

Review

Optimization-Based Approaches to Control of Probabilistic Boolean Networks

Koichi Kobayashi ^{1,*} and Kunihiko Hiraishi ²

¹ Graduate School of Information Science and Technology, Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo 060-0814, Hokkaido, Japan

² School of Information Science, Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan; hira@jaist.ac.jp

* Correspondence: k-kobaya@ssi.ist.hokudai.ac.jp; Tel.: +81-11-706-6452

Academic Editors: Tatsuya Akutsu and Henning Fernau

Received: 30 September 2016; Accepted: 20 February 2017; Published: 22 February 2017

Abstract: Control of gene regulatory networks is one of the fundamental topics in systems biology. In the last decade, control theory of Boolean networks (BNs), which is well known as a model of gene regulatory networks, has been widely studied. In this review paper, our previously proposed methods on optimal control of probabilistic Boolean networks (PBNs) are introduced. First, the outline of PBNs is explained. Next, an optimal control method using polynomial optimization is explained. The finite-time optimal control problem is reduced to a polynomial optimization problem. Furthermore, another finite-time optimal control problem, which can be reduced to an integer programming problem, is also explained.

Keywords: integer programming; gene regulatory network; polynomial optimization; probabilistic Boolean network

1. Introduction

It is important to understand biological behavior as a system. For this purpose, systems biology has been established as a new research field [1,2]. One of the practical applications in systems biology is gene therapy technologies (see, e.g., [3]). In gene therapy, the expression of certain genes must be controlled, and such a problem can be formulated as a control problem. In other words, control of gene regulatory networks corresponds to therapeutic interventions realized by radiation, chemotherapy, and so on. Thus, control theory of gene regulatory networks is important for developing medical services in the future. Furthermore, from the viewpoint of experimental study, several results on control of gene regulatory networks have been obtained in [4–7]. For example, in [5], it was shown that cellular behavior can be regulated by control. These results suggest that feedback control of biological systems can be implemented. Motivated by the above background, we are developing theory of control methods for gene regulatory networks.

In developing a control method of gene regulatory networks, first, it is important to consider what mathematical model is utilized. A gene regulatory network is modeled by discrete dynamical systems (e.g., Bayesian networks, Boolean networks (BNs) [8]), continuous dynamical systems (e.g., differential/difference equations), or hybrid dynamical systems (e.g., piecewise affine models) (see, e.g., [9] for further details). In control theory, BNs and piecewise affine models are frequently utilized (see, e.g., [10] for control of a biological system modeled by a piecewise affine model). The control problem of hybrid systems is generally reduced to a mixed integer programming problem, in which continuous variables and binary variables are decision variables, and the computation time to solve it is frequently too long. Hence, a class of gene regulatory networks is limited to those with a small network. A BN is a mathematical model whose interactions between genes are expressed by Boolean

functions [8]. That is, gene expression is approximated by a binary value (0 or 1). Since a BN is too simple, there is a possibility that biological behavior is not precisely modeled (see, e.g., [11]). On the other hand, this model can be relatively applied to large-scale systems. This advantage is focused on, and a BN is utilized in the first step of developing control theory of gene regulatory networks. Furthermore, a probabilistic Boolean network (PBN) [12] and a context-sensitive PBN [13,14] have been proposed as the extended models of BNs. In a PBN, at each discrete time, a Boolean function is randomly chosen from the candidates of Boolean functions. In a context-sensitive PBN, the time of choosing a Boolean function is also selected randomly. Hence, a context-sensitive PBN is a generalized version of a PBN.

In the last decade, many results on control of BNs and PBNs have been obtained so far (see, e.g., [15–32]). Especially, analysis and control methods using the semi-tensor product (STP) method have been widely studied (see, e.g., [33–43]). In the review paper [44], many methods for PBNs, e.g., construction and inference methods, structural intervention methods, and control methods, have been explained in a comprehensive manner. Of course, it is difficult to understand details of individual methods by reading only [44]. It is also significant to restrict topics and explain details. In this review paper, our previously proposed methods [23,24] in optimal external control of PBNs, i.e., a polynomial optimization approach and an integer programming approach are explained. Comparing our methods with other existing methods, the main advantage of our methods is that a relatively large-scale PBN can be handled. In many existing control methods (e.g., [19–21,30,32,36]), transition probability matrices with $2^n \times 2^n$ must be manipulated (n is the number of genes). As a result, a class of PBNs is restricted (see also Section 5 for further details). In our methods, transition probability matrices are not used. Moreover, since several solvers are available, optimization-based approaches are useful. To the best of our knowledge, such approaches have not been studied sufficiently, and their development in the future is necessary.

First, we introduce an optimal control method using polynomial optimization. In this method, the expected value of the state in PBN is modeled by a polynomial system, which can be directly obtained by Boolean functions and information on probabilities. Then, the finite-time optimal control problem is reduced to a polynomial optimization problem. Using e.g., SparsePOP [45], which is available on MATLAB R2006b or later, the polynomial optimization problem can be easily described and can be solved. However, there is a weakness that the polynomial optimization problem for large-scale PBNs cannot be solved owing to memory (but, comparing this method with other methods, a relatively large-scale PBN can be handled). In addition, the computational complexity of the finite-time optimal control problem of a PBN is Σ_2^P -hard [17], which is harder than NP-hard (Non-deterministic Polynomial time hard). Next, we introduce an optimal control method using integer programming. Instead of the expected value of the cost function, we focus on upper and lower bounds of the cost function under a certain constraint on probabilities. Then, the optimal control problem is reduced to an integer linear programming problem.

This review paper is organized as follows. In Section 2, PBNs are briefly explained. In Section 3, a control method using polynomial optimization is explained. In Section 4, a control method using integer programming is explained. In Section 5, discussion on numerical experiments is done. In Section 7, we conclude this review 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/vector X , let X^\top denote the transpose of X . Let I and 0 denote the identity matrix with an appropriate size and the zero matrix/vector/scalar, respectively. For an event A , let $P(A)$ denote the probability that A occurs. For two events A, B , let $P(A|B)$ denote the conditional probability of A given B . For two events A, B , let $E[A|B]$ denote the conditional expected value of A given B .

2. Probabilistic Boolean Networks

In this section, we introduce a probabilistic Boolean network (PBN).

Consider the following PBN:

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

where $x = [x_1 \ x_2 \ \dots \ x_n]^\top \in \{0,1\}^n$ is the state (e.g., the expression of genes), and $u = [u_1 \ u_2 \ \dots \ u_m]^\top \in \{0,1\}^m$ is the control input (e.g., the expression of genes), i.e., the value of u can be arbitrarily determined, $k \in \{0,1,2,\dots\}$ is the discrete time. The function $f^{(i)} : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^1$ is a given Boolean function consisting of logical operators such as AND (\wedge), OR (\vee), XOR (\oplus), and NOT (\neg). In deterministic Boolean networks, $x(k+1)$ is uniquely determined for given $x(k)$ and $u(k)$. In PBNs, the candidates of $f^{(i)}(x(k), u(k))$ are given, and for each x_i , selecting one Boolean function is probabilistically independent at each time. The candidates of $f^{(i)}(x(k), u(k))$ are denoted by $f_j^{(i)}(x(k), u(k)), j = 1, 2, \dots, l(i)$, and the probability that $f_j^{(i)}(x(k), u(k))$ is selected is denoted by

$$c_j^{(i)} = P(f^{(i)}(x(k), u(k)) = f_j^{(i)}(x(k), u(k))) \in [0, 1].$$

Then, the relation

$$\sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad (2)$$

must be satisfied.

