IEICE TRANS. INF. & SYST., VOL.E95–D, NO.12 DECEMBER 2012

PAPER

Integer Programming-Based Approach to Attractor Detection and Control of Boolean Networks

Tatsuya AKUTSU[†a)], ***Member***, **Yang ZHAO**[†], ***Nonmember***, **Morihiro HAYASHIDA**[†], ***and*** **Takeyuki TAMURA**[†], ***Members***

SUMMARY The Boolean network (BN) can be used to create discrete mathematical models of gene regulatory networks. In this paper, we consider three problems on BNs that are known to be NP-hard: detection of a singleton attractor, finding a control strategy that shifts a BN from a given initial state to the desired state, and control of attractors. We propose integer programming-based methods which solve these problems in a unified manner. Then, we present results of computational experiments which suggest that the proposed methods are useful for solving moderate size instances of these problems. We also show that control of attractors is Σ_2^p-hard, which suggests that control of attractors is harder than the other two problems.

key words: *Boolean networks, genetic networks, attractors, integer programming, nonlinear discrete systems*

1. Introduction

Many kinds of genes, proteins and molecules interact with each other in living cells. Among the various kinds of interactions, those between genes play an important role in living cells. Sets of these interactions are often modeled as *gene regulatory networks* (*genetic networks*, in short), and an understanding of genetic networks has become one of the key topics in systems biology. Furthermore, development of methods for control of genetic networks is becoming important because of its potential applications to drug discovery and treatment of intractable diseases [21].

In order to develop control methods for genetic networks, we need a mathematical model of genetic networks. Furthermore, since genetic networks contain highly non-linear components, we need a non-linear model. The *Boolean network* (BN, in short) is one of the well studied non-linear models of genetic networks [19]. BN is a very simple model: each node (e.g., gene) takes either 0 (inactive) or 1 (active) and the states of nodes change according to regulation rules given as Boolean functions. Two types of BNs have been mainly studied: *synchronous BNs* and *asynchronous BNs*, depending on whether or not the states of nodes are updated synchronously. Though asynchronous BNs might be more appropriate as a model of genetic networks, we focus on synchronous BNs in this paper because previous control studies on BNs focused on synchronous BNs and detection of a singleton attractor is equivalent in both models. Though synchronous BNs might be too simple as a model of genetic networks, they are useful for modeling and analyzing certain properties of real genetic networks. For example, these were applied to analysis of *D. melanogaster* embryo development [24], and analysis of robustness of genetic (sub)networks of *S. cerevisiae*, *E. coli*, *B. subtilis*, *D. melanogaster* and *A. thaliana* [5].

Extensive studies have been done on the distribution of *attractors* in randomly generated BNs [19], [28] mainly because it is considered that different attractors correspond to different cell types [19]. Recently, several methods have been developed for efficiently finding and/or enumerating attractors in BNs [10]–[12], [14], [17], [32], whereas it is known that finding a singleton attractor (i.e., a steady state) is NP-hard [1]. Devloo et al. developed a method using transformation to a constraint satisfaction problem [11]. Garg et al. developed a method based on Binary Decision Diagrams (BDDs) [14]. Irons developed a method that makes use of small subnetworks [17]. Dubrova and Tesienko developed a method to enumerate all attractors using a solver for the Boolean satisfiability (SAT) problem [12]. De Jong and Page also developed a SAT-based method for finding steady states in piecewise-linear differential equation models for genetic networks [10], which may also be applied to attractor detection for BNs. Inoue analyzed theoretical relationships between BNs and logic programs and suggested the use of SAT solvers for detecting attractors [16]. However, theoretical analysis of the average case or worst case complexity was not performed in these studies. Zhang et al. developed algorithms for enumerating singleton attractors and small attractors and analyzed the average case time complexities of these algorithms [32]. Akutsu et al., Melkman et al., and Tamura et al. developed algorithms with guaranteed worst case time complexities for detection of singleton attractors for BNs with restricted Boolean functions [4], [27], [31].

For *control of BNs*, not many studies have been done. In order to avoid BNs from falling into chaotic states, Luque and Solé studied effects of periodic controls [26]. However, their methods cannot be applied to detailed control of BNs. Inspired from works on control of the probabilistic Boolean network [9], Akutsu et al. studied control of BNs [2]. They formalized control of a BN as a problem of finding 0-1 sequences to external nodes which lead a BN from a given initial state to the desired state [2]. They showed that this control problem is NP-hard in general but can be solved

Manuscript received October 13, 2011.
Manuscript revised May 1, 2012.
[†]The authors are with Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji-shi, 611–0011 Japan.
a) E-mail: takutsu@kuicr.kyoto-u.ac.jp
DOI: 10.1587/transinf.E95.D.2960

in polynomial time if BNs have tree-structures. Langmead and Jha developed a method based on model checking and successfully applied the method to find control policies for an existing model of fruit fly embryo development [24]. Though their method might be useful, it seems not easy to incorporate scores into their model. Cheng and his colleagues have recently been extensively studying control and related problems by utilizing the concept of semi-tensor product [7], [8]. Though their theory is deep and might be useful, it is difficult to apply their current methods to large-scale BNs.

On the other hand, it is not easy to perform detailed controls, or to give the target state (i.e., desired state) precisely and/or uniquely. Therefore, Hayashida et al. introduced the problem of *control of attractors* in BNs [15]. In this problem, some of genes are selected and controlled (maybe by gene disruption and/or overexpression) so that the minimum score of the attractor states is maximized, where a score function, which measures the quality of global states, is given in advance. Though we developed several recursive algorithms for the problem based on the methodology in [32], it is still difficult to solve the problem for non-small BNs [15].

In this paper, we propose *integer programming*-based methods which solve the above three problems on BNs in a unified manner. In each case, an instance of the original problem is transformed into *integer linear program(s)* (ILP(s)) and then an existing solver is applied. One of the advantages of the proposed approach is its simplicity and flexibility because ILP allows us to add various constraints. To be shown in the following sections, these problems are transformed into ILPs in similar and systematic ways. In order to study the efficiency of the proposed methods, we performed computational experiments by mainly using artificially generated BNs. The results suggest that the proposed methods can be applied to medium-size BNs. Though ILP is applied to all three methods, it is not directly applicable to control of attractors. In order to reveal the reason, we show that control of attractors is Σ_2^p-hard, which may suggest that control of attractors is computationally harder than the other two problems and ILP.

It is to be noted that Chen et al., and Kobayashi and Hiraishi recently proposed ILP-based methods for control of probabilistic Boolean networks (PBNs) [6], [22], [23], where PBN is a probabilistic extension of BN. There are some similarities between our proposed method and their methods. However, a preliminary conference version of this paper [3] appeared earlier than their papers [6], [22], [23], and they did not treat attractor detection or control of attractors. It should also be noted that this version was considerably enhanced from [3]: additional heuristic is introduced to improve the efficiency for the case of K = 3, more detailed computational experiments are performed, and a proof of Theorem 1 is given.

2. Problems

In this section, we briefly review BN [19] and then define three problems: ATTRACTOR DETECTION [32], BN CONTROL [2], and ATTRACTOR CONTROL [15].

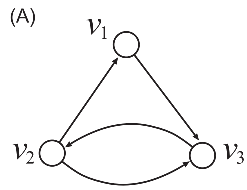
2.1 Boolean Network

A *Boolean network* $G(V, F)$ consists of a set $V = \{v_1, \ldots, v_n\}$ of *nodes* and a list $F = (f_1, \ldots, f_n)$ of *Boolean functions*, 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 each node v_i. We use $x \wedge y$, $x \vee y$, $x \oplus y$, $\overline{x}$ to denote logical AND of x and y, logical OR of x and y, exclusive OR of x and y, and logical NOT of x, respectively. We use $IN(v_i)$ to denote the set of input nodes $v_{i_1}, \ldots, v_{i_k}$ to v_i. Each node takes either 0 or 1 at each discrete time t, and the state of node v_i at time t is denoted by $v_i(t)$. Then, the state of node v_i at time $t+1$ is determined by $v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_{k_i}}(t))$. Here we let $\mathbf{v}(t) = [v_1(t), \ldots, v_n(t)]$, which is called a *Gene Activity Profile* (GAP) at time t. We also write $v_i(t+1) = f_i(\mathbf{v}(t))$ to denote the regulation rule for v_i and $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t))$ to denote the regulation rule for the whole BN.

