Title	An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks
Author(s)	Kobayashi, Koichi; Hiraishi, Kunihiko
Citation	Automatica, 47(6): 1260-1264
Issue Date	2011-02-21
Type	Journal Article
Text version	author
URL	http://hdl.handle.net/10119/10790
Rights	NOTICE: This is the author's version of a work accepted for publication by Elsevier. Koichi Kobayashi and Kunihiko Hiraishi, Automatica, 47(6), 2011, 1260-1264, http://dx.doi.org/10.1016/j.automatica.2011.01.035
Description	

An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks

Koichi Kobayashi [a], Kunihiko Hiraishi [a],

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

Abstract

A Boolean network is one of the models of biological networks such as gene regulatory networks, and has been extensively studied. In particular, a probabilistic Boolean network (PBN) is well known as an extension of Boolean networks, but in the existing methods to solve the optimal control problem of PBNs, it is necessary to compute the state transition diagram with 2^n nodes for a given PBN with n states. To avoid this computation, an integer programming-based approach is proposed for a context-sensitive PBN (CS-PBN), which is a general form of PBNs. In the proposed method, a CS-PBN is transformed into a linear system with binary variables, and the optimal control problem is reduced to an integer linear programming problem. By a numerical example, the effectiveness of the proposed method is shown.

Key words: context-sensitive probabilistic Boolean networks; optimal control; integer programming; biological networks.

1 Introduction

In recent years, there have been a lot of studies in analysis and control of biological networks such as gene regulatory networks. One of the final goals in these studies is to find a method for a suitable medication which can be used for drug discovery and cancer treatment [10]. In order to deal with such a system, it is important to consider a simple model, and various models have been developed so far. In particular, Boolean networks [8] are well known as one of the models, and have been extensively studied (see e.g., [2]). In this model, dynamics such as interactions between genes are expressed by a set of Boolean functions. So this model is simple, and can be applied to large-scale systems. In addition, since the behavior of biological networks is probabilistic by effects of noise, it is appropriate that Boolean functions are randomly decided at each time. From this viewpoint, a probabilistic Boolean network (PBN) has been proposed in [14]. Furthermore, a context-sensitive PBN (CS-PBN) in which the deciding time is randomly selected has been proposed as a general form of PBNs [7,12].

In a similar way to standard dynamical systems, CS-PBNs including PBNs have the state and the control input. We assume that the value (0 or 1) of the control input can be arbitrarily determined. The control input in biological networks has the following significance. For example, the value of the control input expresses whether a stimulus is given to a cell. Then the control input is designed to obtain the state trajectory that transits from the initial state to the desired one. So the control input can represent the current status of therapeutic interventions, which are realized by radiation, chemotherapy, and so on. Thus, in order to develop gene therapy technologies (see e.g., [9,13]) in future, it is important to consider control methods of CS-PBNs. Motivated by the above discussion, control methods of CS-PBNs have been developed so far [7,12]. However in these existing works, the state transition diagram with 2^n nodes must be computed for a CS-PBN with n states. This is a crucial weakness.

In this paper, for CS-PBNs, we propose a new control method in which the state transition diagram is not computed. In the proposed method, first, a linear state equation and linear inequalities with binary variables are derived from given Boolean functions expressing dynamics. A random decision of Boolean functions is expressed as a discrete-time Markov chain. This chain is also expressed as a linear form by using binary variables. Therefore, a CS-PBN is expressed as a constrained linear system with binary variables. Then the problem of finding a control

⋆ This paper was not presented at any IFAC meeting. This work was partially supported by Grant-in-Aid for Young Scientists (B) 20760278 and Scientific Research (C) 21500009.
Corresponding author K. Kobayashi. Tel. +81-761-51-1282. Fax +81-761-51-1149.

Email addresses: k-kobaya@jaist.ac.jp (Koichi Kobayashi), hira@jaist.ac.jp (Kunihiko Hiraishi).

input minimizing the lower bound of the cost function is rewritten as an integer linear programming problem. By using the proposed method, for CS-PBNs such that the existing method cannot be applied, we can derive the control input within the practical computation time.