First, we present a biological example of (deterministic) Boolean networks (BNs).

Example 1. Consider the following BN of an apoptosis network (see also Figure 1):

$$\begin{cases} x_1(k+1) = \neg x_2(k) \wedge u(k), \\ x_2(k+1) = \neg x_1(k) \wedge x_3(k), \\ x_3(k+1) = x_2(k) \vee u(k), \end{cases} \quad (3)$$

where x_1 is the concentration level (high or low) of the inhibitor of apoptosis proteins (IAP), x_2 is the concentration level of the active caspase 3 (C3a), and x_3 is the concentration level of the active caspase 8 (C8a). The control input u corresponds to the concentration level of the tumor necrosis factor (TNE, a stimulus). This model is introduced in [46], and is a simplified version of an apoptosis network model in [47]. In this model, $x_2(k) = 0$ implies cell survival, and $x_1(k) = 0, x_2(k) = 1$ imply cell death.

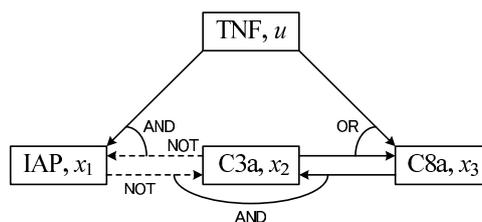


Figure 1. Simplified model of an apoptosis network.

Next, we present a simple example of PBNs.

Example 2. Consider the PBN with three states and single control input. Boolean functions and probabilities are given by

$$\begin{aligned}
 f^{(1)} &= \begin{cases} f_1^{(1)} = x_3(k) \wedge u(k), & c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_3(k), & c_2^{(1)} = 0.2, \end{cases} \\
 f^{(2)} &= f_1^{(2)} = x_1(k) \wedge \neg x_3(k), \quad c_1^{(2)} = 1.0, \\
 f^{(3)} &= \begin{cases} f_1^{(3)} = x_1(k) \wedge \neg x_2(k), & c_1^{(3)} = 0.7, \\ f_2^{(3)} = x_2(k) \wedge u(k), & c_2^{(3)} = 0.3, \end{cases}
 \end{aligned}$$

where $l(1) = 2, l(2) = 1$ and $l(3) = 2$ hold, and we see that the relation (2) is satisfied. Next, consider the state trajectories. Then, for $x(0) = [0 \ 0 \ 0]^T$ and $u(0) = 1$, we can obtain

$$\begin{aligned}
 P(x(1) = [0 \ 0 \ 0]^T \mid x(0) = [0 \ 0 \ 0]^T) &= 0.8, \\
 P(x(1) = [1 \ 0 \ 0]^T \mid x(0) = [0 \ 0 \ 0]^T) &= 0.2.
 \end{aligned}$$

In this example, the cardinality of the finite state set $\{0, 1\}^3$ is given by $2^3 = 8$, and we can obtain the state transition diagram of Figure 2 by computing the transition from each state under $u(0) = 0$. In Figure 2, the number assigned to each node denotes x_1, x_2, x_3 (elements of the state), and the number assigned to each arc denotes the transition probability from one state to other state. Note here that, for simplicity, the state transitions from only $x(k) = [0 \ 0 \ 0]^T, [0 \ 0 \ 1]^T, [0 \ 1 \ 0]^T, [1 \ 1 \ 0]^T$ are illustrated in Figure 2.

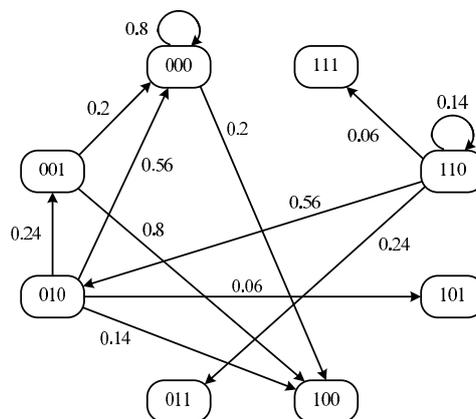


Figure 2. State transition diagram under $u(0) = 1$.

3. Polynomial Optimization Approach

In this section, a polynomial optimization approach to optimal control of PBNs is reviewed. The advantage of this approach is that implementation is easy by using e.g., MATLAB.

3.1. Problem Formulation

For PBN (1), consider the following finite-time optimal control problem.

Problem 1. Suppose that, for PBN (1), the initial state $x(0) = x_0$ and the control time N are given. Then, find a control input sequence $u(0), u(1), \dots, u(N - 1)$ minimizing the following cost function:

$$J = E \left[\sum_{k=0}^{N-1} \{Qx(k) + Ru(k)\} + Q_f x(N) \mid x(0) = x_0 \right], \tag{4}$$

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

In this problem, we use some linear function with respect to the state and the control input as a cost function. Noting that the relation $\delta^2 = \delta$ holds for a binary variable δ , it is unnecessary to add the quadratic term such as $x_i^2(k)$. Moreover, one of the main purposes in control of gene regulatory networks is that the expression of a certain gene is inhibited or activated (see e.g., [20]). Hence, simple linear cost function (4) is sufficient.

3.2. Transformation of a PBN into a Polynomial System

Consider modeling the dynamics of the expected value of the state by a polynomial system. First, the relation between a Boolean function and a polynomial is explained as a preparation. Next, the result for a general PBN is explained. Finally, a simple example is shown.

As a preparation, the following lemma [48] is introduced.

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

- (i) $\neg\delta_1$ is equivalent to $1 - \delta_1$,
- (ii) $\delta_1 \wedge \delta_2$ is equivalent to $\delta_1\delta_2$,
- (iii) $\delta_1 \vee \delta_2$ is equivalent to $\delta_1 + \delta_2 - \delta_1\delta_2$,
- (iv) $\delta_1 \oplus \delta_2$ is equivalent to $\delta_1 + \delta_2 - 2\delta_1\delta_2$.

By Lemma 1, a given Boolean function can be transformed into a polynomial on the real number field. For example, $\delta_1 \vee \neg\delta_2$ is equivalently transformed into $\delta_1 + (1 - \delta_2) - \delta_1(1 - \delta_2) = 1 - \delta_2 + \delta_1\delta_2$.

