Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 952310, 10 pages
http://dx.doi.org/10.1155/2015/952310

Research Article

Optimal Control of Probabilistic Logic Networks and Its Application to Real-Time Pricing of Electricity

Koichi Kobayashi[1] and Kunihiko Hiraishi[2]

[1] *Graduate School of Information Science and Technology, Hokkaido University, Sapporo 060-0814, Japan*
[2] *School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan*

Correspondence should be addressed to Koichi Kobayashi; k-kobaya@ssi.ist.hokudai.ac.jp

Received 1 July 2015; Revised 18 October 2015; Accepted 16 November 2015

Academic Editor: Qingling Zhang

In analysis and control of large-scale complex systems, a discrete model plays an important role. In this paper, a probabilistic logic network (PLN) is considered as a discrete model. A PLN is a mathematical model where multivalued logic functions are randomly switched. For a PLN with two kinds of control inputs, the optimal control problem is formulated, and an approximate solution method for this problem is proposed. In the proposed method, using a matrix-based representation for a PLN, this problem is approximated by a mixed integer linear programming problem. In application, real-time pricing of electricity is studied. In real-time pricing, electricity conservation is achieved by setting a high electricity price. A numerical example is presented to show the effectiveness of the proposed method.

1. Introduction

In the last decade, control of large-scale complex systems has attracted considerable attention, where complex nonlinear dynamics are taken into account. In such cases, one of the natural approaches is to approximately solve the control problem of such complex systems using simplification/abstraction techniques (see, e.g., [1]). Specifically, a discrete abstract model plays an important role in analysis and control of large-scale complex systems.

Several discrete models have been proposed so far (see, e.g., [2]). Petri nets, Bayesian networks, and automata-based models are frequently adopted as a mathematical model in analysis of large-scale complex systems. Although Boolean networks (BNs) [3] are frequently adopted as a model for analysis, BNs are recently adopted as a mathematical model in control of large-scale complex systems. In a BN, the state takes a binary value (0 or 1), and the dynamics such as interactions of states are modeled by a set of Boolean functions. In [4], it is pointed out that since a BN is too simple, the behavior of large-scale complex systems such as gene regulatory networks cannot be appropriately modeled. One of the methods to overcome this criticism is to extend Boolean functions to multivalued logic functions. From this viewpoint, several modeling methods using multivalued logic functions have been studied so far (see, e.g., [5–8]). Furthermore, since the behavior of large-scale complex systems is frequently stochastic by the effects of noise, a probabilistic logic network (PLN), which is an extension of a probabilistic BN (PBN) [9], has been studied so far (see, e.g., [10, 11]). In a PLN, a multivalued logic function is randomly selected from the candidates of functions. We adopt a PLN as a discrete model.

In this paper, first, we formulate the optimal control problem for the PLN with two kinds of control inputs. One of control inputs is a conventional control input, which is included in multivalued logic functions. The other is called here a structural control input, which is included in a given discrete probability distribution. By the structural control input, a discrete probability distribution is controlled. Such a control input has been considered in, for example, [12–15].

Next, an approximate solution method for the optimal control problem is proposed. Solution methods have been studied in [10]. However, large sized matrices must be handled. From this fact, it is difficult to directly apply the existing methods to large-scale systems. Hence, it is important to consider a new solution method. To derive

an approximate solution method, a matrix-based representation of PLNs is proposed. The authors have proposed a matrix-based representation of BNs [16] and that of probabilistic BNs [13]. However, multivalued logic functions have not been studied so far. Using a matrix-based representation, the original problem is approximated by a mixed integer linear programming (MILP) problem. In the case where there is no conventional control input, the original problem is approximated by a linear programming problem. In [17], it has been proven that the optimal control problem of PBNs is Σ_2^p-hard problem, which is much harder than an NP-complete problem. Hence, approximating the optimal control problem of PLNs by an MILP problem is useful for solving that of a wider class of PLNs.

A PLN has several applications such as gene regulatory networks and power systems. As already explained, BNs and PBNs are well known as a model of gene regulatory networks. Logical models with multivalued variables are also used in theoretical analysis of gene regulatory networks. A PLN can be regarded as a generalized model of these mathematical models. In this paper, as one of the other applications, we consider real-time pricing of electricity (see, e.g., [18–21]).

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. Real-time pricing 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 consumers 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, consumers 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., [21]). The authors have proposed in [13] a method to model decision making of consumers using a PBN. However, the state of a consumer is given by a binary value, that is, 0 (a consumer conserves electricity) or 1 (a consumer normally uses electricity). A PLN is appropriate as a more precise model of consumers.

Notation. For the nonnegative integer $i \geq 2$, define the finite set $\mathscr{D}_i := \{0, 1, \ldots, i-1\}$. For the n-dimensional vector $x = [x_1 \ x_2 \ \cdots \ x_n]^\top$ and the index set $\mathscr{I} = \{i_1, i_2, \ldots, i_m\} \subseteq \{1, 2, \ldots, n\}$, define $[x_i]_{i\in\mathscr{I}} := [x_{i_1} \ x_{i_2} \ \cdots \ x_{i_m}]^\top$. 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 $\mathscr{J} = \{j_1, j_2, \ldots, j_p\} \subseteq \{1, 2, \ldots, q\}$, define $\bigotimes_{j\in\mathscr{J}} := y_{j_1} \otimes y_{j_2} \otimes \cdots \otimes y_{j_p}$. For example, for q two-dimensional vectors $z_1, z_2, \ldots, z_q$ and $\mathscr{J} = \{1, 5\}$, we can obtain

$$\bigotimes_{j\in\mathscr{J}} z_j = z_1 \otimes z_5 = \begin{bmatrix} z_1^{(1)} \\ z_1^{(2)} \end{bmatrix} \otimes \begin{bmatrix} z_5^{(1)} \\ z_5^{(2)} \end{bmatrix} = \begin{bmatrix} z_1^{(1)} z_5^{(1)} \\ z_1^{(1)} z_5^{(2)} \\ z_1^{(2)} z_5^{(1)} \\ z_1^{(2)} z_5^{(2)} \end{bmatrix}, \quad (1)$$

where $z_j^{(i)}$ is the ith element of z_j. Let $1_{m\times n}$ denote the $m \times n$ matrix whose elements are all one. Let I_n and $0_{m\times n}$ denote the $n \times n$ identity matrix and the $m \times n$ zero matrix, respectively. For simplicity of notation, we sometimes use the symbol 0 instead of $0_{m\times n}$, and the symbol I instead of I_n.

