Title	Optimal Control of Boolean Biological Networks Modeled by Petri Nets
Author(s)	Kobayashi, Koichi; Hiraishi, Kunihiko
Citation	IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E96-A(2): 532-539
Issue Date	2013-02-01
Type	Journal Article
Text version	publisher
URL	http://hdl.handle.net/10119/11469
Rights	Copyright (C)2013 IEICE. Koichi Kobayashi and Kunihiko Hiraishi, IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, E96-A(2), 2013, 532-539. http://www.ieice.org/jpn/trans_online/
Description	

PAPER *Special Section on Mathematical Systems Science and its Applications*

Optimal Control of Boolean Biological Networks Modeled by Petri Nets

Koichi KOBAYASHI[†a] *and* **Kunihiko HIRAISHI**[†], *Members*

SUMMARY A Boolean network model is one of the models of gene regulatory networks, and is widely used in analysis and control. Although a Boolean network is a class of discrete-time nonlinear systems and expresses the synchronous behavior, it is important to consider the asynchronous behavior. In this paper, using a Petri net, a new modeling method of asynchronous Boolean networks with control inputs is proposed. Furthermore, the optimal control problem of Petri nets expressing asynchronous Boolean networks is formulated, and is reduced to an integer programming problem. The proposed approach will provide us one of the mathematical bases of control methods for gene regulatory networks.

key words: *asynchronous behavior, Boolean networks, integer programming, optimal control, Petri nets*

1. Introduction

In recent years, there have been a lot of studies on modeling, analysis, and control of biological networks such as gene regulatory networks and metabolic networks in both the control theory community and the theoretical biology community. In particular, it is important to develop control methods of gene regulatory networks as a basis of gene therapy technologies in the future. In theoretical studies on control of gene regulatory networks, the control input is given as the concentration of certain genes, and we assume that its value can be arbitrarily manipulated (see e.g., [1], [3], [7], [8], [12]–[14]). It is difficult at the current stage to implement such genes. On the other hand, in [15], feedback control of synthetic biological circuits has been implemented, and the experimental result in which cellular behavior is regulated by control has been obtained. In this experiment, the control input is given as the light pulses. This result suggests that control methods of biological networks can be realized, and the control input may be given as the status of interventions to a cell.

Biological networks are in general expressed 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, hybrid systems (piecewise affine models), and Petri nets have been developed so far (see e.g., [10]). In control problems, Boolean networks and hybrid systems are frequently used [1], [3], [7], [12], [14]. However, in the hybrid systems-based ap-

proach, a class of biological networks are limited to low-dimensional systems, because the computation time to solve the control problem is too long. In Boolean networks, dynamics such as interactions between genes are expressed by Boolean functions [11]. There is a criticism that a Boolean network is too simple as a model of biological networks, but this model can be relatively applied to large-scale systems. Although a Boolean network is a class of discrete-time nonlinear systems and expresses the synchronous behavior, it is important to consider the asynchronous behavior for adequately representing a variety of time scales. Some methods for expressing asynchronous Boolean networks (ABNs) have been already proposed (see e.g., [8], [19]). In [19], ABNs are modeled by non-deterministic dynamical systems, and the asynchronous behavior is expressed as the probabilistic behavior. However, only n combinations for n genes are considered (see also Example 1). In [8], it is assumed that a combination at each time is periodically given in advance, but it is difficult to consider many combinations. Although it is desirable to consider all (2^n) combinations, in the existing methods it is difficult to consider all combinations.

On the other hand, a Petri net is well known as a model expressing the asynchronous behavior [16], [24]. A Petri net is a class of directed bipartite graphs, in which the nodes represent transitions and places. The methods to express asynchronous Boolean networks as Petri nets have been proposed in [5], [18]. However, in these methods, only constant control inputs are considered, and these methods cannot be directly applied to the control problem with dynamical control inputs.

Thus in this paper, the optimal control problem of asynchronous Boolean networks modeled by Petri nets is discussed. First, based on the method proposed in [5] and the notation of external input places [9], [20], we propose a new method to transform asynchronous Boolean networks with control inputs into Petri nets with external input places. Next, the optimal control problem is formulated, and is reduced to an integer linear programming (ILP) problem. In addition, the biological significance is also discussed by using a simple example. Finally, the effectiveness of the proposed approach is shown by a numerical example on a WNT5A network [22]. The proposed approach provides us a new control method of gene regulatory networks.

This paper is organized as follows. In Sect. 2, synchronous Boolean networks and asynchronous Boolean networks are introduced. In Sect. 3, Petri nets expressing asyn-

chronous Boolean networks are derived. In Sect. 4, the optimal control problem is addressed. In Sect. 5, the effectiveness is shown by a numerical example. In Sect. 6, we conclude this paper.

Notation: Let $\mathcal{R}$ denote the set of real numbers. Let $\{0, 1\}^{m \times n}$ denote the set of $m \times n$ matrices, which consists of elements 0 and 1. For the finite set M, let $|M|$ denote the number of elements. For a matrix X, let X^T denote the transpose of X.

2. Boolean Biological Networks

First, consider synchronous Boolean networks (SBNs). A general form of SBNs is given by

$$x(k + 1) = f_a(x(k)) \tag{1}$$

where $x \in \{0, 1\}^n$ is the state (e.g., the concentration of genes), $k = 0, 1, 2, \ldots$ is the discrete time. $f_a : \{0, 1\}^n \to \{0, 1\}^n$ is a given Boolean function with logical operators such as AND ($\wedge$), OR ($\vee$), and NOT ($\neg$). Since the SBN (1) is deterministic, $x(k + 1)$ is uniquely determined for a given $x(k)$.

