Applied Mathematics, 2016, 7, 1734-1747
http://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Verification of Real-Time Pricing Systems Based on Probabilistic Boolean Networks

Koichi Kobayashi[1]*, Kunihiko Hiraishi[2]

[1]Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan
[2]School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan
Email: *k-kobaya@ssi.ist.hokudai.ac.jp

How to cite this paper: Kobayashi, K. and Hiraishi, K. (2016) Verification of Real-Time Pricing Systems Based on Probabilistic Boolean Networks. *Applied Mathematics*, **7**, 1734-1747.
http://dx.doi.org/10.4236/am.2016.715146

Received: July 26, 2016
Accepted: September 13, 2016
Published: September 16, 2016

Abstract

In this paper, verification of real-time pricing systems of electricity is considered using a probabilistic Boolean network (PBN). In real-time pricing systems, electricity conservation is achieved by manipulating the electricity price at each time. A PBN is widely used as a model of complex systems, and is appropriate as a model of real-time pricing systems. Using the PBN-based model, real-time pricing systems can be quantitatively analyzed. In this paper, we propose a verification method of real-time pricing systems using the PBN-based model and the probabilistic model checker PRISM. First, the PBN-based model is derived. Next, the reachability problem, which is one of the typical verification problems, is formulated, and a solution method is derived. Finally, the effectiveness of the proposed method is presented by a numerical example.

Keywords

Model Checking, Probabilistic Boolean Networks, Real-Time Pricing

1. Introduction

In recent years, there has been growing interest in energy and the environment. For problems on energy and the environment such as energy saving, several approaches have been studied (see, e.g., [1] [2]). In this paper, we focus on real-time pricing systems of electricity. 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 (see, e.g., [3]-[6]). 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., [6]). In order to realize more precisely pricing, it is necessary to use a mathematical model of consumers.

On the other hand, in order to deal with complex systems such as power systems and gene regulatory networks, it is one of the appropriate methods to approximate a complex system by a discrete abstract model (see, e.g., [7]). In addition, human decision making is also complex, and is modeled by a discrete model (see, e.g., [8]). Thus, in analysis and control of complex systems and those with human decision making, a discrete model plays an important role. Several discrete models have been proposed so far (see, e.g., [9]). In this paper, we focus on a Boolean network (BN) [10]. 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 addition, the behavior of complex systems is frequently stochastic by the effects of noise. From this viewpoint, a probabilistic BN (PBN) has been proposed in [11]. In a PBN, a Boolean function is randomly decided at each time among the candidates of Boolean functions.

Under the above backgrounds, the authors have proposed in [12] the PBN-based model of real-time pricing systems. In this model, decision making of electric consumers is modeled by a PBN. That is, decisions of a consumer are modeled by Boolean functions, and one of decisions is selected probabilistically. Selection probabilities are controlled by the price at each time. In [12], an approximate algorithm for solving the optimal control problem has been proposed. However, analysis and verification using the PBN-based model have not been considered.

In this paper, we propose a verification method of real-time pricing systems using the PBN-based model and the probabilistic model checker PRISM [13]. Using PRISM, we can verify whether this system satisfies the specification described by probabilistic computation tree logic (PCTL) [14] or not. The reachability problem is considered as one of the typical verification problems, and a numerical example is presented. The proposed method provides us a basic of model-based design of real-time pricing systems.

In Section 2, the outline of real-time pricing systems studied in this paper is explained. In Section 3, the PBN-based model is explained. In Section 4, the verification problem is formulated. In Section 5, a solution method using PRISM is proposed. In Section 6, a numerical example is presented. In Section 7, we conclude this paper.

Notation: For the n-dimensional vector $x = \left[x_1 \ \ x_2 \ \ \cdots \ \ x_n\right]^{\mathrm{T}}$ and the index set $\mathcal{I} = \{i_1, i_2, \cdots, i_m\} \subseteq \{1, 2, \cdots, n\}$, $[x_i]_{i\in\mathcal{I}} := \left[x_{i_1} \ \ x_{i_2} \ \ \cdots \ \ x_{i_m}\right]^{\mathrm{T}}$ is defined.

2 Real-Time Pricing Systems

In this section, we explain the outline of real-time pricing systems studied in this paper.