2. Probabilistic Logic Network

First, we explain a multivalued logic network (MVLN). This system is defined by

$$\begin{aligned} x_1(k+1) &= f^{(1)}\left(\left[x_j(k)\right]_{j\in\mathscr{N}_1}, \left[u_j(k)\right]_{j\in\mathscr{M}_1}\right), \\ x_2(k+1) &= f^{(2)}\left(\left[x_j(k)\right]_{j\in\mathscr{N}_2}, \left[u_j(k)\right]_{j\in\mathscr{M}_2}\right), \\ &\vdots \\ x_n(k+1) &= f^{(n)}\left(\left[x_j(k)\right]_{j\in\mathscr{N}_n}, \left[u_j(k)\right]_{j\in\mathscr{M}_n}\right), \end{aligned} \quad (2)$$

where $x_i \in \mathscr{D}_{r(i)}$, $i = 1, 2, \ldots, n$, $x := [x_1 \ x_2 \ \cdots \ x_n]^\top$ is the state, and $u_i \in \mathscr{D}_{s(i)}$, $i = 1, 2, \ldots, m$, $u := [u_1 \ u_2 \ \cdots \ u_m]^\top$, is the control input. In addition, $r(i)$ and $s(i)$ are given, and $k \in \{0, 1, 2, \ldots\}$ is the discrete time. The sets $\mathscr{N}_i \subseteq \{1, 2, \ldots, n\}$ and $\mathscr{M}_i \subseteq \{1, 2, \ldots, m\}$ are given index sets, and the function

$$f^{(i)} : \prod_{j\in\mathscr{N}_i} \mathscr{D}_{r(j)} \times \prod_{j\in\mathscr{M}_i} \mathscr{D}_{s(j)} \longrightarrow \mathscr{D}_{r(i)} \quad (3)$$

is a given multivalued logic function.

Next, we explain a probabilistic logic network (PLN) (see [10, 11] for further details). A PLN is a generalized version of a probabilistic Boolean network proposed in [9]. In a PLN, the candidates of $f^{(i)}$ are given, and, for each x_i, selecting one Boolean function is probabilistically independent at each time. Let

$$f_l^{(i)}\left(\left[x_j(k)\right]_{j\in\mathscr{N}_i}, \left[u_j(k)\right]_{j\in\mathscr{M}_i}\right), \quad l = 1, 2, \ldots, q(i) \quad (4)$$

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

$$c_l^{(i)} := \text{Prob}\left(f^{(i)} = f_l^{(i)}\right). \quad (5)$$

Then, the relation $\sum_{l=1}^{q(i)} c_l^{(i)} = 1$ must be satisfied. Probabilistic distributions are derived from experimental results.

We present a simple example.

Example 1. Consider the PLN with two states $x_1 \in \mathscr{D}_2$, $x_2 \in \mathscr{D}_3$ ($r(1) = 2$, $r(2) = 3$) and no control inputs. Suppose that $q(1) = q(2) = 2$ holds. Functions $f_1^{(1)}$, $f_2^{(1)}$ and $f_1^{(2)}$, $f_2^{(2)}$ are given by the truth table in Table 1. Probabilities are given by $c_1^{(1)} = 0.7$, $c_2^{(1)} = 0.3$ and $c_1^{(2)} = 0.8$, $c_2^{(2)} = 0.2$. We see that the relation $\sum_{l=1}^{q(i)} c_l^{(i)} = 1$ is satisfied. In this case, $\mathscr{N}_1 = \mathscr{N}_2 =$

Table 1: Truth table.

x_1	x_2	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$
0	0	0	1	2	0
0	1	0	1	0	0
0	2	1	1	0	1
1	0	1	0	1	2
1	1	0	1	0	0
1	2	0	1	0	1

$\{1, 2\}$ holds. Next, consider the state trajectory. For example, $x(1)$ for $x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top$ can be obtained as

$$\begin{aligned} &\text{Prob}\left(x(1) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top\right) = 0.14, \\ &\text{Prob}\left(x(1) = \begin{bmatrix} 0 & 2 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top\right) = 0.56, \\ &\text{Prob}\left(x(1) = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top\right) = 0.06, \\ &\text{Prob}\left(x(1) = \begin{bmatrix} 1 & 2 \end{bmatrix}^\top \mid x(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^\top\right) = 0.24. \end{aligned} \tag{6}$$

This PLN can be equivalently transformed into the discrete-time Markov chain with six states in which the transition probability matrix is given by

$$P = \begin{bmatrix} 0.14 & 0 & 0.56 & 0.06 & 0 & 0.24 \\ 0.7 & 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 & 0 \\ 0 & 0.24 & 0.06 & 0 & 0.56 & 0.14 \\ 0.7 & 0 & 0 & 0.3 & 0 & 0 \\ 0.56 & 0.14 & 0 & 0.24 & 0.06 & 0 \end{bmatrix}. \tag{7}$$

Using the transition probability matrix expressing a given PLN, we can consider reachability analysis and optimal control. However, in many cases, the size of the matrix P becomes large.

Remark 2. In this paper, we consider the PLN with no time delay. Using additional states, we can easily consider time delays.

3. Problem Formulation

In this section, the optimal control problem for the PLN is formulated. We assume that the value of the control input can be arbitrarily set. However, there is a possibility that there exists no control input satisfying this assumption. In such a case, it is desirable to use the structural control input, which is used in, for example, control theory of gene regulatory networks (see, e.g., [12, 14]). For example, in [12], the structural control input is regarded as an external stimulus, and a discrete probabilistic distribution is switched at each time by the structural control input. In other words, the discrete probabilistic distribution is selected among the set of candidates. In [13], the structural control input is used as the electricity price in real-time pricing systems. By the structural control input, probabilities in a discrete probability distribution are manipulated. Thus, in complex systems, such as gene regulatory networks, power systems, and social systems, it will be desirable to consider a weaker control method using the structural control input. Hence, we consider both the conventional control input and the structural control input.