Next, a recursive representation of a general PBN is explained. Let $\hat{f}^{(i)}$ denote the polynomial that can be obtained from the Boolean function $f^{(i)}$. Let $\hat{f}_j^{(i)}$ denote the polynomial that can be obtained from the Boolean function $f_j^{(i)}$. Hereafter, for simplicity of notation, the condition in $E[x_i(k)|*]$ is omitted.

In [24], the following theorem was obtained.

Theorem 1. Suppose that, for the PBN (1), the initial state $x(0) = x_0$ is given. Then, the expected value of the state, $E[x_i(k)]$, is expressed as the following polynomial system:

$$\Sigma_i : E[x_i(k+1)] = \sum_{j=1}^{l(i)} c_j^{(i)} \hat{f}_j^{(i)}(E[x(k)], u(k)). \tag{5}$$

The outline of the proof is explained as follows. Since a Boolean function is independently switched for each state, $E[x_i x_j] = E[x_i]E[x_j]$, $i \neq j$ holds. In addition, there exists no second-order term of x_i in $\hat{f}_j^{(i)}$. Then, noting that $E[u(k)] = u(k)$ holds, we can obtain the system (5).

Noting that $\hat{f}_j^{(i)}(x(k), u(k))$ is a polynomial, the right-hand side of the system (5) is also a polynomial. Therefore, from Theorem 1, we see that time evolution of the expected value of the state is modeled by a polynomial system.

We present a simple example.

Example 3. Consider the PBN in Example 2 again. Suppose that, for this PBN, $x(0)$ and $u(0)$ are given by $x(0) = x_0 = [0 \ 0 \ 0]^T$ and $u(0) = 0$, respectively. Then, from Figure 2, we can obtain $E[x_1(1)] = 0.8 \cdot 0 + 0.2 \cdot 1 = 0.2$. We can also obtain it in other way. By Lemma 1, $x_3(k) \wedge u(k)$ and $\neg x_3(k)$ are transformed into $x_3(k)u(k)$ and $1 - x_3(k)$, respectively. Hence, we can obtain

$$\begin{aligned} E[x_1(1)] &= 0.8(x_3(0)u(0)) + 0.2(1 - x_3(0)) \\ &= 0.2. \end{aligned}$$

In a similar way, $E[x_2(1)]$ and $E[x_3(1)]$ can be obtained as

$$\begin{aligned} E[x_2(1)] &= x_1(0)(1 - x_3(0)) \\ &= 0, \\ E[x_3(1)] &= 0.7\{x_1(0)(1 - x_2(0))\} + 0.3(x_2(0)u(0)) \\ &= 0, \end{aligned}$$

respectively.

Next, suppose that $u(1)$ is given by $u(1) = 0$, and consider deriving $E[x(2)]$. Then, noting that a Boolean function is independently switched for each state, we can obtain

$$\begin{aligned} E[x_1(2)] &= E[f^{(1)}(x(1), u(1))] \\ &= 0.8E[f_1^{(1)}(x(1), u(1))] + 0.2E[f_2^{(1)}(x(1), u(1))] \\ &= 0.8E[x_3(1)u(1)] + 0.2E[1 - x_3(1)] \\ &= 0.8E[x_3(1)]u(1) + 0.2 - 0.2E[x_3(1)] \\ &= 0.2, \\ E[x_2(2)] &= E[f^{(2)}(x(1), u(1))] \\ &= 1.0E[x_1(1)(1 - x_3(1))] \\ &= E[x_1(1)] - E[x_1(1)]E[x_3(1)] \\ &= 0.2, \\ E[x_3(2)] &= E[f^{(3)}(x(1), u(1))] \\ &= 0.7E[x_1(1) - x_1(1)x_2(1)] + 0.3E[x_2(1)u(1)] \\ &= 0.14. \end{aligned}$$

By recursively repeating, we can obtain $E[x(k)]$, $k \geq 3$.

From this example, we see that the expected value of the state can be expressed as a polynomial system using Theorem 1.

3.3. Reduction to a Polynomial Optimization Problem

Consider reducing Problem 1 to a polynomial optimization problem. Then, the following theorem was obtained in [24].

Theorem 2. Problem 1 is equivalent to the following polynomial optimization problem:

$$\begin{aligned} \text{find} \quad & E[x(k+1)] \in \mathcal{R}^n, \quad u(k) \in \mathcal{R}^m, \quad k = 0, 1, \dots, N-1 \\ \text{min} \quad & \text{Cost function (4)} \\ \text{subject to} \quad & \text{System } \Sigma_i, \quad i = 1, 2, \dots, n, \\ & x(0) = x_0, \\ & u_i(k)(u_i(k) - 1) = 0. \end{aligned}$$

This theorem is immediately obtained from the following two properties: (i) the relation $E[u(k)] = u(k)$ holds; (ii) the fact that $u_i(k)$ is a binary variable is equivalent to the constraint $u_i(k)(u_i(k) - 1) = 0$. We remark here that this constraint is non-convex. From the practical viewpoint, it is desirable to relax this constraint to $0 \leq u_i(k) \leq 1$. Furthermore, from Theorem 1, we see that $E[x(k+1)] \in [0, 1]^n$ always holds. Hence, we set $E[x(k+1)] \in \mathcal{R}^n$ in the above polynomial optimization problem. Finally, $E[x(k+1)]$ may be eliminated from decision variables in the above

problem, but its elimination is not easy. Hence, $E[x(k + 1)]$ is included in a decision variable of the above problem.

A polynomial optimization problem can be solved by using e.g., SparsePOP [45], which is available on MATLAB. In SparsePOP, polynomials can be directly described.

4. Integer Programming Approach

The effectiveness of the polynomial optimization approach is restricted because the computational complexity of Problem 1 is Σ_2^P -hard [17], which is harder than NP-hard. In this section, an integer programming approach is explained. Instead of the expected value of the cost function, we focus on upper and lower bounds of the cost function. Then, the optimal control problem is reduced to an integer linear programming (ILP) problem. This method is a simplified version of our previously proposed method in [23].

4.1. Problem Formulation

First, preparations are given. Suppose that for PBN (1), $\bar{j}(i, k) \in \{1, 2, \dots, l(i)\}$ is given for fixed $i \in \{1, 2, \dots, n\}$ and $k \in \{0, 1, \dots\}$. Then, we define

$$\pi_{\bar{j}}(k) := \prod_{i=1}^n c_{\bar{j}(i,k)}^{(i)}.$$

The parameter $\pi_{\bar{j}}(k)$ implies the probability that the Boolean functions $f_{\bar{j}(1,k)}^{(1)}, f_{\bar{j}(2,k)}^{(2)}, \dots, f_{\bar{j}(n,k)}^{(n)}$ are selected at time k . Furthermore, we define

$$\pi_{\bar{j}}(k_1, k_2) := \prod_{q=k_1}^{k_2} \pi_{\bar{j}}(q).$$

The parameter $\pi_{\bar{j}}(k_1, k_2)$ implies the probability that certain Boolean functions are selected in the time interval $[k_1, k_2]$.

