Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 906717, 8 pages
http://dx.doi.org/10.1155/2014/906717

Research Article

A Probabilistic Approach to Control of Complex Systems and Its Application to Real-Time Pricing

Koichi Kobayashi and Kunihiko Hiraishi

School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan

Correspondence should be addressed to Koichi Kobayashi; k-kobaya@jaist.ac.jp

Received 16 July 2014; Accepted 1 September 2014; Published 17 September 2014

Academic Editor: Rajan Rakkiyappan

Control of complex systems is one of the fundamental problems in control theory. In this paper, a control method for complex systems modeled by a probabilistic Boolean network (PBN) is studied. A PBN is widely used as a model of complex systems such as gene regulatory networks. For a PBN, the structural control problem is newly formulated. In this problem, a discrete probability distribution appeared in a PBN is controlled by the continuous-valued input. For this problem, an approximate solution method using a matrix-based representation for a PBN is proposed. Then, the problem is approximated by a linear programming problem. Furthermore, the proposed method is applied to design of real-time pricing systems of electricity. Electricity conservation is achieved by appropriately determining the electricity price over time. The effectiveness of the proposed method is presented by a numerical example on real-time pricing systems.

1. Introduction

Analysis and control of complex systems such as power systems and gene regulatory networks are one of the fundamental problems in control theory of large-scale systems. In order to deal with such complex systems, it is one of the appropriate methods to approximate a complex system by a discrete abstract model (see, e.g., [1]). On the other hand, human decision making is also complex and is modeled by a discrete model (see, e.g., [2]). Thus, in analysis and control of complex systems and those with human decision making, a discrete model plays an important role.

Several discrete models such as Petri nets, Bayesian networks, automata-based models, and Boolean networks have been proposed so far (see, e.g., [3]). In this paper, we focus on a Boolean network (BN) [4]. In a BN, the state is given by a binary value (0 or 1), and the dynamics are expressed by a set of Boolean functions. Since Boolean functions are used, it is easy to understand the interaction between states. In the field of theoretical biology, there is a criticism that a BN is too simple as a model of gene regulatory networks (see, e.g., [5]), but a BN can be relatively applied to large-scale systems. In addition, since the behavior of complex systems is frequently stochastic by the effects of noise, it is appropriate that a Boolean function is randomly decided at each time among the candidates of Boolean functions. Thus, a probabilistic BN (PBN) has been proposed in [6]. Furthermore, a context-sensitive PBN (CS-PBN) in which the deciding time is randomly selected has been proposed as a general form of PBNs [7, 8]. In this paper, we adopt a probabilistic Boolean network (PBN) as a mathematical model of complex systems.

For a given PBN, we consider the structural control problem (see, e.g., [9–11]). In this problem, a discrete probability distribution is controlled. For example, in [9], a discrete probability distribution at each time is selected among a given set. In this paper, we consider fine control of a discrete probability distribution by using the continuous-valued input. For a newly formulated problem, we propose an approximate solution method. First, a matrix-based representation of BNs proposed in [12] is extended to that of PBNs. Next, using the obtained representation, the original problem is approximated by a linear programming (LP) problem.

Furthermore, as one of the applications, we consider a design method of real-time pricing systems (see, e.g., [13–16]). A real-time pricing system of electricity is a system that charges different electricity prices for different hours of the day and for different days, and is effective for reducing the peak and flattening the load curve. In general, a real-time pricing system consists of one controller deciding the price at each time and multiple electric customers such as commercial facilities and homes. If electricity conservation is needed, then the price is set to a high value. Since the economic load becomes high, customers conserve electricity. Thus, electricity conservation is achieved. In the existing methods, the price at each time is given by a simple function with respect to power consumptions and voltage deviations and so on (see, e.g., [16]). To the best of our knowledge, decision making of customers has not been explicitly considered so far. In order to realize more precisely pricing, it is necessary to use a mathematical model of customers. Thus, decision making of customers is modeled by a PBN, and the problem of finding the price at each time is formulated as a structural control problem. The price corresponds to the continuous-valued input. By a numerical example, the effectiveness of the proposed method is presented.

The proposed framework provides us a basic method for control of complex systems using PBNs.

Notation. For the n-dimensional vector $x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top$ and the index set $\mathcal{I} = \{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$, define $[x_i]_{i\in\mathcal{I}} := \begin{bmatrix} x_{i_1} & x_{i_2} & \cdots & x_{i_m} \end{bmatrix}^\top$. For two matrices A and B, let $A \otimes B$ denote the Kronecker product of A and B. In addition, for q vectors $y_1, y_2, \ldots, y_q$ and the index set $\mathcal{J} = \{j_1, j_2, \ldots, j_p\} \subseteq \{1, 2, \ldots, q\}$, define $\bigotimes_{j\in\mathcal{J}} y_j := y_{j_1} \otimes y_{j_2} \otimes \cdots \otimes y_{j_p}$. For example, for q two-dimensional vectors $z_1, z_2, \ldots, z_q$ and $\mathcal{J} = \{1, 5\}$, we can obtain

$$\begin{aligned} \bigotimes_{j\in\mathcal{J}} z_j &= z_1 \otimes z_5 \\ &= \begin{bmatrix} z_1^{(1)} \\ z_1^{(2)} \end{bmatrix} \otimes \begin{bmatrix} z_5^{(1)} \\ z_5^{(2)} \end{bmatrix} \\ &= \begin{bmatrix} z_1^{(1)} z_5^{(1)} \\ z_1^{(1)} z_5^{(2)} \\ z_1^{(2)} z_5^{(1)} \\ z_1^{(2)} z_5^{(2)} \end{bmatrix}, \end{aligned} \tag{1}$$

where $z_j^{(i)}$ is the ith element of z_j. Finally, let $1_{m\times n}$ denote the $m \times n$ matrix whose elements are all one.