The structural control input $v(k)$ formulated here is added to the probability $c_l^{(i)}$ in (5) as follows:

$$c_l^{(i)}(k) = a_l^{(i)} + b_l^{(i)} v(k), \tag{8}$$

where $v(k) \in [0, 1] \subseteq \mathscr{R}^1$ (the constraint is normalized). For simplicity of discussion, we consider the one-dimensional structural control input, but we may consider the multidimensional structural control input.

Under the above preparation, we consider the following problem.

Problem 3. Suppose that, for the PLN with the conventional control input u and the structural control input v, the initial state $x(0) = x_0$ is given. Then, find $u(0), u(1), \ldots, u(N-1) \in \{0, 1\}^m$ and $v(0), v(1), \ldots, v(N-1) \in [0, 1]$ minimizing the following cost function:

$$\begin{aligned} J = E\Bigg[&\sum_{k=0}^{N-1} \left(Qx(k) + R_1 u(k) + R_2 v(k)\right) \\ &+ Q_f x(N) \mid x(0) = x_0\Bigg], \end{aligned} \tag{9}$$

subject to the following two constraints on $c_l^{(i)}(k)$:

$$\begin{aligned} &\sum_{l=1}^{q(i)} \left(a_l^{(i)} + b_l^{(i)} v(k)\right) = 1, \\ &0 \le a_l^{(i)} + b_l^{(i)} v(k) \le 1. \end{aligned} \tag{10}$$

In the cost function (9), $Q, Q_f \in \mathscr{R}^{1\times n}$, $R_1 \in \mathscr{R}^{1\times m}$, and $R_2 \in \mathscr{R}^1$ are weighting vectors whose element is a nonnegative real number, and $E[\cdot \mid \cdot]$ denotes a conditional expected value.

For simplicity of discussion, we consider the linear cost function (9). We remark that $x \ge 0$, $u \ge 0$, and $v \ge 0$ hold. The desired state x_d, the desired conventional control input u_d, and the desired structural control input v_d may be introduced. Then, in the cost function (9), $x(k)$, $u(k)$, and $v(k)$ may be replaced with $|x(k) - x_d|$, $|u(k) - u_d|$, and $|v(k) - v_d|$, respectively. In the case of $x_i \in \mathscr{D}_2$ and $u_i \in \mathscr{D}_2$ (i.e., probabilistic Boolean networks), Problem 3 can be rewritten as a polynomial optimization problem (see [22] for further details). However, only few results on optimal control of general PLNs have been obtained so far (see, e.g., [10]). In this paper, an approximate solution method for Problem 3 is proposed. Furthermore, Problem 3 is used in model predictive control (MPC). See Section 5.3 for further details.

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

4. Derivation of Approximate Solution Method

In this section, we derive an approximate solution method for Problem 3. First, a matrix-based representation for PLNs is derived. The obtained representation is an extension of a matrix-based representation for probabilistic Boolean networks proposed in [13]. Next, using the matrix-based representation, Problem 3 is approximated by a mixed integer linear programming (MILP) problem.

4.1. Matrix-Based Representation for PLNs. As a preparation, the notation is defined. Binary variables $x_i^0(k), x_i^1(k), \ldots, x_i^{r(i)}(k)$ are introduced. If $x_i(k) = j$ holds, then $x_i^j(k) = 1$ holds; otherwise $x_i^j(k) = 0$ holds. Then, the equality $\sum_{j=1}^{r(i)} x_i^j(k) = 1$ is satisfied. In addition, binary variables $u_i^0(k), u_i^1(k), \ldots, u_i^{s(i)}(k)$ are introduced. If $u_i(k) = j$ holds, then $u_i^j(k) = 1$ holds; otherwise $u_i^j(k) = 0$ holds. Then, the equality $\sum_{j=1}^{s(i)} u_i^j(k) = 1$ is satisfied. Using $x_i^j(k)$, $j = 1, 2, \ldots, r(i)$, and $u_i^j(k)$, $j = 1, 2, \ldots, s(i)$, define

$$\begin{aligned} \overline{x}_i(k) &:= \begin{bmatrix} x_i^0(k) \\ x_i^1(k) \\ \vdots \\ x_i^{r(i)-1}(k) \end{bmatrix} \in \{0,1\}^{r(i)}, \quad i = 1, 2, \ldots, n, \\ \overline{u}_i(k) &:= \begin{bmatrix} u_i^0(k) \\ u_i^1(k) \\ \vdots \\ u_i^{s(i)-1}(k) \end{bmatrix} \in \{0,1\}^{s(i)}, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{11}$$

In the case of $r(i) = 2$, $\overline{x}_i(k)$ can be obtained as

$$\overline{x}_i(k) = \begin{bmatrix} x_i^0(k) \\ x_i^1(k) \end{bmatrix} = \begin{bmatrix} 1 - x_i(k) \\ x_i(k) \end{bmatrix}. \tag{12}$$

First, consider transforming a given MVLN into a matrix-based representation. Then, the matrix-based representation for $x_i(k+1)$ is given by

$$\overline{x}_i(k+1) = A^{(i)} z_i(k), \tag{13}$$

where

$$\begin{aligned} z_i(k) &:= \left(\bigotimes_{j \in \mathcal{N}_i} \overline{x}_j(k) \right) \otimes \left(\bigotimes_{j \in \mathcal{M}_i} \overline{u}_j(k) \right) \in \{0,1\}^{\alpha(i)}, \\ \alpha(i) &:= \prod_{j \in \mathcal{N}_i} r(j) \prod_{j \in \mathcal{M}_i} s(j), \\ A^{(i)} &\in \{0,1\}^{r(i) \times \alpha(i)}. \end{aligned} \tag{14}$$

Table 2: Truth table for x_1.

