

Research Article

Verification and Optimal Control of Context-Sensitive Probabilistic Boolean Networks Using Model Checking and Polynomial Optimization

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 30 August 2013; Accepted 28 October 2013; Published 23 January 2014

Academic Editors: Y. Lai, R. Purohit, and T. Roegnavaldsson

Copyright © 2014 K. Kobayashi and K. Hiraishi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the significant topics in systems biology is to develop control theory of gene regulatory networks (GRNs). In typical control of GRNs, expression of some genes is inhibited (activated) by manipulating external stimuli and expression of other genes. It is expected to apply control theory of GRNs to gene therapy technologies in the future. In this paper, a control method using a Boolean network (BN) is studied. A BN is widely used as a model of GRNs, and gene expression is expressed by a binary value (ON or OFF). In particular, a context-sensitive probabilistic Boolean network (CS-PBN), which is one of the extended models of BNs, is used. For CS-PBNs, the verification problem and the optimal control problem are considered. For the verification problem, a solution method using the probabilistic model checker PRISM is proposed. For the optimal control problem, a solution method using polynomial optimization is proposed. Finally, a numerical example on the WNT5A network, which is related to melanoma, is presented. The proposed methods provide us useful tools in control theory of GRNs.

1. Introduction

Control of gene regulatory networks (GRNs) is one of the significant topics in the field of systems biology, and is also one of the basics of therapeutic interventions (see, e.g., [1]) in the future. Furthermore, in recent years, the experimental result on control of GRNs has been obtained in [2]. That is, feedback control of synthetic biological circuits has been implemented, and the experimental result in which cellular behavior is regulated by control has been obtained. This result suggests that control methods of GRNs can be realized. Motivated by the above backgrounds, we study a control method of GRNs.

GRNs are in general modeled by ordinary/partial differential equations with high nonlinearity and high dimensionality. In order to deal with such a system, it is important to consider a simple model, and various models such as Bayesian networks, Boolean networks (BNs) [3], hybrid systems (piecewise affine models), and Petri nets have been developed so far (see, e.g., [4] for further details). In control problems, BNs and hybrid systems are frequently used [5–10]. In the hybrid systems-based approach, the classes of

GRNs are limited to low-dimensional systems, because the computation time to solve the control problem is too long. In a BN, gene expression is expressed by a binary value (ON or OFF), and dynamics such as interactions between genes are expressed by Boolean functions [3]. In, for example, [11], it is pointed out that a BN is too simple as a model of GRNs. However, there is an advantage that a BN can be relatively applied to large-scale systems. Furthermore, since the behavior of GRNs is 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 Boolean network (PBN) has been proposed in [12], and further, a context-sensitive PBN (CS-PBN) has been proposed as a general form of PBNs [13, 14]. In a CS-PBN, the deciding time is also randomly selected.

In this paper, a CS-PBN is adopted as a model of GRNs, and for CS-PBNs, the verification problem and the optimal control problem are considered. In the verification of PBNs, a solution method using the probabilistic model checker PRISM [15] has been proposed in [16]. However, this PRISM-based method for PBNs has not been extended to that for

CS-PBNs. In optimal control of PBNs and CS-PBNs, many results have been obtained so far (see, e.g., [13, 14, 17–24]). In many existing results, state transition diagrams with 2^n nodes (i.e., $2^n \times 2^n$ transition probability matrices) must be computed for a (CS-)PBN with n genes. In order to compute state transition diagrams, several issues such as memory consumption must be considered in implementation, and it is desirable to directly use a given Boolean function. The authors have proposed in [22, 23] control methods in which state transition diagrams are not computed. Comparing the methods in [22, 23] with other existing results [13, 14, 17–21, 24], the methods in [22, 23] can relatively handle more large-scale GRNs. The method in [22] can be applied to PBNs and CS-PBNs, but the expected value of a given nonnegative function cannot be evaluated as a cost function (objective function). In the method in [23], the expected value of a given nonnegative function can be used as a cost function, and the optimal control problem is reduced to a polynomial optimization problem. However, this method has been proposed for PBNs, and an extension to CS-PBNs has not been discussed so far. Thus, for verification of CS-PBNs, the PRISM-based method for PBNs [16] is extended to that for CS-PBNs. For optimal control of CS-PBNs, a solution method using polynomial optimization [23] is extended to that for CS-PBNs. Furthermore, the effectiveness of the proposed methods is presented by a numerical example on the WNT5A network, which is related to melanoma. The proposed methods provide us useful tools in control theory of GRNs.

This paper is organized as follows. In Section 2.1, a CS-PBN is explained. In Section 2.2, a solution method for the verification problem is proposed. In Section 2.3, a solution method for the optimal problem is proposed. In Section 3, a numerical example is presented. In Section 4, we conclude this paper.