We define the set of *edges* E by $E = \{(v_{i_j}, v_i) \mid v_{i_j} \in IN(v_i)\}$. Then, $G(V, E)$ is a directed graph representing the network topology of a BN. It is worthy to mention that an edge from v_{i_j} to v_i means that v_{i_j} directly affects expression of v_i. The number of input nodes to v_i is called the *indegree* of v_i. We use K to denote the *maximum indegree* of a BN, which greatly affects the computation time.

An example of a BN is given in Fig. 1. Dynamics of a BN is well-described by a *state transition table* and a *state transition diagram* shown in Fig. 1. For example, the fourth row of the table means that if the state of BN is $[0, 1, 1]$ at time t then the state will be $[0, 1, 0]$ at time $t+1$, and the arc from 011 to 010 in the diagram means that if the state of BN

(A)



$v_1(t+1) = \overline{v_2(t)}$
$v_2(t+1) = v_3(t)$
$v_3(t+1) = v_1(t) \wedge v_2(t)$

(B)

time t			time $t+1$		
v_1	v_2	v_3	v_1	v_2	v_3
0	0	0	1	0	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	0	1	1

(C)

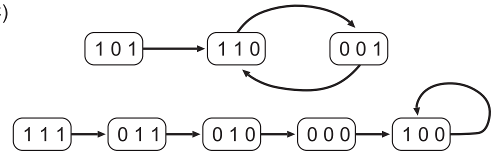


Fig. 1 Example of a Boolean network. Dynamics of BN (A) is well-described by a state transition table (B) and by a state transition diagram (C).

is [0, 1, 1] at time t the state will be [0, 1, 0] at time $t + 1$. It is easily seen that for a BN with n nodes, the state transition table consists of 2^n rows and the state transition diagram has 2^n vertices.

2.2 Detection of an Attractor

Since $\mathbf{v}(t+1)$ is determined from $\mathbf{v}(t)$ in a BN, starting from an initial GAP $\mathbf{v}(0)$, a BN will eventually reach a set of global states (i.e., a directed cycle in the state transition diagram). This set is called an *attractor*. An attractor consisting of only one global state (i.e., $\mathbf{v} = \mathbf{f}(\mathbf{v})$) is called a *singleton attractor*, which corresponds to a fixed point. Otherwise, it is called a *cyclic attractor* with period p if it consists of p global states (i.e., $\mathbf{v}^1 = \mathbf{f}(\mathbf{v}^p) = \mathbf{f}(\mathbf{f}(\mathbf{v}^{p-1})) = \cdots = \mathbf{f}(\mathbf{f}(\cdots \mathbf{f}(\mathbf{v}^1)\cdots)))$. For example, in Fig. 1, 100 is a singleton attractor, and {110, 001} is a cyclic attractor of period 2.

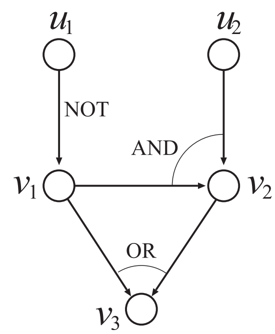
In this paper, the attractor detection problem is defined as a problem of finding an attractors for a given BN. However, it is very difficult to find attractors with long periods. Thus, we consider detection of attractors with period at most some given threshold p_{max}. This problem is defined as below, where we focus on the case of $p_{max} = 1$ (i.e., the *singleton attractor detection problem*) in this paper.

Definition 1: [ATTRACTOR DETECTION]
Instance: a BN and the maximum period p_{max},
Problem: find an attractor with period at most p_{max}. If there does not exist such an attractor, "None" should be the output.

2.3 Control of Boolean Networks

Akutsu et al. introduced a control problem for BNs (BN CONTROL) [2], inspired from the control problem for probabilistic Boolean networks [9]. In BN CONTROL, it is assumed that there exist two types of nodes: *internal nodes* and *external nodes*, where internal nodes correspond to usual nodes (i.e., genes) in a BN and external nodes correspond to control nodes. Let $V = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_{n+m}\}$ be a set of nodes, where $v_1, \ldots, v_n$ are internal nodes and $v_{n+1}, \ldots, v_{n+m}$ are external nodes. For convenience, we use u_i to denote an external node v_{n+i}. Then, states of internal nodes ($v_i(t+1)$ for $i = 1, \ldots, n$) are determined by $v_i(t+1) = f_i(v_{i_1}(t), \ldots, v_{i_{k_i}}(t))$, where each v_{i_k} is either an internal node or an external node. Here, we let $\mathbf{v}(t) = [v_1(t), \ldots, v_n(t)]$ and $\mathbf{u}(t) = [u_1(t), \ldots, u_m(t)]$. We can describe the state transition rule of a BN by $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t), \mathbf{u}(t))$, where $\mathbf{u}(t)$s are determined externally. Then, BN CONTROL is defined as follows (see also Fig. 2).

Definition 2: [BN CONTROL]
Instance: a BN, an initial state of the network for internal nodes $\mathbf{v}^0$, and the desired state of the network for internal nodes $\mathbf{v}^M$ at the M-th time step,
Problem: find a sequence of 0-1 vectors $\langle \mathbf{u}(0), \ldots, \mathbf{u}(M) \rangle$ such that $\mathbf{v}(0) = \mathbf{v}^0$ and $\mathbf{v}(M) = \mathbf{v}^M$. If there does not exist such a sequence, "None" should be the output.



initial (t=0)	0	0	0
desired (t=3)	0	1	1

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

Fig. 2 Example of Control of BN. In this problem, given initial and desired states of internal nodes (v_1, v_2, v_3), it is required to compute a sequence of states of external nodes (u_1, u_2) leading to the desired state.

2.4 Control of Attractors

Hayashida et al. [15] proposed a control problem on attractors in BN. In their definition, external nodes and internal nodes are assumed to be given in advance. However, it might be better to assume that it is not determined in advance which nodes are external nodes. The reason is that recent developments of iPS cells (induced pluripotent stem cells) were achieved by enforcing activation of some genes [30]. Based on this idea, we will define an attractor control problem.

Since m control nodes are selected among the original n nodes, we need to modify the definition of singleton attractors. Suppose that $v_{i_1}, \ldots, v_{i_m}$ are selected as control nodes and Boolean values of $b_{i_1}, \ldots, b_{i_m}$ are assigned to these nodes. Then, $\mathbf{v}$ is called a singleton attractor (in ATTRACTOR CONTROL) if the following conditions (denoted by COND1) are satisfied:

- $v_i = b_i$ if $i \in \{i_1, \ldots, i_m\}$,
- $v_i = f_i(\mathbf{v})$ otherwise.

In addition, we need a score function g from $\{0, 1\}^n$ to the set of real numbers for evaluating how appropriate each attractor state is. Though there is no established score function, we assume for simplicity that g is given as a linear combination of 0-1 values of internal nodes:

$$g(\mathbf{v}) = \sum_{i=1,\ldots,n} \alpha_i \cdot (1 - w_i) \cdot v_i,$$

where α_i are real constants. Giving a large value to α_i means that the gene corresponding to v_i should be expressed in a desired state. If node v_i is selected as a control node, w_i is 1. Otherwise, w_i is 0. This means that the scores of selected control nodes are not taken into account for $g(\mathbf{v})$. Since singleton attractors need not be uniquely determined, we need to maximize the minimum score of singleton attractors, considering the worst case. However, as mentioned later, it is not easy to maximize the minimum score. Thus, by introducing a threshold Θ of the minimum score, we define the problem of control of attractors (ATTRACTOR CONTROL) as follows.

