.*, vol. E96-B, no. 3, 685–712, 2013.

[20] I. Rish and G. Grabarnik, *Sparse Modeling: Theory, Algorithms, and Applications*. Baton Rouge, LA, USA: CRC Press, 2014.

[21] E. Arai and Y. Ueda, "Development of simple estimation model for aggregated residential load by using temperature data in multi-region," in *Proc. 4th Int. Conf. Renew. Energy Res. Appl.*, 2015, pp. 772–776.

[22] "The demand response baseline," Boston, MA, USA, EnerNOC Inc., White Paper, 2009.

[23] S. Park, S. Ryu, Y. Choi, J. Kim, and H. Kim, "Data-driven baseline estimation of residential buildings for demand response," *Energies*, vol. 8, no. 9, pp. 10239–10259, 2015.

[24] S. Lee *et al.*, "Demand response prospects in the South Korean power system," in *Proc. IEEE PES Gen. Meeting*, Providence, RI, USA, 2010, pp. 1–6.

Shun-Ichi Azuma (S'03–M'05–SM'15) received the B.E. degree in electrical engineering from Hiroshima University, Higashihiroshima, Japan, in 1999 and the M.E. and Ph.D. degrees in control engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2001 and 2004, respectively. He was an Assistant Professor with the Department of Systems Science, Graduate School of Informatics, Kyoto University, Uji, Japan, from 2005 to 2011, where he was an Associate Professor from 2011 to 2017. He is currently a Professor with Nagoya University. His research interest includes analysis and control of hybrid systems. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS since 2013, *Automatica* since 2014, and the IEEE CONTROL SYSTEMS LETTERS since 2016.

Daisuke Sato received the B.E. degree in mechanical and aerospace engineering from Nagoya University, Japan, in 2018, where he is currently pursuing the master's degree with the Department of Mechanical Systems Engineering, Graduate School of Engineering.

Koichi Kobayashi (M'07) received the B.E. and M.E. degrees from Hosei University, Japan, in 1998 and 2000, respectively, and the D.E. degree from the Tokyo Institute of Technology, Japan, in 2007. From 2000 to 2004, he was with Nippon Steel Corporation, Japan. From 2007 to 2015, he was an Assistant Professor with the Japan Advanced Institute of Science and Technology. Since 2015, he has been an Associate Professor with the Graduate School of Information Science and Technology, Hokkaido University, Japan. His research interests include discrete event and hybrid systems.

Nobuyuki Yamaguchi (M'02) received the Ph.D. degree in engineering from Hokkaido University, Japan, in 2002. From 2002 to 2015, he was a Research Scientist with the Central Research Institute of Electric Power Industry, Japan. Since 2015, he has been with the Department of Electrical Engineering, Tokyo University of Science, Japan, where he is currently an Associate Professor. He is a member of the Institute of Electrical Engineers of Japan.