Notation: For a matrix M, $\ln M$ denotes the matrix such that the (i, j)-th element is given as the natural logarithm of the (i, j)-th element in M.

2 Context-sensitive probabilistic Boolean networks

First, we introduce a probabilistic Boolean network (PBN). Consider the following PBN

$$x(k+1) = f(x(k), u(k)) \tag{1}$$

where $x \in \{0,1\}^n$ is the state, $u \in \{0,1\}^m$ is the control input, and $k = 0, 1, 2, \ldots$ is the discrete time. $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^n$ is a given Boolean function. The i-th element of the state x and the i-th element of the Boolean function f are denoted by x_i and $f^{(i)}$, respectively. In deterministic Boolean networks, the next state $x(k+1)$ is uniquely determined for given $x(k)$ and $u(k)$. In PBNs, candidates of $f^{(i)}$ are given, and selecting one Boolean function is probabilistically independent at each time. The candidates of $f^{(i)}$ are denoted by $f_j^{(i)}$, $j = 1, 2, \ldots, l(i)$, and the probability that $f_j^{(i)}$ is selected is denoted by $c_j^{(i)} = \mathrm{Prob}(f^{(i)} = f_j^{(i)}) \in [0, 1]$. Then the relation $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$ must be satisfied.

Example 1 *As a simple example, consider the following deterministic Boolean network of an apoptosis network [6]:* $x_1(k+1) = \neg x_2(k) \wedge u(k)$, $x_2(k+1) = \neg x_1(k) \wedge x_3(k)$, *and* $x_3(k+1) = x_2(k) \vee u(k)$, *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 (TNF, 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. From this viewpoint, we consider the following PBN induced by the above system*

$$f^{(1)} = \begin{cases} f_1^{(1)} = \neg x_2(k) \wedge u(k), & c_1^{(1)} = 0.6, \\ f_2^{(1)} = x_1(k), & c_2^{(1)} = 0.4, \end{cases} \tag{2}$$

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

$$f^{(3)} = \begin{cases} f_1^{(3)} = x_2(k) \vee u(k), & c_1^{(3)} = 0.8, \\ f_2^{(3)} = x_3(k), & c_2^{(3)} = 0.2 \end{cases} \tag{4}$$

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 orbits can be expressed as the state transition diagram with* 2^n *nodes.*

Although in PBNs selecting one Boolean function is probabilistically independent at each time, it will be natural to consider that switchings of Boolean functions do not occur frequently, and may depend on the occurrence of an external stimulus. From this viewpoint, a context-sensitive PBN (CS-PBN) has been proposed in [7,12]. In CS-PBNs, the deciding time of Boolean functions is also selected randomly. Hereafter, the probability that Boolean functions are switched at time k is given as $q(k) \in [0, 1]$, and a pair of the system (1) and the probability $q(k)$ is called a CS-PBN.

3 Problem formulation

First, the following two notations are defined. Suppose that $j(i, k) \in \{1, 2, \ldots, l(i)\}$ is given for fixed i-th element of a given Boolean function and time k, and $q(k)$ is also given. Then by $\pi_{j(1,k),j(2,k),\ldots,j(n,k)}(k)$ or $\pi_{\bar{j}}(k)$ for short, denote the probability that the Boolean function $[\ f_{j(1,k)}^{(1)}\ \ f_{j(2,k)}^{(2)}\ \ \cdots\ \ f_{j(n,k)}^{(n)}\ ]^T$ is selected at time k. Furthermore $\pi_{\bar{j}}(k_1, k_2) := \prod_{s=k_1}^{k_2} \pi_{\bar{j}}(s)$ is defined. $\pi_{\bar{j}}(k_1, k_2)$ implies the probability that some sequence of Boolean functions is selected at time interval $[k_1, k_2]$.

Next, consider the following optimal control problem.

