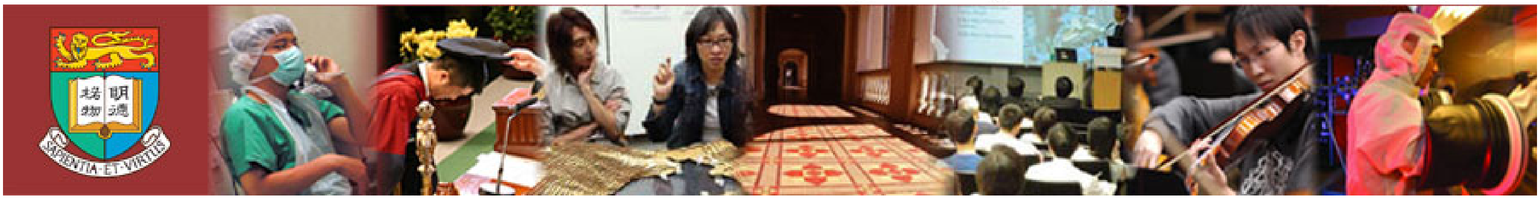


Advances in Complex Systems
Vol. 19, No. 3 (2016) 1650006 (32 pages)
© The Author(s)
DOI: 10.1142/S0219525916500065

FINDING AND ANALYZING THE MINIMUM SET OF DRIVER NODES IN CONTROL OF BOOLEAN NETWORKS

WENPIN HOU

Department of Mathematics, The University of Hong Kong,
Pokfulam Road, Hong Kong 999077
whou@hku.hk

TAKEYUKI TAMURA

Bioinformatics Center, Institute for Chemical Research,
Kyoto University, Kyoto 611-0011, Japan
tamura@kuicr.kyoto-u.ac.jp

WAI-KI CHING

Department of Mathematics, The University of Hong Kong,
Pokfulam Road, Hong Kong 999077
wching@hku.hk

TATSUYA AKUTSU*

Bioinformatics Center, Institute for Chemical Research,
Kyoto University, Kyoto 611-0011, Japan
takutsu@kuicr.kyoto-u.ac.jp

Received 16 April 2016
Accepted 19 July 2016
Published 2 September 2016

We study the minimum number of driver nodes control of which leads a Boolean network (BN) from an initial state to a target state in a specified number of time steps. We show that the problem is NP-hard and present an integer linear programming-based method that solves the problem exactly. We mathematically analyze the average size of the minimum set of driver nodes for random Boolean networks with bounded in-degree and with a small number of time steps. The results of computational experiments using randomly generated BNs show good agreements with theoretical analyses. A further examination in realistic BNs demonstrates the efficiency and generality of our theoretical analyses.

Keywords: Boolean networks; Boolean control; integer linear programming; genetic networks; driver nodes.

*Corresponding author.

1. Introduction

Development of a control theory for complex biological systems is one of the primary goals of systems biology because of its potential applications to drug discovery and treatment of intractable diseases [40, 41]. While a lot of useful results have been obtained for control of linear systems, not so many practical results were obtained for nonlinear systems [12]. It does not seem to be useful to apply existing theories and methods to control of biological systems because biological systems are complex and contain many nonlinear subsystems. Therefore, it is necessary to develop useful control strategies for complex biological systems.

Among various mathematical models proposed for modeling complex and nonlinear biological systems, *Boolean network* (BN) [37, 38] attracts considerable attentions as a nonlinear model of *gene regulatory networks* [5, 7, 10, 11, 24, 76]. Gene regulatory networks are models of sets of interactions among genes, the analysis of which is one of the key topics in systems biology and theoretical biology. BNs have been widely used to model the regulatory networks in mammalian cortical area [31], T cell large granular lymphocyte leukemia [75], mammalian cell cycle [26], and to reproduce the wild-type gene expression patterns [8]. Of course, BNs cannot model all aspects of biological systems. However, it is suggested that BNs provide good approximations to the nonlinear functions in many biological systems [11] and BNs can be used to elucidate how perturbations may alter normal behavior and thus lead to testable predictions [9]. In fact, many BN-based models have been proposed for describing various regulatory behaviors of biological systems as mentioned above (see also [9]). There are two types of BNs: *synchronous BNs* and *asynchronous BNs*. The difference between these two types lies on whether or not the states of nodes are updated synchronously. Asynchronous BNs may be more appropriate to model certain behaviors of biological systems. However, there are various subtypes in asynchronous BNs [9], and there is no widely-used control model of asynchronous BNs. On the other hand, many studies have been done on controllability of synchronous BNs [2, 17, 28, 49, 50, 60, 84]. Therefore, in order to make use of the established control models, we focus on synchronous BNs in this paper.