Definition 3: [ATTRACTOR CONTROL]
Instance: a BN, a score function g, the number of control nodes m, and a threshold Θ,
Problem: find m nodes and a 0-1 assignment to these control nodes for which the minimum score of singleton attractors is no less than Θ. If there do not exist such nodes, "None" should be the output.

Assume that a BN in Fig. 1 is given as an instance of ATTRACTOR CONTROL along with $\alpha_1 = 2$, $\alpha_2 = 1$, $\alpha_3 = 1$, and $m = 1$. If we assign 1 to v_1, there exist two singleton attractors 100 and 111. Since the scores of 100 and 111 are 0 and 2 respectively, the minimum score is 0. If we assign 1 to v_3, there exists one singleton attractor 011 with score 1. If we assign 0 to v_2, there exists one singleton attractor 100 with score 2. With considering the other three cases, we can see that the last case (i.e., assigning 0 to v_2) gives the solution for this instance.

3. Integer Programming-Based Methods

In this section, we present integer programming-based methods for ATTRACTOR DETECTION, BN CONTROL, and ATTRACTOR CONTROL. Integer programming, especially, *integer linear programming* (ILP) is to maximize (or minimize) a linear objective function under linear constraints (i.e., linear equalities and linear inequalities) and the condition that specified variables must take integer values. Since ILP has been widely used for solving various types of NP-hard problems [18] and all the problems in this paper are NP-hard, it is reasonable to try to apply ILP to solving these problems. In the following, each variable takes either 0 or 1 and we identify integer values with Boolean values. Moreover, we use x_i or similar variables to denote a 0-1 variable corresponding to v_i.

3.1 ILP for ATTRACTOR DETECTION

In order to give ILP formalization for ATTRACTOR DETECTION, we need several definitions, many of which are also used in BN CONTROL and ATTRACTOR CONTROL.

We define $\sigma_b(x)$ by $\sigma_b(x) = \begin{cases} x, & \text{if } b = 1, \\ \overline{x}, & \text{otherwise.} \end{cases}$ We note that any Boolean function with k inputs can be represented as

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

We define $\tau_b(x)$ by $\tau_b(x) = \begin{cases} x, & \text{if } b = 1, \\ 1 - x, & \text{otherwise.} \end{cases}$ If $f_i(b_{i_1}, \dots, b_{i_k}) = 1$, we add constraints

$$x_{i,b_{i_1} \dots b_{i_k}} \geq \left(\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j}) \right) - (k-1),$$

$$x_{i,b_{i_1} \dots b_{i_k}} \leq \frac{1}{k} \sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j}),$$

where the first constraint forces $x_{i,b_{i_1} \dots b_{i_k}}$ to be 1 if $\sigma_{b_1}(x_{i_1}) \wedge \cdots \wedge \sigma_{b_k}(x_{i_k})$ is satisfied, and the latter forces $x_{i,b_{i_1} \dots b_{i_k}}$ to be 0 if it is not satisfied. If $f_i(b_{i_1}, \dots, b_{i_k}) = 0$, we simply add a constraint $x_{i,b_{i_1} \dots b_{i_k}} = 0$. These constraints ensure that $x_{i,b_{i_1} \dots b_{i_k}} = 1$ if and only if $f_i(b_{i_1}, \dots, b_{i_k}) \wedge \sigma_{b_1}(x_{i_1}) \wedge \cdots \wedge \sigma_{b_k}(x_{i_k}) = 1$. Finally, for each x_i, we add constraints

$$x_i \leq \sum_{b_{i_1} \dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1} \dots b_{i_k}},$$

$$x_i \geq \frac{1}{2^k} \sum_{b_{i_1} \dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1} \dots b_{i_k}}.$$

These constraints ensure that $x_i = f_i(x_{i_1}, \dots, x_{i_k})$ holds for every x_i, which means that any obtained feasible solution corresponds to a singleton attractor.

Here, we show a simple example. Suppose that x_3 is determined by $x_3 = f_3(x_1, x_2) = x_1 \oplus x_2$. Then, $f_3(x_1, x_2)$ can be represented as

$$\begin{aligned} f_3(x_1, x_2) &= (f_3(0,0) \wedge \overline{x_1} \wedge \overline{x_2}) \vee (f_3(0,1) \wedge \overline{x_1} \wedge x_2) \\ &\quad \vee (f_3(1,0) \wedge x_1 \wedge \overline{x_2}) \vee (f_3(1,1) \wedge x_1 \wedge x_2) \\ &= (\overline{x_1} \wedge x_2) \vee (x_1 \wedge \overline{x_2}). \end{aligned}$$

Then, this Boolean formula is transformed into the following inequalities

$$\begin{aligned} x_{3,00} &= 0, \\ x_{3,01} &\geq (1 - x_1) + x_2 - 1 = x_2 - x_1, \\ x_{3,01} &\leq \frac{1}{2}(1 - x_1 + x_2), \\ x_{3,10} &\geq x_1 + (1 - x_2) - 1 = x_1 - x_2, \\ x_{3,10} &\leq \frac{1}{2}(x_1 + 1 - x_2), \\ x_{3,11} &= 0, \\ x_3 &\leq x_{3,00} + x_{3,01} + x_{3,10} + x_{3,11}, \\ x_3 &\geq \frac{1}{4}(x_{3,00} + x_{3,01} + x_{3,10} + x_{3,11}). \end{aligned}$$

The correctness of the transformation is easily seen from this example.

By putting together all the constraints, the singleton attractor detection problem can be transformed into the following ILP.

Maximize x_1,
Subject to

$x_{i,b_{i_1} \dots b_{i_k}} \geq \left(\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j}) \right) - (k-1)$,
$x_{i,b_{i_1} \dots b_{i_k}} \leq \frac{1}{k} \sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j})$,
for all $i \in [1 \dots n]$ and $b_{i_1} \dots b_{i_k} \in \{0,1\}^k$
such that $f_i(b_{i_1}, \dots, b_{i_k}) = 1$,
$x_{i,b_{i_1} \dots b_{i_k}} = 0$,
for all $i \in [1 \dots n]$ and $b_{i_1} \dots b_{i_k} \in \{0,1\}^k$
such that $f_i(b_{i_1}, \dots, b_{i_k}) = 0$,
$x_i \leq \sum_{b_{i_1} \dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1} \dots b_{i_k}}$,

$$
\begin{aligned}
& x_i \geq \tfrac{1}{2^k} \textstyle\sum_{b_{i_1}\dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\dots b_{i_k}}, \\
& x_i \in \{0,1\}, \quad \text{for all } i \in [1\dots n], \\
& x_{i,b_{i_1}\dots b_{i_k}} \in \{0,1\}, \\
& \qquad\qquad \text{for all } i \in [1\dots n] \text{ and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k.
\end{aligned}
$$

It is to be noted that we do not need an objective function since ATTRACTOR DETECTION is not an optimization problem but a decision problem. Since some objective function is required in order to use ILP, we simply used 'maximize x_1'.

It is possible to extend the above mentioned method for detection of a cyclic attractor with period at most p_{max}. However, for that purpose, we need to introduce much more variables $x_{i,t}$, $x_{i,t,b_{i_1}\dots b_{i_k}}$ for $t = 0, \dots, p_{max} - 1$, as in ILP for BN CONTROL.

