Finding Optimal Control Policy in Probabilistic Boolean Networks with Hard Constraints by Using Integer Programming and Dynamic Programming

Xi Chen*, Tatsuya Akutsu†, Takeyuki Tamura†, Wai-Ki Ching*

*AMAC Laboratory, Department of Mathematics, The University of Hong Kong, Hong Kong, China
e-mail: dlkcissy@hku.hk, wching@hkusua.hku.hk

†Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto, Japan
e-mail: {takutsu,tamura}@kuicr.kyoto-u.ac.jp

***Abstract*—In this paper, we study control problems of Boolean Networks (BNs) and Probabilistic Boolean Networks (PBNs). For BN CONTROL, by applying external control, we propose to derive the network to the desired state within a few time steps. For PBN CONTROL, we propose to find a control sequence such that the network will terminate in the desired state with a maximum probability. Also, we propose to minimize the maximum cost of the terminal state to which the network will enter. Integer linear programming and dynamic programming in conjunction with hard constraints are then employed to solve the above problems. Numerical experiments are given to demonstrate the effectiveness of our algorithms. We also present a hardness result suggesting that PBN CONTROL is harder than BN CONTROL.**

***Keywords*-Boolean networks; probabilistic Boolean networks; optimal control; integer linear programming; dynamic programming**

I. Introduction

Developing efficient algorithms for control of genetic regulatory networks is an important research issue in bioinformatics. A number of formalisms have been developed for modeling genetic regulation processes, such as Bayesian networks, multivariate Markov chain model [1], Boolean networks and probabilistic Boolean networks. A brief review can be found in [2]. Among all the models, Boolean networks (BNs) and their extension probabilistic Boolean networks (PBNs) have received much attention since they are able to capture the switching behavior of the genetic process.

Boolean network (BN) was first introduced by Kauffman [3] in 1969. It is a very simple model: each gene is quantized to only two levels – on and off (represented as 1 and 0). The target gene can be regulated by several genes called its input genes via its Boolean function (predictor function). A BN is said to be well defined if all the input genes and Boolean functions are given. However, a BN is a deterministic model. The randomness comes only from the initial state. Therefore, it is more realistic to extend a BN to a stochastic one, namely, Probabilistic Boolean Network (PBN). In a PBN, instead of having only one Boolean function, each gene can have multiple Boolean functions with selecting probabilities assigned to them. The dynamics of a PBN can be studied and analyzed by the theory of Markov chain. Furthermore, it is possible to control one or more genes in a network such that the whole network is derived into a desired state or a steady-state distribution. Then therapeutic gene intervention or gene control policy can be developed [4], [5], [6].

In this paper, we propose to solve the control problem of PBNs by using integer linear programming and dynamic programming in conjunction with hard constraints. In [7] an integer programming approach has been applied to solve the control problem of PBN. Here we consider adding hard constraints (i.e. adding an upper bound for the number of controls that can be applied to the network [6]) into the problem and propose an integer linear programming based method with hard constraints to solve the control problem of BN and PBN. Introduction of hard constraints is important for medical applications because the number of treatments such as radiation and chemo-therapy is usually limited [6]. Moreover, given the terminal cost for each state, we want to derive the network into the state with the minimized maximum cost by applying external control. This problem is important also for medical applications because we may want to minimize the damage or cost even in the worst case. In addition to development of algorithms, we study the time complexity of control problems for PBN. We prove that both minimizing the maximum cost and minimizing the average cost are Σ_2^p-hard, where the latter problem corresponds to the original control problem for PBN [8]. It is known that control of BN is NP-complete[1] and control of PBN is NP-hard [4]. Since it is believed that Σ_2^p-hard problems are much harder than NP-complete problems [10], this result suggests that control of PBN is much harder than control of BN. Furthermore, this result suggests that such methods as integer linear programming cannot be effectively applied to solve the control problem of PBN because (a decision problem version of) integer linear programming is known

[1] Control of BN is NP-complete if the number of time steps is polynomially bounded. Otherwise, it is PSPACE-complete, which directly follows from the result of [9]. However, it is not usual to consider an exponential number of time steps.

978-1-4244-8305-1/10/$26.00 ©2010 IEEE

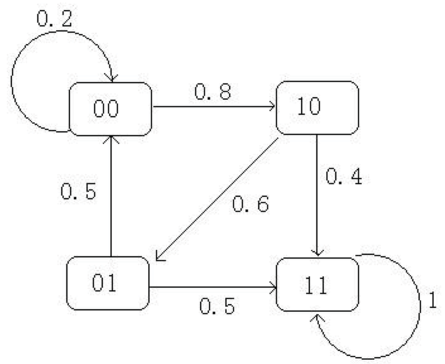


Figure 1. Example of a PBN. Dynamics of the PBN is well described by the state transition probability and the transition diagram.

to be NP-complete [10]. Therefore, integer programming-based approach can only be applied to control of BN [5] and special restricted variants of control of PBN [7].