Many studies have been done on the analyses of BNs. For example, distribution of attractors [7, 24, 38, 76], singleton attractors detection and the average case complexity analyses [30, 81, 83], relationship between network topology and chaotic behavior [10, 11], and inference of BNs from gene expression data [5, 48, 56] have been extensively studied. Stimulated by works on control of the probabilistic Boolean network (PBN) [22, 71], Akutsu *et al.* studied the problem of finding control strategies for BNs and showed it is NP-hard even in considerably restricted cases [2].

Recently, extensive studies have also been done on the controllability and observability of BNs [16, 27, 28, 50, 53–55, 60, 84], based on the concept of *Semi-tensor product matrices* that was proposed by Cheng and his colleagues [17–19]. Zhao *et al.* and Fornasini *et al.* considered the optimal control of BNs over infinite horizon [28, 84]. Li *et al.* investigated the controllability of both BNs and free Boolean sequences in time-variant and time-invariant ways [53, 54]. Laschov *et al.* and Chen *et al.* considered the Mayer-type optimal control problems of BNs [16, 50]. In many of these works, necessary and sufficient conditions of the controllability of Boolean control networks are derived. However, these methods require high computational cost: it is required to handle exponential size matrices. Other methods have also been developed for PBN-like models [1, 45, 46, 67]. Although these methods consider an arbitrary target state, other methods have been proposed for restricting the target states to be attractors. Recently, Zañudo and Albert proposed a method that identifies driver nodes for BNs irrespective of the updating scheme (synchronous/asynchronous) and the number of time steps by using the notion of stable motifs, each of which is defined as a set of nodes that form a minimal strongly connected component of the network and whose states form a partial fixed point [70]. Many years ago, Akutsu *et al.* showed that singleton attractors in a BN can be determined by specifying the states of nodes in the feedback vertex set (FVS) [4], where FVS is a well-known concept in graph theory whose removal leaves the network acyclic. It means that we can drive a BN to any specified singleton attractor by fixing the values of nodes in the FVS. Mochizuki *et al.* extended this FVS-based approach to a general model including a certain type of nonlinear ordinary differential equations and periodic attractors [63]. Zañudo and Albert suggested that their method may require a smaller set of driver nodes than the FVS-based methods do although there is no guarantee that their method can give the smallest set of driver nodes [70]. These attractor-based methods provide practical ways to control BNs. However, the target states are restricted to attractors and the number of time steps cannot be specified.

On the other hand, *structural controllability* has recently attracted considerable attentions in the field of complex networks. One of the main targets in that study is to determine the minimum set of driver nodes control of which leads the system to a desired state, where the set is determined from the structure of networks. Liu *et al.* [61] analyzed the average size of the minimum driver set for random networks and scale-free networks (i.e., networks with power-law degree distributions) by making use of the relationship [58] between structural controllability of a linear system and the maximum matching of a bipartite graph [34]. Following their work, many studies have been done. For example, Nepusz and Vicsek proposed and studied the edge dynamics model [66] in place of the node dynamics model adopted by Liu *et al.* Nacher and Akutsu gave a relationship between structural controllability and the minimum dominating set, a well known concept in graph theory, by using a similar model [64]. Gao *et al.* showed that the minimum number of driver nodes will be overestimated if it is required to control a subset of nodes (or a subsystem) [29]. The problem of how the number of driver nodes changes as the removal fraction of edges changes was discussed in [68, 73, 74], and structurally robust control was discussed in [65]. However, these results cannot be applied to BNs because BNs are highly nonlinear. Therefore, we study in this paper the minimum set of driver nodes control of which leads a given Boolean network from a specified initial state to a specified target state in a specified number of time steps.