Example 4. Consider the PBN in Example 2 again. Suppose $\bar{j}(1, 0) = 1, \bar{j}(2, 0) = 1, \text{ and } \bar{j}(3, 0) = 2$. In other words, $f_1^{(1)}, f_1^{(2)}, f_2^{(3)}$ are given as a Boolean function at time 0. Then, we can obtain

$$\pi_{\bar{j}}(0) = c_1^{(1)} \cdot c_1^{(2)} \cdot c_2^{(3)} = 0.8 \cdot 1 \cdot 0.3 = 0.24.$$

Furthermore, suppose that $\bar{j}(1, 1) = 2, \bar{j}(2, 1) = 1, \bar{j}(3, 1) = 1, \bar{j}(1, 2) = 1, \bar{j}(2, 2) = 1, \text{ and } \bar{j}(3, 2) = 1$. Then, we can obtain

$$\begin{aligned} \pi_{\bar{j}}(1) &= 0.2 \cdot 1 \cdot 0.7 = 0.14, \\ \pi_{\bar{j}}(2) &= 0.8 \cdot 1 \cdot 0.7 = 0.56, \\ \pi_{\bar{j}}(0, 2) &= \pi_{\bar{j}}(0) \cdot \pi_{\bar{j}}(1) \cdot \pi_{\bar{j}}(2) = 0.018816. \end{aligned}$$

Under these preparations, two problems (i.e., the optimal control problem and the performance evaluation problem) are formulated. First, the optimal control problem is formulated as follows.

Problem 2. Suppose that, for PBN (1), the initial state $x(0) = x_0, \rho$ satisfying $0 \leq \rho \leq 1$, the control time N , and the weighting vectors $Q, Q_f \in \mathcal{R}^{1 \times n}, R \in \mathcal{R}^{1 \times m}$ whose element is a non-negative real number are given. Then, solve the following two problems.

Problem 2-A: Find a control input sequence $u(0), u(1), \dots, u(N - 1)$ minimizing the lower bound of the cost function

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

satisfying the following constraint

$$\pi_{\bar{y}}(0, N - 1) \geq \rho. \tag{7}$$

Problem 2-B: Apply the control input sequence obtained in Problem 2-A to the PBN. Then, find an upper bound of the cost function (6).

Let \bar{J}_s and \underline{J}_s denote the obtained upper and lower bounds, respectively. In a similar way to Problem 1, a linear function with respect to x and u is considered as a cost function. Using the target value of the state (the offset vector) $x_d \in \{0, 1\}^n$, $x(i)$ may be replaced by $\hat{x}(i) := x(i) - x_d$. Then, the cost function (6) must be changed to $J = \sum_{i=0}^{N-1} \{ |Q\hat{x}(i)| + Ru(i) \} + |Q_f \hat{x}(N)|$. Furthermore, by the inequality constraint (7), combinations of Boolean functions selected with the low probability can be excluded.

Each element of the control input sequence obtained solving Problem 2 is in general different. In gene regulatory networks, it is difficult to implement such control input sequence. In this case, we must impose constraints for control input sequences. For example, one of the simple methods is that the control input is given by always 1 (active) or 0 (inactive). That is, Problem 2-A is modified as follows: find $u_0 (= u(0) = u(1) = \dots = u(N - 1))$ minimizing the lower bound of the cost function (6) under the constraint (7).

We present a simple example.

Example 5. Consider the PBN with a single control input. Boolean functions and probabilities are given by

$$\begin{aligned} f^{(1)} &= \begin{cases} f_1^{(1)} = x_3(k) \wedge \neg u(k), & c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_3(k) \wedge \neg u(k), & c_2^{(1)} = 0.2, \end{cases} \\ f^{(2)} &= f_1^{(2)} = x_1(k) \wedge \neg x_3(k), \quad c_1^{(2)} = 1.0, \\ f^{(3)} &= \begin{cases} f_1^{(3)} = x_1(k) \wedge \neg x_2(k), & c_1^{(3)} = 0.7, \\ f_2^{(3)} = x_2(k) \vee u(k), & c_2^{(3)} = 0.3, \end{cases} \end{aligned}$$

respectively, where $u(k) \in \{0, 1\}^1$. The initial state and the cost function are given by

$$x(0) = x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \quad J = \sum_{k=0}^1 \{x_3(k) + u(k)\} + x_3(2),$$

respectively. For simplicity of discussion, ρ is given by $\rho = 0$, i.e., the constraint (7) is not imposed.

Consider calculating the cost function explicitly. First, $x_3(0) = 1$ holds. Next, we can obtain the following relations at time 1:

$$\begin{cases} P(x_1(1) = 1 \mid x(0) = x_0) = \neg u(0) = 0.8, \\ P(x_1(1) = 0 \mid x(0) = x_0) = 0.2, \\ P(x_2(1) = 0 \mid x(0) = x_0) = 1.0, \\ P(x_3(1) = 0 \mid x(0) = x_0) = 0.7, \\ P(x_3(1) = 1 \mid x(0) = x_0) = 0.3. \end{cases}$$

Furthermore, $x_3(2)$ can be derived as

$$\begin{cases} P(x_3(2) = \neg u(0) \mid x(0) = x_0) = 0.7 \cdot 0.8 = 0.56, \\ P(x_3(2) = 0 \mid x(0) = x_0) = 0.7 \cdot 0.2 = 0.14, \\ P(x_3(2) = u(1) \mid x(0) = x_0) = 0.3 \cdot 1.0 = 0.3. \end{cases}$$

Noting that $\neg u(0)$ is equivalent to $1 - u(0)$ (see Lemma 1), we can obtain

$$\begin{aligned} P(J = 2 + u(1) \mid x(0) = x_0) &= 0.392, \\ P(J = 1 + u(0) + 2u(1) \mid x(0) = x_0) &= 0.210, \\ P(J = 1 + u(0) + u(1) \mid x(0) = x_0) &= 0.098, \\ P(J = 3 + u(1) \mid x(0) = x_0) &= 0.168, \\ P(J = 2 + u(0) + 2u(1) \mid x(0) = x_0) &= 0.090, \\ P(J = 2 + u(0) + u(1) \mid x(0) = x_0) &= 0.042. \end{aligned}$$

First, consider solving Problem 2. We can obtain the optimal solution of Problem 2-A as $u(0) = u(1) = 0$. The minimum value of the lower bound of the cost function is $\underline{J}_s = 1$. The minimum value of the upper bounds can be obtained as $\bar{J}_s = 3$ (this is the solution of Problem 2-B). Next, consider the case of $u(0) = u(1) = 1$ as an example of other constant input. Then, the upper and lower bound of the cost function can be obtained by 2 and 5, respectively. From this result, we see that a solution for Problem 2-A should be applied to the system.

Problem 2 for a simple PBN can be easily solved. However, solving this problem for a large-scale PBN is hard. Here, we introduce an integer programming approach for solving Problem 2.

4.2. Solution Method for Problem 2-A

In this subsection, a solution method for Problem 2-A is derived. The solution method for Problem 2-B will be explained in the last part of this subsection.