$x_1^0(k)$	$x_1^1(k)$	$x_2^0(k)$	$x_2^1(k)$	$x_2^2(k)$	$x_1^0(k+1)$	$x_1^1(k+1)$
1	0	1	0	0	0	1
1	0	0	1	0	0	1
1	0	0	0	1	0	1
0	1	1	0	0	1	0
0	1	0	1	0	0	1
0	1	0	0	1	0	1

Table 3: Truth table for x_2.

x_1^0	x_1^1	x_2^0	x_2^1	x_2^2	$x_2^0(k+1)$	$x_2^1(k+1)$	$x_2^2(k+1)$
1	0	1	0	0	0	0	1
1	0	0	1	0	1	0	0
1	0	0	0	1	1	0	0
0	1	1	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	0	0	1	1	0	0

The matrix $A^{(i)}$ can be derived from the truth table for $x_i(k+1)$ using $x_i^j(k)$, $j = 1, 2, \ldots, r(i)$, and $u_i^j(k)$, $j = 1, 2, \ldots, s(i)$.

We present a simple example.

Example 4. Consider an MVLN with two states and no control inputs. For $x_1 \in \mathcal{D}_2$, the logic function $f_2^{(1)}$ in Table 1 is given. For $x_2 \in \mathcal{D}_3$, the logic function $f_1^{(2)}$ in Table 1 is given. Using $x_1^0(k), x_1^1(k)$ and $x_2^0(k), x_2^1(k), x_2^2(k)$, logic functions $f_2^{(1)}$ and $f_1^{(2)}$ can be expressed as truth tables in Tables 2 and 3, respectively. From these truth tables, we can obtain the following matrix-based representation of the MVLN:

$$\begin{aligned} \underbrace{\begin{bmatrix} x_1^0(k+1) \\ x_1^1(k+1) \end{bmatrix}}_{\overline{x}_1(k+1)} &= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}}_{A^{(1)}} \underbrace{\begin{bmatrix} x_1^0(k)\, x_2^0(k) \\ x_1^0(k)\, x_2^1(k) \\ x_1^0(k)\, x_2^2(k) \\ x_1^1(k)\, x_2^0(k) \\ x_1^1(k)\, x_2^1(k) \\ x_1^1(k)\, x_2^2(k) \end{bmatrix}}_{z_1(k)}, \\ \underbrace{\begin{bmatrix} x_2^0(k+1) \\ x_2^1(k+1) \\ x_2^2(k+1) \end{bmatrix}}_{\overline{x}_2(k+1)} &= \underbrace{\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A^{(2)}} z_2(k), \end{aligned} \tag{15}$$

where $z_1(k) = z_2(k) (= \overline{x}_1(k) \otimes \overline{x}_2(k))$. 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 [6–8]. The matrix-based representation for one x_i,

that is, (13), may be derived by the method in [6–8]. See also Remark 9.

Next, consider extending the matrix-based representation of an MVLN to that of a PLN. Suppose that the matrix-based representation of the logic function $f_l^{(i)}$ in a given PLN is given by $A_l^{(i)} z_i(k)$. Then, the expected value of $\overline{x}_i(k+1)$ can be obtained as

$$E\left[\overline{x}_i(k+1)\right] = \left(\sum_{l=1}^{q(i)} c_l^{(i)}(k) A_l^{(i)}\right) E\left[z_i(k)\right], \tag{16}$$

where $A_l^{(i)} \in \{0,1\}^{r(i)\times\alpha(i)}$, and the condition $x(0) = x_0$ in the expected value is omitted. We remark here that the jth element of $E[\overline{x}_i(k+1)]$ is the probability that $x_i(k) = j-1$ holds.

We present a simple example.

Example 5. Consider the PLN in Example 1. In this system, there are no control input and no structural control input. Using the matrix-based representation, the expected value of $\overline{x}_i(k+1)$ can be obtained as

$$\begin{aligned} E\left[\overline{x}_1(k+1)\right] &= \left(0.7A_1^{(1)} + 0.3A_2^{(1)}\right) E\left[z_1(k)\right], \\ E\left[\overline{x}_2(k+1)\right] &= \left(0.8A_1^{(2)} + 0.2A_2^{(2)}\right) E\left[z_2(k)\right], \end{aligned} \tag{17}$$

where $z_1(k) = z_2(k) = \overline{x}_1(k) \otimes \overline{x}_2(k)$,

$$\begin{aligned} A_1^{(1)} &= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}, \\ A_2^{(1)} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}, \\ A_1^{(2)} &= \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \\ A_2^{(2)} &= \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{18}$$

4.2. Reduction to an MILP Problem. Using the matrix-based representation (16), consider transforming Problem 3. First, we focus on the structural control input $v(k) \in [0,1]$. Using (8), the system (16) can be rewritten as

$$\begin{aligned} E\left[\overline{x}_i(k+1)\right] &= \left(\sum_{l=1}^{q(i)} a_l^{(i)}(k) A_l^{(i)}\right) E\left[z_i(k)\right] \\ &\quad + \left(\sum_{l=1}^{q(i)} b_l^{(i)}(k) A_l^{(i)}\right) v(k) E\left[z_i(k)\right]. \end{aligned} \tag{19}$$

Noting that the input constraint is imposed as $v(k) \in [0,1]$, the structural control input $v(k)$ can be regarded as the expected value of some binary probabilistic variable $\widetilde{v}(k) \in \{0,1\}$. Then,

$$v(k) E\left[z_i(k)\right] = E\left[z_i(k)\widetilde{v}(k)\right] = E\left[z_i(k)\right] E\left[\widetilde{v}(k)\right] \tag{20}$$

holds. Here, according to the definitions of $\overline{x}_i(k)$ and $\overline{u}_i(k)$, we define

$$\begin{aligned} \widetilde{z}_i(k) &:= z_i(k) \otimes \left[1-\widetilde{v}(k) \quad \widetilde{v}(k)\right]^\top \\ &= \begin{bmatrix} z_i(k)(1-\widetilde{v}(k)) \\ z_i(k)\widetilde{v}(k) \end{bmatrix}. \end{aligned} \tag{21}$$

Thus, we can obtain

$$\begin{aligned} E\left[\overline{x}_i(k+1)\right] &= \left(\sum_{l=1}^{q(i)} a_l^{(i)}(k) A_l^{(i)}\right) E\left[z_i(k)\right] \\ &\quad + \left[0 \quad \sum_{l=1}^{q(i)} b_l^{(i)}(k) A_l^{(i)}\right] E\left[\widetilde{z}_i(k)\right]. \end{aligned} \tag{22}$$

Then, Problem 3 can be equivalently rewritten as the following problem.