The purpose of this study is to develop a practical method using *Integer Linear Programming* (ILP) to select the minimum number of driver nodes and to perform theoretical analyses on the average size of the minimum set of driver nodes, both for BNs. We focus on cases where the maximum in-degree is bounded by some constant K. It is to be noted that most nodes in many real networks are considered to have small in-degree. For example, gene regulatory networks are generally thought to be made up of only a few nodes with high degree (knowns as hubs) and a large majority of poorly connected nodes with low degree. Thus, gene regulatory networks are assumed to have a hierarchical scale free network topology [13]. Many of existing studies on BNs also focused on the cases of bounded in-degree [2, 3, 6, 21].

The paper is organized as follows. In Sec. 2, after reviewing BNs, we define the problem to be discussed and show that it is NP-hard. In Sec. 3, we elucidate an ILP-based method for finding the minimum set of driver nodes for given BNs and initial/target states. In Sec. 4, we mathematically analyze boundary values of the minimum number of control nodes for random BNs: upper bounds for cases of time steps $M = 1$ and $M = 2$. We also analyze the expected number of control nodes needed in final time step. In Sec. 5, we present results of computational experiments. Finally, we conclude with future work.

2. Problem

In this section, we briefly review BNs [38] and then introduce the problem.

2.1. *Boolean network*

A *Boolean network* $G(V, F)$ consists of a set of N *nodes* $V = \{v_1, \ldots, v_N\}$ and a list of *Boolean functions* $F = (f_1, \ldots, f_N)$. The state of v_i at time t is denoted by $v_i(t)$. The *global state* (or simply the *state*, or *Gene Activity Profile* (GAP)) of a BN at time t is denoted by the vector $\mathbf{v}(t) = (v_1(t), \ldots, v_N(t))$. For each node v_i, the *Boolean function* is in the form of $f_i(v_{i1}, \ldots, v_{ik_i})$ where $v_{i1}, \ldots, v_{ik_i}$ are the k_i input nodes of v_i. The number k_i is called the *in-degree* of node v_i. The *Boolean function* f_i is a logical combination of k_i variables. Here, we denote the logical AND of x and y, logical OR of x and y, logical NOT of x, and exclusive OR of x and y by $x \wedge y$, $x \vee y$, $\bar{x}$, and $x \oplus y$, respectively. Then the state of node v_i at time $t+1$ is determined by $v_i(t+1) = f_i(v_{i1}(t), \ldots, v_{ik_i}(t))$. We also write the *transition rule* for v_i as $v_i(t+1) = f_i(\mathbf{v}(t))$ and the transition rule for the whole BN as $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t))$.

2.2. *Minimum set of control nodes in BN control*

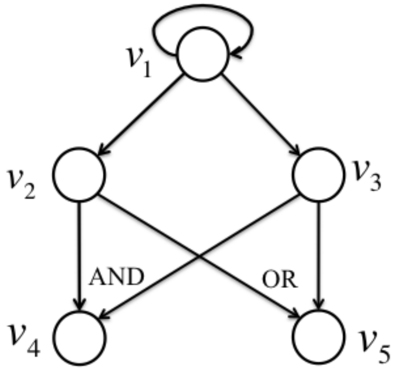
Akutsu *et al.* defined a control problem for BNs called BN CONTROL [2]. In BN CONTROL, nodes are classified into two classes: *internal nodes* and *external nodes* where internal nodes are original usual nodes in the BN and external nodes are control nodes. *Control nodes* (also called *driver nodes*) are nodes by controlling the values of which we can lead the BN to a desired state. There are two ways to obtain control nodes: select from original nodes [6] or add new nodes to the BN [33]. Here, we consider selecting n nodes from the original BN as control nodes for the reason that gene activation has been shown to be critical in control of cells such as iPS cells (induced pluripotent stem cells) generation [80].