As a preparation, we explain one lemma. The product of multiple binary variables such as $\delta_1\delta_2$ can be linearized by using the following lemma [49].

Lemma 2. Suppose that binary variables $\delta_j \in \{0, 1\}$, $j \in \mathcal{J}$ are given, where \mathcal{J} is some index set. Then, $z = \prod_{j \in \mathcal{J}} \delta_j$ is equivalent to the following linear inequalities

$$\sum_{j \in \mathcal{J}} \delta_j - z \leq |\mathcal{J}| - 1, \quad - \sum_{j \in \mathcal{J}} \delta_j + |\mathcal{J}|z \leq 0,$$

where $|\mathcal{J}|$ is the cardinality of \mathcal{J} .

From Lemmas 1 and 2, we see that any Boolean function can be equivalently transformed into a pair of linear functions and linear inequalities. See [48,49] for further details.

Consider transforming Problem 2-A into an ILP problem. Using Lemma 1, $f_j^{(i)}(x(k), u(k))$ in the PBN (1) can be equivalently transformed into a certain polynomial. Let $\hat{f}_j^{(i)}(x(k), u(k))$ denote the obtained polynomial. Then, consider the following system:

$$x_i(k + 1) = \sum_{j=1}^{l(i)} \left\{ \delta_{i,j}(k) \hat{f}_j^{(i)}(x(k), u(k)) \right\}, \tag{8}$$

where $i = 1, 2, \dots, n$ and $\delta_{i,j}(k) \in \{0, 1\}^1$. For $\delta_{i,j}(k)$, we impose the following constraint:

$$\sum_{j=1}^{l(i)} \delta_{i,j}(k) = 1. \tag{9}$$

Next, using $c_j^{(i)}$ and $\delta_{i,j}(k)$, we define

$$\begin{aligned} L_i &:= \left[\ln c_1^{(i)} \quad \ln c_2^{(i)} \quad \dots \quad \ln c_{l(i)}^{(i)} \right], \\ L &:= \left[L_1 \quad L_2 \quad \dots \quad L_n \right], \\ \delta_i(k) &:= \left[\delta_{i,1} \quad \delta_{i,2} \quad \dots \quad \delta_{i,l(i)} \right]^T \in \{0, 1\}^{l(i)}, \\ \delta(k) &:= \left[\delta_1(k) \quad \delta_2(k) \quad \dots \quad \delta_n(k) \right]^T \in \{0, 1\}^l, \end{aligned}$$

where $l := \sum_{i=1}^n l(i)$. Then, the following lemma was obtained in [23].

Lemma 3. *Problem 2-A is equivalent to the following problem.*

Problem 2-C:

$$\begin{aligned} &\text{find } u(k), \delta(k), \quad k = 0, 1, \dots, N - 1 \\ &\text{min Cost function (6)} \\ &\text{subject to System (8), } x(0) = x_0, \\ &\text{Inequality constraint } \sum_{k=0}^{N-1} L\delta(k) \geq \ln \rho, \\ &\text{Equality constraint (9).} \end{aligned}$$

The outline of the proof is explained as follows. By deciding $\delta(k)$, Boolean functions at each time are uniquely determined. Then, we can obtain $\ln \pi_{\bar{y}}(k) = L\delta(k)$. That is, using the logarithm of the probability, $\pi_{\bar{y}}(k)$ can be expressed by a linear form with respect to $\delta(k)$. Hence, we can obtain

$$\ln \pi_{\bar{y}}(0, N - 1) = \sum_{k=0}^{N-1} \ln \pi_{\bar{y}}(k) = \sum_{k=0}^{N-1} L\delta(k).$$

Thus, Problem 2-A is equivalent to Problem 2-C.

We consider further rewriting Problem 2-C. Since the system (8) is a polynomial system, the system (8) can be equivalently transformed into a linear system by using Lemma 2. Hence, the system (8) can be transformed into the following system:

$$\begin{cases} x(k+1) = Ax(k) + B_1u(k) + B_2\delta(k) + B_3z(k), \\ Cx(k) + D_1u(k) + D_2\delta(k) + D_3z(k) \leq E, \end{cases} \tag{10}$$

where $z(k) \in \{0, 1\}^p$ is an auxiliary variable, and the dimension p can be determined from the number of the product of binary variables. For simplicity of notation, we define $B := [B_1 \ B_2 \ B_3]$, $v(k) := [u^T(k) \ \delta^T(k) \ z^T(k)]^T$. In addition, under $x(0) = x_0 \in \{0, 1\}^n$ and $v(k) \in \{0, 1\}^{m+l+p}$, $x(k)$ becomes a binary variable automatically. Hence, we set $x(k) \in \mathcal{R}^n, k \geq 1$.

We present a simple example.

Example 6. Consider the PBN in Example 5 again. Then, the system (8) can be derived as

$$\begin{aligned} x_1(k+1) &= \delta_{1,1}(k) \{x_3(k)(1-u(k))\} + \delta_{1,2}(k) \{(1-x_3(k))(1-u(k))\}, \\ x_2(k+1) &= \delta_{2,1}(k) \{x_1(k)(1-x_3(k))\}, \\ x_3(k+1) &= \delta_{3,1}(k) \{x_1(k)(1-x_2(k))\} + \delta_{3,2}(k) \{x_2(k) + u(k) - x_2(k)u(k)\}. \end{aligned}$$

The equality constraint (9) can be derived as

$$\sum_{j=1}^2 \delta_{1,j}(k) = 1, \quad \delta_{2,1}(k) = 1, \quad \sum_{j=1}^2 \delta_{3,j}(k) = 1.$$

The state equation in the system (10) can be derived as

$$\begin{aligned} x_1(k+1) &= z_{1,1}(k) - z_{1,2}(k) + \delta_{1,2}(k) - z_{1,3}(k) - z_{1,4}(k) + z_{1,5}(k), \\ x_2(k+1) &= x_1(k) - z_{2,1}(k), \\ x_3(k+1) &= z_{3,1}(k) - z_{3,2}(k) + z_{3,3}(k) + z_{3,4}(k) - z_{3,5}(k), \end{aligned}$$

where

$$\begin{aligned} z_{1,1} &= \delta_{1,1}x_3, \quad z_{1,2} = \delta_{1,1}x_3u, \quad z_{1,3} = \delta_{1,2}u, \quad z_{1,4} = \delta_{1,2}x_3, \quad z_{1,5} = \delta_{1,2}x_3u, \\ z_{2,1} &= x_1x_3, \\ z_{3,1} &= \delta_{3,1}x_1, \quad z_{3,2} = \delta_{3,1}x_1x_2, \quad z_{3,3} = \delta_{3,2}x_2, \quad z_{3,4} = \delta_{3,2}u, \quad z_{3,5} = \delta_{3,2}x_2u. \end{aligned}$$

Using Lemma 2, these relations can be equivalently transformed into a set of linear inequalities. Then, $p = 11$ holds. Thus, the linear inequality in the system (10) can be derived.