II. Problems

A. Boolean Networks and Probabilistic Boolean Networks

A Boolean network (BN) is represented by a set of nodes (genes) $V = \{v_1, v_2, \ldots, v_n\}$ and a list of Boolean functions $F = \{f_1, f_2, \ldots, f_n\}$ where a Boolean function $f_i(v_{i_1}, \ldots, v_{i_k})$ with inputs from specified nodes $v_{i_1}, \ldots, v_{i_k}$ is assigned to v_i. We use $IN(v_i)$ to represent the set of input nodes $v_{i_1}, \ldots, v_{i_k}$ to v_i. The number of inputs to v_i is called the *indegree* of v_i. We define K as the *maximum indegree* of a BN.

We define $v_i(t)$ to be the state (0 or 1) of the gene i at time t. The rules of the regulatory interactions among the genes can then be represented by Boolean functions:

$$v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_k}(t)), \quad i = 1, 2, \ldots, n. \quad (1)$$

Here we let $\mathbf{v}(t) = (v_1(t), v_2(t), \ldots, v_n(t))^T$ which is called the *Gene Activity Profile* (GAP). The GAP can take any possible states from the set $S = \{(v_1, v_2, \ldots, v_n)^T : v_i \in \{0, 1\}\}$ and thus totally there are 2^n possible states in the network. We then define

$$z(t) = 1 + \sum_{i=1}^{n} 2^{n-i} v_i(t).$$

As $v_1(t)v_2(t)\ldots v_n(t)$ ranges from $00\ldots0$ to $11\ldots1$, $z(t)$ will take on all values from 1 to 2^n. Clearly, there is a one-to-one map from $x(t)$ to $z(t)$. Hence instead of the binary representation for the global state, one can use equivalent decimal representation $z(t)$.

Since a BN is a deterministic model, it is more realistic to extend it to a probabilistic setting. To extend the concepts of a BN to a stochastic model, for each vertex v_i in a PBN, instead of having only one Boolean function as in BN, there are a multiple of Boolean functions (predictor functions) $f_j^{(i)}(j = 1, 2, \ldots, l(i))$ to be chosen for determining the state of gene v_i and usually $l(i)$ is not very large. The probability of choosing $f_j^{(i)}$ as the predictor function is $c_j^{(i)}$,

$$0 \le c_j^{(i)} \le 1 \quad \text{and} \quad \sum_{j=1}^{l(i)} c_j^{(i)} = 1 \quad \text{for} \quad i = 1, 2, \ldots, n. \quad (2)$$

To estimate the probability $c_j^{(i)}$, one of the efficient methods is called Coefficient of Determination (COD) [11]. It can be used to estimate the probability $c_j^{(i)}$ with real gene expression data sets.

We let f_j be the jth possible realization, where $f_j = (f_{j_1}^{(1)}, f_{j_2}^{(2)}, \ldots, f_{j_n}^{(n)}), \quad 1 \le j_i \le l(i), \quad i = 1, 2, \ldots, n.$ Suppose that the selection of the Boolean function f_{j_i} for each gene i is an independent process, then the probability of choosing the corresponding BN with Boolean functions $f_j = (f_{j_1}^{(1)}, f_{j_2}^{(2)}, \ldots, f_{j_n}^{(n)})$ is given by

$$q_{j_1 j_2 \cdots j_n} = \prod_{i=1}^{n} c_{j_i}^{(i)}. \quad (3)$$

There are at most $N = \prod_{i=1}^{n} l(i)$ different possible realizations of BNs. Fig. 1 gives an example of PBN. We note that the transition process among the states in the set S forms a Markov chain process. Let $\mathbf{a}$ and $\mathbf{b}$ be any two column vectors in the set S. Then the transition probability

$$\begin{aligned} &\mathrm{P}\ \{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}\} = \textstyle\sum_{j=1}^{N} \\ &\mathrm{P}\ \{\mathbf{v}(t+1) = \mathbf{a} \mid \mathbf{v}(t) = \mathbf{b}, \text{the } j\text{th BN is selected}\} \cdot q_j \\ &= \textstyle\sum_{j \in \mathcal{I}} q_j \end{aligned}$$

where $\mathcal{I}$ is the set of BNs of which the transition probability from state $\mathbf{b}$ to state $\mathbf{a}$ is 1.