Notation. Let \mathcal{R} denote the set of real numbers. Let $\{0, 1\}^n$ denote the set of n -dimensional vectors, which consists of elements 0 and 1. For a matrix M , M^T denotes the transpose of M .

2. Materials and Methods

2.1. Context-Sensitive Probabilistic Boolean Networks. First, we introduce a probabilistic Boolean network (PBN). Consider the following PBN:

$$\begin{aligned} x_1(k+1) &= f^{(1)}(k, x(k), u(k)), \\ x_2(k+1) &= f^{(2)}(k, x(k), u(k)), \\ &\vdots \\ x_n(k+1) &= f^{(n)}(k, x(k), u(k)), \end{aligned} \quad (1)$$

where $x = [x_1 \ x_2 \ \cdots \ x_n]^T \in \{0, 1\}^n$ is the state (e.g., the expression of genes), $u = [u_1 \ u_2 \ \cdots \ u_m]^T \in \{0, 1\}^m$ is the control input (e.g., the expression of genes), that is, the

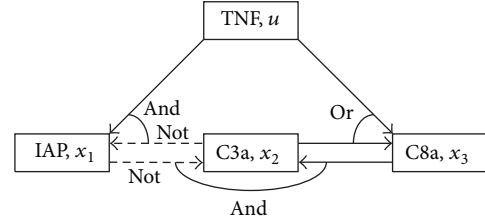


FIGURE 1: Simplified model of an apoptosis network. Activation (solid), inhibition (broken).

value of u can be arbitrarily given, and $k = 0, 1, 2, \dots$ is the discrete time. For a fixed $k \in \{0, 1, \dots\}$, $f^{(i)} : \{0, 1, \dots\} \times \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^1$ is a given Boolean function consisting of logical operators such as AND (\wedge), OR (\vee), and NOT (\neg). In deterministic Boolean networks, $x(k+1)$ is uniquely determined for given k , $x(k)$, and $u(k)$. In PBNs, the candidates of $f^{(i)}(k, x(k), u(k))$ are given, and for each x_i , selecting one Boolean function is probabilistically independent at each time. Let $f_j^{(i)}(x(k), u(k))$, $j = 1, 2, \dots, l(i)$, denote the candidates of $f^{(i)}(k, x(k), u(k))$. The probability that $f_j^{(i)}(x(k), u(k))$ is selected is defined by

$$c_j^{(i)} := \text{Prob}(f^{(i)}(k, x(k), u(k)) = f_j^{(i)}(x(k), u(k))). \quad (2)$$

Then, the following relation:

$$\sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad (3)$$

must be satisfied. Probabilistic distributions are derived from experimental results, but details are one of the future works. Then, a method for inferring a probabilistic Boolean network will be useful (see, e.g., [25]).

Example 1. As a simple example, consider the following deterministic Boolean network of an apoptosis network [26, 27] (see also Figure 1):

$$\begin{aligned} x_1(k+1) &= \neg x_2(k) \wedge u(k), \\ x_2(k+1) &= \neg x_1(k) \wedge x_3(k), \\ x_3(k+1) &= x_2(k) \vee u(k), \end{aligned} \quad (4)$$

where the concentration level (high or low) of the inhibitor of apoptosis proteins (IAP) is denoted by x_1 , the concentration level of the active caspase 3 (C3a) by x_2 , and the concentration level of the active caspase 8 (C8a) by x_3 . The concentration level of the tumor necrosis factor (TNE, a stimulus) is denoted by u and is regarded as the control input. Although Boolean dynamics in the above system are synchronous, both synchronous and asynchronous dynamics will be included.

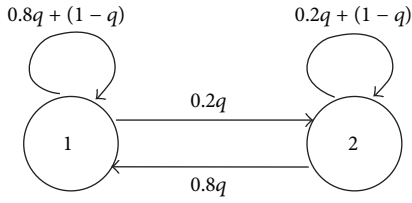


FIGURE 2: Discrete-time Markov chain in $f^{(3)}$.

From this viewpoint, we consider the following PBN induced by the above system:

$$f^{(1)} = \begin{cases} f_1^{(1)} = \neg x_2(k) \wedge u(k), & c_1^{(1)} = 0.6, \\ f_2^{(1)} = x_1(k), & c_2^{(1)} = 0.4, \end{cases} \quad (5)$$

$$f^{(2)} = \begin{cases} f_1^{(2)} = \neg x_1(k) \wedge x_3(k), & c_1^{(2)} = 0.7, \\ f_2^{(2)} = x_2(k), & c_2^{(2)} = 0.3, \end{cases} \quad (6)$$

$$f^{(3)} = \begin{cases} f_1^{(3)} = x_2(k) \vee u(k), & c_1^{(3)} = 0.8, \\ f_2^{(3)} = x_3(k), & c_2^{(3)} = 0.2, \end{cases} \quad (7)$$

where $l(1) = l(2) = l(3) = 2$, and we give $c_j^{(i)}$ satisfying $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$. In addition, all state trajectories can be expressed as the state transition diagram with 2^3 nodes.