To consider the control problems, we add the control input to the SBN (1) as follows:

$$x(k + 1) = f(x(k), u(k)) \tag{2}$$

where $u \in \{0, 1\}^m$ is the control input, i.e., the value of u (e.g., the concentration of genes) can be arbitrarily given, and $f : \{0, 1\}^n \times \{0, 1\}^m \to \{0, 1\}^n$ is a given Boolean function. The i-th element of the state x, the i-th element of the control input u and the i-th element of the Boolean function f are denoted by x_i, u_i and f_i, respectively. Also in the SBN (2), $x(k + 1)$ is uniquely determined for given $x(k)$ and $u(k)$.

Next, consider asynchronous Boolean networks (ABNs). Suppose that a Boolean function assigned to each gene (i.e., x_i) is given by $f_i(x(k), u(k))$, $f_i : \{0, 1\}^n \times \{0, 1\}^m \to \{0, 1\}$. For each state x_i, either

$$x_i(k + 1) = f_i(x(k), u(k))$$

or

$$x_i(k + 1) = x_i(k)$$

is selected at each time. Therefore, the number of combinations of Boolean functions is given by 2^n. In [19], asynchronous Boolean networks are modeled by non-deterministic dynamical systems, and the asynchronous behavior is expressed as the probabilistic behavior. In [8], it is assumed that a combination at each time is periodically given in advance.

Using the result in [19], we show an example of SBNs and ABNs.

Example 1: In [19], the behavior of ABNs is given by the union of the behaviors of the following n SBNs:

$$\Sigma_i : \begin{cases} x_i(k + 1) = f_i(x(k), u(k)), \\ x_j(k + 1) = x_j(k), \\ \quad \forall j \in \{1, 2, \ldots, n\} \setminus \{i\} \end{cases} \tag{3}$$

where $i = 1, 2, \ldots, n$. As a simple example, consider the following SBN of an apoptosis network:

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

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 (TNF, a stimulus) is denoted by u, and is regarded as the control input. This model is described in [6], and is a simplified version of an apoptosis network model in [19]. In this model, $x_2(k) = 0$ implies cell survival, and $x_1(k) = 0, x_2(k) = 1$ imply cell death [6]. Then, by using this model, we can find an initial state and a control input sequence such that the state reaches cell death (or cell survival).

In the case of synchronous Boolean dynamics, state transitions can be computed by directly using (4). For example, for $x(0) = [\ 1\ \ 1\ \ 1\ ]^T$ and $u(k) = 0$, we obtain $x(1) = [\ 0\ \ 0\ \ 1\ ]^T$. By computing the transition from each state, we obtain the state transition diagram in Fig. 1 (left). In Fig. 1, the number assigned to each node denotes x_1, x_2, x_3 (elements of the state).

In the case of asynchronous Boolean dynamics, according to (3), we consider the following three SBNs

$$\Sigma_1 : \begin{cases} x_1(k + 1) = \neg x_2(k) \wedge u(k), \\ x_2(k + 1) = x_2(k), \\ x_3(k + 1) = x_3(k), \end{cases} \tag{5}$$

$$\Sigma_2 : \begin{cases} x_1(k + 1) = x_1(k), \\ x_2(k + 1) = \neg x_1(k) \wedge x_3(k), \\ x_3(k + 1) = x_3(k), \end{cases} \tag{6}$$

$$\Sigma_3 : \begin{cases} x_1(k + 1) = x_1(k), \\ x_2(k + 1) = x_2(k), \\ x_3(k + 1) = x_2(k) \vee u(k). \end{cases} \tag{7}$$

State transitions can be computed by using (5), (6), (7). For example, for $x(0) = [\ 1\ \ 1\ \ 1\ ]^T$ and $u(k) = 0$, we obtain $x(1) = \{[\ 0\ \ 1\ \ 1\ ]^T, [\ 1\ \ 0\ \ 1\ ]^T, [\ 1\ \ 1\ \ 1\ ]^T\}$. In a similar way, by computing the transition from each state, we obtain the state transition diagram in Fig. 1 (right).
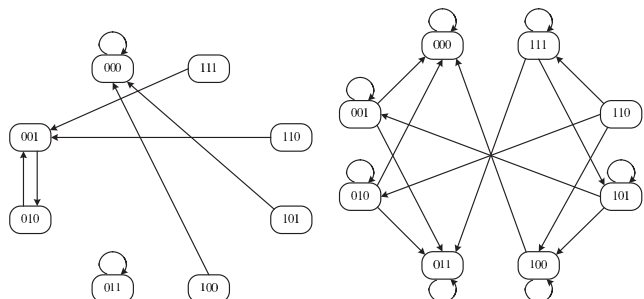

Fig. 1 (Left) State transition diagram of (4) and $u(k) = 0$, (Right) State transition diagram of (5), (6), (7) and $u(k) = 0$.

Comparing the left figure with the right figure in Fig. 1, we see that a part of behaviors is clearly different.　□

Although it is desirable to consider all (2^n) combinations, this is difficult in the method of [19]. Also in the method of [8], it may be difficult to set periodic patterns for a large-scale network. To overcome these technical issues, ABNs are modeled by Petri nets, not multiple SBNs. By using the Petri net-based model, we can consider several situations. One example is shown as follows. Activation/inactivation of a part of a given biological network may be able to be controlled by external stimuli. In the case of inactivation, dynamics are given as $x(k + 1) = 0$. If there exist some patterns in which different parts of a network can be activated/inactivated, then this implies that a part of the asynchronous behavior can be controlled by selecting among 2^n combinations of Boolean functions. In the proposed approach, we can treat such a situation by using the Petri net-based model.