2. Probabilistic Boolean Network

First, we explain a (deterministic) Boolean network (BN). A BN is defined by

$$\begin{aligned} x_1(k+1) &= f^{(1)}\left(\left[x_j(k)\right]_{j\in\mathcal{N}^{(1)}}\right), \\ x_2(k+1) &= f^{(2)}\left(\left[x_j(k)\right]_{j\in\mathcal{N}^{(2)}}\right), \\ &\vdots \\ x_n(k+1) &= f^{(n)}\left(\left[x_j(k)\right]_{j\in\mathcal{N}^{(n)}}\right), \end{aligned} \tag{2}$$

where $x := \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^\top \in \{0,1\}^n$ is the state, and $k = 0, 1, 2, \ldots$ is the discrete time. The set $\mathcal{N}^{(i)} \subseteq \{1, 2, \ldots, n\}$ is a given index set, and the function $f_i : \{0,1\}^{|\mathcal{N}^{(i)}|} \to \{0,1\}^1$ is a given Boolean function consisting of logical operators such as AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). If $\mathcal{N}^{(i)} = \emptyset$ holds, then $x_i(k+1)$ is uniquely determined as 0 or 1.

Next, we explain a probabilistic Boolean network (PBN) (see [6] for further details). In a PBN, the candidates of $f^{(i)}$ are given, and for each x_i, selecting one Boolean function is probabilistically independent at each time. Let

$$f_l^{(i)}\left(\left[x_j(k)\right]_{j\in\mathcal{N}_l^{(i)}}\right), \quad l = 1, 2, \ldots, q(i) \tag{3}$$

denote the candidates of $f^{(i)}$. The probability that $f_l^{(i)}$ is selected is defined by

$$c_l^{(i)} := \mathrm{Prob}\left(f^{(i)} = f_l^{(i)}\right). \tag{4}$$

Then, the following relation

$$\sum_{l=1}^{q(i)} c_l^{(i)} = 1 \tag{5}$$

must be satisfied. Probabilistic distributions are derived from experimental results. Finally, $\mathcal{N}_i$, $i = 1, 2, \ldots, n$ are defined by

$$\mathcal{N}_i := \bigcup_{l=1}^{q(i)} \mathcal{N}_l^{(i)}. \tag{6}$$

We present a simple example.

Example 1. Consider the PBN in which Boolean functions and probabilities are given by

$$\begin{aligned} f^{(1)} &= \begin{cases} f_1^{(1)} = x_3(k), & c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_3(k), & c_2^{(1)} = 0.2, \end{cases} \\ f^{(2)} &= f_1^{(2)} = x_1(k) \wedge \neg x_3(k), \quad c_1^{(2)} = 1.0, \\ f^{(3)} &= \begin{cases} f_1^{(3)} = x_1(k) \wedge \neg x_2(k), & c_1^{(3)} = 0.7, \\ f_2^{(3)} = x_2(k), & c_2^{(3)} = 0.3, \end{cases} \end{aligned} \tag{7}$$

where $q(1) = 2$, $q(2) = 1$, and $q(3) = 2$ hold, $\mathcal{N}_1 = \{3\}$, $\mathcal{N}_2 = \{1, 3\}$, and $\mathcal{N}_3 = \{1, 2\}$ hold, and we see that the relation (5) is satisfied. Next, consider the state trajectory. Then, for $x(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top$, we obtain

$$\begin{aligned} \mathrm{Prob}\left(x(1) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top\right) = 0.8, \\ \mathrm{Prob}\left(x(1) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top\right) = 0.2. \end{aligned} \tag{8}$$

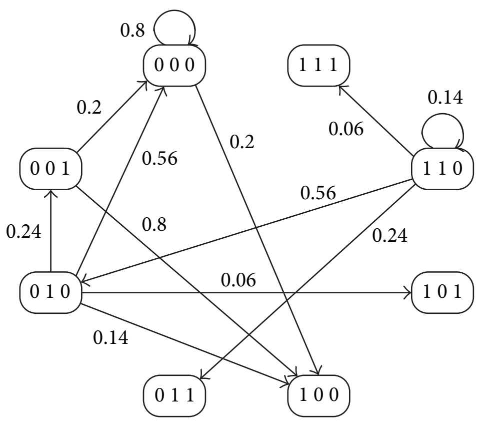

Figure 1: State transition diagram.

In this example, the cardinality of the finite state set $\{0,1\}^3$ is given by $2^3 = 8$, and we obtain the state transition diagram of Figure 1 by computing the transition from each state. In Figure 1, the number assigned to each node denotes x_1, x_2, x_3 (elements of the state), and the number assigned to each arc denotes the transition probability from some state to other state. Note here that, for simplicity, the state transition from only $x(k) = [0\ 0\ 0]^\top, [0\ 0\ 1]^\top, [0\ 1\ 0]^\top, [1\ 1\ 0]^\top$ is illustrated in Figure 1.