In PBNs, we suppose that selecting one Boolean function is probabilistically independent at each time. However, it will be natural to consider the situation that switches of Boolean functions do not occur frequently. From this viewpoint, a context-sensitive PBN (CS-PBN) has been proposed in [13, 14]. In CS-PBNs, the deciding time of Boolean functions is also selected randomly. Hereafter, let $q \in [0, 1]$ denote the probability that Boolean functions are switched at time k , and a pair of the system (1) and q is called a CS-PBN.

To compare CS-PBNs with PBNs, consider (7) as a simple example. In PBNs, a switch of $f_1^{(3)}$ and $f_2^{(3)}$ functions does not depend on the Boolean function at time $k - 1$. In CS-PBNs, a switch of $f_1^{(3)}$ and $f_2^{(3)}$ is decided by the discrete-time Markov chain in Figure 2. In other words, this switch depends on the Boolean function at time $k - 1$. Owing to this difference, a control/verification method for CS-PBNs cannot be directly derived from that for PBNs.

2.2. Verification Using Model Checking. First, the reachability problem is formulated as the verification problem studied in this paper. The reachability problem is one of the typical verification problems. For a given CS-PBN, the output $y(k) = [y_1(k) \ y_2(k) \ \cdots \ y_p(k)]^T \in \{0, 1\}^p$ is defined, where $y_i = x_j$, $i = 1, 2, \dots, p$, $j \in \mathcal{F} \subseteq \{1, 2, \dots, n\}$. We remark that the output does not mean the measured signal. First, the reachability problem is formulated as follows.

Problem 2 (reachability problem). Suppose that, for CS-PBN with the output, the initial state $x(0) = x_0$, the initial Boolean

function $f^{(i)}(0, x(0), u(0)) = f_{j_0(i)}^{(i)}(x(0), u(0))$ ($j_0(i) \in \{1, 2, \dots, l(i)\}$), the control time N , and the target output y_f are given ($u(0)$ is not given). Then, find a maximum probability P_{\max} that $y(k) = y_f$ holds within time N by manipulating a control input sequence $u(0), u(1), \dots, u(N - 1)$.

In the standard 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 $0, 1, \dots, N - 1$. Furthermore, since a CS-PBN has the control input, which can be regarded as a nondeterministic variable, we find a maximum probability satisfying the condition.

Next, we will propose a solution method for Problem 2. As a preparation, the following lemma [28] is introduced.

Lemma 3. 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 on the real number field.

To solve Problem 2, the probabilistic model checker PRISM [15] 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,” and “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 (probabilistic computation tree logic) [29], and we can verify if a given PCTL formula holds. In the “Simulator,” the state trajectories can be simulated.

Now, using PRISM, we propose a method for modeling a given CS-PBN. By modeling a given CS-PBN via PRISM, Problem 2 can be solved. In the PRISM-based method, Boolean functions in a given PBN can be directly used. To explain the PRISM-based method, consider the PBN (5)–(7) in Example 1 and $q = 0.5$. Suppose that the initial state and the initial Boolean function are given by $x_0 = [1 \ 1 \ 1]^T$, $f^{(1)}(0) = f_1^{(1)}$, $f^{(2)}(0) = f_1^{(2)}$, and $f^{(3)}(0) = f_1^{(3)}$ (i.e., $j_0(1) = j_0(2) = j_0(3) = 1$), respectively. By using Lemma 3, each Boolean function can be transformed into some polynomial on the field of real numbers. Then, the PRISM code describing this CS-PBN is shown in Figure 3.