3.2 ILP for BN CONTROL

In order to solve BN CONTROL using ILP, we need to introduce the notion of time. For that purpose, we introduce integer variables $x_{i,t}$ to represent Boolean values $v_i(t)$. Moreover, we use $x_{i,t,b_1\dots b_k}$ which corresponds to $x_{i,b_1\dots b_k}$ in ATTRACTOR DETECTION. Then, based on ILP-formulation for ATTRACTOR DETECTION, we have the following ILP-formulation for BN CONTROL.

$$
\begin{aligned}
& \text{Maximize } \textstyle\sum_{i=1,\dots,n} x_{i,M}, \\
& \text{Subject to} \\
& \quad x_{i,t+1,b_{i_1}\dots b_{i_k}} \geq \left(\textstyle\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j,t})\right) - (k-1), \\
& \quad x_{i,t+1,b_{i_1}\dots b_{i_k}} \leq \tfrac{1}{k} \textstyle\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j,t}), \\
& \qquad\qquad \text{for all } i \in [1\dots n],\ t \in [0\dots M-1] \\
& \qquad\qquad \text{and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k \\
& \qquad\qquad \text{such that } f_i(b_{i_1},\dots,b_{i_k}) = 1, \\
& \quad x_{i,t+1,b_{i_1}\dots b_{i_k}} = 0, \\
& \qquad\qquad \text{for all } i \in [1\dots n],\ t \in [0\dots M-1] \\
& \qquad\qquad \text{and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k \\
& \qquad\qquad \text{such that } f_i(b_{i_1},\dots,b_{i_k}) = 0, \\
& \quad x_{i,t} \leq \textstyle\sum_{b_{i_1}\dots b_{i_k} \in \{0,1\}^k} x_{i,t,b_{i_1}\dots b_{i_k}}, \\
& \quad x_{i,t} \geq \tfrac{1}{2^k} \textstyle\sum_{b_{i_1}\dots b_{i_k} \in \{0,1\}^k} x_{i,t,b_{i_1}\dots b_{i_k}}, \\
& \qquad\qquad \text{for all } i \in [1\dots n] \text{ and } t \in [1\dots M], \\
& \quad x_{i,0} = \mathbf{v}_i^0,\ x_{i,M} = \mathbf{v}_i^M, \qquad \text{for all } i \in [1\dots n], \\
& \quad x_{i,t} \in \{0,1\}, \\
& \qquad\qquad \text{for all } i \in [1\dots n+m] \text{ and } t \in [0\dots M], \\
& \quad x_{i,t,b_{i_1}\dots b_{i_k}} \in \{0,1\}, \\
& \qquad\qquad \text{for all } i \in [1\dots n],\ t \in [1\dots M] \\
& \qquad\qquad \text{and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k.
\end{aligned}
$$

It is to be noted that as in ATTRACTOR DETECTION, we do not need an objective function and thus 'maximize ...' is used as a dummy function. However, it might be more useful in some cases if we define BN CONTROL as an optimization problem by introducing some score for GAP at the target time step. In such a case, we simply remove constraints of $x_{i,M} = \mathbf{v}_i^M$ and instead use an appropriate objective function though it must be a linear combination of variables.

3.3 ILP for ATTRACTOR CONTROL

In ATTRACTOR CONTROL, differently from the above formulations, we need to consider the following two possibilities for each variable v_i:

- v_i is selected as a control node (i.e., v_i corresponds to an external node),
- v_i is not selected as a control node (i.e., v_i becomes an internal node).

In order to choose one from these two possibilities, we introduce additional variables and constraints. Let x_i be given by $x_i = \begin{cases} y_i & \text{if } w_i = 0, \\ z_i & \text{if } w_i = 1. \end{cases}$ This function can be represented by

$$
\begin{aligned}
y_i - w_i \leq\ & x_i \leq y_i + w_i, \\
z_i - (1 - w_i) \leq\ & x_i \leq z_i + (1 - w_i).
\end{aligned}
$$

In this representation, $w_i = 1$ corresponds to the case that x_i is selected as a control node, to which z_i gives 0-1 assignment.

In ATTRACTOR CONTROL, we need to maximize the score for non-control nodes. That is, we need to maximize $\displaystyle\sum_i \alpha_i \cdot (1 - w_i) \cdot x_i$, where we assume without loss of generality that $\alpha_i \geq 0$ (otherwise, we can use $1 - x_i$ instead of x_i). For that purpose, we introduce additional 0-1 variables u_i (this u_i is different from that in BN CONTROL) and put constraints $u_i \leq x_i$ and $u_i \leq 1 - w_i$, and let the objective function be $\sum_i \alpha_i \cdot u_i$.

In the original definition of ATTRACTOR CONTROL, the objective is to maximize the minimum score of singleton attractors. However, since it is quite difficult to directly give an ILP formation for the problem, we begin with an ILP formation for finding a singleton attractor with the maximum score by selecting and controlling m nodes.

By combining the above mentioned inequalities with the ILP formalization for ATTRACTOR DETECTION, we have the following ILP formalization for finding a singleton attractor with the maximum score.

$$
\begin{aligned}
& \text{Maximize } \textstyle\sum_i \alpha_i u_i, \\
& \text{Subject to} \\
& \quad x_{i,b_{i_1}\dots b_{i_k}} \geq \left(\textstyle\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j})\right) - (k-1), \\
& \quad x_{i,b_{i_1}\dots b_{i_k}} \leq \tfrac{1}{k} \textstyle\sum_{j=1,\dots,k} \tau_{b_{i_j}}(x_{i_j}), \\
& \qquad\qquad \text{for all } i \in [1\dots n] \text{ and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k \\
& \qquad\qquad \text{such that } f_i(b_{i_1},\dots,b_{i_k}) = 1, \\
& \quad x_{i,b_{i_1}\dots b_{i_k}} = 0, \\
& \qquad\qquad \text{for all } i \in [1\dots n] \text{ and } b_{i_1}\dots b_{i_k} \in \{0,1\}^k \\
& \qquad\qquad \text{such that } f_i(b_{i_1},\dots,b_{i_k}) = 0, \\
& \quad y_i \leq \textstyle\sum_{b_{i_1}\dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\dots b_{i_k}}, \\
& \qquad\qquad \text{for all } i \in [1\dots n], \\
& \quad y_i \geq \tfrac{1}{2^k} \textstyle\sum_{b_{i_1}\dots b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\dots b_{i_k}}, \\
& \quad y_i - w_i \leq x_i \leq y_i + w_i, \\
& \quad z_i + w_i - 1 \leq x_i \leq z_i - w_i + 1,
\end{aligned}
$$

$u_i \le x_i, \quad u_i \le 1 - w_i,$
for all $i \in [1 \ldots n]$,
$x_i, y_i, z_i, w_i, u_i \in \{0,1\}$, for all $i \in [1 \ldots n]$,
$x_{i,b_{i_1}\ldots b_{i_k}} \in \{0,1\}$,
for all $i \in [1 \ldots n]$ and $b_{i_1} \ldots b_{i_k} \in \{0,1\}^k$,
$\sum_{i=1,\ldots,n} w_i = m$.

In order to solve the original version of ATTRACTOR CONTROL, we repeatedly use this ILP formulation (denoted by ILP-A). Suppose that $V' = (v_{i_1}, \ldots, v_{i_m})$ are selected as control nodes with 0-1 values $B' = (b_{i_1}, \ldots, b_{i_m})$. Then, we find the attractor of the minimum score under these control nodes (i.e., under COND1). This can also be formalized as ILP by modifying the above mentioned ILP formalization as follows. Let $I = \{i_1, \ldots, i_m\}$. We replace the objective function by "Minimize $\sum_{i \notin I} \alpha_i x_i$" and the constraints using u_i by

$x_i = z_i,\ w_i = 1$ for all $i \in I$,
$w_i = 0,$ for all $i \notin I$.

We denote the resulting ILP by ILP-B.