Problem 2 *Suppose that for the CS-PBN, the initial state* $x(0) = x_0$, $\rho \in [0, 1]$, *the control time* N, *and the desired state* $x_d \in \{0,1\}^n$ *are given. Then solve the following two problems.*

Problem A: *For all combinations of Boolean functions satisfying the following constraint*

$$\pi_{\bar{j}}(0, N-1) \geq \rho, \tag{5}$$

find a control input sequence $u(0)$, $u(1)$, $\ldots$, $u(N-1)$ *minimizing the lower bound of the following cost function*

$$J = \sum_{i=0}^{N-1} \{\|W_x \hat{x}(i)\|_p + \|W_u u(i)\|_p\} + \|W_f \hat{x}(N)\|_p$$

where $\hat{x}(i) := x(i) - x_d$, $W_x, W_f \in \mathcal{R}^{n \times n}$, $W_u \in \mathcal{R}^{m \times m}$, *and* $\|\cdot\|_p$ *denotes p-norm of a vector.*

Problem B: *Apply the control input sequence obtained in Problem A to the given CS-PBN. Then for all combinations of Boolean functions satisfying the constraint (5), find the upper bound* $\overline{J}^*$ *of the above cost function.*

Let $\underline{J}^*$ denote the minimum value of the lower bound obtained by solving Problem A. In the existing method [12] for control of CS-PBNs, the expected value of a given cost function is minimized. In also standard control methods of stochastic systems, the expected value is evaluated. However, for CS-PBNs, it is difficult to evaluate the expected value, because all combinations of Boolean functions must be enumerated. So the method in [12] can be applied to only small-scale systems. In this paper, instead of the expected value, the lower bound of a given cost function is minimized, and the control performance is evaluated by using the lower and the upper bounds. Then, if the constraint (5) is not imposed in Problem 2, i.e., $\rho = 0$, then behaviors of CS-PBNs are regarded as uncertain behaviors, and the best and worst performances are derived in Problem 2. However, since combinations of Boolean functions selected with low probability are included, the derived performances may not be appropriate. So in order to exclude such combinations, we impose the constraint (5). See also Section 5 for a method for deciding ρ. Similar problem formulations have been considered in optimal control of stochastic hybrid systems (see e.g. [1,4]).

Example 3 *As a simple example, consider the optimal control problem for the PBN (2), (3), (4) expressing an apoptosis network. Suppose that $q(k)$, ρ, and the initial state are given as $q(k) = 1$, $\rho = 0.05$, and $x(0) = x_0 = [\ 1\ \ 1\ \ 1\ ]^T$, respectively. For this system, find a control strategy such that a stimulus is not applied as much as possible, and cell survival is achieved.* $u = 0$ *implies that a stimulus is not applied to the system, and $x_1 = 1$, $x_2 = 0$ express cell survival [6]. Then as one of appropriate cost functions, we can consider the following cost function: $J = \sum_{i=0}^{2} u(i) + |10(x_1(3)-1)| + |10(x_2(3)-0)|$. Consider to solve Problem 2 with this cost function. Then by simple calculations, we obtain $\underline{J}^* = 1$ and $\overline{J}^* = 21$ for $u(0) = 0, u(1) = 0, u(2) = 1$ or $u(0) = 0, u(1) = 1, u(2) = 0$. From this result, we see that dynamical control is more effective than simple control such that the value of the control input is alway* 0 *or* 1. *Finally, consider the case of $\rho = 0$. In this case, we obtain $\underline{J}^* = 0$ and $\overline{J}^* = 20$ for $u(0) = u(1) = u(2) = 0$. Then the probability that $\underline{J}^* = 0$ is achieved is* 0.0403456*, and is small.*

Hereafter, for simplicity of discussion, we assume that $x_d = [\ 0\ \ 0\ \ \cdots\ \ 0\ ]^T$. In addition, although a quadratic cost function ($p = 2$) can also be considered, we consider the following form of 1-norm (linear) cost functions

$$J = \sum_{i=0}^{N-1} \{Qx(i) + Ru(i)\} + Q_f x(N). \tag{6}$$

where $Q, Q_f \in \mathcal{R}^{1\times n}$, $R \in \mathcal{R}^{1\times m}$ are vectors whose element is a non-negative real number.