3. Problem Formulation

In this section, we formulate the control problem studied in this paper. In the conventional control problem, the control input is added to a given Boolean function. For example, the control input is added as follows: $f^{(i)}([x_j(k)]_{j\in\mathcal{N}^{(i)}}, u(k))$, $u(k) \in \{0,1\}^1$. In general, we assume that the value of the control input can be arbitrarily given. However, there is a possibility that there exists no control input satisfying this assumption. In control of gene regulatory networks, a structural control (or structural intervention) method for PBNs has been proposed so far (see, e.g., [9, 10]). For example, in [9], the discrete probabilistic distribution is switched at each time. In other words, the discrete probabilistic distribution is selected from the set of candidates. On the other hand, in complex systems such as gene regulatory networks, power systems, and social systems, it will be desirable to consider a weaker control method. Thus, in this paper, we consider fine control of probabilities in a discrete probabilistic distribution. This control method can be regarded as a kind of structural control methods.

In the structural control problem formulated here, we assume that the probability $c_l^{(i)}$ in (4) is given by

$$c_l^{(i)}(k) = a_l^{(i)} + b_l^{(i)} u_i(k), \tag{9}$$

where $u := [u_1\ u_2\ \cdots\ u_n]^\top \in [\underline{u}_1, \overline{u}_1] \times [\underline{u}_2, \overline{u}_2] \times \cdots \times [\underline{u}_n, \overline{u}_n] \subseteq \mathscr{R}^n$ is the control input. The set $[\underline{u}_i, \overline{u}_i]$ expresses the input constraint, and $\underline{u}_i, \overline{u}_i \in \mathscr{R}^1$ are given in advance. The parameter $b_l^{(i)}$ expresses elasticity of the probability to the control input. Finding $a_l^{(i)}$ and $b_l^{(i)}$ is the important problem, and will be focused on in future efforts. Of course, we must find $u_i(k)$ such that $c_l^{(i)}(k)$ satisfies (5) and

$$0 \le a_l^{(i)} + b_l^{(i)} u_i(k) \le 1, \quad l = 1, 2, \ldots, q(i). \tag{10}$$

In addition, the dimension of the control input may be less than the dimension n of the state.

Under the above preparation, we consider the following problem.

Problem 2. Suppose that for the PBN with (9), the lower and upper bounds of input constraints $\underline{u}_i$, $\overline{u}_i$, and the initial state $x(0) = x_0$ are given. Then, find a control input sequence $u(0), u(1), \ldots, u(N-1) \in [\underline{u}_1, \overline{u}_1] \times [\underline{u}_2, \overline{u}_2] \times \cdots \times [\underline{u}_n, \overline{u}_n]$ minimizing the cost function

$$\begin{aligned} J = E\Bigg[&\sum_{k=0}^{N-1} \{Qx(k) + Ru(k)\} \\ &+ Q_f x(N) \mid x(0) = x_0 \Bigg] \end{aligned} \tag{11}$$

under the constraints (5) and (10), where $Q, Q_f \in \mathscr{R}^{1\times n}$, $R \in \mathscr{R}^{1\times m}$ are weighting vectors whose element is a nonnegative real number, and $E[\cdot \mid \cdot]$ denotes a conditional expected value.

The linear cost function (11) is appropriate from the following reason. For a binary variable $\delta \in \{0,1\}$, the relation $\delta^2 = \delta$ holds. That is, in the cost function, the quadratic term such as $x_i^2(k)$ is not necessary.

According to the result in [17], Problem 2 can be rewritten as a polynomial optimization problem. However, in the case of large-scale PBNs, it will be difficult to solve a polynomial optimization problem. In this paper, an approximate solution method for Problem 2 is proposed.

Hereafter, the condition $x(0) = x_0$ in the conditional expected value is omitted.

4. Solution Method

In this section, we derive an approximate solution method for Problem 2. First, a matrix-based representation for PBNs is derived. The obtained representation is an extension of a matrix-based representation for BNs proposed in [12]. Next, using the matrix-based representation, an approximate solution method for Problem 2 is derived.

4.1. Matrix-Based Representation for PBNs. As a preparation, the notation is defined. Binary variables $x_i^0(k)$ and $x_i^1(k)$ are introduced. If $x_i(k) = 0$ holds, then $x_i^0(k) = 1$ holds; otherwise $x_i^0(k) = 0$ holds. If $x_i(k) = 1$ holds, then $x_i^1(k) = 1$ holds; otherwise $x_i^1(k) = 0$ holds. Then, the equality $x_i^0(k) + x_i^1(k) = 1$ is satisfied. Using $x_i^0(k)$ and $x_i^1(k)$, consider transforming the BN (2) into a matrix-based representation.

First, we explain the outline of a matrix-based representation by using a simple example.

TABLE 1: Truth tables for $x_i(k+1)$, $i = 1, 2$.