In order to avoid examining the previously examined (V', B'), we need to modify ILP-A. This can also be accomplished by introducing some linear inequalities stating that the solution must be different from the previously obtained solutions (given as sets of explicit node-value pairs). Let $\mathbf{x}^j = (x_1^{(j)}, x_2^{(j)}, \ldots, x_n^{(j)})$ be the jth control previously found, where we let $x_i^{(j)} = z_i$ if $w_i = 1$, otherwise $x_i^{(j)} = -1$. Then, for each j, we add the following linear constraint:

$$\sum_{x_i^{(j)} \neq -1} \left(\delta(x_i^{(j)}, 1)(-z_i - w_i) + \delta(x_i^{(j)}, 0)(z_i - w_i) \right) \ge 1 - \sum_{x_i^{(j)} \neq -1} (1 + x_i^{(j)}),$$

where $\delta(x,y)$ is the delta function (i.e., $\delta(x,y) = 1$ if and only if $x = y$). This inequality means that the following must hold for at least one i:

- if $x_i^{(j)} = 1$, either $z_i = 0$ or $w_i = 0$ holds,
- otherwise, either $z_i = 1$ or $w_i = 0$ holds.

In the following, ILP-A' denotes this modified version. Using the following procedure, we can solve ATTRACTOR CONTROL.

1. Repeat steps 2-3.
2. Find (V', B') which maximizes the score of a singleton attractor using ILP-A' under the condition that (V', B') is different from any of previously examined nodes/values pairs. If the maximum score is less than Θ, output "None" and halt.
3. Compute the minimum score of singleton attractors for (V', B') using ILP-B. If the minimum score is no less than Θ, output (V', B') and halt.

Though this procedure may repeat exponentially many times in the worst case, it is expected that this procedure does not repeat so many times because the expected number of singleton attractors (per (V', B')) is small regardless of n and thus it is expected in many cases that the maximum score matches with the minimum score [15].

4. Computational Experiments

In order to evaluate the efficiency of the proposed integer programming-based methods, we performed computational experiments by mainly using randomly generated BNs. In order to generate random BNs, for each node, we randomly selected a specified number of input nodes and randomly assigned Boolean functions. All of computational experiments were done on a PC with a Xeon 5470 3.33 GHz CPU and 10 GB RAM running under the LINUX (version 2.6.16) operating system. We used ILOG CPLEX (version 11.2, http://www.ilog.com/products/cplex/) for solving integer linear programs. For each problem, we examined the cases of maximum indegree 2 and 3 (i.e., $K = 2$ and $K = 3$).

4.1 Improvement Using More Direct Encoding

Though we presented a general encoding scheme for ILP in Sect. 3, we can use a more direct encoding scheme for the case of maximum indegree 2, which is useful for reducing CPU time. In the following, we present this scheme for ATTRACTOR DETECTION, where we present rules for positive literals only for AND, OR and XOR because modifications for other cases are straight-forward.

$$\begin{array}{ll}
v_i = b: & x_i = b, \\
v_i = v_j: & x_i - x_j = 0, \\
v_i = \overline{v}_j: & x_i + x_j = 1, \\
v_j \wedge v_k: & x_j + x_k - 1 \ \le\ x_i \ \le\ \frac{1}{2}x_j + \frac{1}{2}x_k, \\
v_j \vee v_k: & \frac{1}{2}x_j + \frac{1}{2}x_k \ \le\ x_i \ \le\ x_j + x_k, \\
v_j \oplus v_k: & x_i - x_j + x_k \ge 0,\ \ x_i + x_j - x_k \ge 0, \\
 & x_i - x_j - x_k \le 0,\ \ x_i + x_j + x_k \le 2.
\end{array}$$

Though it might be possible to use this kind of encoding scheme for $K \ge 3$, it would be quite complicated. Instead, we can add the following rules for $K = 3$ if $f_i(v_j, v_k, v_h)$ has a special form (modification for cases of $f_i(v_j, b, v_h)$ and $f_i(v_j, v_k, b)$ are straight-forward), where this technique can be extended for $K > 3$.

$$\begin{array}{ll}
f_i(0, v_k, v_h) = 0: & x_i - x_j \le 0, \\
f_i(0, v_k, v_h) = 1: & x_i + x_j \ge 1, \\
f_i(1, v_k, v_h) = 0: & x_i + x_j \le 1, \\
f_i(1, v_k, v_h) = 1: & x_i - x_j \ge 0.
\end{array}$$

In addition to the above modifications, there is some minor implementation issue. For the case of $K = 3$, we cannot represent $1/K$ exactly. Thus, we replaced

$$x_{i,b_{i_1}\ldots b_{i_3}} \le \frac{1}{3} \sum_{j=1,\ldots,3} \tau_{b_{i_j}}(x_{i_j})$$

Table 1 CPU time (sec.) on ATTRACTOR DETECTION for BNs of $K = 2$.

n	2000	4000	6000	8000	10000
time (sec)	0.016	0.026	0.037	0.055	0.071

Table 2 CPU time (sec.) on ATTRACTOR DETECTION (original version) for BNs of $K = 3$ with varying the ratio of indegree 3 nodes.

	n				
ratio	30	60	90	120	150
20%	0.005	0.003	0.006	0.007	0.010
40%	0.006	0.010	0.050	0.024	0.201
60%	0.010	0.112	0.794	1.193	3.207
80%	0.034	0.502	3.214	9.248	39.249
100%	0.055	1.942	20.014	113.409	262.473

Table 3 CPU time (sec.) on ATTRACTOR DETECTION (improved version) for BNs of $K = 3$ with varying the ratio of indegree 3 nodes.

	n					
ratio	30	60	90	120	150	180
20%	0.004	0.005	0.005	0.007	0.008	0.008
40%	0.006	0.008	0.012	0.012	0.020	0.016
60%	0.007	0.011	0.020	0.168	0.310	0.384
80%	0.008	0.071	0.259	1.198	9.343	20.342
100%	0.017	0.637	2.886	14.536	47.223	310.985

by

$$x_{i,b_{i_1}\ldots b_{i_3}} \leq 0.333 \sum_{j=1,\ldots,3} \tau_{b_{i_j}}(x_{i_j}) + 0.1.$$

4.2 Results on ATTRACTOR DETECTION

We applied the proposed method for ATTRACTOR DETECTION to randomly generated BNs. In Table 1, the average CPU time (per BN) is shown against 20 randomly generated BNs with $K = 2$ and $K = 3$, respectively. For the case of $K = 2$, we examined the improved method (described in Sect. 4.1). For the case of $K = 3$, we examined the original method (described in Sect. 3.1) and the improved method.

For BNs with $K = 2$, we examined cases of $n = 2000$, 4000, 6000, 8000 and 10000. The result is shown in Table 1.

For the original method for BNs with $K = 3$, we examined cases of $n = 30$, 60, 90, 120 and 150 with varying the ratio of indegree 3 nodes (note that K is the maximum indegree). For the improved method for BNs with $K = 3$, we examined cases of $n = 30$, 60, 90, 120, 150 and 180 also with varying the ratio of indegree 3 nodes. The results are shown in Table 2 and Table 3.

From Table 1, it is seen that ATTRACTOR DETECTION is solved very quickly for the case of $K = 2$. It also seems that CPU time increases near linearly with the size of BNs. This result is a bit surprising because ATTRACTOR DETECTION is known to be NP-hard even for the case of $K = 2$ [31]. One of the reasons might be that 6 among 16 possible Boolean functions of $K = 2$ are constant or unary functions and thus the values of many nodes are determined quickly. Another possible reason is that randomly generated BNs for $K = 2$ may not have large connected components or may have small connectivities and thus detection of a singleton attractor may be easy.

It is seen from Tables 2 and 3 that the ratio of indegree 3 nodes significantly affects the computation time. For the cases of low ratios of indegree 3 nodes, it is difficult to see the tendency of increase of CPU time with the number of nodes. However, we can observe from the case of 100% ratio of indegree 3 nodes that CPU time increases exponentially with the number of nodes. We can also see that our proposed method works very well for BNs with $K = 3$ if the ratio of indegree 3 nodes is not high. It is also seen from Tables 2 and 3 that the improved method is much faster than the original method. For example, in the case of (100%, $n = 150$), the improved method took only 47.223 sec. whereas the original method took 262.473 sec.