In line 1, it is described that a given system is a MDP; that is, the control input (in other words, the nondeterministic variable) that must decide is included. In line 2, the probability q is given by $q = 0.5$. In lines 3–7, the discrete-time Markov chain such as Figure 2 is modeled for $f^{(1)}$. The probabilistic variable $d1$ corresponds to $j \in \{1, 2\}$ in $f_j^{(i)}$. In line 4, $f^{(1)}(0)$ is given by $f^{(1)}(0) = f_1^{(1)}$. In lines 5–6, the behavior of $d1$ is modeled. In line 5, it is described that if $d_1 = 1$ holds, then

```

(1) mdp
(2)   const double q=0.5;
(3) module CSPBNm1
(4)   d1:[1..2] init 1;
(5)   [CSPBN] d1 = 1 -> 0.6*q+(1-q):(d1' = 1) + 0.4*q:(d1' = 2);
(6)   [CSPBN] d1 = 2 -> 0.6*q:(d1' = 1) + 0.4*q+(1-q):(d1' = 2);
(7) endmodule
(8) module CSPBN1
(9)   [CSPBN] x1:[0..1] init 1;
(10)  [CSPBN] d1 = 1 -> 1.0:(x1' = u-x2*u);
(11)  [CSPBN] d1 = 2 -> 1.0:(x1' = x1);
(12) endmodule
(13) module CSPBNm2
(14)  d2:[1..2] init 1;
(15)  [CSPBN] d2 = 1 -> 0.7*q+(1-q):(d2' = 1) + 0.3*q:(d2' = 2);
(16)  [CSPBN] d2 = 2 -> 0.7*q:(d2' = 1) + 0.3*q+(1-q):(d2' = 2);
(17) endmodule
(18) module CSPBN2
(19)  [CSPBN] x2:[0..1] init 1;
(20)  [CSPBN] d2 = 1 -> 1.0:(x2' = x3-x1*x3);
(21)  [CSPBN] d2 = 2 -> 1.0:(x2' = x2);
(22) endmodule
(23) module CSPBNm3
(24)  d3:[1..2] init 1;
(25)  [CSPBN] d3 = 1 -> 0.8*q+(1-q):(d3' = 1) + 0.2*q:(d3' = 2);
(26)  [CSPBN] d3 = 2 -> 0.8*q:(d3' = 1) + 0.2*q+(1-q):(d3' = 2);
(27) endmodule
(28) module CSPBN3
(29)  [CSPBN] x3:[0..1] init 1;
(30)  [CSPBN] d3 = 1 -> 1.0:(x3' = x2+u-x2*u);
(31)  [CSPBN] d3 = 2 -> 1.0:(x3' = x3);
(32) endmodule
(33) module input
(34)  u:[0..1] init 0;
(35)  [CSPBN] u = 0 -> (u' = 0);
(36)  [CSPBN] u = 0 -> (u' = 1);
(37)  [CSPBN] u = 1 -> (u' = 0);
(38)  [CSPBN] u = 1 -> (u' = 1);
(39) endmodule

```

FIGURE 3: PRISM code expressing the CS-PBN.

the next state d_1' is 1 with the probability $0.6q + (1 - q)$ and 2 with the probability $0.4q$. In lines 8–12, $f^{(1)}$ is modeled. In line 9, it is described that x_1 takes a binary value, and the initial value of x_1 is given by 1. In line 10, $f_1^{(1)}$ is modeled. In line 11, $f_2^{(1)}$ is modeled. In a similar way, $f^{(2)}$ is modeled in lines 13–22, and $f^{(3)}$ is modeled in lines 23–32. In CS-PBNs, a discrete probabilistic distribution is given for each $f^{(i)}$. Hence, $f^{(i)}$, $i = 1, 2, 3$, must be modeled separately. To associate with each module, [CSPBN] is described. Finally, in lines 33–39, the property of the control input is described as a nondeterministic variable. Note that the initial value of the control input $u(0)$ must be given (see line 34). Hence, PRISM must be executed for two cases of $u(0) = 0$ and $u(0) = 1$.

The above explanation is the outline of the PRISM-based modeling method. Based on the above example, we propose a procedure for deriving the PRISM code expressing a general CS-PBN.

2.2.1. Procedure for Modeling CS-PBNs

Step 1. Transform each Boolean function into a polynomial on the real number field by using Lemma 3. The obtained Boolean functions are denoted by $\hat{f}_j^{(i)}$.

Step 2. Describe that a given system is a MDP, and give q .

Step 3. Describe modules CSPBNm i and CSPBN i , $i = 1, 2, \dots, n$, as follows:

module CSPBNm i

$d_i : [1..l(i)]$ init $j_0(i)$;

[CSPBN] $d_i = 1 \rightarrow c_1^{(i)}q + (1 - q) : (d_i' = 1) + c_2^{(i)}q : (d_i' = 2) + \dots + c_{l(i)}^{(i)}q : (d_i' = l(i))$;

[CSPBN] $d_i = 2 \rightarrow c_1^{(i)}q : (d_i' = 1) + c_2^{(i)}q + (1 - q) : (d_i' = 2) + \dots + c_{l(i)}^{(i)}q : (d_i' = l(i))$;

```

:
[CSPBN]  $d_i = l(i) \rightarrow c_1^{(i)}q : (d'_i = 1) + c_2^{(i)}q : (d'_i = 2) + \dots + c_{l(i)}^{(i)}q + (1 - q) : (d'_i = l(i));$ 
endmodule
module CSPBN i
     $x_i : [0..1]$  init  $x_i(0);$ 
    [CSPBN]  $d_i = 1 \rightarrow 1.0 : (x'_i = \tilde{f}_1^{(i)}(x, u));$ 
    [CSPBN]  $d_i = 2 \rightarrow 1.0 : (x'_i = \tilde{f}_2^{(i)}(x, u));$ 
    :
    [CSPBN]  $d_i = l(i) \rightarrow 1.0 : (x'_i = \tilde{f}_{l(i)}^{(i)}(x, u));$ 
endmodule
    
```