(a)

$x_2(k)$	$x_1(k+1)$
0	1
1	0

(b)

$x_1(k)$	$x_2(k+1)$
0	0
1	1

Example 3. Consider the following BN:

$$\begin{aligned} x_1(k+1) &= \neg x_2(k), \\ x_2(k+1) &= x_1(k), \\ x_3(k+1) &= x_1(k) \wedge \neg x_2(k), \end{aligned} \tag{12}$$

where $\mathcal{N}^{(1)} = \{2\}$, $\mathcal{N}^{(2)} = \{1\}$, and $\mathcal{N}^{(3)} = \{1, 2\}$. Then, we can obtain the truth table for each $x_i(k+1)$. See Tables 1 and 2. From these truth tables, we can obtain the following matrix-based representation:

$$\begin{aligned}
\begin{bmatrix} x_1^0(k+1) \\ x_1^1(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{A^{(1)}} \begin{bmatrix} x_2^0(k) \\ x_2^1(k) \end{bmatrix}, \\
\begin{bmatrix} x_2^0(k+1) \\ x_2^1(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A^{(2)}} \begin{bmatrix} x_1^0(k) \\ x_1^1(k) \end{bmatrix}, \\
\begin{bmatrix} x_3^0(k+1) \\ x_3^1(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{A^{(3)}} \begin{bmatrix} x_1^0(k)\, x_2^0(k) \\ x_1^0(k)\, x_2^1(k) \\ x_1^1(k)\, x_2^0(k) \\ x_1^1(k)\, x_2^1(k) \end{bmatrix},
\end{aligned} \tag{13}$$

where each element of $A^{(i)}$, $i = 1, 2, 3$ is given by a binary value (0 or 1), and a sum of all elements in each column of $A^{(i)}$ is equal to 1.

Such a matrix-based representation has been proposed in also [18, 19]. However, in the representation proposed in [18, 19], matrices with the size of $2^n \times 2^n$ must be manipulated (n is the dimension of the state). In the matrix-based representation proposed in [12], matrices with the size of $2 \times 2^{|\mathcal{N}_i|}$ are manipulated for each x_i. Thus, the proposed representation enables us to model a BN using matrices with the smaller size.

TABLE 2: Truth table for $x_3(k+1)$.

$x_1(k)$	$x_2(k)$	$x_3(k+1)$
0	0	0
0	1	0
1	0	1
1	1	0

Consider a general case. Define

$$\overline{x}_i(k) := \begin{bmatrix} x_i^0(k) \\ x_i^1(k) \end{bmatrix} \left(= \begin{bmatrix} 1 - x_i(k) \\ x_i(k) \end{bmatrix} \right). \tag{14}$$

Then, the matrix-based representation for $x_i(k+1)$ is given by

$$\overline{x}_i(k+1) = A^{(i)} \bigotimes_{j \in \mathcal{N}_i} \overline{x}_j(k), \tag{15}$$

where $A^{(i)} \in \{0,1\}^{2 \times 2^{|\mathcal{N}_i|}}$ and $\bigotimes_{j \in \mathcal{N}_i} \overline{x}_j(k) \in \{0,1\}^{2^{|\mathcal{N}_i|}}$. The matrix $A^{(i)}$ can be derived from the following procedure.

Procedure for Deriving $A^{(i)}$ in (15)

Step 1. Derive a truth table for $x_i(k+1)$.

Step 2. Based on the obtained truth table, assign $x_i(k+1) = 0$ or $x_i(k+1) = 1$ for each element of $\bigotimes_{j \in \mathcal{N}_i} \overline{x}_j(k)$.

Step 3. Express the assignment obtained in Step 2 by a row vector. Denote the obtained row vector by $\overline{A}^{(i)} \in \{0,1\}^{1 \times 2^{|\mathcal{N}_i|}}$.

Step 4. Derive $A^{(i)}$ as

$$A^{(i)} = \begin{bmatrix} 1_{1 \times 2^{|\mathcal{N}_i|}} - \overline{A}^{(i)} \\ \overline{A}^{(i)} \end{bmatrix}. \tag{16}$$

Next, consider extending the matrix-based representation of BNs to that of PBNs. First, using a simple example, we explain the outline.

Example 4. Consider the PBN in Example 1. Using the matrix-based representation, the expected value of $\overline{x}_i(k+1)$ can be obtained as

$$E[\overline{x}_1(k+1)] = \left(0.8 \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A_1^{(1)}} + 0.2 \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}}_{A_2^{(1)}} \right) \times \begin{bmatrix} E[x_2^0(k)] \\ E[x_2^1(k)] \end{bmatrix},$$

$$E\left[\bar{x}_2(k+1)\right] = \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{A_1^{(2)}} \begin{bmatrix} E\left[x_1^0(k)\, x_3^0(k)\right] \\ E\left[x_1^0(k)\, x_3^1(k)\right] \\ E\left[x_1^1(k)\, x_3^0(k)\right] \\ E\left[x_1^1(k)\, x_3^1(k)\right] \end{bmatrix},$$

$$E\left[\bar{x}_3(k+1)\right] = \left(0.7 \underbrace{\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{A_1^{(3)}} + 0.3 \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{A_2^{(3)}} \right) \times \begin{bmatrix} E\left[x_1^0(k)\, x_2^0(k)\right] \\ E\left[x_1^0(k)\, x_2^1(k)\right] \\ E\left[x_1^1(k)\, x_2^0(k)\right] \\ E\left[x_1^1(k)\, x_2^1(k)\right] \end{bmatrix}, \tag{17}$$

where the condition $x(0) = x_0$ is omitted. In this representation, the matrices $A_1^{(1)}$ and $A_2^{(1)}$ correspond to the Boolean functions $f_1^{(1)}$ and $f_2^{(1)}$, respectively. In a similar way, $A_1^{(2)}$, $A_1^{(3)}$, and $A_2^{(3)}$ correspond to the Boolean functions $f_1^{(2)}$, $f_1^{(3)}$, and $f_2^{(3)}$, respectively.