The tendency that attractor detection for $K = 2$ is much easier than that for $K = 3$ is also observed by Devloo et al. [11] and by de Jong and Page [10] though there are some differences between our model and their models. It is also to be noted that our method is much faster than their methods in the case of $K = 2$ and is comparable to the method by Devloo et al. in the case of $K = 3$, but is slower than the method by Jong and Page in the case of $K = 3$.

Table 4 CPU time (sec.) on BN CONTROL for BNs of $K = 2$.

$n/m = M$	100/10	500/50	1000/100	1500/150
time	0.003	0.916	33.770	328.568

4.3 Results on BN CONTROL

For BN CONTROL, we randomly generated BNs with external control nodes, initial global states, and control sequences, from which we computed the target desired states. Then, we generated ILP instances from these BNs and these initial and desired states (without giving control sequences). As a result, we could obtain the desired control sequences for all cases. For BNs of $K = 2$, we examined the cases of $(n, m, M) = (100, 10, 10)$, $(500, 50, 50)$, $(1000, 100, 100)$, and $(1500, 150, 150)$, where n, m, M are the numbers of internal nodes, external nodes, and the target time step, respectively. For BNs of $K = 3$, we examined the cases of $(n, m, M) = (30, 3, 3)$, $(60, 6, 6)$, $(90, 9, 9)$, $(120, 12, 12)$, $(150, 15, 15)$ and $(180, 18, 18)$, with varying the ratio of indegree 3 nodes. For both of $K = 2$ and $K = 3$, we employed the improved encoding schemes, and we only examined the cases of $m = M$.

For each case, the average CPU time (per BN) over 20 trials is shown in Tables 4 and 5, respectively. It is seen from Table 4 that the proposed method is fast for BNs of $K = 2$. It is also seen from Table 5 that the proposed method is fast if the ratio of indegree 3 nodes is not large. The reason of this big difference might be the same as that for ATTRACTOR DETECTION.

4.4 Results on ATTRACTOR CONTROL

For ATTRACTOR CONTROL, we also examined the cases

Table 5 CPU time (sec.) on BN CONTROL for BNs of $K = 3$ with varying the ratio of indegree 3 nodes, where '-' means that an experiment was not finished within 5 days.

	$n/m = M$					
ratio	30/3	60/6	90/9	120/12	150/15	180/18
20%	0.006	0.013	0.026	0.056	0.237	0.017
40%	0.006	0.017	0.048	0.388	4.157	16.798
60%	0.008	0.023	0.102	8.615	-	-
80%	0.009	0.006	0.010	4914.6	-	-
100%	0.012	0.041	97.74	-	-	-

Table 6 Results on ATTRACTOR CONTROL.

$K = 2$	n/m	800/80	1200/120	1600/160	2000/200
	time (sec)	14.194	19.057	42.537	61.124
	#pos/#rep	7/3.86	7/2.29	7/3.15	7/2.58
	#neg/#rep	4/1.00	1/1.00	1/1.00	1/1.00
$K = 3$	n/m	100/10	120/12	140/14	160/16
	time (sec)	75.386	27.761	81.246	309.263
	#pos/#rep	4/3.0	9/1.56	7/2.58	5/1.80
	#neg/#rep	4/1.0	1/1.00	3/1.00	2/1.00

of $K = 2$ and $K = 3$. In Table 6, the following information is shown against 10 randomly generated BNs with $K = 2$ and $K = 3$:

- time: average CPU time (seconds) per BN,
- #pos/#rep: the number of trials (i.e., BNs) for which the desired attractors were found, and the average number of repetitions per each of such trials (recall that we need to solve ILP instances repeatedly for ATTRACTOR CONTROL),
- #neg/#rep: the number of trials (i.e., BNs) for which it was found that there did not exist the desired attractors, and the average number of repetitions per each of such trials,

where we set $\alpha_i = 1$ for all $i \in \{1, \ldots, n\}$ and $\Theta = 0.65n$, and we set the maximum number of repetitions to be 20. It is to be noted that in some cases (i.e., $10 - \#pos - \#neg$ cases), the procedure could not decide within 20 repetitions whether or not there existed the desired attractors. The reason why $\Theta = 0.65n$ was selected is that there almost always existed the desired attractors if $\Theta \ll 0.65n$ and there seldom existed the desired attractors if $\Theta \gg 0.65n$ in our preliminary computational experiments.

It is seen from Table 6 that the proposed method is at least useful for BNs of $K = 2$ with up to 2000 nodes and for BNs of $K = 3$ with up to 140 nodes. It is also seen that the numbers of repetitions required to decide the existence of the desired attractors are usually small though these numbers are large (more than 20) in some cases. The reason why CPU time for (120, 12) is smaller than that for (100, 10) in $K = 3$ is that the former case required less number of repetitions.

In addition to artificially generated BNs, we applied the proposed method for ATTRACTOR CONTROL to a 10-gene WNT5A network (see Fig. 3) [20], which is closely related to metastatic melanoma. Since only topology of the network is given in [20], we assigned a random

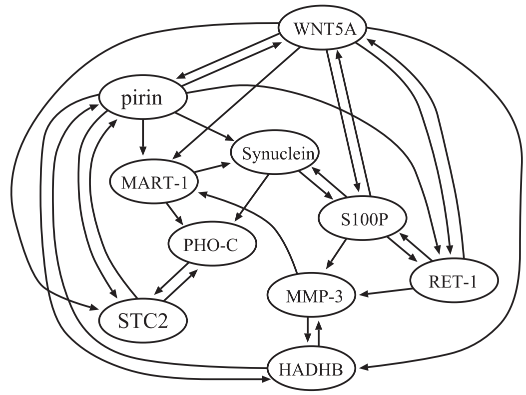


Fig. 3 Structure of 10 gene WNT5A network [20].

Boolean function to each node, where each node has 3 inputs and each Boolean function has exactly 3 relevant inputs. We generated 10 BNs in this way and took the average CPU time, #pos/#rep, #neg/#rep, where we let $\alpha_i = 1$ for all i, $n = 10$, $m = 2$ and $\Theta = 0.6$. The result was: 0.080 (sec), #pos/#rep = 8/2.13, #neg/#rep = 2/3.5. This result suggests that the proposed method might work in reasonable CPU time for real (but not large) networks though further studies (in particular, assignment of biologically adequate Boolean functions) should be done in order to evaluate the usefulness of the method from a biological viewpoint.

5. Complexity of ATTRACTOR CONTROL

In the above, we did not give a direct transformation from ATTRACTOR CONTROL to ILP. Here, we show that it is not plausible to give such a direct transformation.

Theorem 1: ATTRACTOR CONTROL is Σ_2^p-hard.

(Proof) Let $\psi(\mathbf{x}, \mathbf{y})$ be a 3-DNF (disjunction of conjunctions each of which consisting of 3 literals) over variables $\mathbf{x} = (x_1, \ldots, x_{n_1})$ and $\mathbf{y} = (y_1, \ldots, y_{n_2})$. Then, it is known that deciding whether or not $(\exists\mathbf{x})(\forall\mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ is true is Σ_2^p-complete [29]. We show a polynomial time reduction from this problem to ATTRACTOR CONTROL.

From a given $\psi(\mathbf{x}, \mathbf{y})$, we construct a BN as follows (see also Fig. 4). Let m_1 be the number of terms in $\psi(\mathbf{x}, \mathbf{y})$. Then, we let $V = \{v_1, v_2, \ldots, v_{n_1+n_2+m_1+1}\}$. For $i = 1, \ldots, n_1$, v_i corresponds to x_i. For $i = 1, \ldots, n_2$, v_{n_1+i} corresponds to y_i. For $i = 1, \ldots, m_1$, $v_{n_1+n_2+i}$ corresponds to the ith term of $\psi(\mathbf{x}, \mathbf{y})$, where the ith term is represented as $l_{i_1} \wedge l_{i_2} \wedge l_{i_3}$.