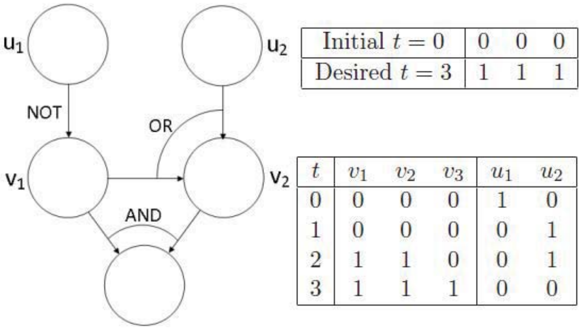
Here we let

$$q_j = q_{j_1 j_2 \cdots j_n} \quad \text{and} \quad j = j_1 + \sum_{i=2}^{n} \left((j_i - 1)(\prod_{k=1}^{i-1} l(k)) \right). \quad (4)$$

We can then use both of them when there is no confusion. By letting $\mathbf{a}$ and $\mathbf{b}$ take all the possible states in S, one can get the transition probability matrix of the Markov chain (or the PBN). The transition probability matrix can be written as

$$A = \sum_{j=1}^{N} q_j A_j \quad (5)$$

where A_j is the corresponding transition matrix of the jth BN and q_j is the probability of choosing the jth BN. We remark that there are at most $N2^n$ nonzero entries in the transition probability matrix A and this means the matrix is sparse, i.e., having a lot of zero entries.



Initial $t = 0$	0	0	0
Desired $t = 3$	1	1	1

t	v_1	v_2	v_3	u_1	u_2
0	0	0	0	1	0
1	0	0	0	0	1
2	1	1	0	0	1
3	1	1	1	0	0

Figure 2. Example of BN CONTROL. In this problem, given the initial and desired states of internal nodes (v_1, v_2, v_3), it is required to find a sequence of control inputs (u_1, u_2) leading to the desired state.

B. Control of BN with Hard Constraints

Akutsu et al. introduced a control problem for BNs in [4]. In BN CONTROL, there are two types of nodes: *internal nodes* and *external nodes*, where internal nodes correspond to usual nodes in a BN and external nodes correspond to control nodes. Let a set V of $n+m$ nodes be $V = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+m}\}$, where $v_1, \ldots, v_n$ are internal nodes and $v_{n+1}, \ldots, v_{n+m}$ are control nodes. Then the states of internal nodes at time $t+1$ are represented by

$$v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_k}(t)), \quad i = 1, 2, \ldots, n. \quad (6)$$

where each v_{i_j} is either an internal node or a control node. Fig. 2 gives an example of BN CONTROL. Here we let $\mathbf{v}(t) = [v_1(t), v_2(t), \ldots, v_n(t)]$ and $\mathbf{u}(t) = [v_{n+1}(t), v_{n+2}(t), \ldots, v_{n+m}(t)]$. If

$$v_{n+i}(t) - v_{n+i}(t+1) \neq 0, \text{ for some } i \in \{1, \ldots, m\}$$

then we say that the external control is applied once to the network. Thus the number of controls applied to network is equal to

$$\sum_{t=0}^{M-1} \sum_{i=1}^{m} |v_{n+i}(t) - v_{n+i}(t+1)|.$$

Then the control problem of BN under hard constraints is as follows:

Definition 1: Suppose an initial state of the network is $\mathbf{v}^0$ and the desired state of the network is $\mathbf{v}^M$, find a control sequence $\langle \mathbf{u}(0), \mathbf{u}(1), \ldots, \mathbf{u}(M) \rangle$ such that $\mathbf{v}(0) = \mathbf{v}^0$ and $\mathbf{v}(M) = \mathbf{v}^M$, and the maximum number of controls applied to the network during the finite time period M is H.

C. Finding the Optimal Path with Hard Constraints

In a PBN, for each time step t, the network will choose one of the possible BNs (e.g., j_t-th possible BN) with the corresponding selecting probability q_{j_t} and enter into the next state $\mathbf{v}(t+1)$ from $\mathbf{v}(t)$. Given the initial state $\mathbf{v}^0$ and the desired state $\mathbf{v}^M$, we can define the probability of a path with $\mathbf{v}(0) = \mathbf{v}^0$ and $\mathbf{v}(M) = \mathbf{v}^M$ as $\prod_{t=0}^{M-1} q_{j_t}$. By applying external control to the network, we can derive the network into desired state $\mathbf{v}^M$ with different path probabilities. Then the problem of maximizing the highest probability of a path with the initial state $\mathbf{v}^0$ and the terminal (desired) state $\mathbf{v}^M$ can be described as follows:

Definition 2: Suppose an initial state of the network is $\mathbf{v}^0$ and the desired state of the network is $\mathbf{v}^M$, find a control sequence $\langle \mathbf{u}(0), \ldots, \mathbf{u}(M) \rangle$ such that the probability of the path with the initial state $\mathbf{v}^0$ and the terminal (desired) state $\mathbf{v}^M$ is maximized, and the maximum number of controls applied to the network during the finite time period M is H.