Using the system (10), consider rewriting Problem 2-C as an ILP problem. First, equality constraint (9) is embedded in the inequality constraint in the system (10). From the state equation in the system (10), we can obtain

$$x(k) = A^k x_0 + \sum_{i=1}^k A^{i-1} B v(k-i).$$

Using this expression, we can obtain

$$\bar{x} = \bar{A}x_0 + \bar{B}\bar{v}, \tag{11}$$

where $\bar{x} := [x^\top(0) \ x^\top(1) \ \dots \ x^\top(N)]^\top$, $\bar{v} := [v^\top(0) \ v^\top(1) \ \dots \ v^\top(N-1)]^\top$ and

$$\bar{A} = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{N-1}B & \dots & AB & B \end{bmatrix}.$$

From the linear inequality in the system (10), we can obtain

$$\bar{C}\bar{x} + \bar{D}\bar{v} \leq \bar{E}, \tag{12}$$

where

$$\bar{C} = \begin{bmatrix} C & & 0 \\ & \ddots & \vdots \\ & & C & 0 \end{bmatrix}, \bar{D} = \begin{bmatrix} D & & \\ & \ddots & \\ & & D \end{bmatrix}, \bar{E} = \begin{bmatrix} E \\ \vdots \\ E \end{bmatrix}.$$

Next, consider the linear inequality $\sum_{k=0}^{N-1} L\delta(k) \geq \ln \rho$ in Problem C. From $\delta(k) = Wv(k)$, $W := [0_{l \times m} \ I_l \ 0_{l \times m}]$, this inequality can be rewritten as

$$-\bar{L}\bar{W}\bar{v} \leq -\ln \rho,$$

where $\bar{L} = [L \ \dots \ L]$ and

$$\bar{W} = \begin{bmatrix} W & & \\ & \ddots & \\ & & W \end{bmatrix}.$$

Cost function (6) can be rewritten as

$$J = \bar{Q}\bar{x} + \bar{R}\bar{v}, \tag{13}$$

where $\bar{Q} := [Q \ \dots \ Q \ Q_f]$ and $\bar{R} := [R \ \dots \ R]$. By substituting the state Equation (11) into the inequality (12) and the cost function (13), we can obtain the ILP problem. In [23], this result was obtained as the following theorem.

Theorem 3. *Problem 2-C is equivalent to the following ILP problem.*

Problem 2-D:

$$\begin{aligned} & \text{find } \bar{v} \in \{0, 1\}^{(m+l+p)N} \\ & \text{min } (\bar{R} + \bar{Q}\bar{B})\bar{v} + \bar{Q}\bar{A}x_0 \\ & \text{subject to } \begin{bmatrix} \bar{C}\bar{B} + \bar{C} \\ -\bar{L}\bar{W} \end{bmatrix} \bar{v} \leq \begin{bmatrix} \bar{E} - \bar{C}\bar{A}x_0 \\ -\ln \rho \end{bmatrix}. \end{aligned}$$

Problem 2-D can be solved by using a suitable free/commercial solver.

Finally, we explain a solution method for Problem 2-B. In Problem 2-B, we assume that $u(k)$ (i.e., certain elements of \bar{v}) is given. Under this assumption, Problem 2-B can be transformed into an ILP problem by replacing “min” with “max” in Problem 2-D.

5. Discussion

In this section, from the viewpoint of numerical experiments, we compare two introduced methods [23,24] with other existing methods.

In many existing methods, transition probability matrices with $2^n \times 2^n$ in discrete-time Markov chains must be manipulated (see, e.g., [19–21,30,32,36]). In Example 2, the transition probability matrix with $u(0) = 0$ can be obtained as

$$\begin{bmatrix} 0.8 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 \\ 0.2 & 0 & 0 & 0 & 0.8 & 0 & 0 & 0 \\ 0.56 & 0.24 & 0 & 0 & 0.14 & 0.06 & 0 & 0 \\ 0.14 & 0.06 & 0 & 0 & 0.56 & 0.24 & 0 & 0 \\ 0 & 0 & 0.24 & 0.56 & 0 & 0 & 0.06 & 0.14 \\ 0.06 & 0.14 & 0 & 0 & 0.24 & 0.56 & 0 & 0 \\ 0 & 0 & 0.56 & 0.24 & 0 & 0 & 0.14 & 0.06 \\ 0.14 & 0.06 & 0 & 0 & 0.56 & 0.24 & 0 & 0 \end{bmatrix}.$$

See also the state transition diagram in Figure 1. In recent years, analysis and control methods for BNs and PBNs using the semi-tensor product approach have been widely studied (see, e.g., [34,36]). In this approach, matrices with $2^n \times 2^n$ must be manipulated. It is hard to manipulate matrices with such size. In MATLAB, matrices with $2^{16} \times 2^{16}$ cannot be generated. Although matrices with $2^{15} \times 2^{15}$ can be generated, computation of the product of matrices requires a long time. This is a weakness of many existing methods.

In two methods introduced in this review paper, matrices with such size are not manipulated. As a result, we can compute the optimal control problem. In [24], we presented a numerical experiment for the polynomial optimization approach. Then, we reported that for a PBN with 15 states and three control inputs, the computation times for solving Problem 1 in the cases of $N = 2, 3, 4, 5$ were 0.6 s ($N = 2$), 19.6 s ($N = 3$), 218.8 s ($N = 4$), and 334.4 s ($N = 5$), respectively. In this numerical experiments, we used SparsePOP [45] for solving a polynomial optimization problem. The computer environment is CPU: Intel Core i7 1.2 GHz, Memory: 4 GB, Windows 7 Professional 64 bit. In the integer programming approach, we presented in [23] a numerical experiment for a context-sensitive PBN with 15 states and three control inputs. Then, we reported that the computation time for solving Problem 2-A with $N = 10$ was 18.96 s. In this numerical experiments, we used ILOG CPLEX 11.0 for solving an ILP problem. The computer environment is CPU: Intel Core 2 Duo CPU 3.0 GHz, Memory: 4 GB, Windows 7 Professional 64 bit. We remark that it is hard to solve the optimal control problem of a PBN with such size by using the other existing methods such as [20,21,34,36]. Thus, two methods introduced here enable us to compute the optimal control input for relatively large-scale PBNs.

6. Open Problems in Control Theory of Probabilistic Boolean Networks

We have explained our previously proposed methods, and we have compared these methods with other existing methods. In this section, we suggest open problems in control theory of PBNs. Here, we consider not only optimal control methods but also other control methods.

Although our previously proposed methods can handle large-scale PBNs, these methods are not scalable. Developing scalable control methods is important. Then, decomposition of networks is important (see, e.g., [50,51]). For BNs, we consider only one network structure (directed graph) such as Figure 1. On the other hand, for PBNs, we must consider a set of directed graphs. Hence, the problem of decomposing given networks in a PBN is more difficult than that in a BN. This is one of the challenging problem in control of PBNs. Moreover, the optimal control problems for PBNs obtained from decomposed graphs can be solved fast, but their solutions may not be an optimal solution for the optimal control problem of a given PBN. As a result, the optimality of the cost function is degraded. Now, some analysis of the approximation accuracy is required.