3. Petri Net-Based Modeling of Boolean Biological Networks

First, the outline of Petri nets is explained. A Petri net is one of the model of concurrent and distributed systems, and is a directed bipartite graph consisting of four components: places, transitions, arcs, and tokens. These components are denoted by circles, rectangles, arrows, and dots, respectively. Figure 2 shows a simple example of Petri nets. Places, transitions, arcs express the structure of a given Petri net. Each transition has input places and output places, and direct connections between two places or two transitions are not allowed. The state of Petri nets is given by the distribution of tokens on places, and is changed by firing of transitions. If input places contain at least one token, then the transition can fire by consuming one token of input places, one token is added in output places. In the example of Fig. 2, the transitions t_1, t_2, and t_3 may fire. If the transition t_1 fires, then one token in p_2 is moved to p_1. See e.g., [16], [24] for further information on Petri nets.

Now, let us consider expressing an ABN as a Petri net by using each Boolean function f_i in the SBN (2). A Petri net expressing an ABN has been proposed in [5], [18], but only constant control inputs can be considered, and dynamical control in which the value of the control input is switched
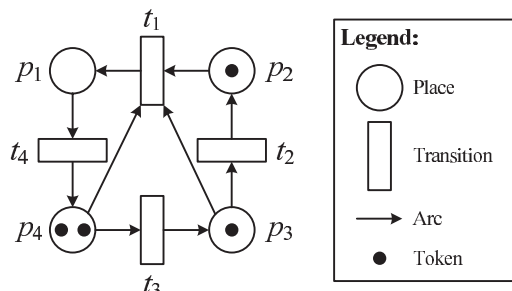
cannot be realized. In this paper, as an extension of the method in [5], we propose a modeling method of a Petri net expressing an ABN with dynamical control inputs.

We introduce the following notation. By $\mathcal{I}(f_j)$, $j = 1, 2, \ldots, n$, denote a finite set consisting of the state variable and the control input variable included in the Boolean function f_j. In the example of (4), we obtain $\mathcal{I}(f_1) = \{x_2, u\}$, $\mathcal{I}(f_2) = \{x_1, x_3\}$, and $\mathcal{I}(f_3) = \{x_2, u\}$. Next, we define a logical parameter $K_j(X) \in \{0, 1\}$, $X \subseteq \mathcal{I}(f_j)$, $j = 1, 2, \ldots, n$. $K_j(X)$ is given as the value of $f_j(x, u)$ under the condition that the value of each variable included in X is '1' and the value of each variable included in $\mathcal{I}(f_j) \setminus X$ is '0'. In the example of (4), we can obtain

$$\begin{cases} K_1(\emptyset) = 0, & K_1(\{x_2\}) = 0, \\ K_1(\{u\}) = 1, & K_1(\{x_2, u\}) = 0, \end{cases}$$

$$\begin{cases} K_2(\emptyset) = 0, & K_2(\{x_1\}) = 0, \\ K_2(\{x_3\}) = 1, & K_2(\{x_1, x_3\}) = 0, \end{cases}$$

$$\begin{cases} K_3(\emptyset) = 0, & K_3(\{x_2\}) = 1, \\ K_3(\{u\}) = 1, & K_3(\{x_2, u\}) = 1. \end{cases}$$

Next, consider deriving a Petri net expressing Boolean networks. In the derived Petri net, the number of places is given as $2(n + m)$, that is, for each x_i in (2), two places x_i and $\overline{x}_i$ are prepared. In a similar way, for each u_i in (2), two places u_i and $\overline{u}_i$ are prepared. $\overline{x}_i$ and $\overline{u}_i$ are called complementary places [5]. The number of transitions is given as $\sum_{i=1}^{n} 2^{|\mathcal{I}(f_i)|}$. In the example of (4), the number of transitions is given as $2^2 + 2^2 + 2^2 = 12$. From the property of Boolean networks, the following assumptions are made.

Assumption 1: The maximum number of tokens in each place is equal to 1.

Assumption 2: A sum of the number of tokens in x_i (u_i) and that in $\overline{x}_i$ ($\overline{u}_i$) is equal to 1.

In addition, suppose that u_i and $\overline{u}_i$ are given as an external input place [9], [20]. In u_i and $\overline{u}_i$, a token is arbitrary generated, but the above two assumptions must be satisfied.

Finally, we remark that the case of a self-regulator (i.e., $x_i \in \mathcal{I}(f_i)$) is slightly different to the case of $x_i \notin \mathcal{I}(f_i)$. If $x_i \in X$ holds, then $K_i(X) = 1$ does not lead any change on x_i. If $x_i \notin X$ holds, then $K_i(X) = 0$ does not lead any change on x_i.

Under the above preparations, we define a Petri net expressing an ABN. In [5], the Petri net expressing an ABN without the control input is defined. The following definition gives the Petri net expressing an ABN with the control input, and is an extension of the definition in [5].

Definition 1: For a given SBN (2), the Petri net N_c expressing an ABN is defined as follows:

$$N_c = (P \cup P_c, T, Pre, Post) \tag{8}$$

where

- $P = \{x_1, \overline{x}_1, x_2, \overline{x}_2, \ldots, x_n, \overline{x}_n\}$ is the set of places,

- $P_c = \{u_1, \overline{u}_1, u_2, \overline{u}_2, \ldots, u_m, \overline{u}_m\}$ is the set of external

Fig. 2　A simple example of Petri nets.

input places,

- $T = \{t_{x_i,X}, i = 1, 2, \ldots, n, X \subseteq \mathcal{I}(f_i)\}$ is the set of transitions,

- $Pre : (P \cup P_c) \times T \rightarrow \{0, 1\}$ is the mapping defining arcs between places and transitions,

- $Post : T \times (P \cup P_c) \rightarrow \{0, 1\}$ is the mapping defining arcs between transitions and places.