Step 4. Describe the control input $u_i, i = 1, 2, \dots, m$, as follows:

```

module input i
     $u_i : [0..1]$  init  $u_i(0);$ 
    [PBN1]  $u_i = 0 \rightarrow (u'_i = 0)$ 
    [PBN1]  $u_i = 0 \rightarrow (u'_i = 1)$ 
    [PBN1]  $u_i = 1 \rightarrow (u'_i = 0)$ 
    [PBN1]  $u_i = 1 \rightarrow (u'_i = 1)$ 
endmodule
    
```

Finally, consider solving Problem 2. For solving this problem, we use Pmax prepared in PRISM. For example, suppose that $y = [y_1 \ y_2]^T$ and $y_f = [0 \ 1]^T$. Then, in PRISM, Problem 2 can be described by

$$Pmax =? [F <= N (y1 = 0) \& (y2 = 1)]. \quad (8)$$

Therefore, we see that Problem 2 can be solved using PRISM. The control input sequence $u(0), u(1), \dots, u(N - 1)$ 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 CS-PBN.

2.3. Optimal Control Using Polynomial Optimization. Consider the following problem.

Problem 4 (optimal control problem). Suppose that, for CS-PBN, the initial state $x(0) = x_0$, the initial Boolean function $f^{(i)}(0, x(0), u(0)) = f_{j_0(i)}^{(i)}(x(0), u(0)) (j_0(i) \in \{1, 2, \dots, l(i)\})$, and the control time N are given ($u(0)$ is not given). Then, find a control input sequence $u(0), u(1), \dots, u(N - 1)$ minimizing the cost function

$$J = E \left[\sum_{k=0}^{N-1} \{Qx(k) + Ru(k)\} + Q_f x(N) \mid x(0) = x_0 \right], \quad (9)$$

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

According to the following two reasons, the linear cost function (9) is appropriate. (i) 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. (ii) In control of GRNs, expression of a certain gene is frequently focused (see, e.g., [19]). That is, in the cost function, the quadratic term such as $x_i(k)x_j(k), i \neq j$, is not necessary.

For a PBN, the authors have derived the following recursive representation of the expected value of the state:

$$E[x_i(k+1)] = \sum_{j=1}^{l(i)} c_j^{(i)} \tilde{f}_j^{(i)}(E[x(k)], u(k)), \quad (10)$$

where the condition $x(0) = x_0$ in the expected value is omitted. See [23] for further details. In this paper, this representation is extended to that in CS-PBNs.

First, we present a simple example. Consider the PBN (5)–(7) in Example 1. Suppose that the initial state and the initial Boolean function are given by $x_0 = [1 \ 1 \ 1]^T, f^{(1)}(0) = f_1^{(1)}, f^{(2)}(0) = f_1^{(2)}$, and $f^{(3)}(0) = f_1^{(3)}$, respectively.

Consider deriving the expected value of the state at time $k = 1$. Suppose that $u(0) = 0$. Since the Boolean function at time $k = 0$ is given, the state at time $k = 1$ is uniquely derived as

$$E[x(1) \mid x(0) = x_0, u(0) = 0] = x(1) = [0 \ 0 \ 1]^T. \quad (11)$$

Hereafter, the condition such as $x(0) = x_0, u(0) = 0$ in the expected value is omitted. Next, consider deriving the expected value of the state at time $k = 2$. We remark that, for each x_i , the discrete-time Markov chain such as Figure 2 can be obtained. For example, the probability that $f_1^{(1)}$ is selected at time $k = 1$ is $0.6q + (1 - q)$, and the probability that $f_2^{(1)}$ is selected at time $k = 1$ is $0.4q$. Suppose that $u(1) = 1$. Then, we can obtain

$$\begin{aligned}
 E[x_1(2)] &= E[f^{(1)}(1, x(1), u(1))] \\
 &= (0.6q + (1 - q)) E[f_1^{(1)}(x(1), u(1))] \\
 &\quad + 0.4q E[f_2^{(1)}(x(1), u(1))] \\
 &= (0.6q + (1 - q)) \cdot (1 - 0) \cdot 1 + 0.4q \cdot 0 \\
 &= 0.6q + (1 - q), \\
 E[x_2(2)] &= E[f^{(2)}(1, x(1), u(1))] \\
 &= (0.7q + (1 - q)) E[f_1^{(2)}(x(1), u(1))] \\
 &\quad + 0.3q E[f_2^{(2)}(x(1), u(1))] \\
 &= (0.7q + (1 - q)) \cdot (1 - 0) \cdot 1 + 0.3q \cdot 0 \\
 &= 0.7q + (1 - q),
 \end{aligned}$$

$$\begin{aligned}
E[x_3(2)] &= E[f^{(3)}(1, x(1), u(1))] \\
&= (0.8q + (1 - q))E[f_1^{(3)}(x(1), u(1))] \\
&\quad + 0.2qE[f_2^{(3)}(x(1), u(1))] \\
&= (0.8q + (1 - q)) \cdot (0 + 1 - 0 \cdot 1) + 0.2q \cdot 1 \\
&= 1.0.
\end{aligned} \tag{12}$$