Figure 1 shows an illustration of real-time pricing systems studied in this paper. This system consists of one controller and multiple electric consumers such as commercial facilities and homes. For an electric consumer, we suppose that each consumer can

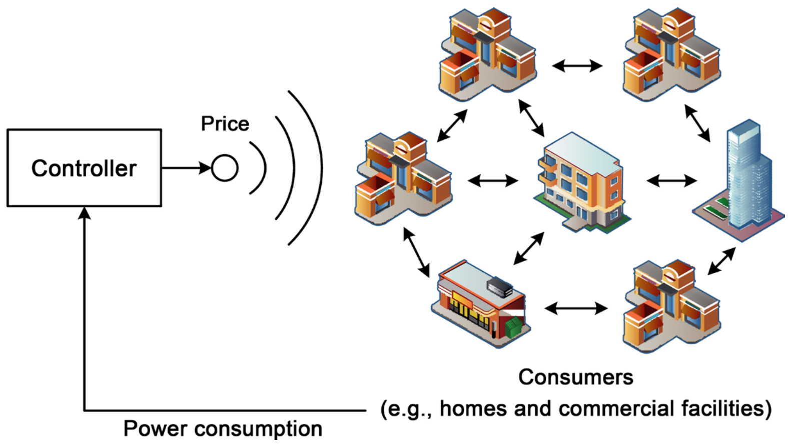

Figure 1. Illustration of real-time pricing systems.

monitor the status of electricity conservation of other consumers. In other words, the status of some consumer affects that of other consumers. 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, *i.e.*, the status of electricity conservation, the controller determines the price at each time. 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. The price does not depend on each consumer, and is uniquely determined.

In this paper, decision making of electric consumers is modeled by a probabilistic Boolean network (PBN). Here, we suppose that each electric consumer has candidates of a decision in electricity conservation, and one of candidates is selected probabilistically depending on the electricity price at the current time. In such a case, it is appropriate to adopt the PBN-based model. In this paper, the property of real-time pricing systems can be verified using the PBN-based model.

3. Modeling Using Probabilistic Boolean Networks

In this section, first, we explain the outline of PBNs. Next, each consumer in real-time pricing systems is modeled by a PBN.

3.1. Probabilistic Boolean Networks

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

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

where $x := \left[x_1\ x_2\ \cdots\ x_n\right]^{\mathrm{T}} \in \{0,1\}^n$ is the state, and $k = 0,1,2,\cdots$ is the discrete time. The set $\mathcal{N}^{(i)} \subseteq \{1,2,\cdots,n\}$ is a given index set, and the function

$f_i : \{0,1\}^{\left|\mathcal{N}^{(i)}\right|} \to \{0,1\}^1$ is a given Boolean function consisting of logical operators such as AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). If $\mathcal{N}^{(i)} = \varnothing$ holds, then $x_i(k+1)$ is uniquely determined as 0 or 1.

Next, we explain a probabilistic Boolean network (PBN) (see [11] 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

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

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

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

Then, the following relation

$$\sum_{l=1}^{q(i)} c_l^{(i)} = 1 \tag{2}$$

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

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

We show a simple example.

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

$$f^{(1)} = \begin{cases} f_1^{(1)} = x_3(k),\ c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_3(k),\ c_2^{(1)} = 0.2, \end{cases}$$

$$f^{(2)} = f_1^{(2)} = x_1(k) \wedge \neg x_3(k),\ c_1^{(2)} = 1.0,$$

$$f^{(3)} = \begin{cases} f_1^{(3)} = x_1(k) \wedge \neg x_2(k),\ c_1^{(3)} = 0.7, \\ f_2^{(3)} = x_2(k),\ c_2^{(3)} = 0.3, \end{cases}$$

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 (2) is satisfied. Next, consider the state trajectory. Then, for $x(0) = [0\ 0\ 0]^{\mathrm{T}}$, we can obtain

$$\mathrm{Prob}\left(x(1) = [0\ 0\ 0]^{\mathrm{T}} \mid x(0) = [0\ 0\ 0]^{\mathrm{T}}\right) = 0.8,$$

$$\mathrm{Prob}\left(x(1) = [1\ 0\ 0]^{\mathrm{T}} \mid x(0) = [0\ 0\ 0]^{\mathrm{T}}\right) = 0.2.$$

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 2 by computing the transition from each state. In Figure 2, 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 $x(k) = [0\ 0\ 0]^{\mathrm{T}}, [0\ 0\ 1]^{\mathrm{T}}, [0\ 1\ 0]^{\mathrm{T}}, [1\ 1\ 0]^{\mathrm{T}}$ is illustrated in Figure 2. □

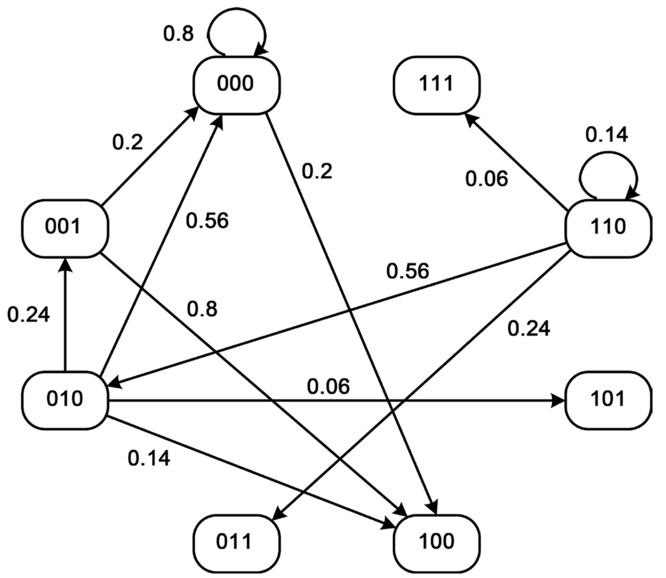

Figure 2. State transition diagram.