Let $V = \{v_1, \ldots, v_n, v_{n+1}, \ldots, v_N\}$ be the set of all nodes in a BN, where $v_1, \ldots, v_n$ are control nodes and $v_{n+1}, \ldots, v_N$ are internal nodes. For convenience, we denote $\mathbf{u}(t) = (u_1(t), \ldots, u_n(t)) = (v_1(t), \ldots, v_n(t))$ and $\mathbf{v}'(t) = (v_{n+1}(t), \ldots, v_N(t))$. The number n is the number of control nodes selected to drive the BN to a desired state. The transition rule for the whole BN becomes $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{u}(t), \mathbf{v}'(t))$. For a global state $\mathbf{v}$, a global state $\mathbf{w}$ is called a *predecessor* of $\mathbf{v}$ if $\mathbf{v} = \mathbf{f}(\mathbf{w})$. That is, there exists an edge from $\mathbf{w}$ to $\mathbf{v}$ in the state transition diagram of a given BN if $\mathbf{w}$ is a predecessor of $\mathbf{v}$. Then the problem of Minimum Set of Driver Nodes is defined as follows.

Definition 1 (Minimum Set of Driver Nodes). Instance: a BN with N nodes, an initial state of the network $\mathbf{v}^0$, and a desired state of the network $\mathbf{v}^M$ at the M-th time step.

Problem: find the minimum set of driver nodes. By controlling the sequence $\langle \mathbf{u}(0), \ldots, \mathbf{u}(M) \rangle$ we have $\mathbf{v}(0) = \mathbf{v}^0$ and $\mathbf{v}(M) = \mathbf{v}^M$. The minimum number of driver nodes is denoted as N.

Example 1. Here follows an example to illustrate this problem. Recall that the problem is to find the minimum number of control nodes for a given BN, $\mathbf{v}^0$, $\mathbf{v}^M$, and M. For example, we consider a BN with 5 nodes (see Fig. 1(a)) defined by



$$v_1(t+1) = v_1(t),$$

$$v_2(t+1) = v_1(t),$$

$$v_3(t+1) = v_1(t),$$

$$v_4(t+1) = v_2(t) \wedge v_3(t),$$

$$v_5(t+1) = v_2(t) \vee v_3(t).$$

(a) (b)

Fig. 1. Example of BN.

Table 1. Transition diagrams resulted from different controls.

(a)

t	v_1	v_4	v_2	v_3	v_5
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	1	1	0
3	1	1	1	1	1

(b)

t	v_1	v_2	v_3	v_4	v_5
0	0	0	0	0	0
1	1	0	0	0	0
2	1	1	1	0	0
3	1	1	1	1	1

transition rules shown in Fig. 1(b). (1) Suppose that the initial global state is $\mathbf{v}^0 = (0,0,0,0,0)$, the desired global state is $\mathbf{v}^M = (1,1,1,1,1)$, and the target time step is $M = 3$. Then, if we select $\{v_1, v_4\}$ as the set of control nodes, we would have transition diagram Table 1(a).

Note that we can freely assign 0/1 values to $v_4(t)$ (for $t = 1, 2, \ldots$) regardless of the value assigned to $v_1(t)$. For convenience, we consider a node is selected as a control node at time t if the original Boolean function is not applied at the first time. For example, v_1 is selected as a control node at $t = 1$, and v_4 is selected as a control node at $t = 2$ in Table 1(a). Note that it is possible that multiple nodes are selected as control nodes at the same time step.

Therefore, in this case, the number of control nodes is 2 and the set of control nodes is $\{v_1, v_4\}$.

(2) However, if we select $\{v_1\}$ as the set of control nodes, we would have the state transitions from $\mathbf{v}^0$ to $\mathbf{v}^M$ as Table 1(b). Here, we note that if the set of control nodes is empty (i.e., the number of control nodes is 0), $\mathbf{v}(0) = \mathbf{v}(1) = \mathbf{v}(2) = \cdots = (0,0,0,0,0)$ holds and thus we cannot reach $\mathbf{v}^M = (1,1,1,1,1)$ at $M = 3$.