Finally, we remark that the probability that some Boolean function $f_j^{(i)}$ is selected is time-varying. For example, the probability that $f_1^{(1)}$ is selected at time $k = 2$ is $(0.6q + (1 - q))^2 + 0.4q \cdot 0.6q$, and the probability that $f_2^{(1)}$ is selected at time $k = 2$ is $(0.6q + (1 - q)) \cdot 0.4q + 0.4q \cdot (0.2q + (1 - q))$.

Next, consider a general case. From the observation of the above example, we can obtain the following recursive representation:

$$\begin{aligned}
E[x_i(k+1)] &= \sum_{j=1}^{l(i)} d_j^{(i)}(k) \tilde{f}_j^{(i)}(E[x(k)], u(k)), \\
d^{(i)}(k+1) &= P^{(i)} d^{(i)}(k),
\end{aligned} \tag{13}$$

where $d^{(i)}(k) = [d_1^{(i)}(k) \ d_2^{(i)}(k) \ \cdots \ d_{l(i)}^{(i)}(k)]^\top$ and

$$P^{(i)} = \begin{bmatrix} c_1^{(i)}q + (1 - q) & c_2^{(i)}q & \cdots & c_{l(i)}^{(i)}q \\ c_1^{(i)}q & c_2^{(i)}q + (1 - q) & \cdots & c_{l(i)}^{(i)}q \\ \vdots & \vdots & \ddots & \vdots \\ c_1^{(i)}q & c_2^{(i)}q & \cdots & c_{l(i)}^{(i)}q + (1 - q) \end{bmatrix}. \tag{14}$$

Therefore, Problem 4 can be reduced to the following polynomial optimization problem:

$$\begin{aligned}
&\text{find} && E[x(k+1)] \in \mathcal{R}^n, \quad u(k) \in \mathcal{R}^m, \\
&&& k = 0, 1, \dots, N-1, \\
&\text{min} && \text{Cost function (9)}, \\
&\text{subject to} && \text{System (13)}, \quad i = 1, 2, \dots, n, \\
&&& x(0) = x_0, \\
&&& u_i(k)(u_i(k) - 1) = 0.
\end{aligned} \tag{15}$$

The constraint $u_i(k)(u_i(k) - 1) = 0$ guarantees that $u(k)$ is a binary variable. A polynomial optimization problem can be solved by using a suitable solver such as SparsePOP [30].

3. Results and Discussion

In this section, we present a numerical example on the WNT5A network. First, the WNT5A network is explained. Next, computational results are presented.

3.1. WNT5A Network. Consider the GRN with the gene WNT5A, which is related to melanoma. A Boolean network model is given by

$$\begin{aligned}
x_1(k+1) &= \neg x_6(k), \\
x_2(k+1) &= (\neg x_2(k) \wedge x_4(k) \wedge x_6(k)) \\
&\quad \vee \{x_2(k) \wedge (x_4(k) \vee x_6(k))\}, \\
x_3(k+1) &= \neg x_7(k), \\
x_4(k+1) &= x_4(k), \\
x_5(k+1) &= x_2(k) \vee \neg x_7(k), \\
x_6(k+1) &= x_3(k) \vee x_4(k), \\
x_7(k+1) &= \neg x_2(k) \vee x_7(k),
\end{aligned} \tag{16}$$

where the concentration level (high or low) of the gene WNT5A is denoted by x_1 , the concentration level of the gene pirin by x_2 , the concentration level of the gene S100P by x_3 , the concentration level of the gene RET1 by x_4 , the concentration level of the gene MART1 by x_5 , the concentration level of the gene HADHB by x_6 , and the concentration level of the gene STC2 by x_7 . See [31] for further details.