Problem 6. Suppose that, for the PLN with the conventional control input u and the structural control input v, the initial state $x(0) = x_0$ is given. Then, find $u(0), u(1), \ldots, u(N-1) \in \{0,1\}^m$ and $v(0), v(1), \ldots, v(N-1) \in [0,1]$ minimizing the cost function (9) subject to (10) and (22).

By a simple calculation, Problem 6 can be rewritten as a polynomial optimization problem. However, it will be difficult to solve a polynomial optimization problem for large-scale PLNs. In this paper, we focus on the relations among $\overline{x}_i(k)$, $\overline{u}_i(k)$, and $z_i(k)$ of (14) and $\widetilde{z}_i(k)$ of (21). Using these relations, we derive the relaxed problem for Problem 6. The relaxed problem is an MILP problem and can be solved faster than a polynomial optimization problem.

First, we present a simple example.

Example 7. Consider $E[z_1(k)]$ in Example 5. Then, noting that $E[x_2^0(k)] + E[x_2^1(k)] + E[x_2^2(k)] = 1$ holds, we can obtain

$$\begin{aligned} &\begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} E\left[x_1^0(k)\right] E\left[x_2^0(k)\right] \\ E\left[x_1^0(k)\right] E\left[x_2^1(k)\right] \\ E\left[x_1^0(k)\right] E\left[x_2^2(k)\right] \\ E\left[x_1^1(k)\right] E\left[x_2^0(k)\right] \\ E\left[x_1^1(k)\right] E\left[x_2^1(k)\right] \\ E\left[x_1^1(k)\right] E\left[x_2^2(k)\right] \end{bmatrix}}_{E[z_1(k)]} \\ &= E\left[x_1^0(k)\right]. \end{aligned} \tag{23}$$

In a similar way, we can obtain

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} E\left[z_1(k)\right] = \begin{bmatrix} E\left[x_1^1(k)\right] \\ E\left[x_2^0(k)\right] \\ E\left[x_2^1(k)\right] \\ E\left[x_2^2(k)\right] \end{bmatrix}. \tag{24}$$

In addition, a sum of all elements in $E[z_1(k)]$ is equal to 1. Equalities obtained can be used as constraints in the relaxed problem.

Next, consider a general case. Noting that $E[\overline{u}_j(k)] = \overline{u}_j(k)$ holds, we can obtain the following constraints for $E[z_i(k)]$:

$$\begin{aligned} &E\left[\overline{x}_{j_x}(k)\right] = C_{j_x} E\left[z_i(k)\right], \\ &j_x \in \mathcal{N}_i = \left\{j_{x1}, j_{x2}, \ldots, j_{x|\mathcal{N}_i|}\right\}, \ j_{x1} < j_{x2} < \cdots < j_{x|\mathcal{N}_i|}, \\ &\overline{u}_{j_u}(k) = D_{j_u} E\left[z_i(k)\right], \\ &j_u \in \mathcal{M}_i = \left\{j_{u1}, j_{u2}, \ldots, j_{u|\mathcal{M}_i|}\right\}, \ j_{u1} < j_{u2} < \cdots < j_{u|\mathcal{M}_i|}, \end{aligned} \tag{25}$$

where

$$\begin{aligned}
C_{j_{x1}} &= \begin{bmatrix} 1_{1\times\alpha(i)r(j_{x1})^{-1}} & & 0 \\ & \ddots & \\ 0 & & 1_{1\times\alpha(i)r(j_{x1})^{-1}} \end{bmatrix} \in \{0,1\}^{r(j_{x1})\times\alpha(i)}, \\
C_{j_{x2}} &= \begin{bmatrix} C'_{j_{x2}} & \cdots & C'_{j_{x2}} \end{bmatrix} \in \{0,1\}^{r(j_{x2})\times\alpha(i)}, \\
C'_{j_{x2}} &= \begin{bmatrix} 1_{1\times\alpha(i)(r(j_{x1})r(j_{x2}))^{-1}} & & & 0 \\ & 1_{1\times\alpha(i)(r(j_{x1})r(j_{x2}))^{-1}} & & \\ & & \ddots & \\ 0 & & & 1_{1\times\alpha(i)(r(j_{x1})r(j_{x2}))^{-1}} \end{bmatrix} \in \{0,1\}^{r(j_{x2})\times\alpha(i)(r(j_{x1})r(j_{x2}))^{-1}}, \\
&\vdots \\
C_{j_{x|\mathcal{N}_i|}} &= \begin{bmatrix} C'_{j_{x|\mathcal{N}_i|}} & \cdots & C'_{j_{x|\mathcal{N}_i|}} \end{bmatrix} \in \{0,1\}^{r(j_{x|\mathcal{N}_i|})\times\alpha(i)}, \\
C'_{j_{x|\mathcal{N}_i|}} &= \begin{bmatrix} 1_{1\times\alpha(i)(r(j_{x1})r(j_{x2})\cdots r(j_{x|\mathcal{N}_i|}))^{-1}} & & 0 \\ & \ddots & \\ 0 & & 1_{1\times\alpha(i)(r(j_{x1})r(j_{x2})\cdots r(j_{x|\mathcal{N}_i|}))^{-1}} \end{bmatrix} \in \{0,1\}^{r(j_{x2})\times\alpha(i)(r(j_{x1})r(j_{x2})\cdots r(j_{x|\mathcal{N}_i|}))^{-1}}, \\
D_{j_{u1}} &= \begin{bmatrix} D'_{j_{u1}} & \cdots & D'_{j_{u1}} \end{bmatrix} \in \{0,1\}^{r(j_{u1})\times\alpha(i)}, \\
D'_{j_{u1}} &= \begin{bmatrix} 1_{1\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1}))^{-1}} & & 0 \\ & \ddots & \\ 0 & & 1_{1\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1}))^{-1}} \end{bmatrix} \in \{0,1\}^{r(j_{x2})\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1}))^{-1}}, \\
&\vdots \\
D_{j_{u|\mathcal{M}_i|}} &= \begin{bmatrix} D'_{j_{u|\mathcal{M}_i|}} & \cdots & D'_{j_{u|\mathcal{M}_i|}} \end{bmatrix} \in \{0,1\}^{r(j_{u|\mathcal{M}_i|})\times\alpha(i)}, \\
D'_{j_{u|\mathcal{M}_i|}} &= \begin{bmatrix} 1_{1\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1})\cdots r(j_{u|\mathcal{M}_i|}))^{-1}} & & 0 \\ & \ddots & \\ 0 & & 1_{1\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1})\cdots r(j_{u|\mathcal{M}_i|}))^{-1}} \end{bmatrix} \\
&\in \{0,1\}^{r(j_{x2})\times\alpha(i)(r(j_{x1})\cdots r(j_{x|\mathcal{N}_i|})r(j_{u1})\cdots r(j_{u|\mathcal{M}_i|}))^{-1}}.
\end{aligned} \tag{26}$$