In general, using the matrix-based representation, the expected value of $\bar{x}_i(k+1)$ can be obtained as

$$E\left[\bar{x}_i(k+1)\right] = \left(\sum_{l=1}^{q(i)} c_l^{(i)}(k)\, A_l^{(i)} \right) \bigotimes_{j \in \mathcal{N}_i} E\left[\bar{x}_j(k)\right], \tag{18}$$

where $A_l^{(i)} \in \{0,1\}^{2 \times 2^{|\mathcal{N}_i|}}$ and $\bigotimes_{j \in \mathcal{N}_i} \bar{x}_j(k) \in \{0,1\}^{2^{|\mathcal{N}_i|}}$. The matrix $A_l^{(i)}$ can be derived from the above procedure.

4.2. Reduction to a Linear Programming Problem. Using the matrix-based representation (18), consider transforming Problem 2. First, Problem 2 can be rewritten as the following problem.

Problem 5. Find $u(0), u(1), \ldots, u(N-1)$ minimizing the cost function (11) subject to (5), (10), (18), and the input constraint.

In a similar way to Problem 2, Problem 5 is rewritten as a polynomial optimization problem. In this paper, we focus on the structure of $\bigotimes_{j \in \mathcal{N}_i} E[\bar{x}_j(k)]$ and derive the relaxed problem for Problem 5. The relaxed problem is reduced to a linear programming (LP) problem, which can be solved faster than a polynomial optimization problem.

First, we present an example.

Example 6. Consider the matrix-based representation obtained in Example 4. We remark that the discrete probabilistic distribution for each x_i is independent. Then, in (17), we can obtain

$$\begin{aligned} &E\left[x_1^0(k)\, x_2^0(k)\right] + E\left[x_1^0(k)\, x_2^1(k)\right] \\ &\quad = E\left[x_1^0(k)\right]\left(E\left[x_2^0(k)\right] + E\left[x_2^1(k)\right]\right) \\ &\quad = E\left[x_1^0(k)\right]. \end{aligned} \tag{19}$$

In a similar way, we can obtain

$$\begin{aligned} E\left[x_1^1(k)\, x_2^0(k)\right] + E\left[x_1^1(k)\, x_2^1(k)\right] &= E\left[x_1^1(k)\right], \\ E\left[x_1^0(k)\, x_2^0(k)\right] + E\left[x_1^1(k)\, x_2^0(k)\right] &= E\left[x_2^0(k)\right], \\ E\left[x_1^0(k)\, x_2^1(k)\right] + E\left[x_1^1(k)\, x_2^1(k)\right] &= E\left[x_2^1(k)\right]. \end{aligned} \tag{20}$$

In addition,

$$\begin{aligned} &E\left[x_1^0(k)\, x_3^0(k)\right] + E\left[x_1^0(k)\, x_3^1(k)\right] + E\left[x_1^1(k)\, x_3^0(k)\right] \\ &\quad + E\left[x_1^1(k)\, x_3^1(k)\right] = 1 \end{aligned} \tag{21}$$

holds. The obtained equalities are linear with respect to $E[x_1^0(k)x_2^0(k)]$, $E[x_1^0(k)x_2^1(k)]$, $E[x_1^1(k)x_2^0(k)]$, and $E[x_1^1(k)x_2^1(k)]$, and $E[x_1^0(k)]$, $E[x_1^1(k)]$, $E[x_2^0(k)]$, and $E[x_2^1(k)]$. Hence, these can be used as constraints in the relaxed problem.

Next, consider a general case. Define

$$z_i(k) := \bigotimes_{j \in \mathcal{N}_i} E\left[\bar{x}_j(k)\right] \in [0,1]^{2^{|\mathcal{N}_i|}}. \tag{22}$$

Then, (18) can be rewritten as

$$\begin{aligned} E\left[\bar{x}_i(k+1)\right] &= \left(\sum_{l=1}^{q(i)} a_l^{(i)} A_l^{(i)} \right) z_i(k) \\ &\quad + \left(\sum_{l=1}^{q(i)} b_l^{(i)} A_l^{(i)} \right) w_i(k), \end{aligned} \tag{23}$$

where $w_i(k) := u_i(k) z_i(k) \in [0,1]^{2^{|\mathcal{N}_i|}}$. The relation between $E[\bar{x}_i(k)]$ and $z_i(k)$ is given by

$$E\left[\bar{x}_j(k)\right] = C_j z_i(k), \quad j \in \mathcal{N}_i. \tag{24}$$

For $\mathcal{N}_i = \{j_1, j_2, \ldots, j_{|\mathcal{N}_i|}\}$, $j_1 < j_2 < \cdots < j_{|\mathcal{N}_i|}$, matrices C_j, $j \in \mathcal{N}_i$ can be derived as

$$\begin{aligned} C_{j_1} &= \begin{bmatrix} 1_{1 \times 2^{|\mathcal{N}_i|-1}} & 0_{1 \times 2^{|\mathcal{N}_i|-1}} \\ 0_{1 \times 2^{|\mathcal{N}_i|-1}} & 1_{1 \times 2^{|\mathcal{N}_i|-1}} \end{bmatrix}, \\ C_{j_2} &= \begin{bmatrix} 1_{1 \times 2^{|\mathcal{N}_i|-2}} & 0_{1 \times 2^{|\mathcal{N}_i|-2}} & 1_{1 \times 2^{|\mathcal{N}_i|-2}} & 0_{1 \times 2^{|\mathcal{N}_i|-2}} \\ 0_{1 \times 2^{|\mathcal{N}_i|-2}} & 1_{1 \times 2^{|\mathcal{N}_i|-2}} & 0_{1 \times 2^{|\mathcal{N}_i|-2}} & 1_{1 \times 2^{|\mathcal{N}_i|-2}} \end{bmatrix}, \\ &\vdots \\ C_{j_{|\mathcal{N}_i|}} &= \begin{bmatrix} 1 & 0 & 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & 0 & 1 & \cdots & 0 & 1 \end{bmatrix}. \end{aligned} \tag{25}$$