Next, suppose that the control input u is given by x_2 (the concentration level of the gene pirin), according to the discussion in [19]. By replacing x_2 and x_3, x_4, \dots, x_7 with u and x_2, x_3, \dots, x_6 , respectively, we can obtain the following model:

$$\begin{aligned}
x_1(k+1) &= f_d^{(1)}(x(k), u(k)) = \neg x_5(k), \\
x_2(k+1) &= f_d^{(2)}(x(k), u(k)) = \neg x_6(k), \\
x_3(k+1) &= f_d^{(3)}(x(k), u(k)) = x_3(k), \\
x_4(k+1) &= f_d^{(4)}(x(k), u(k)) = \neg x_6(k) \vee u(k), \\
x_5(k+1) &= f_d^{(5)}(x(k), u(k)) = x_2(k) \vee x_3(k), \\
x_6(k+1) &= f_d^{(6)}(x(k), u(k)) = x_6(k) \vee \neg u(k).
\end{aligned} \tag{17}$$

Furthermore, we add the probabilistic behavior as follows:

$$\begin{aligned}
&x_i(k+1) \\
&= \begin{cases} f_d^{(i)}(x(k), u(k)), & \text{with the probability } c_1, \\ x_i(k), & \text{with the probability } c_2, \end{cases} \tag{18}
\end{aligned}$$

where $l(i) = 2$ holds. Thus, we can obtain the PBN model expressing a WNT5A network.

3.2. Computational Result on Verification. Consider solving Problem 4. For the PBN (18), we assume that c_1 and c_2 are given by $c_1 = 0.5$ and $c_2 = 0.5$, respectively. In the WNT5A network, it is important to inhibit the concentration level x_1 of the gene WNT5A [32]. From this fact, we set $y = x_1$ and $y_f = 0$. The initial state is given by $x_0 = [1 \ 0 \ 0 \ 1 \ 0 \ 0]^\top$.

The initial Boolean function is given by $f_d^{(i)}$. In addition, we set $N = 5$.

Next, we show the computation result. Then, we can obtain $P_{\max} = 0.7215$ for $q = 0.3$, $P_{\max} = 0.6587$ for $q = 0.5$, and $P_{\max} = 0.6489$ for $q = 0.7$. It is desirable that P_{\max} is close to 1. Hence, we see that the performance is degraded for a larger q .

3.3. Computational Result on Optimal Control. Consider solving Problem 4. For the PBN (18), we assume that c_1 and c_2 are given by $c_1 = 0.8$ and $c_2 = 0.2$, respectively. Since the concentration level x_1 must be inhibited, the weights Q , Q_f , and R in Problem 4 are given by

$$\begin{aligned} Q &= [1 \ 0 \ 0 \ 0 \ 0 \ 0], & R &= 1, \\ Q_f &= [10 \ 0 \ 0 \ 0 \ 0 \ 0], \end{aligned} \quad (19)$$

respectively. The initial state is given by $x_0 = [1 \ 1 \ 0 \ 1 \ 0 \ 0]^T$. The initial Boolean function is given by $f_d^{(i)}$. In addition, we set $N = 5$ and $q = 0.3$.

Next, we show the computation result. By solving Problem 4, we can obtain $u(0) = u(1) = 1$, $u(2) = u(3) = u(4) = 0$. The expected value of the state at each time is obtained as

$$\begin{aligned} E[x(1)] &= [1 \ 1 \ 0 \ 1 \ 1 \ 0]^T, \\ E[x(2)] &= [0.06 \ 1 \ 0 \ 1 \ 1 \ 0]^T, \\ E[x(3)] &= [0.0061 \ 1 \ 0 \ 1 \ 1 \ 0]^T, \\ E[x(4)] &= [0.0008 \ 0.22 \ 0 \ 0.22 \ 1 \ 0.9866]^T, \\ E[x(5)] &= [0.0001 \ 0.0448 \ 0 \ 0.0448 \ 0.3385 \ 0.998]^T. \end{aligned} \quad (20)$$

Hence, we see that the concentration level x_1 of the gene WNT5A is inhibited with time.

In addition, the optimal value J^* of the cost function was 5.23. For $q = 0.5$ and $q = 0.7$, we can obtain $J^* = 5.41$ and $J^* = 5.57$, respectively. From these values, we see that the performance is degraded for a larger q .

4. Conclusions

In this paper, we discussed verification and optimal control for a context-sensitive probabilistic Boolean network (CS-PBN), which is one of the models for gene regulatory networks (GRNs). In verification, the PRISM-based method for PBNs [16] was extended to that for CS-PBNs. In optimal control, the optimal control method for PBNs [23] was extended to that for CS-PBNs. A CS-PBN is a generalized version of a PBN, and it enables us to consider several situations. Furthermore, as a numerical example, we considered the WNT5A network, which is related to melanoma. The proposed methods provide us useful tools in control theory of GRNs.

In recent years, a stochastic Boolean network [33] has been proposed as a new representation of PBNs. In addition, to simplify a given Boolean network, the Karnaugh map

realization of a Boolean network has been proposed in [34]. These modeling methods will be useful for reducing the computational burden. Future efforts will focus on applying these modeling methods to the control problem and the verification for CS-PBNs.

Conflict of Interests

The authors declare that they have no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was partially supported by Grant-in-Aid for Young Scientists (B) 23760387.