Then, we assign the following functions to V:

$$v_i(t+1) = \overline{v_i(t)}, \quad \text{for } i = 1, \ldots, n_1,$$
$$v_{n_1+i}(t+1) = v_{n_1+i}(t), \quad \text{for } i = 1, \ldots, n_2,$$
$$v_{n_1+n_2+i}(t+1) = l_{i_1}(t) \wedge l_{i_2}(t) \wedge l_{i_3}(t), \quad \text{for } i = 1, \ldots, m_1,$$
$$v_{n_1+n_2+m_1+1}(t+1) = \bigvee_{i \in \{1,\ldots,m_1\}} v_{n_1+n_2+i}(t).$$

Finally, we let $m = n_1$, $\alpha_{n_1+n_2+m_1+1} = 1$, $\alpha_i = 0$ for $i <$

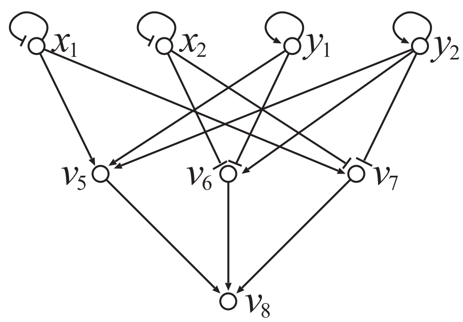


Fig. 4 Reduction from $(\exists \mathbf{x})(\forall \mathbf{y})((x_1 \wedge y_1 \wedge y_2) \vee (\overline{x}_2 \wedge \overline{y}_1 \wedge y_2) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{y}_2))$ to ATTRACTOR CONTROL. In this figure, x_1, x_2, y_1 and y_2 are identified with v_1, v_2, v_3 and v_4, respectively. Usual and T-shape arrows denote positive and negative controls, respectively. In this case, ATTRACTOR CONTROL has a solution by selecting x_1 and x_2 as control nodes with assigning $(x_1, x_2) = (1, 0)$.

$n_1 + n_2 + m_1 + 1$, and $\Theta = 1$.

Then, we can see that ATTRACTOR CONTROL has a solution iff $(\exists \mathbf{x})(\forall \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ is true. First, suppose that $(\exists \mathbf{x})(\forall \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ is true for an assignment of $\mathbf{x} = (b_1, \ldots, b_{n_1})$. Then, it is straight-forward to see that ATTRACTOR CONTROL has a solution by an assignment of $(v_1, \ldots, v_{n_1}) = (b_1, \ldots, b_{n_1})$.

Next, suppose that ATTRACTOR CONTROL has a solution. Then, we can see that $v_1, \ldots, v_{n_1}$ must be selected as control nodes since $v_i(t+1) = \overline{v_i(t)}$ are assigned to these nodes. Furthermore, for any assignment on $v_{n_1+1}, \ldots, v_{n_2}$, the states of $v_{n_1+n_2+1}, \ldots, v_{n_1+n_2+m_1+1}$ satisfying the condition of a singleton attractor are uniquely determined. Since $g(\mathbf{v})$ is determined only by the value of $v_{n_1+n_2+m_1+1}$ and $g(\mathbf{v}) \geq 1$ must hold, $v_{n_1+n_2+m_1+1}$ takes 1 (in a singleton attractor) for each assignment on $v_{n_1+1}, \ldots, v_{n_2}$. Therefore, $(\exists \mathbf{x})(\forall \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ is true.

Since the reduction can obviously be done in polynomial time, we have the theorem. □

It is to be noted that the theorem holds for BNs with bounded indegree 3 by encoding the large OR node (i.e., $v_{n_1+n_2+m+1}$) using a binary tree.

Since ILP belongs to NP and it is widely believed that Σ_2^p is not equal to NP [13], it is not plausible that there exists a polynomial time reduction from ATTRACTOR CONTROL to ILP. It is worthy to note that both detection of a singleton attractor and control of BN are trivially in NP (assuming that M is polynomial of n and m). Therefore, this result also suggests that control of attractors is computationally harder than the other two problems.

The technique introduced here was applied later to show Σ_2^p-hardness of control of PBNs [6].

6. Discussions and Conclusions

In this paper, we have presented integer programming-based methods that solve attractor detection, control, and attractor control problems for BNs in a unified manner. Since these problems are NP-hard, it is reasonable to use integer programming. The results of computational experiments suggest that the proposed methods are useful for solving moderate size instances of these problems. Though the proposed methods might not be the fastest, they are simple and easy to implement and have room for various kinds of extensions and modifications. For example, one might want to limit the number of nodes whose values are 1 to be less than some constant. Such a constraint can be easily posed by adding a linear constraint, whereas it seems very difficult to give such a constraint using SAT formula.

The proposed methods are not fast for BNs of $K = 3$. Furthermore, the number of variables increases exponentially against K. Therefore, our method cannot be applied to BNs with large indegree. However, the results on attractor detection suggest that the methods might still be useful if the ratio of nodes of indegree more than 2 is not large. It is also suggested that CPU time for the case of $K \geq 3$ might be significantly improved if we employ other encoding schemes. Furthermore, it might be possible to develop much more efficient encoding schemes if we can restrict the types of Boolean functions used. Therefore, development of such encoding schemes is left as future work.

For attractor control, we could not transform an original instance to a single ILP instance, and instead employed a procedure which iteratively generates and solves ILP instances. The reason why we could not find a transformation method is that the attractor control problem is Σ_2^p-hard, whereas it is known that decision problem versions of ILP, attractor detection, and control of BN are in NP. It is to be noted that Σ_2^p-hardness of the attractor control problem suggests that this problem is not in NP (under the assumption of $P \neq NP$). This result also suggests that attractor control is computationally harder than attractor detection and control of BN.

In attractor detection and control of BN problems, the proposed ILP-based methods output just one solution. However, it is possible to modify the methods for enumerating all solutions by iteratively executing ILP with putting a constraint that a new solution must be different from previous solutions.

Such an approach may provide another way to cope with Σ_2^p-hardness of the attractor control problem. As discussed in Sect. 3.3, attractor control can be directly transformed into ILP if we only need to maximize the score of one singleton attractor, and the number of singleton attractors may be small in many real networks. By combining these facts, we modify the objective of attractor control from optimization of the minimum score to maximization of the k-th largest score of singleton attractors. Then, it seems possible to obtain direct transformation to ILP by making h copies of ILP formulation in Sect. 3.3 (with keeping the same control nodes to these h copies), putting constraints that the resulting h singleton attractors must be different and the scores $s_1, \ldots, s_h$ must satisfy $s_1 \geq s_2 \geq \cdots \geq s_h$, and letting the objective to maximize s_h. If the number of singleton attractors is at least h when appropriately selecting m nodes as control nodes, the resulting ILP outputs an optimal solution under this modified definition. Otherwise, ILP fails to output a solution, and then we should examine lower values of h. Although this method might not work efficiently in practice because of the size of the resulting ILP, it might be worthy to explore this approach.

Recently, Liu et al. studied the controllability of complex networks [25], in terms of structural controllability, which is based on linear control theory but can be determined only from network structure. By making use of a relationship between structural controllability and maximum matching, they analytically showed that dense and homogeneous networks can be controlled using a few control nodes whereas sparse inhomogeneous networks, which appear in many real networks, require many control nodes. Since maximum matching can be computed in polynomial time, they could also analyze the structural controllability of many real networks. Although the purpose of our study is similar to theirs, the models are very different. BN is a non-linear discrete model whereas they considered linear systems. Furthermore, it seems very difficult for BNs to obtain a good characterization of the structural controllability because it is known that control problem is NP-hard [2], which motivated us to develop the proposed ILP-based method. However, there is a possibility that real networks have some useful structural properties that make control easy. Therefore, studies to discover such properties should be done.