4 Proposed solving method

In this section, a solving method for Problem 2 will be proposed. After the outline of a solving method for Problem A is explained by a very simple example, a general case is considered.

4.1 Simple example

Consider a single-state and single-input system. Boolean functions are given as

$$f^{(1)} = \begin{cases} f_1^{(1)} = x_1(k) \wedge u(k), & c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_1(k), & c_2^{(1)} = 0.2. \end{cases} \tag{7}$$

Then a random decision of Boolean functions can be expressed as the discrete-time Markov chain (DT-MC) of Fig. 1, where the label of each node implies the index of candidates $f_1^{(1)}, f_2^{(1)}$ of Boolean functions, and the weight $p_{ij}(k)$ from node i to j is defined as $p_{ij}(k) := \text{Prob}(f^{(1)} = f_j^{(1)} \text{ at } k \mid f^{(1)} = f_i^{(1)} \text{ at } k-1)$.

Next, we will explain a modeling method of (7) and the DT-MC of Fig. 1. From the fact [1] in [15], Boolean functions in (7) can be transformed into polynomials on the real number field. Then consider the following system

$$x_1(k+1) = \delta_{1,1}(k)\{x_1(k)u(k)\} + \delta_{1,2}(k)\{1 - x_1(k)\} \tag{8}$$

where $\delta_{1,1}(k), \delta_{1,2}(k)$ are binary variables satisfying

$$\delta_{1,1}(k) + \delta_{1,2}(k) = 1. \tag{9}$$

If $\delta_{1,1}(k) = 1$ is satisfied, then $x_1(k)u(k)$ corresponding to $f_1^{(1)}$ is selected. In a similar way, if $\delta_{1,2}(k) = 1$ is satisfied, then $1 - x_1(k)$ corresponding to $f_2^{(1)}$ is selected. This technique is frequently used in control of hybrid systems [3]. Furthermore, from (8) we obtain

$$x_1(k+1) = z_{1,1}(k) + \delta_{1,2}(k) - z_{1,2}(k) \tag{10}$$

where $z_{1,1} = \delta_{1,1}x_1 u$ and $z_{1,2} = \delta_{1,2}x_1$.

To express the DT-MC of Fig. 1, a binary variable is assigned to each arc. In the DT-MC of Fig. 1, we use four binary variables $\delta_{1,11}(k)$, $\delta_{1,12}(k)$, $\delta_{1,21}(k)$, $\delta_{1,22}(k)$. Then by defining the relation $\delta_{1,pq}(k) := \delta_{1,p}(k-1)\delta_{1,q}(k)$, we have

$$\delta_{1,1}(k) = \delta_{1,11}(k) + \delta_{1,21}(k), \tag{11}$$
$$\delta_{1,2}(k) = \delta_{1,12}(k) + \delta_{1,22}(k). \tag{12}$$

[1] For two binary variables δ_1, δ_2, the following relations hold: (i) $\neg\delta_1$ is equivalent to $1 - \delta_1$, (ii) $\delta_1 \vee \delta_2$ is equivalent to $\delta_1 + \delta_2 - \delta_1\delta_2$, and (iii) $\delta_1 \wedge \delta_2$ is equivalent to $\delta_1\delta_2$.

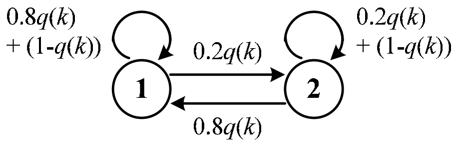

Fig. 1. Discrete-time Markov chain in CS-PBNs

By using a binary variable $\delta_{1,pq}(k)$, dynamics of the DT-MC of Fig. 1 can be expressed as the following input-output relation at each node:

$$\delta_{1,11}(k+1) + \delta_{1,12}(k+1) = \delta_{1,11}(k) + \delta_{1,21}(k), \tag{13}$$
$$\delta_{1,21}(k+1) + \delta_{1,22}(k+1) = \delta_{1,12}(k) + \delta_{1,22}(k). \tag{14}$$

We may also use the inequality-based representation [3]. However, the use of the above input-output relation is desirable in the sense that the computation time for solving the optimal control problem is decreased (see [11]).