The functions Pre and $Post$ are defined as follows:
(i) Case of $x_i \notin \mathcal{I}(f_i)$ (x_i is not a self-regulator): For a given transition $t_{x_i,X}$, the following terms are defined (all the other terms are equal to zero):

$$Pre(x_i, t_{x_i,X}) = Post(t_{x_i,X}, \overline{x}_i) = 1 - K_i(X),$$
$$Pre(\overline{x}_i, t_{x_i,X}) = Post(t_{x_i,X}, x_i) = K_i(X),$$
$$Pre(x_j, t_{x_i,X}) = Post(t_{x_i,X}, x_j) = 1, \quad \forall x_j \in X,$$
$$Pre(\overline{x}_j, t_{x_i,X}) = Post(t_{x_i,X}, \overline{x}_j) = 1,$$
$$\forall x_j \in \mathcal{I}(f_i) - X,$$
$$Pre(u_j, t_{x_i,X}) = 1, \quad \forall u_j \in X,$$
$$Pre(\overline{u}_j, t_{x_i,X}) = 1, \quad \forall u_j \in \mathcal{I}(f_i) - X.$$

(ii) Case of $x_i \in \mathcal{I}(f_i)$ (x_i is a self-regulator): Consider a given transition $t_{x_i,X}$.
If $x_i \in X$, then only the case of $K_i(X) = 0$ is considered. Therefore, the following terms are defined:

$$Pre(x_i, t_{x_i,X}) = Post(t_{x_i,X}, \overline{x}_i) = 1,$$
$$Pre(x_j, t_{x_i,X}) = Post(t_{x_i,X}, x_j) = 1,$$
$$\forall x_j \in X, \quad x_j \neq x_i,$$
$$Pre(\overline{x}_j, t_{x_i,X}) = Post(t_{x_i,X}, \overline{x}_j) = 1,$$
$$\forall x_j \in \mathcal{I}(f_i) - X,$$
$$Pre(u_j, t_{x_i,X}) = 1, \quad \forall u_j \in X,$$
$$Pre(\overline{u}_j, t_{x_i,X}) = 1, \quad \forall u_j \in \mathcal{I}(f_i) - X.$$

If $x_i \notin X$, then only the case of $K_i(X) = 1$ is considered. Therefore, the following terms are defined:

$$Pre(\overline{x}_i, t_{x_i,X}) = Post(t_{x_i,X}, x_i) = 1,$$
$$Pre(x_j, t_{x_i,X}) = Post(t_{x_i,X}, x_j) = 1, \quad \forall x_j \in X,$$
$$Pre(\overline{x}_j, t_{x_i,X}) = Post(t_{x_i,X}, \overline{x}_j) = 1,$$
$$\forall x_j \in \mathcal{I}(f_i) - X, \quad x_j \neq x_i,$$
$$Pre(u_j, t_{x_i,X}) = 1, \quad \forall u_j \in X,$$
$$Pre(\overline{u}_j, t_{x_i,X}) = 1, \quad \forall u_j \in \mathcal{I}(f_i) - X.$$

□

In the above definition, a sum of the number of tokens in u_i and that in $\overline{u}_i$ becomes zero by firing some transition. In this case, to satisfy Assumption 1 and Assumption 2, a token is generated in either u_i or $\overline{u}_i$.
We show a simple example.

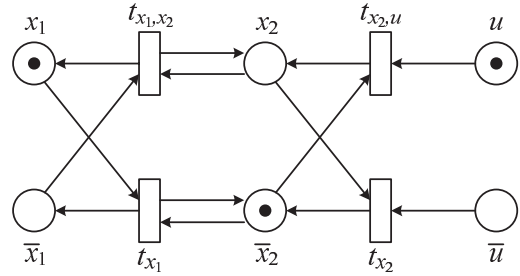**Example 2:** Consider the following simple SBN:

Fig. 3 Petri net expressing an ABN.

$$\begin{cases} x_1(k + 1) = x_2(k), \\ x_2(k + 1) = u(k) \end{cases} \tag{9}$$

From $x_1(k + 1) = x_2(k)$, we obtain $K_1(\emptyset) = 0$ and $K_1(\{x_2\}) = 1$. In a similar way, from $x_2(k+1) = u(k)$, we obtain $K_2(\emptyset) = 0$ and $K_2(\{u\}) = 1$. Then we consider four transitions $t_{x_1,\emptyset}$, $t_{x_1,\{x_2\}}$, $t_{x_2,\emptyset}$, and $t_{x_2,\{u\}}$. We denote these transitions by t_{x_1}, t_{x_1,x_2}, t_{x_2}, and $t_{x_2,u}$, respectively. Then we obtain the Petri net in Fig. 3, where the placement of tokens represents $x_1(k) = 1$, $x_2(k) = 0$, and $u(k) = 1$. In this case, transitions t_{x_1} and $t_{x_2,u}$ can fired. For example, one token on the place x_1 is moved to the place $\overline{x}_1$ by firing t_{x_1}, and one token on the place $\overline{x}_2$ stays.

In addition, suppose that one token is included in place $\overline{x}_1$, $\overline{x}_2$, and u. Then the transition $t_{x_2,u}$ may fire. If the transition $t_{x_2,u}$ fires, then one token is moved from $\overline{x}_2$ and u to x_2. A pair of x_2 and $\overline{x}_2$ satisfies Assumption 1 and Assumption 2, but a pair of u and $\overline{u}$ does not satisfy Assumption 2. So one token must be added in either u and $\overline{u}$ with fire.