References

- [1] H. Santos-Rosa and C. Caldas, "Chromatin modifier enzymes, the histone code and cancer," *European Journal of Cancer*, vol. 41, no. 16, pp. 2381–2402, 2005.
- [2] A. Miliadis-Argeitis, S. Summers, J. Stewart-Ornstein et al., "In silico feedback for in vivo regulation of a gene expression circuit," *Nature Biotechnology*, vol. 29, no. 12, pp. 1114–1116, 2011.
- [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] 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.
- [5] T. Akutsu, M. Hayashida, W.-K. Ching, and M. K. Ng, "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 670–679, 2007.
- [6] S.-I. Azuma, E. Yanagisawa, and J.-I. Imura, "Controllability analysis of biosystems based on piecewise-affine systems approach," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 139–152, 2008.
- [7] K. Kobayashi, J.-I. Imura, and K. Hiraishi, "Polynomial-time algorithm for controllability test of a class of Boolean biological networks," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2010, Article ID 210685, 12 pages, 2010.
- [8] K. Kobayashi and K. Hiraishi, "Optimal control of Boolean biological networks modeled by Petri nets," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E96-A, no. 2, pp. 532–539, 2013.
- [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] C. J. Langmead and S. K. Jha, "Symbolic approaches for finding control strategies in boolean networks," *Journal of Bioinformatics and Computational Biology*, vol. 7, no. 2, pp. 323–338, 2009.
- [11] 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.
- [12] 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.

- [13] 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.
- [14] 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.
- [15] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: verification of probabilistic real-time systems," in *Proceedings of the 23rd International Conference on Computer Aided Verification*, vol. 6806 of *Lecture Notes in Computer Science*, pp. 585–591, Springer, 2011.
- [16] K. Kobayashi and K. Hiraishi, "Symbolic approach to verification and control of deterministic/probabilistic Boolean networks," *IET Systems Biology*, vol. 6, no. 6, pp. 215–222, 2012.
- [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," *International Journal of Data Mining and Bioinformatics*, vol. 7, no. 3, pp. 321–343, 2013.
- [18] W.-K. Ching, S.-Q. Zhang, Y. Jiao, T. Akutsu, N.-K. Tsing, and A. S. Wong, "Optimal control policy for probabilistic Boolean networks with hard constraints," *IET Systems Biology*, vol. 3, no. 2, pp. 90–99, 2009.
- [19] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1-2, pp. 169–191, 2003.
- [20] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: the imperfect information case," *Bioinformatics*, vol. 20, no. 6, pp. 924–930, 2004.
- [21] S. Z. Denic, B. Vasic, C. D. Charalambous, and R. Palanivelu, "Robust control of uncertain context-sensitive probabilistic Boolean networks," *IET Systems Biology*, vol. 3, no. 4, pp. 279–295, 2009.
- [22] K. Kobayashi and K. Hiraishi, "An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks," *Automatica*, vol. 47, no. 6, pp. 1260–1264, 2011.
- [23] 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.
- [24] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2375–2387, 2006.
- [25] S. Marshall, L. Yu, Y. Xiao, and E. R. Dougherty, "Inference of a probabilistic boolean network from a single observed temporal sequence," *Eurasip Journal on Bioinformatics and Systems Biology*, vol. 2007, Article ID 32454, 15 pages, 2007.
- [26] M. Chaves, "Methods for qualitative analysis of genetic networks," in *Proceedings of the European Control Conference*, pp. 671–676, 2009.
- [27] L. Tournier and M. Chaves, "Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics," *Journal of Theoretical Biology*, vol. 260, no. 2, pp. 196–209, 2009.
- [28] H. P. Williams, *Model Building in Mathematical Programming*, John Wiley & Sons, 4th edition, 1999.
- [29] F. Ciesinski and M. Groesser, "On probabilistic computation tree logic," in *Validation of Stochastic Systems*, vol. 2925 of *Lecture Notes in Computer Science*, pp. 147–188, 2004.
- [30] SparsePOP, <http://www.is.titech.ac.jp/~kojima/SparsePOP/SparsePOP.html>.
- [31] Y. Xiao and E. R. Dougherty, "The impact of function perturbations in Boolean networks," *Bioinformatics*, vol. 23, no. 10, pp. 1265–1273, 2007.
- [32] A. T. Weeraratna, Y. Jiang, G. Hostetter et al., "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," *Cancer Cell*, vol. 1, no. 3, pp. 279–288, 2002.
- [33] J. Liang and J. Han, "Stochastic Boolean networks: an efficient approach to modeling gene regulatory networks," *BMC Systems Biology*, vol. 6, article 113, 2012.
- [34] R. K. Layek, A. Datta, and E. R. Dougherty, "From biological pathways to regulatory networks," *Molecular BioSystems*, vol. 7, no. 3, pp. 843–851, 2011.