Furthermore, by using $\delta_{1,pq}(k)$, we can calculate $\ln \pi_{\bar{j}}(k)$. Noting that $\delta_{1,11}(k) + \delta_{1,12}(k) + \delta_{1,21}(k) + \delta_{1,22}(k) = 1$ holds from (9), (11) and (12), we obtain

$$\begin{aligned}
&\ln \pi_{\bar{j}}(k) = L_1(k)\delta_1^a(k), \\
&\quad L_1(k) := \ln \left[\, p_{1,11}(k) \;\; p_{1,12}(k) \;\; p_{1,21}(k) \;\; p_{1,22}(k) \,\right], \\
&\quad \delta_1^a(k) := \left[\, \delta_{1,11}(k) \;\; \delta_{1,12}(k) \;\; \delta_{1,21}(k) \;\; \delta_{1,22}(k) \,\right]^T .
\end{aligned}$$

So by the use of natural logarithm, the constraint (5) in Problem A can be expressed as the following linear inequality: $\ln \pi_{\bar{j}}(0, N-1) = \sum_{i=0}^{N-1} L_1(i)\delta_1^a(i) \geq \ln \rho$. Thus Problem A can be equivalently rewritten as the following problem:

$$\begin{aligned}
\text{find} \quad & u(k), \delta_{1,1}(k), \delta_{1,2}(k), \delta_1^a(k), \\
& z_{1,1}(k), z_{1,2}(k), k = 0, 1, \ldots, N-1 \\
\min \quad & \text{Cost function (6)} \\
\text{subject to} \quad & \text{System (10)}, x(0) = x_0, \\
& \text{Equality constraint} \\
& \qquad \text{(9),(11),(12),(13),(14)}, \\
& \text{Inequality constraint} \\
& \qquad \ln \pi_{\bar{j}}(0, N-1) \geq \ln \rho, \\
& \qquad z_{1,1}(k) = \delta_{1,1}(k)x_1(k)u(k), \\
& \qquad z_{1,2}(k) = \delta_{1,2}(k)x_1(k).
\end{aligned}$$

Since by the result described in [5], $z_{1,1} = \delta_{1,1}x_1u$ and $z_{1,2} = \delta_{1,2}x_1$ can be expressed as linear inequalities, this problem is reduced to an integer linear programming (ILP) problem.

4.2 Solving method for Problem A

Based on the above discussion, we will derive a solving method for Problem A under a general setting. First, by using the fact in [15], $f_j^{(i)}(x(k), u(k))$ in (1) can be equivalently transformed into some polynomial. The obtained polynomial is denoted by $\hat{f}_j^{(i)}(x(k), u(k))$. Then consider the following system

$$x_i(k+1) = \sum_{j=1}^{l(i)} \left\{ \delta_{i,j}(k)\hat{f}_j^{(i)}(x(k), u(k)) \right\} \tag{15}$$

where $i = 1, 2, \ldots, n$ and $\delta_{i,j}(k) \in \{0, 1\}^1$. In (15), probabilistic behaviors are not considered, but (15) expresses the switching of the function $\hat{f}_j^{(i)}$ at each time. So we must impose the following constraint

$$\sum_{j=1}^{l(i)} \delta_{i,j}(k) = 1, \quad i = 1, 2, \ldots, n. \tag{16}$$

For simplicity of notation, by $\delta^v(k) \in \{0, 1\}^{l_v}$ denote a vector consisting of all $\delta_{i,j}(k)$, where $l_v := \sum_{i=1}^n l(i)$.