Finally, a matrix representation of Pre and $Post$ in (8) is obtained as follows.

$$Pre = \begin{array}{c} \\ \\ \end{array} \begin{array}{cccc} t_{x_1} & t_{x_1,x_2} & t_{x_2} & t_{x_2,u} \\ \end{array} \\ \left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{array} \right] \begin{array}{c} x_1 \\ \overline{x}_1 \\ x_2 \\ \overline{x}_2 \\ u \\ \overline{u} \end{array}$$

$$Post = \begin{array}{cccccc} x_1 & \overline{x}_1 & x_2 & \overline{x}_2 & u & \overline{u} \\ \end{array} \\ \left[\begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{array} \right] \begin{array}{c} t_{x_1} \\ t_{x_1,x_2} \\ t_{x_2} \\ t_{x_2,u} \end{array}$$

□

4. Optimal Control

For the Petri net (8) expressing an ABN, consider the optimal control problem. First, the state equation of the Petri net (8) is derived. Next, the optimal control problem is formulated.

4.1 State Equation of Petri Nets

First, we introduce the notation. By $x_i(k)$, $\overline{x}_i(k)$, $u_i(k)$, $\overline{u}_i(k) \in$

$\{0, 1\}$, denote existence or non-existence of a token in place $x_i, \bar{x}_i, u_i, \bar{u}_i \in \{0, 1\}$ at time k. Since $x_i(k), \bar{x}_i(k), u_i(k), \bar{u}_i(k)$ are binary variables, Assumption 1 is satisfied. To satisfy Assumption 2, $u_i(k) + \bar{u}_i(k) = 1$ is imposed. Also, $x_i(k) + \bar{x}_i(k) = 1$ must be imposed, but this condition is satisfied under the initial condition $x_i(0) + \bar{x}_i(0) = 1$. Next, by $t_{x_i, X_j}(k) \in \{0, 1\}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, 2^{|\mathcal{I}(f_i)|}$, denote the fire in the transition t_{x_i, X_j}. If $t_{x_i, X_j}(k) = 1$, then the transition t_{x_i, X_j} fires at time k. Otherwise, t_{x_i, X_j} does not fire. Finally, we define

$$x(k) = [\ x_1(k)\ \ \bar{x}_1(k)\ \ \cdots\ \ x_n(k)\ \ \bar{x}_n(k)\ ]^T,$$
$$u(k) = [\ u_1(k)\ \ \bar{u}_1(k)\ \ \cdots\ \ u_m(k)\ \ \bar{u}_m(k)\ ]^T,$$
$$t(k) = \Big[\ t_{x_1, X_1}(k)\ \ \cdots\ \ t_{x_1, X_{2^{|\mathcal{I}(f_1)|}}}(k)$$
$$\cdots\ \ t_{x_n, X_1}(k)\ \ \cdots\ \ t_{x_n, X_{2^{|\mathcal{I}(f_n)|}}}(k)\ \Big]^T.$$

According to the conventional result in Petri nets (see e.g., [16], [24]), we can obtain the following state equation of Petri nets (8)

$$\left[\begin{array}{c} x(k+1) \\ u(k+1) \end{array}\right] = \left[\begin{array}{c} x(k) \\ u(k) \end{array}\right] + B_a t(k),$$

$$Pre\ t(k) \le \left[\begin{array}{c} x(k) \\ u(k) \end{array}\right]$$

where $B_a := Post^T - Pre$. If the number of firing transitions at each time is equal to or less than 1, then the inequality condition $[\ 1\ \cdots\ 1\ ]t(k) \le 1$ must be imposed. Furthermore, from the obtained state equation, we can obtain the following state equation

$$x(k+1) = Ax(k) + Bv(k), \tag{10}$$
$$Cx(k) + Dv(k) \le E \tag{11}$$

where $v(k) := [\ u^T(k)\ t^T(k)\ ]^T$, and

$$A = I_{2n},$$
$$B = \left[\ 0_{2n \times 2m}\ \ [I_{2n}\ 0_{2n \times 2m}]B_a\ \right],$$
$$C = \left[\begin{array}{c} -I_{2n} \\ 0_{2m \times 2n} \end{array}\right],$$
$$D = \left[\ \left[\begin{array}{c} 0_{2n \times 2m} \\ -I_{2m} \end{array}\right]\ \ Pre\ \right],$$
$$E = 0_{2(n+m) \times 1}.$$

4.2 Optimal Control Problem

Next, for the state Eqs. (10), (11) expressing the Petri net (8), consider the following optimal control problem.

Problem 1: For the state Eqs. (10), (11) expressing the Petri net (8), suppose that the initial state $x(0) = x_0$ satisfying Assumption 1 is given. Then find an input sequence $v(0), v(1), \ldots, v(N-1)$ minimizing the linear cost function

$$J = \sum_{k=0}^{N-1} \{Qx(k) + Rv(k)\} + Q_f x(N) \tag{12}$$

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

Suppose that there exist genes such that expression must be inhibited. For the states corresponding to these genes, high weights are set. That is, certain elements of Q are given as a high value. In addition, if it is desirable that the value of the control input is zero, then certain elements of R are given as a high value. By minimizing J, which is given according to the above policy, inhibition of gene expression and derivation of the desirable control input sequence can be achieved.

For simplicity of discussion, a linear function with respect to x and u is considered as a cost function, but a quadratic cost function may be used. In addition, suppose that the desired state $x_d \in \{0, 1\}^{n_d}$ is given. Then the state $x(k)$ must be replaced to $\hat{x}(k) := x(k) - x_d$, and the cost function (12) is also replaced to $J = \sum_{i=k}^{N-1} \{Q|\hat{x}(k)| + Rv(k)|\} + Q_f|\hat{x}(N)|$. Although a longer N is desirable, the computation time to solve Problem 1 must be also considered.