D. Minimizing the Maximum Cost

Suppose that a PBN with n internal nodes $\mathbf{v}(t) = [v_1(t), v_2(t), \ldots, v_n(t)]$ and m control nodes $\mathbf{u}(t) = [v_{n+1}(t), v_{n+2}(t), \ldots, v_{n+m}(t)]$. Let

$$z_t = 1 + \sum_{i=1}^{n} 2^{n-i} v_i$$

which is the state of network at time step t, and

$$u_t = 1 + \sum_{i=1}^{m} 2^{m-i} v_{n+i}$$

which is the control input of network at time step t. In a PBN, even if the network starts with the given initial state $z(0)$, the subsequent states will be random since the PBN is a stochastic model. That is, the terminal state z_M could take any possible values from 1 to 2^n. We assign a terminal cost $C_M(z_M)$ to each of states z_M at time step M. Note that, depending on the particular PBN and the control input used in each step, it is possible that the network can not enter some of the states at time step M. We define $C_t(z_t)$ as the maximum cost of which, beginning from z_t at time step t, the network can reach at the terminal time step. The problem of minimizing the maximum cost can be described as follows:

Definition 3: Given the terminal cost $C_M(z_M)$ for each of states $z_M \in \{1, 2, \ldots, 2^n\}$ at time step M, by applying external control, minimize the maximum cost $C_0(z_0)$ beginning from the given initial state z_0, and the maximum number of controls applied to the network is H.

III. Algorithms

A. ILP with Hard Constraints for BN CONTROL

In order to give ILP formalization for BN CONTROL, we first introduce several definitions.

Let $x_{i,t}$ represent the Boolean value $v_i(t)$. Define

$$\sigma_b(x) = \begin{cases} x, \text{ if } b = 1. \\ \bar{x}, \text{otherwise.} \end{cases} \quad (7)$$

Then any Boolean function $f_i(x_{i_1,t},\ldots,x_{i_k,t})$ is equivalent to

$$f_i(x_{i_1,t},\ldots,x_{i_k,t}) = \bigvee_{b_{i_1}\ldots b_{i_k}\in\{0,1\}^k}\{f_i(b_{i_1},\ldots,b_{i_k})\wedge \sigma_{b_1}(x_{i_1,t})\wedge\cdots\wedge\sigma_{b_k}(x_{i_k,t})\} \tag{8}$$

Then we define binary variable $h_{i,t}\in\{0,1\}$ ($i=n+1,\ldots,n+m$) as the node control variable. If $h_{i,t}=1$, we say the node i changes its value at time step t. Since the maximum number of controls applied to the network during the finite time period is H, we have

$$\sum_{t=0}^{M-1}\sum_{i=n+1}^{n+m} h_{i,t}\le H.$$

Also, we define $\tau_b(x)$ as

$$\tau_b(x)=\begin{cases}x, & \text{if } b=1.\\ 1-x, & \text{otherwise.}\end{cases} \tag{9}$$

Then the ILP-Formulation for the BN CONTROL based on the method of [5] is as follows:

Maximize

$$\sum_{i=1}^{N} x_{i,M}$$

S.T.

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}} \ge \sum_{j\in\{1,2,\ldots,k\}}\tau_{b_j}(x_{i_j,t})-(k-1) \tag{10}$$

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}} \le \frac{1}{k}\sum_{j\in\{1,2,\ldots,k\}}\tau_{b_j}(x_{i_j,t}) \tag{11}$$

for all $i\in\{1,2,\ldots,n\}$, $t\in\{1,2,\ldots,M-1\}$ and $b_{i_1}\ldots b_{i_k}\in\{0,1\}^k$ such that $f_i(b_{i_1},\ldots,b_{i_k})=1$.