3.2. Model of Consumers

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

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

The binary value of x_i is determined by power consumption of consumer i. In addition, we assume that the probability $c_l^{(i)}$ is time-varying and is changed by the price at each time. That is, the probability is given by

$$c_l^{(i)}(k) = a_l^{(i)} + b_l^{(i)} u(k),$$

where $u(k) \in \mathcal{U} \subset \mathcal{R}^1$ is the price (the control input). We assume that the set $\mathcal{U}$ is a finite set, and for any $u \in \mathcal{U}$, two conditions (2) and $0 \le c_l^{(i)}(k) \le 1$ hold. The Boolean function $f_l^{(i)}$ must be derived depending on real situations and experimental results. In this paper, as one of examples, we consider the following situation, which will mimic a real situation.

Let $\mathcal{D}_i \subseteq \{1, 2, \cdots, n\}$, $i = 1, 2, \cdots, n$ denote the set of consumers, which affect to consumer i. We assume that there exists one leader in the local area. The state of a leader is given by x_1. Then, for consumer i, we consider the following model Σ_i:

$$x_i(k+1) = \begin{cases} f_1^{(i)} = 1, \; c_1^{(i)}(k) = a_1^{(i)} + b_1^{(i)} u(k), \\ f_2^{(i)} = 0, \; c_2^{(i)}(k) = a_2^{(i)} + b_2^{(i)} u(k), \\ f_3^{(i)} = x_i(k), \; c_3^{(i)}(k) = a_3^{(i)} + b_3^{(i)} u(k), \\ f_4^{(i)} = g^{(i)}\left(\left[x_j(k)\right]_{j \in \mathcal{D}_i}\right), \; c_4^{(i)}(k) = a_4^{(i)} + b_4^{(i)} u(k), \\ f_5^{(i)} = x_1(k), \; c_5^{(i)}(k) = a_5^{(i)} + b_5^{(i)} u(k), \end{cases} \tag{3}$$

The Boolean functions $f_1^{(i)}$ and $f_2^{(i)}$ imply that consumer *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 consumer *i* is changed depending on the other consumers. The Boolean function $f_5^{(i)}$ implies that the state of consumer *i* is changed depending on the leader. Thus, decision making of consumers can be modeled by a PBN. Of course, we may use other Boolean functions.

4. Problem Formulation

In this section, the verification problem described by probabilistic computation tree logic (PCTL) is formulated for the PBN-based model of consumers (see **Appendix A** for details on PCTL).

Here, the reachability problem is formulated as one of the typical problems. For the system Σ_i, $i = 1, 2, \cdots, n$ given by (3), the output $y(k) = \left[y_1(k) \; y_2(k) \; \cdots \; y_p(k) \right]^{\mathrm{T}} \in \{0,1\}^p$ is defined, where $y_i = x_j$, $j \in \{1, 2, \cdots, n\}$. We remark that the output is not the measured signal. First, the reachability problem is given.

Problem 1. *Suppose that for the system* Σ_i, $i = 1, 2, \cdots, n$ *given by* (3), *the initial state* $x(0) = x_0$, *the control time N, and the target output* y_f *are given. Then, find a maximum probability p satisfying*

$$\mathcal{P}_{\le p}\left(\mathrm{F}^{\le N} \left[y(k) = y_f \right] \right)$$

by manipulating a control input sequence $u(0), u(1), \cdots, u(N-1)$.

Let $P_{\max}$ denote the maximum probability obtained by solving this problem. In this problem, we find a maximum probability that $y(k) = y_f$ holds within time *N*. In the conventional reachability problem, only terminal time is focused, and it is checked whether $y(N) = y_f$ holds or not. In this paper, we focus on not only terminal time N but also other times $0, 1, \cdots, N-1$. Since the system has the control input, we find a maximum probability satisfying the condition. In the case where peak demand is focused on, $y(k) = y_f$ may be replaced with $y(k) \le \gamma$, where γ is a given constant.

Furthermore, by solving Problem 1 at each time, a kind of model predictive control (MPC) can be realized (see Section 5.3 for further details).

5. Solution Method Using PRISM

In this section, we consider a solution method for Problem 1 using the probabilistic model checker PRISM [13].