Therefore, in this case, the minimum set of driver nodes for this instance is $\{v_1\}$ and the minimum number of driver nodes is 1. It is to be noted that in our model, the minimum set of driver nodes depends on $\mathbf{v}^0$, $\mathbf{v}^M$, and M although it is determined independent of these factors in the case of structural controllability of linear systems [58, 61]. If we need to determine the minimum set of driver nodes independently of these factors, a large number of driver nodes would be required in most cases because there exist many states not having predecessors and, to be discussed in Sec. 4, not a small number of driver nodes are required in order to cope with even one such state.

It has been proved that the problem of finding control strategies for BNs is NP-hard even if the number of control nodes is 1 (Theorem 4 in [2]). We can use the same reduction to prove that the problem of finding the minimum set of driver nodes is NP-hard. To this end, it is enough to let x_1 in the proof to be a constant node that always outputs 0. Then, there exists a satisfiable assignment if and only if $\{x_1\}$ is the minimum set of driver nodes. Therefore, we have:

Proposition 1. *Minimum Set of Driver Nodes for Boolean networks is NP-hard.*

It is to be noted that although the minimum driver set problem is NP-hard for BNs, the problem for structural controllability of linear systems can be solved in polynomial time because the problem can be reduced to the maximum matching problem for bipartite graphs [58]. It should also be noted that the Boolean values of the driver nodes are updated synchronously. Therefore, there is no ordering on the driver nodes, which also holds in the case of structural controllability of linear systems [58, 61].

Here, we briefly explain the differences between our proposed model and the attractor-based control methods [4, 63, 70], using the example given in the above. Recall that $\{v_1\}$ is the set of driver nodes in our model. $\{v_1\}$ is also the set of driver nodes in each of the attractor-based control methods, regardless of the initial and target states. However, if $\mathbf{v}^0 = (0,0,0,1,1)$, $\mathbf{v}^M = (0,0,0,0,0)$, and $M = 1$, we do not need any driver node (i.e., the minimum number of driver nodes is 0). On the other hand, if $\mathbf{v}^0 = (0,0,0,0,0)$, $\mathbf{v}^M = (1,1,1,1,1)$, and $M = 1$, we need to control all nodes (i.e., the minimum number of driver nodes is 5). It is to be noted that we cannot specify $\mathbf{u}(0)$ but can only specify $\mathbf{u}(1)$ because $\mathbf{v}(0) = \mathbf{v}^0$ must be satisfied in our definition. These examples show that the minimum set of driver nodes depends on the initial and target states and the number of time steps, whereas the set of driver nodes is uniquely determined in each of the attractor-based control methods. These also show that there exist cases in which we need more nodes than those given by the attractor-based control methods if the number of time steps to reach the target state is restricted. Therefore, we cannot directly compare our model with the attractor-based control methods. Although the attractor-based control methods have many practical merits, our model might be useful if there exist some restrictions on the number of time steps (e.g., duration of therapy) and/or the target state is not an attractor (but can be a new stable state due to driver nodes) of a given BN.

3. Integer Linear Programming-Based Method

In this section, we elucidate an integer programming-based method for our proposed problem. Integer programming [77] is a mathematical optimization program in which some or all variables are restricted to be integers. When it refers to ILP [72], all of the objective function and constraints are linear. Since ILP has been shown to be very powerful in solving a variety of NP-hard problems [36] and the minimum driver set problem for BNs is NP-hard, it is reasonable to apply ILP.

We employ the framework introduced in [6] although considerable modifications are needed in order to select the minimum set of driver nodes. To give the ILP formalization, we need some definitions, many of which have been introduced in [6].