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}}=0 \tag{12}$$

for all $i\in\{1,2,\ldots,n\}$, $t\in\{1,2,\ldots,M-1\}$, and $b_{i_1}\ldots b_{i_k}\in\{0,1\}^k$ such that $f_i(b_{i_1},\ldots,b_{i_k})=0$.

$$x_{i,t}\le\sum_{b_{i_1}\ldots b_{i_k}\in\{0,1\}^k} x_{i,t,b_{i_1}\ldots b_{i_k}} \tag{13}$$

$$x_{i,t}\ge\frac{1}{2^k}\sum_{b_{i_1}\ldots b_{i_k}\in\{0,1\}^k} x_{i,t,b_{i_1}\ldots b_{i_k}} \tag{14}$$

for all $i\in\{1,2,\ldots,n\}$ and $t\in\{1,2,\ldots,M\}$.

$$x_{i,0}=v_i^0,\quad x_{i,M}=v_i^M$$

$$x_{i,t}-x_{i,t+1}\le h_{i,t},\quad x_{i,t+1}-x_{i,t}\le h_{i,t}$$

for all $i\in\{n+1,\ldots,n+m\}$ and $t\in\{0,\ldots,M-1\}$.

$$\sum_{t=0}^{M-1}\sum_{i=n+1}^{n+m} h_{i,t}\le H$$

for all $i\in\{1,2,\ldots,n\}$ and $t\in\{1,2,\ldots,M\}$.

Note that $x_{i,t}$ and $x_{i,t,b_{i_1}\ldots b_{i_k}}$ and $h_{i,t}$ are binary values. Here (10) to (14) are to make sure that $x_{i,t+1}=f_i(x_{i_1,t},\ldots,x_{i_k,t})$ is satisfied for all $i\in\{1,2,\ldots,n\}$ and $t\in\{1,2,\ldots,M-1\}$.

B. ILP with Hard Constraints for PBN CONTROL

To extend the above ILP formulation for PBN CONTROL, we define $y_{r,t}$ as the selection variable. If $y_{r,t}=1$, we say the rth BN is selected at time step t. Otherwise, we say it is not selected at time step t. Then we have

$$\sum_{r=1}^{R} y_{r,t}=1,\ \text{for } t=1,2,\ldots,M-1$$

Here R is the total number of possible realizations for the PBN. Define $f_{i,r}$ as the Boolean function for node v_i when the r-th BN is selected. Let $P=(p_1,p_2,\ldots,p_R)$ be the selecting probabilities for the R possible realizations.

Then the objective function for the PBN control with hard constraints is as follows:

Maximize

$$\sum_{t=0}^{M-1}\sum_{r=1}^{R}-\log(p_r)\cdot y_{r,t}$$

For the constraints, we revise (10) to (12) as follows:

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}}\ge\sum_{j\in\{1,2,\ldots,k\}}\tau_{b_j}(x_{i_{j_r},t})-(k-1)+y_{r,t}-1 \tag{15}$$

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}}\le\frac{1}{k}\sum_{j\in\{1,2,\ldots,k\}}\tau_{b_j}(x_{i_{j_r},t})-y_{r,t}+1 \tag{16}$$

for all $i\in\{1,2,\ldots,n\}$, $t\in\{1,2,\ldots,M-1\}$, $b_{i_1}\ldots b_{i_k}\in\{0,1\}^k$ and $r\in\{1,2,\ldots,R\}$ such that $f_{i,r}(b_{i_1},\ldots,b_{i_k})=1$.

$$x_{i,t+1,b_{i_1}\ldots b_{i_k}}\le 1-y_{r,t} \tag{17}$$

for all $i\in\{1,2,\ldots,n\}$, $t\in\{1,2,\ldots,M-1\}$, $b_{i_1}\ldots b_{i_k}\in\{0,1\}^k$ and $r\in\{1,2,\ldots,R\}$ such that $f_{i,r}(b_{i_1},\ldots,b_{i_k})=0$.

Then we also add the following constraints:

$$\sum_{r=1}^{R} y_{r,t}=1,\ \text{for } t=1,2,\ldots,M-1,\ \text{and } y_{r,t}\in\{0,1\}.$$

for all $t\in\{1,2,\ldots,M\}$ and $r\in\{1,2,\ldots,R\}$.

The other constraints are the same as the ones in ILP-Formulation for the BN CONTROL.

Here (13) to (17) are to make sure that $x_{i,t}$ is regulated by the chosen BN and its corresponding Boolean functions for all $i\in\{1,2,\ldots,n\}$, $t\in\{1,2,\ldots,M-1\}$ and $r\in\{1,2,\ldots,R\}$.

C. Minimizing the Maximum Cost

Define $J(z_t, h_t)$ as the minimized maximum terminal cost $C_M(z_M)$ when the state is z_t, and the remaining number of external controls is h_t, at time step t. Define $u(z_t, h_t)$ as the control function when the state is z_t and the remaining number of external controls is h_t at time step t. Let $F(z_t, u_t)$ be the set of states at time step $t+1$ that can be reached from z_t with control u_t. Then dynamic programming for the PBN control with hard constraints is as follows:

Step 0: Set $t = M$; $J(z_M, h_M) = C_M(z_M)$ for all $h_M = \{0, \ldots, H\}$.