5.1. Preparation: Transformation of Boolean Functions

As a preparation, the following lemma [15] is introduced.

Lemma 1. *Consider two binary variables* δ_1, δ_2. *Then the following relations hold.*

i) $\neg \delta_1$ is equivalent to $1 - \delta_1$.

ii) $\delta_1 \vee \delta_2$ is equivalent to $\delta_1 + \delta_2 - \delta_1 \delta_2$.

iii) $\delta_1 \wedge \delta_2$ is equivalent to $\delta_1\delta_2$.

For example, $\delta_1 \vee \neg\delta_2$ is equivalently transformed into $\delta_1 + (1-\delta_2) - \delta_1(1-\delta_2) = 1 - \delta_2 + \delta_1\delta_2$. By using this lemma, a Boolean function can be transformed into a polynomial with binary variables.

5.2. Description in PRISM

To solve Problem 1 and the verification problem described by PCTL formulas, the probabilistic model checker PRISM is used. PRISM supports a discrete-time Markov chain (DT-MC), a continuous-time Markov chain (CT-MC), and a Markov decision process (MDP). PRISM consists of three parts: "Model", "Properties", "Simulator". In the "Model" part, a given probabilistic system is described using the PRISM language. In the "Properties" part, the property specification language incorporates temporal logic such as PCTL, and we can verify whether a given PCTL formula holds or not. In the "Simulator", the state trajectories can be simulated.

Using PRISM, consider modeling the system Σ_i, $i = 1, 2, \cdots, n$ given by (3). To explain the PRISM-based method, consider the following model of three consumers:

$$x_1(k+1) = \begin{cases} f_1^{(1)} = 1,\ c_1^{(1)}(k) = 0.1, \\ f_2^{(1)} = 0,\ c_2^{(1)}(k) = 0.025u(k), \\ f_3^{(1)} = x_1(k),\ c_3^{(1)}(k) = 0.9 - 0.1u(k), \\ f_4^{(1)} = x_2(k) \wedge x_3(k),\ c_4^{(1)}(k) = 0.05u(k), \\ f_5^{(1)} = x_1(k),\ c_5^{(1)}(k) = 0.025u(k), \end{cases}$$

$$x_2(k+1) = \begin{cases} f_1^{(2)} = 1,\ c_1^{(2)}(k) = 0.1, \\ f_2^{(2)} = 0,\ c_2^{(2)}(k) = 0.025u(k), \\ f_3^{(2)} = x_1(k),\ c_3^{(2)}(k) = 0.9 - 0.1u(k), \\ f_4^{(2)} = x_1(k) \wedge x_3(k),\ c_4^{(2)}(k) = 0.05u(k), \\ f_5^{(2)} = x_1(k),\ c_5^{(2)}(k) = 0.025u(k), \end{cases}$$

$$x_3(k+1) = \begin{cases} f_1^{(3)} = 1,\ c_1^{(3)}(k) = 0.1, \\ f_2^{(3)} = 0,\ c_2^{(3)}(k) = 0.025u(k), \\ f_3^{(3)} = x_1(k),\ c_3^{(3)}(k) = 0.9 - 0.1u(k), \\ f_4^{(3)} = x_1(k) \wedge x_2(k),\ c_4^{(3)}(k) = 0.05u(k), \\ f_5^{(3)} = x_1(k),\ c_5^{(3)}(k) = 0.025u(k). \end{cases}$$

In addition, $\mathcal{U}$ is given by $\mathcal{U} = \{3, 4, 5\}$. Then, the PRISM source code describing this system is shown as follows.

```
01: mdp
02: module RTP1
03:     x1: [0..1] init 1;
04:     [RTP] u=3 -> 0.1:(x1'=1) + 0.075:(x1'=0) + 0.6:(x1'=x1) + 0.15:(x1'=x2*x3)
                 + 0.075:(x1'=x1)
```

```
05:     [RTP] u=4 -> 0.1:(x1'=1) + 0.1:(x1'=0) + 0.5:(x1'=x1) + 0.2:(x1'=x2*x3)
                          + 0.1:(x1'=x1)
06:     [RTP] u=5 -> 0.1:(x1'=1) + 0.125:(x1'=0) + 0.4:(x1'=x1) + 0.25:(x1'=x2*x3)
                          + 0.125:(x1'=x1)
07: endmodule
08: module RTP2
09:     x2:[0..1] init 1;
10:     [RTP] u=3 -> 0.1:(x2'=1) + 0.075:(x2'=0) + 0.6:(x2'=x2) + 0.15:(x2'=x1*x3)
                           + 0.075:(x2'=x1)
11:     [RTP] u=4 -> 0.1:(x2'=1) + ... (omit)
12:     [RTP] u=5 -> 0.1:(x2'=1) + ... (omit)
13: endmodule
14: module RTP3
15:     x3:[0..1] init 1;
16:     [RTP] u=3 -> 0.1:(x3'=1) + 0.075:(x3'=0) + 0.6:(x3'=x3) + 0.15:(x3'=x1*x2)
                           + 0.075:(x3'=x1)
17:     [RTP] u=4 -> 0.1:(x3'=1) + ... (omit)
18:     [RTP] u=5 -> 0.1:(x3'=1) + ... (omit)
19: endmodule
20: module input
21:     u:[3..5] init 3;
22:     [RTP] u=3 -> (u'=3);
23:     [RTP] u=3 -> (u'=4);
24:     [RTP] u=3 -> (u'=5);
25:     [RTP] u=4 -> (u'=3);
26:     [RTP] u=4 -> (u'=4);
27:     [RTP] u=4 -> (u'=5);
28:     [RTP] u=5 -> (u'=3);
29:     [RTP] u=5 -> (u'=4);
30:     [RTP] u=5 -> (u'=5);
31: endmodule.
```