We define two functions $\sigma_b(x)$ and $\tau_b(x)$ by

$$\sigma_b(x) = \begin{cases} x & \text{if } b = 1, \\ \bar{x} & \text{otherwise,} \end{cases} \quad \text{and} \quad \tau_b(x) = \begin{cases} x & \text{if } b = 1, \\ 1 - x & \text{otherwise,} \end{cases}$$

where $b \in \{0,1\}$. Note that any Boolean function with k inputs can be written as

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

To ensure that $x_i = f_i(x_{i_1}, \ldots, x_{i_k})$ holds for every node v_i, we need to do two things:

(1) We need constraints

$$x_i \le \sum_{b_{i_1},\ldots,b_{i_k} \in \{0,1\}^k} x_{i,b_{i_1}\cdots b_{i_k}},$$

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

which ensure any obtained feasible binary solution equals to the value of v_i. If $f_i(x_{i_1}, \ldots, x_{i_k}) = 0$, we simply consider $x_{i,b_{i_1},\ldots,b_{i_k}} = 0$. The definition of $x_{i,b_{i_1}\cdots b_{i_k}}$ is given in (2) as follows.

(2) Ensure that $x_{i,b_{i_1}\cdots b_{i_k}} = 1$ (or 0) is equivalent to $f_i(b_{i_1}, \ldots, b_{i_k}) \wedge \sigma_{b_1}(x_{i_1}) \wedge \cdots \wedge \sigma_{b_k}(x_{i_k}) = 1$ (or 0). We add constraints

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

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

where τ is defined at the beginning of this section.

Moreover, let $x_{i,t}$ represent Boolean values $v_i(t)$. Use $x_{i,t,b_1\cdots b_k}$ to represent the corresponding value of $x_{i,b_1\cdots b_k}$ at time step t. Let $x_{i,t} = y_{i,t}$ if $w_i = 0$ and $x_{i,t} = z_{i,t}$ if $w_i = 1$. In this representation, $w_i = 1$ corresponds to the case that x_i is selected as a control node, to which $z_{i,t}$ gives 0–1 assignment for $x_{i,t}$. These constraints can be represented by

$$y_{i,t} - w_i \le x_{i,t} \le y_{i,t} + w_i,$$

$$z_{i,t} - (1 - w_i) \le x_{i,t} \le z_{i,t} + (1 - w_i).$$

Note that we aim at minimizing $\sum_{i=1}^N w_i$ which gives the number of control nodes.

By putting all the constraints together, we have the following ILP-formulation for Minimum Driver Nodes Set problem.

Minimize $\sum_{i=1}^N w_i$,
Subject to

$x_{i,t+1,b_{i_1}\cdots bi_k} \ge (\sum_{j=1}^k \tau_{b_{i_j}}(x_{i_j,t})) - (k-1)$,
$x_{i,t+1,b_{i_1}\cdots bi_k} \le \frac{1}{k} \sum_{j=1}^k \tau_{b_{i_j}}(x_{i_j,t})$
for all $i \in [1, \ldots, N]$, $t \in [0, \ldots, M-1]$ and $b_{i_1} \cdots bi_k \in \{0,1\}^k$

s.t. $f_i(b_{i_1}, \ldots, bi_k) = 1$,

$x_{i,t+1,b_{i_1}\cdots bi_k} = 0$

for all $i \in [1, \ldots, N]$, $t \in [0, \ldots, M-1]$ and $b_{i_1} \cdots bi_k \in \{0,1\}^k$

s.t. $f_i(b_{i_1}, \ldots, bi_k) = 0$,

$y_{i,t} \leq \sum_{b_{i_1}\cdots bi_k \in \{0,1\}^k} x_{i,t,b_{i_1}\cdots bi_k}$,

$y_{i,t} \geq \frac{1}{2^k} \sum_{b_{i_1}\cdots bi_k \in \{0,1\}^k} x_{i,t,b_{i_1}\cdots bi_k}$

for all $i \in [1, \ldots, N]$, $t \in [0, \ldots, M]$,

$y_{i,t} - w_i \leq x_{i,t} \leq y_{i,t} + w_i$,

$z_{i,t} - (1 - w_i) \leq x_{i,t} \leq z_{i,t} + (1 - w_i)$

for all $i \in [1, \ldots, N]$, $t \in [0, \ldots, M]$,

$x_{i,t}, y_{i,t}, z_{i,t}, w_i \in \{0,1\}$

for all $i \in [1, \ldots, N]$,

$x_{i,0} = \mathbf{v}_i^0$, $x_{i,M} = \mathbf{v}_i^M$,

for all $i \in [1, \ldots, N]$,

$x_{i,t,b_{i_1}\cdots bi_k} \in \{0,1\}$

for all $i \in [1, \ldots, N]$, $t \in [0, \ldots, M]$ and $b_{i_1} \cdots b_{i_k} \in \{0,1\}^k$.