In Problem 1, $v(0), v(1), \ldots, v(N-1)$ are free binary variables (of course, the inequality constraint (11 must be satisfied). However, there is a possibility that a given biological system does not satisfy this assumption. Then suppose that some candidates of input sequences are given. In Problem 1, the optimal input sequence minimizing the cost function (12) is selected among the set of the candidates $\mathcal{B} \subseteq \{0, 1\}^{m_d N}$. This extension is easy. In this sense, Problem 1 can be applied to optimal control of asynchronous Boolean networks such that the updating time of each state is given in advance. Of course, this problem can also be applied to optimal control of SBNs. Thus Problem 1 includes several situations.

We show an example for setting weighting vectors from the biological viewpoint.

Example 3: Consider the Boolean network expressing an apoptosis network (4) in Example 1 again. From (4), we obtain the Petri net (8) with 6 places, 2 external input places, and 12 transitions. In addition, from the obtained Petri net, we obtain the state Eqs. (10), (11). For the obtained state equation, we consider to find a control strategy such that a stimulus is not applied as much as possible, and cell survival is achieved. $u(k) = 0$ implies that a stimulus is not applied to the system, and $x_1(k) = 1$, $x_2(k) = 0$ express cell survival. Then as one of appropriate cost functions, we can consider the following cost function

$$J = \sum_{k=0}^{N-1} \{10|x_1(k) - 1| + 10|x_2(k) - 0| + u(k)\}$$
$$+ 100|x_1(N) - 1| + 100|x_2(N) - 0|.$$

By the appropriate coordinate transformation, this cost function can be rewritten as the form of (12). See also Sect. 5.2 and [7], [8], [13] for biological examples on the optimal control problems. □

Finally, by using the state Eqs. (10), (11), Problem 1

can be written as the following integer linear programming (ILP) problem: find $v(k)$, $k = 1, 2, \ldots, N - 1$ minimizing the cost function (12) subject to the system (10), (11). The ILP problem can be solved by using a suitable solver such as IBM ILOG CPLEX [23].

4.3 Discussion on the Effectiveness of the Petri Net-Based Modeling

One of the simple methods for modeling of ABNs is to select one combination at each time among all (2^n) combinations of Boolean functions. In the case of (9) in Example 2, we obtain the following $2^n = 4$ combinations, i.e.,

$$\Sigma_1 : \quad x_1(k+1) = x_1(k), \quad x_2(k+1) = x_2(k),$$
$$\Sigma_2 : \quad x_1(k+1) = x_2(k), \quad x_2(k+1) = x_2(k),$$
$$\Sigma_3 : \quad x_1(k+1) = x_1(k), \quad x_2(k+1) = u(k),$$
$$\Sigma_4 : \quad x_1(k+1) = x_2(k), \quad x_2(k+1) = u(k).$$

From these subsystems, we can obtain the following system expressing an ABN:

$$x_1(k+1) = (\delta_1(k) + \delta_3(k))x_1(k)$$
$$+ (1 - \delta_1(k) - \delta_3(k))x_2(k), \quad (13)$$
$$x_2(k+1) = (\delta_1(k) + \delta_2(k))x_2(k)$$
$$+ (1 - \delta_1(k) - \delta_2(k))u(k) \quad (14)$$

where $\delta_1(k), \delta_2(k), \delta_3(k)$ are binary variables satisfying $\delta_1(k) + \delta_2(k) + \delta_3(k) \leq 1$, and correspond to $\Sigma_1, \Sigma_2, \Sigma_3$, respectively. Furthermore, $z_1 := \delta_1 x_1$, $z_2 := \delta_3 x_1$, $z_3 := \delta_1 x_2$, $z_4 := \delta_3 x_2$, $z_5 := \delta_2 x_2$, $z_6 := \delta_1 u$, and $z_7 := \delta_2 u$ are defined, and the result[†] proposed in [4] is applied to $z_1, z_2, \ldots, z_7$. Thus we can obtain the linear system, where the number of input decision variables is $1 + 3 + 7 = 11$. This method is called here a direct approach. In the case using the state Eqs. (10), (11), $\dim v = 6$ is derived.

In general, in the direct approach, the dimension of binary variables to switch Boolean functions and the control input is given as $m + 2^n - 1$. In the proposed Petri net-based approach, the dimension v in (10), (11) is given as $2m + \sum_{i=1}^n 2^{|\mathcal{I}(f_i)|}$. In real gene regulatory networks, it is well known that $|\mathcal{I}(f_i)|$ is relatively smaller than n (see e.g., [2]). For example, consider the case of $n = 10$, $m = 3$, and $|\mathcal{I}(f_i)| = 3$. Then we can obtain $2^n - 1 + m = 1026$ and $2m + \sum_{i=1}^n 2^{|\mathcal{I}(f_i)|} = 86$ are obtained.

Thus, in modeling of real gene regulatory networks, it is not appropriate to use the direct approach, and the proposed Petri net-based method provides us a simpler model of ABNs.

5. Numerical Example

In this section, we show the effectiveness of the proposed approach by using a numerical example with a WNT5A network [22]. First, a Boolean network model of a WNT5A network is explained. Next, we show the computation result.

5.1 WNT5A Network

Consider a gene regulatory network with the gene WNT5A. The gene WNT5A is related to melanoma, which is a kind of skin cancers. It is known that overexpression of the gene WNT5A is closely related to tumor growth. Thus it important to consider inhibiting the concentration level of the gene WNT5A (see also [21]).