In line 1, it is described that a given system is an MDP, *i.e.*, the control input (in other words, the nondeterministic variable) that must decide is included. In lines 2-7, the dynamics for x_1 (consumer 1) are modeled. In line 3, it is described that x_1 takes a binary value, and the initial value of x_1 is given by $x_1(0)=1$. In line 4, if $u(k)=3$ holds, then the value of x_1 at the next time is given by 1 with the probability 0.1, 0 with the probability 0.075, x_1 (*i.e.*, the state is not changed) with the probability 0.6, x_2x_3 (corresponding to $x_2(k)\wedge x_3(k)$ [1]) with the probability 0.15, and x_1 with the probability 0.15. Similarly, in line 5, the case of $u(k)=4$ is described. In line 6, the case of $u(k)=5$ is described. In lines 8-13, the dynamics for x_2 (consumer 2) are modeled. In lines 14-19, the dynamics for x_3 (consumer 3) are modeled. In this sys-

[1]In PRISM, given Boolean functions may be directly used (see http://www.prismmodelchecker.org/ for further details).

tem, a discrete probabilistic distribution is given for each x_i. Hence, in PRISM, the dynamics for each x_i must be modeled separately. In lines 20-31, the property of the control input is described as a nondeterministic variable. We note here that the initial value of the control input must be given (see line 21). Finally, to associate with each module, [RTP] is described in lines 4-6, 10-12, 16-18, 22-30.

From the above example, we see that the system Σ_i, $i = 1, 2, \cdots, n$ given by (3) can be described by PRISM. Finally, we present a procedure for deriving the PRISM source code as follows. In the following procedure, without loss of generality, the input set $\mathcal{U}$ is given by $\mathcal{U} = \{1, 2, \cdots, |\mathcal{U}|\}$.

Derivation Procedure of PRISM Source Code:

Step 1: Transform each Boolean function into a polynomial with binary variables by using Lemma 1. Let $\hat{f}_l^{(i)}$ denote the obtained polynomial.

Step 2: Describe that a given system is an MDP.

Step 3: Compute the probability $c_l^{(i)}$ for each element of $\mathcal{U}$. Let $c_{l,p}^{(i)}$ denote the probability for $p \in \mathcal{U}$.

Step 4: Describe module RTP *i*, $i = 1, 2, \cdots, n$ as follows.

module RTP *i*;

$x_i: [0..1]$ init $x_i(0)$;

[RTP] $u = 1 \to c_{1,1}^{(i)} : \left(x_i' = \hat{f}_1^{(i)}\right) + \cdots + c_{5,1}^{(i)} : \left(x_i' = \hat{f}_5^{(i)}\right)$;

$\vdots$

[RTP] $u = |\mathcal{U}| \to c_{1,|\mathcal{U}|}^{(i)} : \left(x_i' = \hat{f}_1^{(i)}\right) + \cdots + c_{5,|\mathcal{U}|}^{(i)} : \left(x_i' = \hat{f}_5^{(i)}\right)$;

endmodule.

Step 5: Describe the control input *u* as follows.

module input

u: $[1..|\mathcal{U}|]$ init $u(0)$;

[RTP] $u_i = 1 \to (u_i' = 1)$;

$\vdots$

[RTP] $u_i = 1 \to (u_i' = |\mathcal{U}|)$;

$\vdots$

[RTP] $u_i = |\mathcal{U}| \to (u_i' = 1)$;

$\vdots$

[RTP] $u_i = |\mathcal{U}| \to (u_i' = |\mathcal{U}|)$;

endmodule.

The above procedure is the improved version of the procedure proposed in [16].

5.3. Verification and Application to MPC

Several properties described by PCTL formulas can be verified by using the obtained model on PRISM. We use the "Properties" part in PRISM.