Next, a random decision of $\hat{f}_j^{(i)}$ is expressed as a DT-MC for each i. Then using $c_j^{(i)}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, l(i)$, the transition probability matrix expressing a DT-MC can be derived as

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

By $P_i^{(p,q)}(k)$ denote the (p, q)-th element in $P_i(k)$. Then we define the following row vector of size $l(i)^2$:

$$\begin{aligned}
L_i(k) := \ln \Big[\, & P_i^{(1,1)}(k) \;\; P_i^{(1,2)}(k) \;\cdots\; P_i^{(1,l(i))}(k) \\
& P_i^{(2,1)}(k) \;\; P_i^{(2,2)}(k) \;\cdots\; P_i^{(2,l(i))}(k) \;\cdots \\
& P_i^{(l(i),1)}(k) \;\; P_i^{(l(i),2)}(k) \;\cdots\; P_i^{(l(i),l(i))}(k) \,\Big].
\end{aligned}$$

Furthermore, we assign a binary variable $\delta_{i,pq}(k) := \delta_{i,p}(k-1)\delta_{i,q}(k)$ to each arc of the derived DT-MC. From the definition, the relation between $\delta_{i,pq}(k)$ and $\delta_{i,j}(k)$ must satisfy the following equality constraint:

$$\delta_{i,j}(k) = \sum_{p=1}^{l(i)} \delta_{i,pj}(k) \tag{17}$$

where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, l(i)$. In addition, using $\delta_{i,pq}(k)$, dynamics of the DT-MC are expressed as the following input-output relation at each node:

$$E_i \delta_i^a(k+1) = F_i \delta_i^a(k) \tag{18}$$

where $E_i, F_i \in \{0,1\}^{l(i) \times l(i)^2}$ and

$$\delta_i^a := [\ \delta_{i,11}\ \ \delta_{i,12}\ \ \cdots\ \ \delta_{i,1l(i)}\ \ \delta_{i,21}\ \ \delta_{i,22}\ \ \cdots\ \ \delta_{i,2l(i)}\ \ \cdots\ \ \delta_{i,l(i)1}\ \ \delta_{i,l(i)2}\ \ \cdots\ \ \delta_{i,l(i)l(i)}\ ]^T \in \{0,1\}^{l(i)^2}.$$

Using $L_i(k)$ and $\delta_i^a(k)$, ln $\pi_{\bar{j}}(k)$ is derived as ln $\pi_{\bar{j}}(k) = \sum_{i=1}^{n} L_i(k)\delta_i^a(k) = L(k)\delta^a(k)$, where $L(k) := [\ L_1(k)\ L_2(k)\ \cdots\ L_n(k)\,]$, $\delta^a(k) := [\,\delta_1^a(k)\ \delta_2^a(k)\ \cdots\ \delta_n^a(k)\,]^T \in \{0,1\}^{l_a}$, and $l_a := \sum_{i=1}^{n} l(i)^2$. Therefore, the constraint (5) in Problem A can be expressed as the following linear inequality with respect to $\delta^a(i)$:

$$\sum_{i=0}^{N-1} L(i)\delta^a(i) \geq \ln \rho. \tag{19}$$

From the above, we obtain the following lemma.

Lemma 4 *Problem A is equivalent to the following problem.*

Problem C:

find	$u(k), \delta^v(k), \delta^a(k), \quad k = 0, 1, \ldots, N-1$
min	Cost function (6)
subject to	System (15), $x(0) = x_0$,
	Equality constraint (16), (17), (18)
	Inequality constraint (19)