Step 1: $t := t - 1$.

Step 2: For any $z_t \in \{1, \ldots, 2^n\}$ and $h_t \in \{0, \ldots, H\}$, compute

$$J(z_t, h_t) = \min_{u_t \in \{1,\ldots,2^m\}}$$
$$\begin{cases} \max\limits_{z_{t+1} \in F(z_t,u_t)} J(z_{t+1}, h_t), \text{ if } u_t = u(z_{t+1}, h_t), \\ \max\limits_{z_{t+1} \in F(z_t,u_t)} J(z_{t+1}, h_t - 1), \text{ otherwise.} \end{cases}$$

and

$$u(z_t, h_t) = \text{argmin}_{u_t \in \{1,\ldots,2^m\}}$$
$$\begin{cases} \max\limits_{z_{t+1} \in F(z_t,u_t)} J(z_{t+1}, h_t), \text{ if } u_t = u(z_{t+1}, h_t), \\ \max\limits_{z_{t+1} \in F(z_t,u_t)} J(z_{t+1}, h_t - 1), \text{ otherwise.} \end{cases}$$

Step 3: If $t > 0$, go back to step 1; Otherwise, stop.

In the above, $u_t \neq u_{t+1}$ is counted as one control where we need to modify the algorithm for the case that the number of controls is defined as before. Finally, we take $\min_{h_0 \in \{0,\ldots,H\}} J(z_0, h_0)$ for computing the minimized maximum cost.[2] Though we do not consider costs of control operations in this paper, it is possible to extend the algorithm so that costs of control operations are taken into account.

IV. Complexity analysis

In this section, we give some analysis on the complexity of minimizing the maximum cost and minimizing the average cost. In order to analyze the time complexity, we assume that a PBN is not given in the matrix form A of (5) but in the form of $f_j^{(i)}$s and $c_j^{(i)}$s because A is of exponential size and thus it is almost meaningless to discuss the time complexity if we use A. Furthermore, we assume that it is only required to output $\mathbf{u}(0)$ for given z_0 and PBN (otherwise, we should output $\mathbf{u}(t)$s for an exponential number of GAPs). Then, we can keep both the sizes of input and output polynomial of n and thus can discuss the time complexity with respect to the network size. Furthermore, we assume that the number of time steps (i.e., M) is polynomially bounded. Otherwise, both BN CONTROL and PBN CONTROL would be PSPACE-hard [9], [12]. Since it is not realistic to consider an exponential number of time steps, this is a reasonable assumption.

For the complexity of minimizing the maximum cost, we have the following theorem by using a polynomial time reduction similar to that in [5].

Theorem 1: Minimizing the maximum cost in control of PBN is $\sum_2^p$-hard.

Proof: As in [5], we use a polynomial time reduction from the quantified Boolean formula problem for 3-DNF (disjunction of conjunctions of which consisting 3 literals). Let $\psi(x, y)$ be a 3-DNF over variables $x = (x_1, \ldots, x_{m_1})$ and $y = (y_1, \ldots, y_{n_1})$. Then it is known that deciding whether or not $(\exists x)(\forall y)\psi(x, y)$ is true is $\sum_2^p$-complete [10], [13]. We show a polynomial time reduction from this problem to minimization of the maximum cost.

From a given $\psi(x, y)$, we construct a PBN as follows. Let n_2 be the number of terms in $\psi(x, y)$. Then we let $V = \{v_1, v_2, \ldots, v_{n_1+n_2+2m_1+1}\}$. For $i = 1, \ldots, n_1$, v_i corresponds to y_i. For $i = 1, \ldots, n_2$, v_{n_1+i} corresponds to the ith term of $\psi(x, y)$ where the ith term is represented as $l_{i_1} \wedge l_{i_2} \wedge l_{i_3}$. For $i = 1, \ldots, m_1$, both $v_{n_1+n_2+1+i}$ and $v_{n_1+n_2+m_1+1+i}$ correspond to x_i. Then, we assign the following Boolean functions to V:

$$\begin{aligned}
&v_i(t+1) = v_i(t) \text{ with probability } 0.5, \text{ for } i = 1, \ldots, n_1, \\
&v_i(t+1) = \bar{v}_i(t) \text{ with probability } 0.5, \text{ for } i = 1, \ldots, n_1, \\
&v_{n_1+i}(t+1) = l_{i_1} \wedge l_{i_2} \wedge l_{i_3}, \text{ for } i = 1, \ldots, n_2, \\
&v_{n_1+n_2+1+i}(t+1) = v_{n_1+n_2+m_1+1+i}(t), \\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for } i = 1, \ldots, m_1, \\
&v_{n_1+n_2+1}(t+1) = \textstyle\bigvee_{i \in \{1,\ldots,n_2\}} v_{n_1+i}(t),
\end{aligned} \tag{18}$$