4. Theoretical Analysis

The purpose of this section is to analyze the distribution of the minimum number of driver nodes for random Boolean networks (with in-degree bound) and for random initial states. However, it is quite difficult to obtain a general result because no simple property is known for characterizing the minimum set of driver nodes, different from the case of linear structural controllability studies [58, 61]. Therefore, we focus on cases with a small number (e.g., 1 and 2) of time steps between the initial and target states.

Recall that we denote the number of all nodes as N, the number of control nodes as n, and the number of time steps needed from the initial state $\mathbf{v}^0$ to the target state $\mathbf{v}^M$ as M. Although the set of driver nodes depends on $\mathbf{v}^0$, $\mathbf{v}^M$, and M, we assume here for simplicity that it is independent of $\mathbf{v}^0$ and $\mathbf{v}^M$. Fix the initial state. Then, the number of possible control sequences is $(2^n)^M$ because the number of possible 0–1 assignments to the driver nodes is 2^n at each time step. Since there exist 2^N possible target states, in order to drive the BN to any specified target state, we have

$$(2^n)^M \geq 2^N. \tag{2}$$

However, it just gives a lower bound on n with respect to N and M, and we may have a smaller bound if the set of driver nodes depends on $\mathbf{v}^0$ and $\mathbf{v}^M$.

By analyzing Eq. (2) we can get: for a fixed N, n decreases as M increases; for a fixed M, n increases as N increases.

In this section, we show that the minimum number of control nodes n has an upper bound for fixed $M = 1$ and $M = 2$, and a lower bound for a fixed N, where the set of control nodes depends on $\mathbf{v}^0$ and $\mathbf{v}^M$. We find two functions $u(N, L)$

(Eq. (16)) and $l(N)$ (Eq. (18)) as the upper and lower bounds estimation of the minimum number of control nodes.

4.1. *Upper bound of n*

In the following, we will show upper bounds of the number of control nodes n for cases $M = 1$ and $M = 2$.

4.1.1. *Case of $M = 1$*

When $M = 1$ which means the desired state is required to be achieved in one time step, we need to compare $\mathbf{f}(\mathbf{v}^0)$ and $\mathbf{v}^1$. Let $\mathbf{v}_i$ denote the ith coordinate value of $\mathbf{v}$. Only if $\mathbf{f}_i(\mathbf{v}^0) \neq \mathbf{v}_i^1$ we will select v_i as a control node. Since the probability of $\mathbf{f}_i(\mathbf{v}^0) \neq \mathbf{v}_i^1$ is $\frac{1}{2}$ for a random BN and a random initial state (including states of control nodes), the expected number of nodes selected as control nodes is upper bounded by $\sum_{i=0}^{N} i\binom{N}{i}(\frac{1}{2})^N = \frac{N}{2}$.

4.1.2. *Case of $M = 2$*

When $M = 2$ which means the desired state is required to be achieved at $t = 2$, we need to consider two possibilities: the target state has or does not have predecessors in the state transition diagram. Figure 2 shows how we select and assign values to control nodes. Here we assume: (1) predecessors are distributed uniformly at