Consider solving Problem 1 (the reachability problem). Then, we use $P_{\max}$ prepared in PRISM. Suppose $y = [y_1 \ y_2]^{\mathrm{T}}$ and $y_f = [0 \ 1]^{\mathrm{T}}$. Then in PRISM, this problem is described by

$$P_{\max} = ?\left[\mathrm{F} <= \mathrm{N}\left(y_1 = 0\right) \& \left(y_2 = 1\right)\right].$$

This implies that find a maximum probability $P_{\max}$ satisfying the following condition: at time $k = 0,1,\cdots,N$, the number of times that $y(k) = y_f$ holds is greater than or equal to 1, *i.e.*, this code expresses the reachability problem itself.

From the above results, we see that the verification problem can be easily implemented by using PRISM. The control input sequence $u(0),u(1),\cdots,u(N-1)$ is obtained simultaneously, but in PRISM 4.0.3, the obtained control input sequence cannot be displayed except for the case of $N = \infty$. In the case of $N = \infty$, the discrete-time Markov chain can be obtained as the closed-loop system of a given system. The control input sequence can be obtained by exploratory analysis using the simulator in PRISM. Otherwise, this sequence can be obtained by solving the control problem such as the optimal control problem. In both cases, the verification result will be useful.

On the other hand, the problem of finding $P_{\max}$ and a control input sequence can be regarded as a kind of the control problem. Noting that the initial value of the control input must be given, a kind of MPC can be realized by the following procedure.

[Procedure of MPC]

Step 1: Set $t = 0$, and determine the current state $x(t) = x_t$ according to power consumption.

Step 2: Find the current control input $u^*(t)$ maximizing $P_{\max}$. That is, for each $u(t) \in \mathcal{U}$, solve Problem 1.

Step 3: Apply only the control input at *t*, *i.e.*, $u^*(t)$, to the plant.

Step 4: Set $t := t+1$, determine $x(t) = x_t$ according to power consumption, and go to Step 2.

6. Numerical Example

We present a numerical example. For Σ_i, $i = 1,2,\cdots,n$ given by (3), parameters are given as follows:

$$n = 8,$$

$$\mathcal{D}_1 = \{2, n\},$$

$$\mathcal{D}_i = \{i-1, i+1\},\ i = 2,3,\cdots,n-1,$$

$$\mathcal{D}_8 = \{1, n-1\},$$

$$a_1^{(i)} = 0.1,\ b_1^{(i)} = 0,$$

$$a_2^{(i)} = 0,\quad b_2^{(i)} = 0.025,$$

$$a_3^{(i)} = 0.9,\ b_3^{(i)} = -0.1,$$

$$a_4^{(i)} = 0,\quad b_4^{(i)} = 0.05,$$

$$a_5^{(i)} = 0,\quad b_4^{(i)} = 0.025,$$

$$\mathcal{U} = \{3,4,5,6,7\}.$$

We remark that for any $u \in \mathcal{U}$, two conditions (2) and $0 \le c_l^{(i)}(k) \le 1$ hold. The Boolean function $g^{(i)}$ is given by

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

In Problem 1, the control time N, the output, and the target output are given by

$$N=10,$$
$$y_i(k)=x_i(k),\ i=1,2,\cdots,n,$$
$$y_f=\begin{bmatrix}0 & \cdots & 0\end{bmatrix}^{\mathrm{T}}.$$

In this example, we consider the following cases:

- Case 1: The initial state is given by $x_i(0)=1$ (all consumers normally use electricity).
  Case 1-1: The initial input is given by $u(0)=3$.
  Case 1-2: The initial input is given by $u(0)=7$.
- Case 2: The initial state is given by $x_4(0)=0$ and $x_i(0)=1$, $i\neq 4$ (only consumer 4 conserves electricity).
  Case 2-1: The initial input is given by $u(0)=3$.
  Case 2-2: The initial input is given by $u(0)=7$.
- Case 3: The initial state is given by $x_1(0)=0$ and $x_i(0)=1$, $i\neq 1$ (only consumer 1 (leader) conserves electricity).
  Case 3-1: The initial input is given by $u(0)=3$.
  Case 3-2: The initial input is given by $u(0)=7$.

Next, we present the computation result. Table 1 shows $P_{\max}$ for each case. By checking $P_{\max}$, we can verify the status of electricity conservation. If $P_{\max}$ is large, then there is a trend that consumers conserve electricity. From Table 1, we see the following facts:

1) It is desirable that the initial input (price) is given by $u(0)=7$.
2) Even if one consumer, who is not the leader, conserves electricity, then a contribution to electricity conservation is small.
3) If the leader conserves electricity, then a contribution to electricity conservation is large.

Thus, using the PBN-based model, we can analyze real-time pricing systems in a quantitative way.