In a similar way, we can obtain the following constraints for $E[\widetilde{z}_i(k)]$:

$$\begin{aligned}
&E\left[\overline{x}_{j_x}(k)\right] = \widetilde{C}_{j_x} E\left[\widetilde{z}_i(k)\right], \\
&j_x \in \mathcal{N}_i = \left\{j_{x1}, j_{x2}, \ldots, j_{x|\mathcal{N}_i|}\right\}, \ j_{x1} < j_{x2} < \cdots < j_{x|\mathcal{N}_i|}, \\
&\quad \overline{u}_{j_u}(k) = \widetilde{D}^1_{j_u} E\left[\widetilde{z}_i(k)\right], \\
&j_u \in \mathcal{M}_i = \left\{j_{u1}, j_{u2}, \ldots, j_{u|\mathcal{N}_i|}\right\}, \ j_{u1} < j_{u2} < \cdots < j_{u|\mathcal{M}_i|}, \\
&\begin{bmatrix} 1 - v(k) \\ v(k) \end{bmatrix} = \widetilde{D}^2_v E\left[\widetilde{z}_i(k)\right].
\end{aligned} \tag{27}$$

From the definition of $\widetilde{z}_i(k)$, that is, (21), the matrices $\widetilde{C}_{j_x}, \widetilde{D}^1_{j_u}$ can be obtained by replacing $\alpha(i)$ with $2\alpha(i)$ in the matrices C_{j_x}, D_{j_u}. The matrix $\widetilde{D}^2_v$ can be obtained as

$$\widetilde{D}^2_v = \begin{bmatrix} I_2 & \cdots & I_2 \end{bmatrix} \in \{0,1\}^{2\times 2\alpha(i)}. \tag{28}$$

Thus, we can obtain the following problem as a relaxed problem of Problem 6.

Problem 8. Suppose that, for the PLN with the conventional control input u and the structural control input v, the initial state $x(0) = x_0$ is given. Then, find $u(0), u(1), \ldots, u(N-1) \in \{0,1\}^m$ and $v(0), v(1), \ldots, v(N-1) \in [0,1]$ minimizing the cost function (9) subject to (10) and (22)–(27).

By a simple calculation, Problem 8 can be equivalently rewritten as an MILP problem. In this MILP problem, continuous decision variables are $v(k)$, $E[z_i(k)]$, $E[\widetilde{z}_i(k)]$, $k = 0, 1, \ldots, N-1$, and binary decision variables are $\overline{u}_i(k)$, $k = 0, 1, \ldots, N-1$. By solving the MILP problem, we can evaluate the lower bound of the optimal value of the cost function in Problem 3. In this paper, we provide only an approximate solution method. However, since the control input is obtained by solving an MILP problem, the proposed solution method enables us to solve the optimal control problem for several classes of PLNs. In the case where there is no conventional control input, Problem 8 with only the structural control input is equivalent to a linear programing problem.

Remark 9. In analysis/control methods in [6–8], the matrix-based representation for x, which can be obtained from that for one x_i, is used. As a result, the matrix with the size of $\prod_{i=1}^n r(i) \times (\prod_{i=1}^n r(i) \prod_{i=1}^m s(i))$ must be manipulated. In the proposed method, matrix $A^{(i)}$ with the size of $r(i)\times\alpha(i)$ is used ($\alpha(i) = \prod_{j\in\mathcal{N}_i} r(j) \prod_{j\in\mathcal{N}_i} s(j)$). Thus, the proposed method enables us to use only matrices with the smaller size.

Remark 10. The difference between Problems 3 and 8 can be evaluated using null spaces of matrices $C_j, D_j, \widetilde{C}_j, \widetilde{D}^1_j$, and $\widetilde{D}^2_j$. Details are one of future works. If $E[z_i(k)]$ and $E[\widetilde{z}_i(k)]$ can be uniquely derived from (25) and (27), then Problems 3 and 8 are equivalent.

5. Application to Real-Time Pricing of Electricity

In this section, we consider real-time pricing of electricity as an application of optimal control of PLNs. First, the outline of real-time pricing is explained. Next, decision making of consumers is modeled by a PLN. Finally, a numerical example is presented.

5.1. Outline. The outline of real-time pricing systems studied in this paper is illustrated in Figure 1. This system is composed of one controller (the independent system operator, ISO) and multiple electric consumers. If electricity conservation is needed, then the ISO decides to set the price to a high value. To decrease the economic load, consumers conserve electricity. As a result, electricity conservation is achieved. Thus, real-time pricing plays an important role in energy management systems.

In the existing methods of real-time pricing, the price at each time is given by a simple function with respect to power consumptions and voltage deviations and so on [21]. However, it is important to determine the electricity price based on prediction using a mathematical model of electricity consumption. From this viewpoint, in [23, 24], probabilistic models for real-time pricing have been proposed. On the other hand, in [13], a method to model decision making of consumers by a PBN has been proposed. However, the state of a consumer is given by a binary value, that is, 0 (a consumer conserves electricity) or 1 (a consumer normally uses electricity). In this paper, we consider extending the method in [13] to the method using a PLN. As a mathematical model of consumers, a PLN is more appropriate than a PBN. It is one of the future efforts to discuss the relation between probabilistic models in [23, 24] and a PLN.