Let $z_i^{(j)}(k)$ and $w_i^{(j)}(k)$ denote the jth element of $z_i(k)$ and $w_i(k)$, respectively. Then, we can obtain

$$\sum_{j=1}^{2^{|\mathcal{N}_i|}} z_i^{(j)}(k) = 1. \tag{26}$$

From $w_i(k) := u_i(k)z_i(k)$, we can obtain

$$\sum_{j=1}^{2^{|\mathcal{N}_i|}} w_i^{(j)}(k) = u_i(k). \tag{27}$$

In addition, we introduce the following constraints:

$$\begin{gathered} \underline{u}_i \le u_i(k) \le \overline{u}_i, \\ \underline{u}_i z_i(k) \le w_i(k) \le \overline{u}_i z_i(k), \\ 0 \le z_i(k) \le 1, \\ 0 \le w_i(k) \le 1. \end{gathered} \tag{28}$$

Thus, we can obtain the following problem as a relaxed problem of Problem 5.

Problem 7. Find $u(0), u(1), \ldots, u(N-1)$ minimizing the cost function (11) subject to (5), (10), and (23)–(28).

By a simple calculation, Problem 7 can be equivalently rewritten as the LP problem with $u_i(k)$, $z_i(k)$, and $w_i(k)$ as decision variables. By solving Problem 7, we can evaluate the lower bound of the optimal value of the cost function in Problem 7. In this paper, only an approximate solution method is provided. However, since the control input can be obtained by solving an LP problem, Problem 7 can be solved fast.

Finally, according to the receding horizon policy (see, e.g., [20]), we present the procedure of model predictive control (MPC).

Procedure of MPC

Step 1. Set $t = 0$.

Step 2. Measure the state $x(t)$.

Step 3. Derive $u(t), u(t+1), \ldots, u(t+N-1)$ by solving Problem 7.

Step 4. Apply only $u(t)$ to the system.

Step 5. Set $t+1 \rightarrow t$, and return to Step 2.

5. Application to Design of Real-Time Pricing Systems

In this section, we consider a design method of real-time pricing systems as an application of structural control of PBNs. First, the outline of real-time pricing systems of electricity is explained. Next, the PBN-based model of real-time pricing systems is derived. Finally, a numerical example is presented.

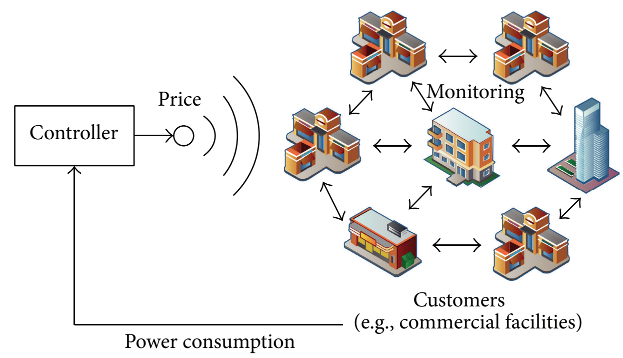

Figure 2: Illustration of real-time pricing systems.

5.1. Outline. Figure 2 shows an illustration of real-time pricing systems studied in this paper. This system consists of one controller and multiple electric customers such as commercial facilities and homes. For an electric customer, we suppose that each customer can monitor the status of electricity conservation of other customers. In other words, the status of some customer affects that of other customers. For example, in commercial facilities, we suppose that the status of rival commercial facilities can be checked by lighting, Blog, Twitter, and so on. Depending on power consumption, that is, the status of electricity conservation, the controller determines the price. If electricity conservation is needed, then the price is set to a high value. Since the economic load becomes high, customers conserve electricity. Thus, electricity conservation is achieved. The price does not depend on each customer and is uniquely determined.

5.2. Model. Consider modeling the set of customers as a PBN. The number of customers is given by n. We assume that the state of customer $i \in \{1, 2, \ldots, n\}$ is binary and is denoted by x_i. The state implies

$$x_i = \begin{cases} 0 & \text{customer } i \text{ conserves electricity,} \\ 1 & \text{customer } i \text{ normally uses electricity.} \end{cases} \tag{29}$$

The binary value of x_i is determined by power consumption of customer i. Let $\mathcal{D}_i \subseteq \{1, 2, \ldots, n\}$, $i = 1, 2, \ldots, n$ denote the set of customers, which affect customer i. In addition, we assume that there exists one leader in the local area. The state of a leader is given by x_1. Then, for customer i, we consider the following PBN as one of the situations:

$$x_i(k+1) = \begin{cases} f_1^{(i)} = 1, & c_1^{(i)} = a_1^{(i)} + b_1^{(i)} u(k), \\ f_2^{(i)} = 0, & c_2^{(i)} = a_2^{(i)} + b_2^{(i)} u(k), \\ f_3^{(i)} = x_i(k), & c_3^{(i)} = a_3^{(i)} + b_3^{(i)} u(k), \\ f_4^{(i)} = g^{(i)}\left(\left[x_j(k)\right]_{j \in \mathcal{D}_i}\right), & c_4^{(i)} = a_4^{(i)} + b_4^{(i)} u(k), \\ f_5^{(i)} = x_1(k), & c_5^{(i)} = a_5^{(i)} + b_5^{(i)} u(k), \end{cases} \tag{30}$$