The Boolean network $x(k + 1) = f_a(x(k))$ is given by

$$x_1(k+1) = \neg x_5(k),$$
$$x_2(k+1) = \neg x_6(k),$$
$$x_3(k+1) = x_3(k),$$
$$x_4(k+1) = \neg x_6(k) \lor u(k),$$
$$x_5(k+1) = x_2(k) \lor x_3(k),$$
$$x_6(k+1) = x_6(k) \lor \neg u(k)$$

where the concentration level (high or low) of the gene WNT5A is denoted by x_1, the concentration level of the gene S100P by x_2, the concentration level of the gene RET1 by x_3, the concentration level of the gene MART1 by x_4, the concentration level of the gene HADHB by x_5, and the concentration level of the gene STC2 by x_6. See [22] for further details. According to discussion on [7], we suppose that in the BN model of a WNT5A network, the control input u is given by the concentration level of the gene pirin.

5.2 Computation Result

First, consider deriving a Petri net expressing the Boolean network model of a WNT5A network. Then we can derive the Petri net with 14 places and 15 transitions. See Appendix for details of matrices $Pre, Post$.

For the obtained state equation, consider solving Problem 1. As a purpose of control, we suppose that the following properties are desirable: (i) the concentration level of the gene WNT5A, x_1 is inactive, (ii) the concentration levels of STC and pirin, i.e., x_6 and u are active. The property (i) is introduced according to [21]. The property (ii) is artificially given for verifying the effectiveness of control. To realize these properties, Q, Q_f, R in the cost function (12) are given as

$$Q = Q_f = [\ \underbrace{10\ 0}_{Q_1}\ 0\ \cdots\ 0\ \underbrace{0\ 10}_{Q_6}\ ],$$
$$R = [\ \underbrace{0\ 1}_{R_1}\ 0\ \cdots\ 0\ ].$$

$Q_1 = [\ 10\ 0\ ]$ corresponds to x_1 and $\bar{x}_1$. By setting the weight for x_1 bigger than that for $\bar{x}_1$, the property (i), i.e., $x_1 = 0$ and $\bar{x}_1 = 1$ will be achieved. Q_6 and R_1 are also set in a similar way. In addition, N in the cost function (12) are

---

[†]For binary variables $\delta_1, \delta_2, \ldots, \delta_n \in \{0, 1\}$, the product $z = \delta_1 \delta_2 \cdots \delta_n$ is equivalent to a pair of $\sum_{i=1}^n \delta_i - z \leq n-1$ and $-\sum_{i=1}^n \delta_i + nz \leq 0$.

Table 1 Time sequences of the focusing states x_1, x_6, and the control input u.

Time	0	1	2	3	4	5	6	7	8	9	10
x_1	1	1	0	0	0	0	0	0	0	0	0
x_6	0	0	0	0	0	0	1	1	1	1	1
u	1	1	1	1	1	0	1	1	1	1	–

given as $N = 10$, and the initial state is given as

$$x_0 = [\,1\ 0\ 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1\,]^T.$$

Next, consider imposing constraints on fires of transitions. The set of transitions are separated as follows:

$$T = T_a \cup T_b \cup T_c,$$
$$T_a = \{t_{x_4,\{u\}},\, t_{x_4,\{x_6,u\}},\, t_{x_4,\{x_6\}},\, t_{x_5,\{x_2\}},\, t_{x_5,\{x_3\}}\},$$
$$T_b = \{t_{x_1,\emptyset},\, t_{x_1,\{x_5\}},\, t_{x_2,\emptyset},\, t_{x_2,\{x_6\}},\, t_{x_4,\emptyset}\},$$
$$T_c = \{t_{x_5,\{x_2,x_3\}},\, t_{x_5,\emptyset},\, t_{x_6,\emptyset},\, t_{x_6,\{x_6,u\}},\, t_{x_6,\{u\}}\}.$$

In a similar way to the method in [8], one transition included in T_a is allowed to fire at each time, one transition included in T_b is allowed to fire at only time $1, 3, 5, 7, 9$, and one transition included in T_c is allowed to fire at only time $5, 9$. Of course, a firing transition at each time may be uniquely given. Thus we can obtain the ILP problem with $17N = 170$ binary variables. Note that in the case using the direct approach explained in Sect. 4.3, the optimal control problem is rewritten as the ILP problem with $202N = 2020$. From this, we see that the size of the ILP problem becomes smaller by the proposed approach.

Next, we show the computation result. Table 1 shows the time sequences of the state x_1, x_6, and the control input u. Here, the transition $t_{x_4,\{x_6\}}$ fired at time 0, the transition $t_{x_1,\{x_5\}}$ fired at time 1, and the transition $t_{x_6,\emptyset}$ fired at time 5. From this result, we see that the above properties (i) and (ii) are achieved by appropriately selecting a firing sequence and a control input sequence. In addition, this result suggests that there is a possibility that the concentration level of WNT5A can be controlled.

Finally, the computation time for solving Problem 1 was 15 [msec], where we used IBM ILOG CPLEX 11.0 [23] as an ILP solver on the computer with the Intel Core 2 Duo 3.0 GHz processor and the 4 GB memory. In the case of $N = 20$, the computation time was 20 [msec].

6. Conclusion

In this paper, we have discussed optimal control of asynchronous Boolean networks with control inputs. First, we have proposed a method to transform a Boolean network with control inputs into a Petri net with external input places. Next, the optimal control problem has been formulated. This problem is a general formulation including several biological situations. Finally, the effectiveness of the proposed approach is shown by a numerical example. The proposed approach will become one of the mathematical bases toward control of biological networks in the future.

One of the future works is to apply the proposed approach to several biological systems. From the practical viewpoint, an extension to probabilistic Boolean networks proposed in [17] is also important. In addition, for large-scale Boolean networks, the computation time to solve the problem will be long. So it is significant to consider to reduce the computation time to solve the problem. Then logic minimization is also important [18].