To express the inequality constraint (5) in Problem A as a linear form, the natural logarithm of the probability is used in Problem C. The system (15) is a polynomial nonlinear system, but by using the result described in [5], the system (15) and the equality/inequality constraints in Problem C can be equivalently transformed into the following constrained linear system:

$$\begin{cases} x(k+1) = Ax(k) + Bv(k), \\ Cx(k) + Dv(k) \leq E \end{cases} \tag{20}$$

where $v(k) = [\ u^T(k)\ \ \delta^T(k)\ \ z^T(k)\ ]^T$, and $\delta(k) := [\ (\delta^v(k))^T\ \ (\delta^a(k))^T\ ]^T \in \{0,1\}^l$, $l := l_v + l_a$. In addition, $z(k) \in \{0,1\}^p$ is an auxiliary variable, and p is determined from the number of the product of binary variables. In (20), $x(k)$ becomes a binary variable thanks to $x(0) = x_0 \in \{0,1\}^n$, $v(k) \in \{0,1\}^{m+l+p}$. So we set $x(k) \in \mathcal{R}^n$, $k \geq 1$. By using (20), we obtain the following theorem immediately.

Theorem 5 *Problem C is equivalent to the ILP problem with $(m+l+p)N$ binary variables.*

The obtained ILP problem can be solved by using a suitable solver. In the case using quadratic cost functions, the ILP problem is replaced to an integer quadratic programming problem. Finally, consider to derive a solving method for Problem B. In Problem C, assume that $u(k)$ is given as the control input obtained by solving Problem A, and replace "min" to "max". Then by solving the replaced ILP problem, $\overline{J}^*$ can be derived.

5 Numerical example

Consider a CS-PBN with 15 states and 3 control inputs, which is derived based on random graphs. See [16] for details of Boolean functions, $q(k)$, weighting vectors, and the initial state. From a given CS-PBN, we obtain the system (20) with $n = 15$, $m = 3$, $l = 90$, $p = 103$. The number of inequalities in (20) is 386. To our knowledge, CS-PBNs with such a size have not been considered so far. Furthermore, we stress that for this CS-PBN the existing method in [12] cannot be implemented using the standard environment (e.g., MATLAB), because it is necessary to compute 2^m matrices with size $2^n \times 2^n$.

Next, we consider how to decide ρ in (5). In control of stochastic systems, the expected value of a given cost function is frequently minimized. Then the cost for combinations of Boolean functions that the realized probability is high is dominant. Based on this fact, ρ is given as the mean probability that some combination of Boolean functions is selected at $[0, N-1]$. In this example, for $N = 2, 3, \ldots, 10$, we obtain $\rho = 2.9 \times 10^{-6}, 1.8 \times 10^{-8}, 2.0 \times 10^{-11}, 9.4 \times 10^{-15}, 5.7 \times 10^{-17}, 8.1 \times 10^{-19}, 2.1 \times 10^{-21}, 1.3 \times 10^{-23}, 6.2 \times 10^{-27}$, respectively.

We show the computation result. In this simulation, we used ILOG CPLEX 11.0 as an ILP solver on the computer with Windows Vista 32-bit, the Intel Core 2 Duo CPU 3.0GHz and the 4GB memory. $\underline{J}_0$, $\overline{J}_0$ denote the lower and the upper bounds of the cost function satisfying $u(k) = 0$ and the constraint (5). Then Table 1 shows $\underline{J}_0$, $\overline{J}_0$ and $\underline{J}^*$, $\overline{J}^*$. Table 2 shows the computation time. Focusing on both the difference between $\underline{J}_0$ and $\underline{J}^*$ and the difference between $\overline{J}_0$ and $\overline{J}^*$, the effectiveness of control synthesis is clear for $N = 4, 5, 6$. From this result, we see that minimizing the lower bound of the cost function is effective. On the other hand, although for $N = 7, 8, 9, 10$ the lower bound of the cost function is improved by designing the control input, the upper bound is not improved. This is because for a large N the number of combinations of Boolean functions is very large, and several cases are included. Then we will indicate that for such a case the effectiveness of minimizing the expected value of the cost function is also low. In

Table 1
Lower and upper bounds of the cost function

N	2	3	4	5	6	7	8	9	10
$\underline{J}_0$	22	22	22	27	30	32	34	35	36
$\overline{J}_0$	136	157	176	189	200	211	224	236	247
$\underline{J}^*$	21	20	22	23	23	24	25	26	27
$\overline{J}^*$	136	157	157	179	191	210	222	234	247