7. Conclusions

In this paper, using a probabilistic Boolean network (PBN), we discussed verification of

Table 1. Computation result.

Case	$P_{\max}$
Case 1-1	0.6248
Case 1-2	0.6630
Case 2-1	0.6455
Case 2-2	0.6828
Case 3-1	0.7454
Case 3-2	0.7756

Scientific Research Publishing

real-time pricing systems of electricity. The PBN-based model and PRISM enable us an easy and convenient verification. As one of the verification problems, the reachability problem was considered. In addition, application to model predictive control was also discussed. The proposed method provides us verification/control methods for real-time pricing systems.

There are several open problems. It is significant to develop the identification method of Boolean functions and parameters $a_l^{(i)}, b_l^{(i)}$ in (3). Once Boolean functions and parameters can be obtained, the proposed method enables us quantitative analysis. Furthermore, for large-scale systems, there is a possibility that PRISM does not work. In such a case, we may use the assume-guarantee verification technique [17], which is one of the compositional verification techniques. Details are one of the future efforts. It is also significant to consider extending a PBN to a probabilistic system with multi-valued logic functions (see e.g., [18]-[21] for further details about such probabilistic systems). Since the PBN-based model expresses human decision making in the purchasing behavior, the proposed method is related to analysis of the consumer behavior in economics. It is important to clarify the relation between the proposed method and the existing method in economics. The proposed method is the first step toward mathematical analysis of the consumer behavior.

Acknowledgements

This research was partly supported by JST, CREST and Grant-in-Aid for Scientific Research (C) 26420412.

References

[1] Camacho, E.F., Samad, T., Garcia-Sanz, M. and Hiskens, I. (2011) Control for Renewable Energy and Smart Grids. In: Samad, T. and Annaswamy, A.M., Eds., *The Impact of Control Technology*, IEEE Control Systems Society, New York, USA.

[2] Ruihua, Z., Yumei, D. and Yuhong, L. (2010) New Challenges to Power System Planning and Operation of Smart Grid Development in China. 2010 *International Conference on Power System Technology*, Hangzhou, China, 24-28 October 2010, 1-8.

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

[4] Roozbehani, M., Dahleh, M. and Mitter, S. (2010) On the Stability of Wholesale Electricity Markets under Real-Time Pricing. *Proceedings of the* 49*th IEEE Conference on Decision and Control*, Atlanta, USA, 15-17 December 2010, 1911-1918.
http://dx.doi.org/10.1109/cdc.2010.5718173

[5] Samadi, P., Mohsenian-Rad, A.-H., Schober, R., Wong, V.W.S. and Jatskevich, J. (2010) Optimal Real-Time Pricing Algorithm Based on Utility Maximization for Smart Grid. *Proceedings of the* 2010 *First IEEE International Conference on Smart Grid Communications*, Gaithersburg, USA, 4-6 October 2010, 415-420.
http://dx.doi.org/10.1109/SMARTGRID.2010.5622077

[6] Vivekananthan, C., Mishra, Y. and Ledwich, G. (2013) A Novel Real Time Pricing Scheme for Demand Response in Residential Distribution Systems. *Proceedings of the* 38*th Annual*

Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10-13 November 2013, 1954-1959.

[7] Tabuada, P. (2009) Verification and Control of Hybrid Systems. Springer, New York, USA. http://dx.doi.org/10.1007/978-1-4419-0224-5

[8] Adomi, M., Shikauchi, Y. and Ishii, S. (2010) Hidden Markov Model for Human Decision Process in a Partially Observable Environment. *Proceedings of the* 20*th International Conference on Artificial Neural Networks*, *LNCS* 6353, Thessaloniki, Greece, 15-18 September 2010, 94-103. http://dx.doi.org/10.1007/978-3-642-15822-3_12

[9] Jong, H.D. (2002) Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. *Journal of Computational Biology*, **9**, 67-103. http://dx.doi.org/10.1089/10665270252833208

[10] Kauffman, S.A. (1969) Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets. *Journal of Theoretical Biology*, **22**, 437-467. http://dx.doi.org/10.1016/0022-5193(69)90015-0

[11] Shmulevich, I., Dougherty, E.R., Kim, S. and Zhang, W. (2002) Probabilistic Boolean Networks: A Rule-Based Uncertainty Model for Gene Regulatory Networks. *Bioinformatics*, **18**, 261-274. http://dx.doi.org/10.1093/bioinformatics/18.2.261

[12] Kobayashi, K. and Hiraishi, K. (2014) A Probabilistic Approach to Control of Complex Systems and Its Application to Real-Time Pricing. *Mathematical Problems in Engineering*, **2014**, Article ID: 906717.

[13] Kwiatkowska, M., Norman, G. and Parker, D. (2011) PRISM 4.0: Verification of Probabilistic Real-Time Systems. *Proceedings of the* 23*rd International Conference on Computer Aided Verification*, LNCS 6806, Snowbird, USA, 14-20 July 2011, 585-591.