We have employed BNs as a model of genetic networks. Though BNs are useful to model certain types of real genetic networks [5], [24], they might be too simple to model other types of genetic networks. On the other hand, real values or non 0-1 integer values can be handled in ILP. Therefore, it might be possible to extend BNs so that certain kinds of real values and/or non 0-1 integer values can be handled. In particular, by introducing real valued variables, it might be possible to cope with some uncertainty as suggested in [6], [22], [23], where uncertainty or stochasticity plays an important role in some biological systems. Such extensions of BNs along with efficient ILP formalizations might be useful and should be studied further.

References

[1] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, “A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions,” Genome Informatics, no.9, pp.151–160, 1998.
[2] T. Akutsu, M. Hayashida, W.-K. Ching, W.-K., 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.
[3] T. Akutsu, M. Hayashida, and T. Tamura, “Integer programming-based methods for attractor detection and control of Boolean networks,” Proc. The combined 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, pp.5610–5617, 2009.
[4] T. Akutsu, A.A. Melkman, T. Tamura, and M. Yamamoto, “Determining a singleton attractor of a Boolean network with nested canalyzing functions,” J. Computational Biology, vol.18, pp.1275–1290, 2011.
[5] E. Balleza, E.R. Alvarez-Buylla, A. Chaos, S. Kauffman, I. Shmulevich, and M. Aldana. “Critical dynamics in genetic regulatory networks: examples from four kingdoms,” PLoS One, vol.3, e2456, 2008.
[6] X. Chen, T. Akutsu, T. Tamura, and W.-K. Ching, “Finding optimal control policy in probabilistic Boolean networks with hard constraints by using integer programming and dynamic programming,” International Journal of Data Mining and Bioinformatics, to appear.
[7] D. Cheng and H. Qi, “Controllability and observability of Boolean control networks,” Automatica, vol.45, pp.1659–1667, 2009.
[8] D. Cheng, “Input-state approach to Boolean networks,” IEEE Trans. Neural Netw., vol.20, pp.512–521, 2009.
[9] 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.
[10] H. de Jong and M. Page, “Search for steady states of piecewise-linear differential equation models of genetic regulatory networks,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol.5, no.2, pp.208–222, 2008.
[11] V. Devloo, P. Hansen, and M. Labbé, “Identification of all steady states in large networks by logical analysis,” Bulletin of Mathematical Biology, vol.65, pp.1025–1051, 2003.
[12] E. Dubrova and M. Tesienko, “A SAT-based algorithm for finding attractors in synchronous Boolean networks,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol.8, no.5, pp.1393–1399, 2011.
[13] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, New York, 1979.
[14] A. Garg, A. DiCara, I. Xenarios, L. Mendoza, and G. DeMicheli, “Synchronous versus asynchronous modeling of gene regulatory networks,” Bioinformatics, vol.24, pp.1917–1925, 2008.
[15] M. Hayashida, T. Tamura, T. Akutsu, S.-Q. Zhang, and W.-K. Ching, “Algorithms and complexity analyses for control of singleton attractors in Boolean networks,” EURASIP Journal on Bioinformatics and Systems Biology, vol.2008, 521407, 2008.
[16] K. Inoue, “Logic programming for Boolean networks,” Proc. 22nd International Joint Conference on Artificial Intelligence, pp.924–930, 2011.
[17] D.J. Irons, “Improving the efficiency of attractor cycle identification in Boolean networks,” Physica D, vol.217, pp.7–21, 2006.
[18] J.K. Karlof (ed.), Integer Programming: Theory And Practice, CRC Press, Boca Raton, Florida, 2005.
[19] S.A. Kauffman, The Origins of Order: Self-organization and Selection in Evolution, Oxford Univ. Press, New York, 1993.
[20] S. Kim, H. Li, E.R. Dougherty, N. Cao, Y. Chen, M. Bittner, and E.B. Suh, “Can Markov chain models mimic biological regulations?” J. Biological Systems, vol.10, pp.337–357, 2002.
[21] H. Kitano, “Cancer as a robust system: implications for anticancer therapy,” Nature Reviews Cancer, vol.4, pp.227–235, 2004.
[22] K. Kobayashi and K. Hiraishi. “An integer programming approach to control problems in probabilistic Boolean networks,” Proc. 2010 American Control Conference, pp.6710–6715, 2010.
[23] K. Kobayashi and K. Hiraishi, “An integer programming approach to optimal control problems in context-sensitive probabilistic Boolean networks,” Automatica, vol.47, pp.1260–1264, 2011.
[24] C.J. Langmead and S.K. Jha, “Symbolic approaches for finding control strategies in Boolean networks,” J. Bioinformatics and Computational Biology, vol.7, pp.323–338, 2009.
[25] Y.-Y. Liu, J.-J. Slotin, and A.-L. Barabási, “Controllability of complex networks,” Nature, vol.473, pp.167–173, 2011.
[26] B. Luque and R.V. Solé, “Controlling chaos in random Boolean networks,” Europhysics Letters, vol.37, pp.597–602, 1997.
[27] A.A. Melkman, T. Tamura, and T. Akutsu, “Determining a singleton attractor of an AND/OR Boolean network in $O(1.587^n)$ time,” Inf. Process. Lett., vol.110, pp.565–569, 2010.
[28] B. Samuelsson and C. Troein, “Superpolynomial growth in the number of attractors in Kauffman networks,” Phys. Rev. Lett., vol.90, 098701, 2003.
[29] L.J. Stockmeyer, “The polynomial-time hierarchy,” Theor. Comput.

Sci., vol.3, pp.1–22, 1976.
[30] K. Takahashi, K. Tanabe, M. Ohnuki, M. Narita, T. Ichisaka, K. Tomoda, and S. Yamanaka, "Induction of pluripotent stem cells from adult human fibroblasts by defined factors," Cell, vol.131, pp.1–12, 2007.
[31] T. Tamura and T. Akutsu, "Detecting a singleton attractor in a Boolean network utilizing SAT algorithms," IEICE Trans. Fundamentals, vol.E92-A, no.2, pp.493–501, Feb. 2009.
[32] S.-Q. Zhang, M. Hayashida, T. Akutsu, W.-K. Ching, and M.K. Ng, "Algorithms for finding small attractors in Boolean networks," EURASIP Journal on Bioinformatics and Systems Biology, vol.2007, 20180, 2007.

Tatsuya Akutsu received his M.Eng degree in Aeronautics in 1986 and a Dr. Eng. degree in Information Engineering in 1989 both from University of Tokyo, Japan. From 1989 to 1994, he was with Mechanical Engineering Laboratory, Japan. He was an associate professor in Gunma University from 1994 to 1996 and in Human Genome Center, University of Tokyo from 1996 to 2001 respectively. He joined Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan as a professor in Oct. 2001. His research interests include bioinformatics and discrete algorithms.

Yang Zhao received her M.Inf. degree from Graduate School of Informatics, Kyoto University, in 2011. She has been a Ph.D course student in Graduate School of Informatics, Kyoto University since 2011. Her research interests include development of computational methods for analyzing chemical molecules and bioimformatics analysis of genomic data.

Morihiro Hayashida received his M.Sc. degree in Information Science from University of Tokyo, Japan, and his Ph.D. degree in Informatics from Kyoto University, Japan, in 2005. He is currently an assistant professor in Laboratory of Biological Information Networks, Bioinformatics Center, Institute for Chemical Research, Kyoto University. His research interests include functional analysis of proteins and development of computational methods for bioinformatics.

Takeyuki Tamura received B.E., M.E. and Ph.D. degrees in informatics from Kyoto University, Japan, in 2001, 2003, and 2006, respectively. He joined Bioinformatics Center, Institute for Chemical Research, Kyoto University as a postdoctoral fellow in April, 2006. He works as an assistant professor from December, 2007. His research interests are bioinformatics and the theory of combinatorial optimization.