5.2. Model. Consider modeling the set of consumers as a PLN. The number of consumers is given by n. We suppose that the state x_i for consumer $i \in \{1, 2, \ldots, n\}$ is d-valued; that is, $x_i \in \mathcal{D}_d$ (for simplicity, d does not depend on i). The state implies

$$x_i = \begin{cases} 0 & \text{consumer } i \text{ conserves electricity}, \\ 1, & \\ \vdots & \\ d-1 & \text{consumer } i \text{ maximally uses electricity}. \end{cases} \tag{29}$$

The value of x_i is determined from power consumption of consumer i. For consumer i, we consider the following PLN:

$$x_i(k+1) = \begin{cases} f_1^{(i)} = x_i(k) + 1, \ c_1^{(i)}(k) = a_1^{(i)} + b_1^{(i)} v(k) & (x_i(k) \neq d-1), \\ f_2^{(i)} = x_i(k), \ c_2^{(i)}(k) = a_2^{(i)} + b_2^{(i)} v(k), & \\ f_3^{(i)} = x_i(k) - 1, \ c_3^{(i)}(k) = a_3^{(i)} + b_3^{(i)} v(k) & (x_i(k) \neq 0). \end{cases} \tag{30}$$

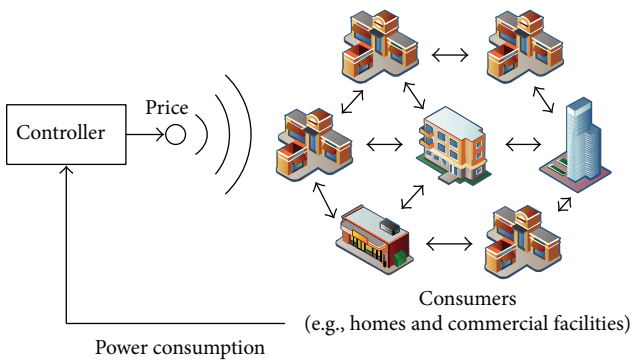

FIGURE 1: Illustration of real-time pricing systems.

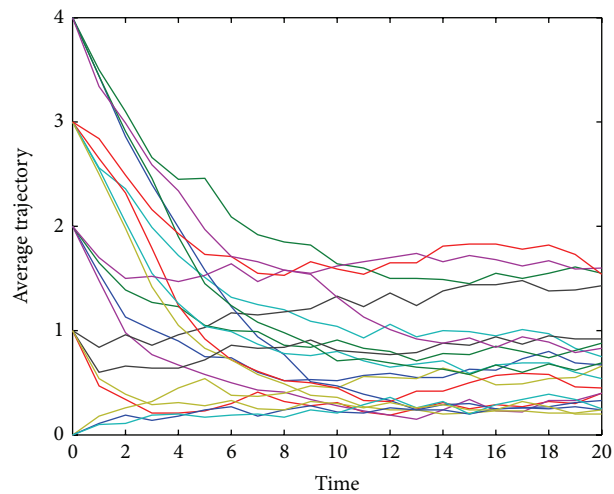

FIGURE 2: Average state trajectory of 100 samples.

In this model, there is no conventional control input, and the structural control input $v(k)$ corresponds to the electricity price. The function $f_1^{(i)}$ implies that power consumption of consumer i increases. The function $f_2^{(i)}$ implies that power consumption of consumer i is not changed. The function $f_3^{(i)}$ implies that power consumption of consumer i decreases. Thus, decision making of consumers can be modeled by a PLN. In [23, 24], it has been shown that the behavior of power consumption is probabilistic based on experimental data. Hence, it is appropriate to adopt a PLN as a model of consumers. We remark that the above functions are an example of models for decision making. Depending on real situations, we may use other logic functions.

Using the PLN-based model obtained, we consider the following problem:

(i) find a time sequence of the price such that consumers conserve electricity as much as possible. However, it is not desirable that the price is too high.

The condition that consumers 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 3 by appropriately setting the weights Q and R.

5.3. Numerical Example. We present a numerical example. Here, Problem 8 is repeatedly solved according to the following procedure of model predictive control (MPC).

Procedure of MPC

Step 1. Set $t = 0$, and give $x(0) = x_0$.

Step 2. Derive $v(t), v(t+1), \ldots, v(t+N-1)$ by solving Problem 8.

Step 3. Apply only $v(t)$ to the system.

Step 4. According to $E[x_i(k+1)]$ (i.e., the discrete probability distribution for $x_i(k+1)$), obtained by using $v(t)$, derive $x_i(k+1) \in \mathcal{D}_i$ randomly. Set $t+1 \rightarrow t$, and return to Step 2.

Consider the case of $n = 20$ and $d = 5$. See the Appendix for parameters $a_l^{(i)}$ and $b_l^{(i)}$. For $v(k)$, the constraint $v(k) \in [0.2, 0.6]$ is imposed. Under this constraint, (10) are satisfied. Parameters in Problem 3 are given as follows:

$$\begin{aligned} &x(0) \\ &= [2\ 4\ 1\ 3\ 2\ 0\ 1\ 4\ 2\ 3\ 3\ 2\ 3\ 1\ 0\ 4\ 3\ 0\ 4\ 1], \\ &N = 10, \\ &Q = 1_{1\times 20}, \\ &R_2 = 50. \end{aligned} \tag{31}$$

Next, we present the computation result. According to the above procedure of MPC, 100 samples of the state trajectory in the time interval $[0, 20]$ were generated. Figure 2 shows the average trajectory of 100 samples. Figure 3 shows one sample of the price (the structural control input) obtained. From these figures, we see that the state becomes small by manipulating the price. In this example, 2000 LP problems were solved. Then, the worst computation time was 0.2548 (sec), and the mean computation time was 0.1297 (sec), where we used Gurobi Optimizer 5.6.2 as an LP solver. Thus, Problem 8 can be solved fast.

Finally, we discuss the optimality. Consider solving Problem 8 once under the above setting. When the obtained control input is applied to the system, the value of the cost function in Problem 3 was 542.8. In the case of $v(k) = 0.2$ (i.e., the constant input), the value of the cost function in Problem 3 was 663.9. In the case of $v(k) = 0.6$, the value of the cost function in Problem 3 was 556.1. From these values, we see that the obtained control input is more effective than trivial control inputs.