where we identify $v_i(t)$ with y_i and $v_{n_1+n_2+1+i}(t)$ with x_i in $l_{i_1} \wedge l_{i_2} \wedge l_{i_3}$. Finally, we let $n = n_1 + n_2 + m_1 + 1$, $m = m_1$, $\mathbf{v}(0) = (0, 0, \ldots, 0)$, and $M = 3$.[3] The cost is given by $J = 0$ if $v_{n_1+n_2+1}(M) = 1$. Otherwise $J = 1$.

Then we see that the minimum of the maximum cost is 0 iff $(\exists x)(\forall y)\psi(x, y)$ is true. First, suppose $(\exists x)(\forall y)\psi(x, y)$ is true for an assignment of $x = (b_1, b_2, \ldots, b_{m_1})$. Then, it is straight-forward to see that the maximum cost is 0 by the control input $(v_{n_1+n_2+m_1+2}(0), \ldots, v_{n_1+n_2+2m_1+1}(0)) = (b_1, b_2, \ldots, b_{m_1})$.

Next, suppose that the minimum of the maximum cost is 0. Since the minimum cost J is only determined by $v_{n_1+n_2+1}(M)$ and $J = 0$ must hold, $v_{n_1+n_2+1}(M) = 1$ holds for any assignment on $v_1(1), \ldots, v_{n_1}(1)$. Therefore, $(\exists x)(\forall y)\psi(x, y)$ is true. ■

We modify the proof of Theorem 1 for proving Σ_2^p-hardness of the original control problem of PBN (i.e., minimizing the average cost). For that purpose, we use the same reduction as in the proof of Theorem 1. Then, we note

[2] We need to modify the algorithm if there exist multiple u_ts giving the minimum cost.

[3] It is to be noted that our hardness result holds for the case of $M = 3$. The reductions in [9], [12] are not directly applicable to such a case (i.e., small M).

Table I
THE RESULTS OF EXPERIMENT 1

	1	2	3	4	5	Average time
ILP method	0.38	4.22	3.66	0.69	6.54	3.10
DP method	20.81	5.93	5.75	7.77	9.53	9.96

Table II
THE RESULTS OF EXPERIMENT 2

	1	2	3	4	5
Maximum cost without control	245	252	249	244	239
Minimized maximum cost	193	194	177	173	163

that for any x, the average cost is 0 iff $(\forall y)\psi(x,y)$ is true (i.e., the average cost is greater than 0 iff $\psi(x,y)$ is false for some y). By utilizing this observation, we can prove the following:

Theorem 2: Minimizing the average cost in control of PBN is $\sum_2^p$-hard.

V. COMPUTATIONAL EXPERIMENTS

In this section, we give some computational experiments to demonstrate the effectiveness of our algorithms. Computational experiments were done on a PC with a Intel core 2 Duo CPU (p8600 2.4GHz) and 2GB RAM running under the Windows XP professional 2002 operating system. For solving integer linear programs, we used ILOG CPLEX (version 11.2,http://www.ilog.com/products/cplex/) with a Xeon 5470 3.33GHz CPU and 10GB RAM running under the LINUX (version 2.6.16) operating system.

A. Experiment 1: Comparison of ILP and DP for the Optimal Path Problem

For comparison of ILP and DP methods, we consider a 10-gene example with $K = 3$, where genes $v_1, \ldots, v_8$ are internal nodes and genes u_1, u_2 are control nodes. The PBN is consisting of 8 possible BNs with the selecting probabilities given by

$$\mathbf{q} = (0.1, 0.1, 0.3, 0.05, 0.05, 0.2, 0.15, 0.05).$$

Let $M = 5$ and $H = 3$. Suppose the initial sate is $00\ldots0$ and the desired state is $11\ldots1$. We conducted the experiments five times with different sets of Boolean functions which were randomly generated. By using the ILP method given in Section III-B and the revised DP method for finding the optimal path, we got the same results. Table I gives the CPU time (sec) for both methods. From the table, we see that the ILP method is faster than the DP method though PCs used for ILP and DP are not the same.

B. Experiment 2: Minimizing the Maximum Cost

For evaluating the DP method presented in Section III-C, we apply the DP method to minimize the maximum cost for a 10-gene example with 8 internal nodes and 2 control nodes. The PBN is consisting of 4 possible BNs and the Boolean functions are randomly generated. Suppose the initial state is $(0, 0, \ldots, 0)$, and the terminal cost is given by $C_M(z_M) = z_M$. Let $M = 3$ and $H = 2$. We conducted the experiments five times. The results are given in Table II.