It is also important to consider what can be obtained from graph structure alone. In [22], controllability of BNs is characterized using a time sequence of graph structure. In [16], singleton attractors (fixed points) of BNs are characterized from graph structure only. In [52,53], a feedback vertex set of a graph is focused. A feedback vertex set of a graph is a set of vertices whose removal results an acyclic graph. For a given graph, several feedback vertex sets can be obtained, but in particular, a minimum feedback vertex set in which the number of vertices is minimum is important. A minimum feedback vertex set is also not given uniquely. It was shown in [52,53] that in the system given by a differential equation, a certain property of all states can be characterized by analyzing the states relating to a certain feedback vertex set. However, the related results for BNs and PBNs have not been obtained so far.

In our previously proposed methods explained in this paper, a controller must be implemented by the personal computer that can solve a polynomial optimization problem and an integer programming problem (of course, actuators for interacting genes are necessary, but this topic is out of scope of this review paper). We may consider the case where a controller must be described by a certain Boolean function. In such a case, the algebraic representation using the STP and the matrix-based representation [54,55] are useful. Details are one of the open problems.

Since BNs and PBNs are synchronous models, an extension to asynchronous models is important. Then, it is appropriate to utilize a Petri net representation. A Petri net is well known as a model of asynchronous behavior (see, e.g., [56] for further details), and is also used as a model of biological networks (see, e.g., [9,57]). In [58], the authors have proposed a Petri net representation for modeling asynchronous BN. In the case of PBNs, a stochastic Petri net may be used as a model of asynchronous PBNs, but further discussion is required.

Finally, the state and the control input in BNs and PBNs are binary. There is a possibility that biological behavior is not precisely modeled (see, e.g., [11]). Then, it may be appropriate that some of states/inputs are given by continuous-valued variables. A dynamical system with binary variables and continuous variables is called a hybrid dynamical system (see, e.g., [59]). Further research on modeling and control methods is needed.

Thus, in control theory of PBNs, there are many remaining open problems. Further development of many control methods is required.

7. Conclusions

The optimal control problem of PBNs is one of the significant problems in systems biology. This problem can be rewritten as the optimal control problem of discrete-time Markov chains, which has been extensively studied. However, the size of Markov chains exponentially grows with the state of PBNs. Hence, it is important to consider a solution method in which Markov chains are not computed. In this review paper, our previously proposed methods (i.e., the polynomial optimization approach and the integer programming approach) in control of PBNs were explained as computation methods that do not require Markov chains. In addition, we compared these methods with the other existing methods. Finally, we explained open problems in control theory of PBNs.

Acknowledgments: This research was partly supported by Grant-in-Aid for Scientific Research (C) 26420412.

Author Contributions: Koichi Kobayashi and Kunihiko Hiraishi wrote the paper. Both authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kitano, H. Systems biology: A brief overview. *Science* **2002**, *295*, 1662–1664.
2. Kitano, H. Computational systems biology. *Nature* **2002**, *420*, 206–210.
3. Santos-Rosa, H.; Caldas, C. Chromatin modifier enzymes, the histone code and cancer. *Eur. J. Cancer* **2005**, *41*, 2381–2402.
4. Menolascina, F.; Fiore, G.; Orabona, E.; de Stefano, L.; Ferry, M.; Hasty, J.; di Bernardo, M.; di Bernardo, D. In-vivo real-time control of protein expression from endogenous and synthetic gene networks. *PLoS Comput. Biol.* **2014**, *10*, doi:10.1371/journal.pcbi.1003625.
5. Miliadis-Argeitis, A.; Summers, S.; Stewart-Ornstein, J.; Zuleta, I.; Pincus, D.; El-Samad, H.; Khammash, M.; Lygeros, J. In silico feedback for in vivo regulation of a gene expression circuit. *Nat. Biotechnol.* **2011**, *29*, 1114–1116.
6. Pathak, G.P.; Vrana, J.D.; Tucker, C.L. Optogenetic control of cell function using engineered photoreceptors. *Biol. Cell* **2013**, *105*, 59–72.
7. Uhlerdorf, J.; Miermont, A.; Delaveau, T.; Charvin, G.; Fages, F.; Bottani, S.; Batta, G.; Hersen, P. Long-term model predictive control of gene expression at the population and single-cell levels. *Proc. Natl. Acad. Sci. USA* **2012**, *109*, 14271–14276.
8. Kauffman, S.A. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.* **1969**, *22*, 437–467.
9. Jong, H.D. Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.* **2002**, *9*, 67–103.
10. Azuma, S.; Yanagisawa, E.; Imura, J. Controllability analysis of biosystems based on piecewise affine systems approach. *IEEE Trans. Autom. Control* **2008**, *53*, 139–152.