[14] Ciesinski, F. and Groesser, M. (2004) On Probabilistic Computation Tree Logic. In: Baier, C., Haverkort, B.R., Hermanns, H., Katoen, J.-P. and Siegle, M., Eds., *Validation of Stochastic Systems*, Springer, Berlin Heidelberg,147-188.

[15] Williams, H.P. (1999) Model Building in Mathematical Programming. 4th Edition, John Wiley & Sons, Ltd., New York, USA.

[16] Kobayashi, K. and Hiraishi, K. (2012) Symbolic Approach to Verification and Control of Deterministic/Probabilistic Boolean Networks. *IET Systems Biology*, **6**, 215-222. http://dx.doi.org/10.1049/iet-syb.2012.0018

[17] Kwiatkowska, M., Norman, G., Parker, D. and Qu, H. (2010) Assume-Guarantee Verification for Probabilistic Systems. 16*th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, *LNCS* 6015, Paphos, Cyprus, 20-28 March 2010, 23-37.

[18] Cheng, D. and Xu, X. (2013) Bi-Decomposition of Multi-Valued Logical Functions and Its Applications. *Automatica*, **49**, 1979-1985. http://dx.doi.org/10.1016/j.automatica.2013.03.013

[19] Kobayashi, K. and Hiraishi, K. (2015) Optimal Control of Probabilistic Logic Networks and Its Application to Real-Time Pricing of Electricity. *Mathematical Problems in Engineering*, **2015**, Article ID: 952310.

[20] Zhao, Y., Li, Z. and Cheng, D. (2011) Optimal Control of Logical Control Networks. *IEEE Transactions on Automatic Control*, **56**, 1766-1776. http://dx.doi.org/10.1109/TAC.2010.2092290

[21] Zhao, Y. and Cheng, D. (2012) Controllability and Stabilizability of Probabilistic Logical Control Networks. *Proceedings of the* 51*st IEEE Conference on Decision and Control*, Maui, USA, 10-13 December 2012, 6729-6734. http://dx.doi.org/10.1109/cdc.2012.6427395

Appendix A. Probabilistic Computation Tree Logic

In classical propositional logic, truth-value of 0 (false) or 1 (true) is time-invariant. Temporal logic is an extension of propositional logic, and deals with time evolution of truth-value. Since a PBN is a discrete-time system, we also consider temporal logic in discrete-time. First, computation tree logic (CTL) is explained as a class of temporal logics. Next, we introduce probabilistic CTL (PCTL) (see [14] for further details).

In CTL, logical operators and temporal operators are used. The logical operators usually consist of $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$. The temporal operators consists of quantifiers over paths A, E and path-specific quantifiers F, G, X, U. CTL formulas, state formulas, and path formulas are defined as follows:

1) Propositional variables and propositional constants (true or false) are state formulas.
2) If ϕ, ψ are state formulas, then $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \rightarrow \psi$, and $\phi \leftrightarrow \psi$ are also state formulas.
3) If ϕ is path formula, then Eϕ and Aϕ are state formulas.
4) If ϕ, ψ are state formulas, then Xϕ, Fϕ, Gϕ, and ϕUψ are path formulas.
5) All state and path formulas consist of the above formulas, and all CTL formulas consist of state formulas.

Next, suppose that ϕ, ψ are given as propositional variables. Then the meaning of each quantifier over paths is explained as follows:

- Aϕ: ϕ has to hold on all paths starting from the current state (All).
- Eϕ: there exists at least one path starting from the current state where ϕ holds (Exists).

Furthermore, the meaning of each path-specific quantifier is also explained as follows:

- Fϕ: ϕ eventually has to hold (somewhere on the subsequent path) (Finally).
- Gϕ: ϕ has to hold on the entire subsequent path (Globally).
- Xϕ: ϕ has to hold at the next state (neXt).
- ϕUψ: ϕ has to hold until at some position ψ holds. This implies that ψ will be verified in the future.

In PCTL, the notion of probability is added in CTL, that is, for the CTL formula ϕ, consider $\mathcal{P}_{\rhd\lhd p}(\phi)$, $\rhd\lhd \in \{\leq, <, \geq, >\}$, $p \in [0,1]$. For example, $\mathcal{P}_{\leq p}(\phi)$ implies that if ϕ is true with the probability that is less than or equal to p, then $\mathcal{P}_{\leq p}(\phi)$ is true, otherwise $\mathcal{P}_{\leq p}(\phi)$ is false.

Finally, the temporal operator F is improved to $\mathrm{F}^{\leq N}$. For the propositional variable ϕ, $\mathrm{F}^{\leq N}\phi$ implies that ϕ eventually has to hold until time N.