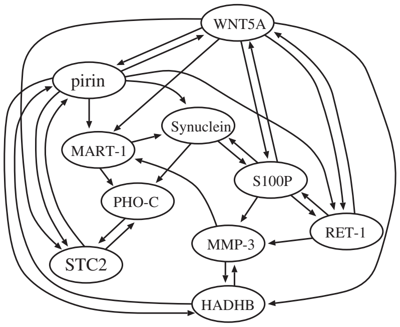


Figure 3. Structure of 10-gene WNT5A network [14].

C. Experiment 3: On Realistic Network

We also conducted an experiment based on a 10-gene WNT5A network (see Fig. 3, which is extracted from [14]). There are 10 nodes in this network, we assume WNT5A, pirin, MART-1, synnclein, PHO-C, S100P, STC2 and MMP-3 are internal nodes, and RET-1 and HADHB are control nodes. The initial state is $(0, 0, \ldots, 0)$, and the desired state is $(1, 1, \ldots, 1)$. Let $M = 10$ and $H = 3$. The Boolean functions are randomly generated and the PBN is consisting of eight possible BNs with the following selection probabilities:

$$\mathbf{q} = (0.0011491, 0.2165696, 0.2018161, 0.0740315,$$
$$0.0587344, 0.0304030, 0.1348942, 0.2824019).$$

By using ILP with hard constraints, the probability of optimal path is 1.2306×10^{-8}. The external control is applied three times to the network.

Then we consider minimizing the maximum cost for the initial state $(0, 0, \ldots, 0)$. Also, the terminal cost is given by $C_M(z_M) = z_M$. Let $M = 3$ and $H = 2$. Without applying external control, the maximum cost was 256. By using DP method, the maximum cost was reduced to 172.

VI. CONCLUSION

In this paper, we have presented ILP-based methods for control of BN and for finding an optimal path for PBN both with hard constraints. We have also presented a DP-based method for finding a control policy that minimizes the maximum cost for PBN under hard constraints, where it uses exponential size tables. The results of preliminary computational experiments suggest that for finding an optimal path for PBN, the ILP-based method is faster than the DP-based method (specialized for this problem). However, the hardness results presented in this paper suggest that

ILP cannot be effectively applied to minimization of the maximum or average cost for PBN.

Acknowledgment

The work of TA was partially supported by MEXT Grant-in-Aid No. 22240009. The work of WC was partially supported by HKGRF Grant No. 7017/07P and HKU CRCG Grants and HKU strategic theme grant on computational sciences.

References

[1] W. Ching, E. Fung, M. Ng, and T. Akutsu, "On construction of stochastic genetic networks based on gene expression sequences," *Journal of Neural Systems*, vol. 15, pp. 297–310, 2005.

[2] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, pp. 69–103, 2002.

[3] S. A. Kauffman, *The Origins of Order: Self-organization and Selection in Evolution*. NY: Oxford Univ. Press, 1993.

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

[5] T. Akutsu, M. Hayashida, and T. Tamura, "Integer programming-based methods for attractor detection and control of Boolean networks," in *Proc. the Combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, 2009, pp. 5610–5617.

[6] W. Ching, S. Zhang, Y. Jiao, T. Akutsu, and N. Tsing, "Optimal control policy for probabilistic Boolean networks with hard constraints," *IET Systems Biology*, vol. 3, pp. 90–99, 2009.

[7] K. Kobayashi and K. Hiraishi, "An integer programming approach to control problems in probabilistic boolean networks," in *Proc. 2010 American Control Conference*, 2010, pp. 6710–6715.

[8] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, pp. 169–191, 2003.

[9] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and R. E. Stearns, "Complexity of reachability problems for finite discrete dynamical systems," *Journal of Computer and System Sciences*, vol. 72, pp. 1317–1345, 2006.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. NY: W. H. Freeman and Company, 1979.

[11] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Machine Learning*, vol. 80, pp. 2219–2235, 2000.

[12] C. L. Barrett, H. B. Hunt III, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, R. E. Stearns, and M. Thakur, "Computational aspects of analyzing social network dynamics," in *Proc. 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 2268–2273.

[13] L. J. Stockmeyer, "The polynomial-time hierarchy," *Theoretical Computer Science*, vol. 3, pp. 1–22, 1976.

[14] S. Kim, H. Li, E. R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E. Suh, "Can Markov chain models mimic biological regulations?" *Journal of Biological Systems*, vol. 10, pp. 337–357, 2002.