where $u(k) \in [\underline{u}, \overline{u}] \subseteq \mathscr{R}^1$ is the control input corresponding to the price. The Boolean functions $f_1^{(i)}$ and $f_2^{(i)}$ imply that customer i forcibly conserves (or does not conserve) electricity. In these cases, time evolution of the state does not depend on the past state. The Boolean function $f_3^{(i)}$ implies that the state is not changed. The Boolean function $f_4^{(i)}$ implies that the state of customer i is changed depending on the other customers. The Boolean function $f_5^{(i)}$ implies that the state of customer i is changed depending on the leader. Thus, decision making of customers can be modeled by a PBN. The above Boolean functions are an example of models for decision making. Depending on real situations, we may use other Boolean functions.

For the PBN-based model obtained, we consider the following problem:

> find a time sequence of the price such that customers conserve electricity as much as possible. However, it is not desirable that the price is too high.

The condition that customers conserve electricity as much as possible can be characterized by $E[x_i]$. In other words, power consumption is expressed by $E[x_i]$. Hence, this problem can be formulated as Problem 2 by appropriately setting the weights Q and R.

5.3. Numerical Example. We present a numerical example. Parameters in the system are given as follows: $n = 8$, $\mathscr{D}_1 = \{2, 8\}$, $\mathscr{D}_i = \{i-1, i+1\}$, $i = 2, 3, \ldots, 7$, $\mathscr{D}_8 = \{1, 7\}$, $a_1^{(i)} = 0.1$, $b_1^{(i)} = 0$, $a_2^{(i)} = 0$, $b_2^{(i)} = 0.25$, $a_3^{(i)} = 0.9$, $b_3^{(i)} = -1$, $a_4^{(i)} = 0$, $b_4^{(i)} = 0.5$, $a_5^{(i)} = 0$, $b_4^{(i)} = 0.25$, $\underline{u} = 0.3$, and $\overline{u} = 0.7$. We remark that under the input constraint $u(k) \in [\underline{u}, \overline{u}]$, (5), and (10) hold. The Boolean function $g^{(i)}$ is given by

$$g^{(i)}\left(\left[x_j(k)\right]_{j \in \mathscr{D}_i}\right) = x_{j_1}(k) \wedge x_{j_2}(k) \wedge \cdots \wedge x_{j_{|\mathscr{D}_i|}}(k), \quad \left\{j_1, j_2, \ldots, j_{|\mathscr{D}_i|}\right\} = \mathscr{D}_i. \tag{31}$$

Parameters in Problem 2 are given as follows: $x(0) = [0\ 1\ \cdots\ 1]^\top$, $N = 15$, $Q = Q_f = [1\ \cdots\ 1]$, and $R = 5$.

Next, we present the computation result. Here, Problem 7 was solved once. Figure 3 shows trajectories of $E[x_i(k)]$. Figure 4 shows trajectories of the control input (the price). From these figures, we see that $E[x_i]$ becomes small by fine adjustment of the control input. In this example, the expected value of each state converges to 0.32.

In addition, when the obtained control input is applied to the system, the value of the cost function in Problem 2 was 79.4311. In the case of $u(k) = 0.3$ (i.e., the constant input), the value of the cost function in Problem 2 was 85.3581. In the case of $u(k) = 0.7$, the value of the cost function in Problem 2 was 87.0247. From these values, we see that the obtained control input is more effective than trivial control inputs. Furthermore, in order to verify the optimality, consider the case of $N = 2$. The optimal control input was derived as $u(0) = 0.7$ and $u(1) = 0.3$ by solving the polynomial programming problem, which is equivalent to Problem 2. On the other hand, the control input obtained by solving Problem 7 was $u(0) = 0.7$ and $u(1) = 0.52$. Thus, there is a possibility that the control input obtained by solving Problem 7 is not optimal for Problem 2.

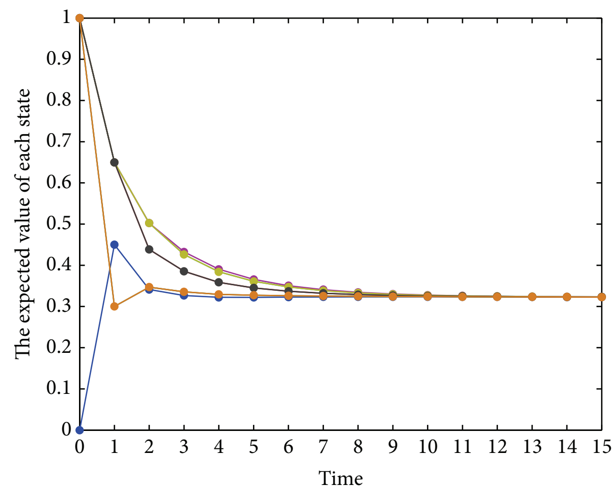

FIGURE 3: The expected value of the state. Some states are indistinguishable.