6. Conclusion

In this paper, we discussed optimal control of probabilistic logic networks (PLNs) and its application to real-time pricing of electricity. First, in the problem formulation, both

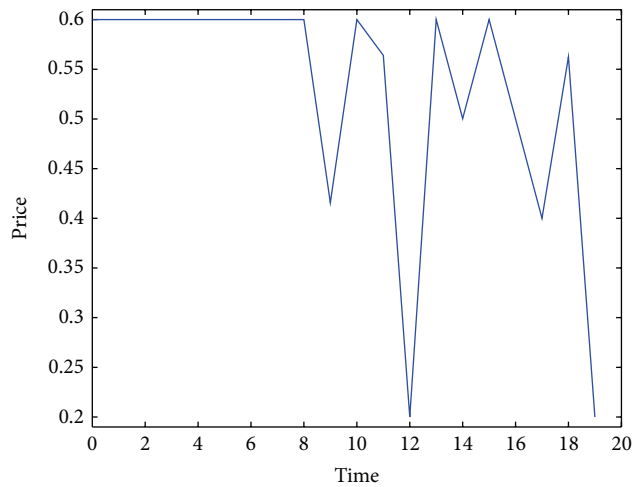

FIGURE 3: One sample of the price obtained.

the conventional control input and the structural control input were introduced. Next, an approximate solution method was proposed. This method is a sophisticated and generalized version of the method proposed in [13]. Finally, we considered real-time pricing of electricity as an application, and we presented a numerical example. The proposed method provides us a new control method for large-scale complex systems.

There are several open problems. It is significant to develop the identification method of PLNs. It is also significant to evaluate the accuracy of an approximation from the theoretical viewpoint.

Appendix

Parameters in Numerical Example

Parameters $a_l^{(i)}$ and $b_l^{(i)}$ in the numerical example are given as follows:

$$\begin{aligned}
&a_1^{(i)} = 0.5,\\
&b_1^{(i)} = -0.5,\\
&a_2^{(i)} = 0.5,\\
&b_2^{(i)} = -0.5,\\
&a_3^{(i)} = 0,\\
&b_3^{(i)} = 1,\\
&\qquad i \in \{1, 6, 11, 16\},\\
&a_1^{(i)} = 0.7,\\
&b_1^{(i)} = -0.5,\\
&a_2^{(i)} = 0.3,\\
&b_2^{(i)} = -0.5,\\
&a_3^{(i)} = 0,\\
&b_3^{(i)} = 1,\\
&\qquad i \in \{2, 7, 12, 17\},\\
&a_1^{(i)} = 0.4,\\
&b_1^{(i)} = -0.5,\\
&a_2^{(i)} = 0.6,\\
&b_2^{(i)} = -0.5,\\
&a_3^{(i)} = 0,\\
&b_3^{(i)} = 1,\\
&\qquad i \in \{3, 8, 13, 18\},\\
&a_1^{(i)} = 0.5,\\
&b_1^{(i)} = -0.4,\\
&a_2^{(i)} = 0.5,\\
&b_2^{(i)} = -0.6,\\
&a_3^{(i)} = 0,\\
&b_3^{(i)} = 1,\\
&\qquad i \in \{4, 9, 14, 19\},\\
&a_1^{(i)} = 0.3,\\
&b_1^{(i)} = -0.3,\\
&a_2^{(i)} = 0.7,\\
&b_2^{(i)} = -0.7,\\
&a_3^{(i)} = 0,\\
&b_3^{(i)} = 1,\\
&\qquad i \in \{5, 10, 15, 20\}.
\end{aligned} \tag{A.1}$$

These parameters in the numerical example are given artificially. Although finding these parameters from real data is important, this topic is one of the future efforts.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This research was partly supported by JST, CREST.

References

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

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

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

[4] 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.

[5] C. Chaouiya, A. Naldi, E. Remy, and D. Thieffry, "Petri net representation of multi-valued logical regulatory graphs," *Natural Computing*, vol. 10, no. 2, pp. 727–750, 2011.

[6] D. Cheng, H. Qi, and Z. Li, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*, Springer, London, UK, 2011.

[7] D. Cheng and X. Xu, "Bi-decomposition of multi-valued logical functions and its applications," *Automatica*, vol. 49, no. 7, pp. 1979–1985, 2013.

[8] Y. Zhao, Z. Li, and D. Cheng, "Optimal control of logical control networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1766–1776, 2011.

[9] 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.

[10] Z. B. Liu, Y. Z. Wang, and H. T. Li, "Two kinds of optimal controls for probabilistic mix-valued logical dynamic networks," *Science China—Information Sciences*, vol. 57, no. 5, pp. 1–10, 2014.

[11] Y. Zhao and D. Cheng, "Controllability and stabilizability of probabilistic logical control networks," in *Proceedings of the 51st IEEE Conference on Decision and Control (CDC '12)*, pp. 6729–6734, Maui, Hawaii, USA, December 2012.

[12] 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.

[13] K. Kobayashi and K. Hiraishi, "A probabilistic approach to control of complex systems and its application to real-time pricing," *Mathematical Problems in Engineering*, vol. 2014, Article ID 906717, 8 pages, 2014.

[14] 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.

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

[16] K. Kobayashi and K. Hiraishi, "ILP/SMT-based method for design of Boolean networks based on singleton attractors," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 6, pp. 1253–1259, 2014.

[17] X. Chen, T. Akutsu, T. Tamura, and W.-K. Ching, "Finding optimal control policy in probabilistic Boolean networks with hard constraints by using integer programming and dynamic programming," in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM '10)*, pp. 240–246, Hong Kong, December 2010.

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

[19] 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, IEEE, Atlanta, Ga, USA, December 2010.

[20] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. S. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm '10)*, pp. 415–420, Gaithersburg, Md, USA, October 2010.

[21] 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.

[22] 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. 95, no. 9, pp. 1512–1517, 2012.

[23] K. Kobayashi and K. Hiraishi, "Algorithm for optimal real-time pricing based on switched Markov chain models," in *Proceedings of the IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT '15)*, pp. 1–5, Washington, DC, USA, Feburary 2015.

[24] I. Maruta and Y. Takarada, "Modeling of dynamics in demand response for real-time pricing," in *Proceedings of the 5th IEEE International Conference on Smart Grid Communications (SmartGridComm '14)*, pp. 806–811, Venice, Italy, November 2014.