References

[1] T. Akutsu, M. Hayashida, W.-K. Ching, and M.K. Ng, "Control of Boolean networks: Hardness results and algorithms for tree structured networks," J. Theoretical Biology, vol.244, pp.670–679, 2007.

[2] M. Andrecut, S.A. Kauffman, and A.M. Madni, "Evidence of scale-free topology in gene regulatory network of human tissues," Int'l Journal of Modern Physics C, vol.19, no.2, pp.283–290, 2008.

[3] S. Azuma, E. Yanagisawa, and J. Imura, "Controllability analysis of biosystems based on piecewise affine systems approach," IEEE Trans. Autom. Control, vol.53, no.1, pp.139–152, 2008.

[4] 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.

[5] C. Chaouiya, E. Remy, P. Ruet, and D. Thieffry, "Qualitative modelling of genetic networks: From logical regulatory graphs to standard Petri nets," Proc. 25th Int'l Conf. on Application and Theory of Petri Nets, LNCS 3099, pp.137–156, 2004.

[6] M. Chaves, "Methods for qualitative analysis of genetic networks," Proc. European Control Conference 2009, pp.671–676, 2009.

[7] A. Datta, A. Choudhary, M.L. Bittner, and E.R. Dougherty, "External control in markovian genetic regulatory networks," Mach. Learn., vol.52, pp.169–191, 2003.

[8] B. Faryabi, J.-F. Chamberland, G. Vahedi, A. Datta, and E.R. Dougherty, "Optimal intervention in asynchronous genetic regulatory networks," IEEE J. Sel. Top. Signal Process., vol.2, no.3, pp.412–423, 2008.

[9] A. Ichikawa and K. Hiraishi, "Analysis and control of discrete event systems represented by Petri nets," Discrete Event Systems: Models and Applications, LNCIS 103, pp.115–134, 1988.

[10] H.D. Jong, "Modeling and simulation of genetic regulatory systems: A literature review," J. Computational Biology, vol.9, pp.67–103, 2002.

[11] S.A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," J. Theoretical Biology, vol.22, pp.437–467, 1969.

[12] K. Kobayashi, J. 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, 210685, 2010.

[13] 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.

[14] C.J. Langmead and S.K. Jha, "Symbolic approaches to finding control strategies in Boolean networks," J. Bioinformatics and Computational Biology, vol.7, no.2, pp.323–338, 2009.

[15] A. Milias-Argeitis, S. Summers, J. Stewart-Ornstein, I. Zuleta, D. Pincus, H. El-Samad, M. Khammash, and J. Lygeros, "*In silico* feedback for *in vivo* regulation of a gene expression circuit," Nature Biotechnology, vol.29, no.12, pp.1114–1116, 2011.

[16] T. Murata, "Petri Nets: Properties, analysis and applications," Proc. IEEE, vol.77, no.4, pp.541–580, 1989.

[17] I. Shmulevich, E.R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene reg-

ulatory networks," Bioinformatics, vol.18, pp.261–274, 2002.

[18] L.J. Steggles, R. Banks, O. Shaw, and A. Wipat, "Qualitatively modelling and analysing genetic regulatory networks: A Petri net approach," Bioinformatics, vol.23, no.3, pp.336–343, 2007.

[19] L. Tournier and M. Chaves, "Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics," J. Theoretical Biology, vol.260, pp.196–209, 2009.

[20] T. Ushio, "Maximally permissive feedback and modular control synthesis in Petri nets with external input places," IEEE Trans. Autom. Control, vol.35, no.7, pp.844–848, 1990.

[21] A.T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. Bittner, and J.M. Trent, "Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma," Cancer Cell, vol.1, pp.279–288, 2002.

[22] Y. Xiao and E.R. Dougherty, "The impact of function perturbations in Boolean networks," Bioinformatics, vol.23, pp.1265–1273, 2007.

[23] http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[24] http://www.informatik.uni-hamburg.de/TGI/PetriNets/

Appendix: Details of Matrices *Pre*, *Post*

We show matrices *Pre*, *Post* in Sect. 5 as follows:

$$
Pre = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0
\end{bmatrix},
$$

$$
Post = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}
$$

where places are arranged in order of $x_1, \bar{x}_1, \ldots, x_6, \bar{x}_6, u, \bar{u}$. In addition, transitions are arranged in order of

$t_{x_1,\emptyset}, t_{x_1,\{x_5\}}$,

$t_{x_2,\emptyset}, t_{x_2,\{x_6\}}$,

$t_{x_4,\emptyset}, t_{x_4,\{x_6\}}, t_{x_4,\{u\}}, t_{x_4,\{x_6,u\}}$,

$t_{x_5,\emptyset}, t_{x_5,\{x_2\}}, t_{x_5,\{x_3\}}, t_{x_4,\{x_2,x_3\}}$,

$t_{x_6,\emptyset}, t_{x_6,\{u\}}, t_{x_6,\{x_6,u\}}$.

Koichi Kobayashi   received the B.E. degree in 1998 and the M.E. degree in 2000 from Hosei University, and the D.E. degree in 2007 from Tokyo Institute of Technology. From 2000 to 2004, he worked at Nippon Steel Corporation. He is currently an Assistant Professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include computational methods for analysis and control of complex dynamical systems. He is a member of the SICE, ISCIE, and IEEE.

Kunihiko Hiraishi   received from the Tokyo Institute of Technology the B.E. degree in 1983, the M.E. degree in 1985, and the D.E. degree in 1990. He is currently a Professor at School of Information Science, Japan Advanced Institute of Science and Technology. His research interests include discrete event systems and formal verification. He is a member of the IEEE, SICE, and IPSJ.