11. Mochizuki, A. An analytical study of the number of steady states in gene regulatory networks. *J. Theor. Biol.* **2005**, *236*, 291–310.
12. Shmulevich, I.; Dougherty, E.R.; Kim, S.; Zhang, W. Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* **2002**, *18*, 261–274.
13. Faryabi, B.; Vahedi, G.; Chamberland, J.-F.; Datta, A.; Dougherty, E.R. Intervention in context-sensitive probabilistic Boolean networks revisited. *EURASIP J. Bioinform. Syst. Biol.* **2009**, *2009*, 360864.
14. Pal, R.; Datta, A.; Bittner, M.L.; Dougherty, E.R. Intervention in context-sensitive probabilistic Boolean networks. *Bioinformatics* **2005**, *21*, 1211–1218.
15. Akutsu, T.; Hayashida, M.; Ching, W.-K.; Ng, M.K. Control of Boolean networks: Hardness results and algorithms for tree structured networks. *J. Theor. Biol.* **2007**, *244*, 670–679.
16. Azuma, S.; Yoshida, T.; Sugie, T. Structural monostability of activation-inhibition Boolean networks. In Proceedings of the 53rd IEEE Conference on Decision and Control, Los Angeles, CA, USA, 15–17 December 2014; pp. 1521–1526.
17. Chen, X.; Akutsu, T.; Tamura, T.; Ching, W.-K. Finding optimal control policy in probabilistic Boolean networks with hard constraints by using integer programming and dynamic programming. In Proceedings of the 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Hong Kong, China, 18–21 December 2010; pp. 240–246.
18. Chen, X.; Jiang, H.; Qiu, Y.; Ching, W.-K. On optimal control policy for probabilistic Boolean network: A state reduction approach. *BMC Syst. Biol.* **2012**, *6*, doi:10.1186/1752-0509-6-S1-S8.
19. Ching, W.-K.; Zhang, S.-Q.; Jiao, Y.; Akutsu, T.; Tsing, N.-K.; Wong, A.S. Optimal control policy for probabilistic Boolean networks with hard constraints. *IET Syst. Biol.* **2009**, *3*, 90–99.
20. Datta, A.; Choudhary, A.; Bittner, M.L.; Dougherty, E.R. External control in Markovian genetic regulatory networks. *Mach. Learn.* **2003**, *52*, 169–191.
21. Datta, A.; Choudhary, A.; Bittner, M.L.; Dougherty, E.R. External control in Markovian genetic regulatory networks: The imperfect information case. *Bioinformatics* **2004**, *20*, 924–930.
22. Kobayashi, K.; Imura, J.; Hiraishi, K. Polynomial-time algorithm for controllability test of a class of Boolean biological networks. *EURASIP J. Bioinform. Syst. Biol.* **2010**, *2010*, 210685.
23. Kobayashi, K.; Hiraishi, K. An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks. *Automatica* **2011**, *47*, 1260–1264.
24. Kobayashi, K.; Hiraishi, K. Optimal control of probabilistic Boolean networks using polynomial optimization. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2012**, *E95-A*, 1512–1517.
25. Kobayashi, K.; Hiraishi, K. Symbolic approach to verification and control of deterministic/probabilistic Boolean networks. *IET Syst. Biol.* **2012**, *6*, 215–222.
26. Kobayashi, K.; Hiraishi, K. Optimal control of gene regulatory networks with effectiveness of multiple drugs: A Boolean network approach. *BioMed Res. Int.* **2013**, *2013*, 246761.
27. Langmead, C.J.; Jha, S.K. Symbolic approaches to finding control strategies in Boolean networks. *J. Bioinform. Comput. Biol.* **2009**, *7*, 323–338.
28. Liu, Q. An optimal control approach to probabilistic Boolean networks. *Physica A* **2012**, *391*, 6682–6689.
29. Lu, J.; Zhong, J.; Li, L.; Ho, D.W.C.; Cao, J. Synchronization analysis of master-slave probabilistic Boolean networks. *Sci. Rep.* **2015**, *5*, 13437.
30. Pal, R.; Datta, A.; Dougherty, E.R. Optimal infinite-horizon control for probabilistic Boolean networks. *IEEE Trans. Signal Process.* **2006**, *54*, 2375–2387.
31. Srihari, S.; Raman, V.; Leong, H.W.; Ragan, M.A. Evolution and controllability of cancer networks: A Boolean perspective. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2014**, *11*, 83–94.
32. Yousefi, M.R.; Datta, A.; Dougherty, E.R. Optimal intervention strategies for therapeutic methods with fixed-length duration of drug effectiveness. *IEEE Trans. Signal Process.* **2012**, *60*, 4930–4944.
33. Cheng, D.; Qi, H. Controllability and observability of Boolean control network. *Automatica* **2009**, *45*, 1659–1667.
34. Cheng, D.; Qi, H.; Li, Z. *Analysis and Control of Boolean Network: A Semi-tensor Product Approach*; Springer: London, UK, 2011.
35. Laschov, D.; Margaliot, M. Controllability of Boolean control networks via the Perron-Frobenius theory. *Automatica* **2009**, *48*, 1218–1223.
36. Li, F.; Sun, J. Controllability of probabilistic Boolean control networks. *Automatica* **2011**, *47*, 2765–2771.
37. Li, F.; Sun, J. Controllability of higher order Boolean control networks. *Appl. Math. Comput.* **2012**, *219*, 158–169.

38. Li, H.; Wang, Y. On reachability and controllability of switched Boolean control networks. *Automatica* **2012**, *48*, 2917–2922.
39. Li, H.; Wang, Y.; Xie, L. Output tracking control of Boolean control networks via state feedback: Constant reference signal case. *Automatica* **2015**, *59*, 54–59.
40. Li, H.; Xie, L.; Wang, Y. On robust control invariance of Boolean control networks. *Automatica* **2016**, *68*, 392–396.
41. Li, F. Pinning control design for the stabilization of Boolean networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1585–1590.
42. Lu, J.; Zhong, J.; Huang, C.; Cao, J. On pinning controllability of Boolean control networks. *IEEE Trans. Autom. Control* **2016**, *61*, 1658–1663.
43. Zhang, L.; Zhang, K. Controllability and observability of Boolean control networks with time-variant delays in states. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1478–1484.
44. Trairatphisan, P.; Mizera, A.; Pang, J.; Tantar, A.A.; Schneider, J.; Sauter, T. Recent development and biomedical applications of probabilistic Boolean networks. *Cell Commun. Signal.* **2013**, *11*, doi:10.1186/1478-811X-11-46.
45. SparsePOP. Available online: <http://www.is.titech.ac.jp/~kojima/SparsePOP/SparsePOP.html> (accessed on 21 February 2017).
46. Chaves, M. Methods for qualitative analysis of genetic networks. In Proceedings of the 2009 European Control Conference (ECC), Budapest, Hungary, 23–26 August 2009; pp. 671–676.
47. Tournier, L.; Chaves, M. Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics. *J. Theor. Biol.* **2009**, *260*, 196–209.
48. Williams, H.P. *Model Building in Mathematical Programming*, 5th ed.; Wiley: Hoboken, NJ, USA, 2013.
49. Cavalier, T.M.; Pardalos, P.M.; Soyster, A.L. Modeling and integer programming techniques applied to propositional calculus. *Comput. Oper. Res.* **1990**, *17*, 561–570.
50. Zhao, Y.; Kim, J.; Filippone, M. Aggregation algorithm towards large-scale Boolean network analysis. *IEEE Trans. Autom. Control* **2013**, *58*, 1976–1985.
51. Zhao, Y.; Ghosh, K.; Cheng, D. Control of large-scale Boolean networks via network aggregation. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 1527–1536.
52. Fiedler, B.; Mochizuki, A.; Kurosawa, G.; Saito, D. Dynamics and Control at Feedback Vertex Sets. I: Informative and Determining Nodes in Regulatory Networks. *J. Dyn. Differ. Equ.* **2013**, *25*, 563–604.
53. Mochizuki, A.; Fiedler, B.; Kurosawa, G.; Saito, D. Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks *J. Theor. Biol.* **2013**, *335*, 130–146.
54. Kobayashi, K.; Hiraishi, K. ILP/SMT-based method for design of Boolean networks based on singleton attractors. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2014**, *11*, 1253–1259.
55. Kobayashi, K.; Hiraishi, K. Design of probabilistic Boolean networks based on network structure and steady-state probabilities. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, doi:10.1109/TNNLS.2016.2572063.
56. Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* **1989**, *77*, 541–580.
57. Chaouiya, C.; Remy, E.; Thieffry, D. Petri net modelling of biological regulatory networks. *J. Discret. Algorithms* **2008**, *6*, 165–177.
58. Kobayashi, K.; Hiraishi, K. Optimal control of Boolean biological networks modeled by Petri nets. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2013**, *E96-A*, 532–539.
59. Lunze, J.; Lamnabhi-Lagarrigue, F. (Eds.) *Handbook of Hybrid Systems Control: Theory, Tools, Applications*; Cambridge University Press: Cambridge, UK, 2009.