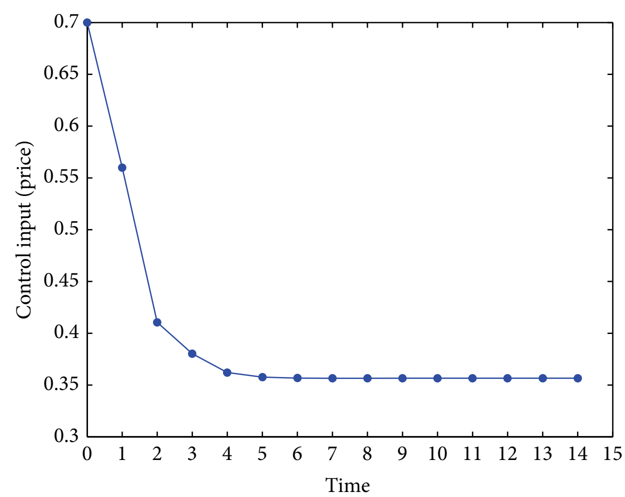

FIGURE 4: The obtained control input (price).

Finally, we discuss the computation time for solving Problem 7. The computation time was 0.6 sec for $N = 15$ and 0.03 sec for $N = 2$, where we used IBM ILOG CPLEX 11.0 as the LP solver. The computation time for solving the polynomial optimization problem for $N = 2$ was 232.2 sec, where we used SparsePOP [21] and MATLAB 32-bit version. In the case of $N \geq 3$, owing to memory warning, the polynomial optimization problem cannot be solved. Thus, although Problem 7 is an approximation of the original problem, Problem 7 can be solved fast.

6. Conclusion

In this paper, we studied control of complex systems modeled by a probabilistic Boolean network (PBN). First, the structural control problem for a PBN was newly formulated. Next, an approximate solution method was proposed based on a matrix-based representation of Boolean functions. Finally, as an application, we considered design of real-time pricing systems of electricity. The proposed method provides us a new control method for complex systems.

There are several open problems. It is significant to consider a method for evaluating the accuracy of an approximation from the theoretical viewpoint. It is also significant to develop an identification method of Boolean functions and parameters $a_l^{(i)}$, $b_l^{(i)}$ in (9). Finally, it will be one of the interesting topics to apply the proposed method to several classes of PBNs, for example, a large-scale PBN with scale-free structure.

Conflict of Interests

The authors declare that they have no conflict of interests.

Acknowledgment

This research was partly supported by JST, CREST.

References

[1] P. Tabuada, *Verification and Control of Hybrid Systems*, Springer, 2009.

[2] M. Adomi, Y. Shikauchi, and S. Ishii, "Hidden Markov model for human decision process in a partially observable environment," in *Proceedings of the 20th International Conference on Artificial Neural Networks*, vol. 6353 of *Lecture Notes in Computer Science*, pp. 94–103, 2010.

[3] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[4] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[5] A. Mochizuki, "An analytical study of the number of steady states in gene regulatory networks," *Journal of Theoretical Biology*, vol. 236, no. 3, pp. 291–310, 2005.

[6] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[7] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, "Intervention in context-sensitive probabilistic boolean networks revisited," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2009, Article ID 360864, 13 pages, 2009.

[8] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty, "Intervention in context-sensitive probabilistic Boolean networks," *Bioinformatics*, vol. 21, no. 7, pp. 1211–1218, 2005.

[9] K. Kobayashi and K. Hiraishi, "Optimal control of gene regulatory networks with effectiveness of multiple drugs: a boolean network approach," *BioMed Research International*, vol. 2013, Article ID 246761, 11 pages, 2013.

[10] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," *Journal of Biological Systems*, vol. 10, no. 4, pp. 431–445, 2002.

[11] Y. Xiao and E. R. Dougherty, "The impact of function perturbations in Boolean networks," *Bioinformatics*, vol. 23, no. 10, pp. 1265–1273, 2007.

[12] K. Kobayashi and K. Hiraishi, "Design of boolean networks based on prescribed singleton attractors," in *Proceedings of the European Control Conference*, pp. 1504–1509, 2014.

[13] S. Borenstein, M. Jaske, and A. Rosenfeld, *Dynamic Pricing, Advanced Metering, and Demand Response in Electricity Markets*, Center for the Study of Energy Markets, University of California, Berkeley, Calif, USA, 2002.

[14] M. Roozbehani, M. Dahleh, and S. Mitter, "On the stability of wholesale electricity markets under real-time pricing," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 1911–1918, December 2010.

[15] A.-H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," in *Proceedings of the Innovative Smart Grid Technologies (ISGT '10)*, pp. 1–10, Gaithersburg, Md, USA, January 2010.

[16] C. Vivekananthan, Y. Mishra, and G. Ledwich, "A novel real time pricing scheme for demand response in residential distribution systems," in *Proceedings of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON '13)*, pp. 1956–1961, Vienna, Austria, November 2013.

[17] K. Kobayashi and K. Hiraishi, "Optimal control of probabilistic Boolean networks using polynomial optimization," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E95-A, no. 9, pp. 1512–1517, 2012.

[18] D. Cheng and H. Qi, "Controllability and observability of Boolean control networks," *Automatica*, vol. 45, no. 7, pp. 1659–1667, 2009.

[19] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Network: A Semi-Tensor Product Approach*, Communications and Control Engineering Series, Springer, London, UK, 2011.

[20] E. F. Camacho and C. B. Alba, *Model Predictive Control*, Springer, 2nd edition, 2007.

[21] SparsePOP, http://www.is.titech.ac.jp/~kojima/SparsePOP/SparsePOP.html.