Table 2
Computation time [sec] to derive $\underline{J}_0$, $\overline{J}_0$ and $\underline{J}^*$, $\overline{J}^*$

N	2	3	4	5	6	7	8	9	10
$\underline{J}_0$	0.02	0.06	0.06	1.01	2.47	4.71	14.87	27.68	35.45
$\overline{J}_0$	0.06	0.08	0.81	1.91	3.07	4.13	9.08	19.78	34.97
$\underline{J}^*$	0.46	1.15	2.70	15.58	4.98	7.10	14.60	31.63	18.96
$\overline{J}^*$	0.07	0.09	1.95	7.72	16.43	12.95	21.29	254.19	98.60

addition, we remark that $\overline{J}_0 \geq \overline{J}^*$ is not in general guaranteed from Problem A and Problem B. Finally, from Table 2, we see that the optimal control problem can be solved within the practical computation time.

6 Conclusion

In this paper, a new control method of context-sensitive probabilistic Boolean networks (CS-PBNs) has been proposed. The proposed method is based on an integer programming problem, and the lower and upper bounds of the cost function are focused. By using the proposed method, for CS-PBNs such that the existing method cannot be applied, the optimal control problem can be solved by using a suitable ILP solver.

The most important future work is to apply the proposed method to several biological systems. In addition, it is important to consider how to determine the switching probability $q(k)$. In [7], the relation between the value of the cost function and $q(k) = q$ has been discussed. Inferring CS-PBNs is also one of the significant topics.

References

[1] F. Adamek, M. Sobotka, and O. Stursberg, Stochastic Optimal Control for Hybrid Systems with Uncertain Discrete Dynamics, *Proc. 4th IEEE Conf. on Automation Science and Engineering*, pp. 23–28, 2008.

[2] 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, pp. 670–679, 2007.

[3] A. Bemporad and M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica*, vol. 35, pp. 407–427, 1999.

[4] A. Bemporad and S. Di Cairano, Optimal control of discrete hybrid stochastic automata, *Hybrid Systems: Computation and Control*, *Lecture Notes in Computer Science 3414*, pp. 417–432, 2005.

[5] T. M. Cavalier, P. M. Pardalos, and A. L. Soyster, Modeling and integer programming techniques applied to propositional calculus, *Computer & Operations Research*, vol. 17, no. 6, pp. 561–570, 1990.

[6] M. Chaves, Methods for Qualitative Analysis of Genetic Networks, *Proc. European Control Conference 2009*, pp. 671–676, 2009.

[7] B. Faryabi, G. Vahedi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, Intervention in Context-Sensitive Probabilistic Boolean Networks Revisited, *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2009, article ID 360864, 2009.

[8] S. A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, *Journal of Theoretical Biology*, vol. 22, pp. 437–467, 1969.

[9] M.-S. Kim *et al.*, DNA demethylation in hormone-induced transcriptional derepression, *Nature*, vol. 461, pp. 1007–1012, 2009.

[10] H. Kitano, Cancer as a robust system: implications for anticancer therapy, *Nature Reviews Cancer*, vol. 4, pp. 227–235, 2004.

[11] K. Kobayashi and J. Imura, Minimality of Finite Automata Representation in Hybrid Systems Control, *Proc. 10th Intl. Conf. on Hybrid Systems: Computation and Control*, *LNCS* 4416, pp. 343–356, Springer, 2007.

[12] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty, Intervention in context-sensitive probabilistic Boolean networks, *Bioinformatics*, vol. 21, pp. 1211–1218, 2005.

[13] H. Santos-Rosa and C. Caldas, Chromatin modifier enzymes, the histone code and cancer, *European Journal of Cancer*, vol. 41, pp. 2381–2402, 2005.

[14] 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, pp. 261–274, 2002.

[15] H. P. Williams, *Model building in mathematical programming*, 4th ed., John Willey & Sons, Ltd, 1999.

[16] http://www.jaist.ac.jp/~k-kobaya/index.html