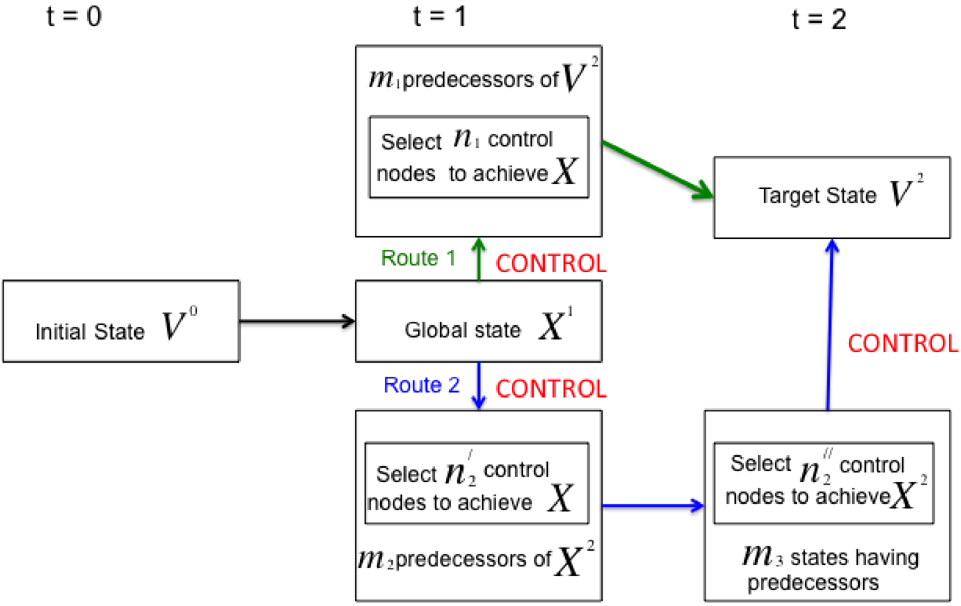

t = 0
t = 1
t = 2
m_1 predecessors of V^2
Select n_1 control nodes to achieve X
Target State V^2
Route 1
CONTROL
Initial State V^0
Global state X^1
Route 2
CONTROL
CONTROL
Select $n_2^{/}$ control nodes to achieve X
m_2 predecessors of X^2
Select $n_2^{//}$ control nodes to achieve X^2
m_3 states having predecessors

Fig. 2. Selection and value assignment of control nodes when $M = 2$.

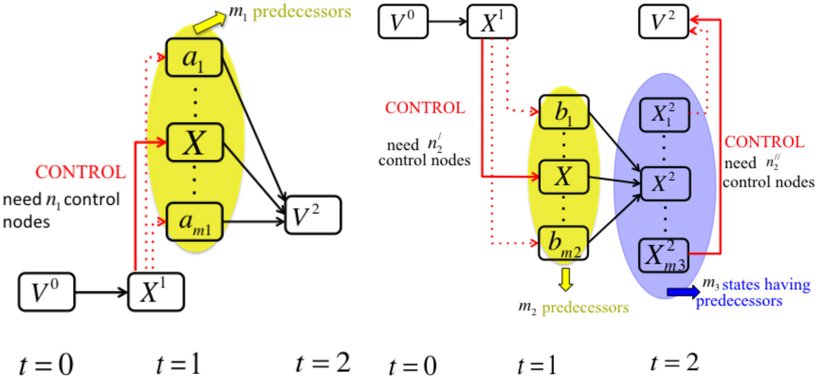
random; (2) once a state has predecessors, the number of its predecessors takes the expected number, i.e., we will have $m_1 = m_2$ in the following discussion; (3) there are m_3 states having predecessors in a BN.

As shown in Fig. 2, Routes 1 and 2 indicate the ways how we select and assign values to control nodes when the target state has and does not have predecessors, respectively.

In Route 1, which occurs when the target state has predecessors, the number of predecessors is denoted as m_1. Denote the global state this BN achieves automatically at $t = 1$ as $\mathbf{X}^1$. We shall select a state among m_1 predecessors from $\mathbf{X}^1$ to which we need the least number of control nodes n_1.

However, if the target state does not have predecessors, we execute Route 2. In this case, we should select control nodes in two steps. Firstly, we shall select one state that has predecessors. According to [3], the number of such states m_3 is at most 2^{N+1-L} where $L = \frac{N}{2^{2^K}+1}$. We shall select the one from which to the target state we need the least control nodes, denoted as $\mathbf{X}^2$. The least number of control nodes selected in this step is denoted as n''_2. Secondly, suppose $\mathbf{X}^2$ has m_2 predecessors, we shall select the predecessors from $\mathbf{X}^1$ to which we need the least control nodes. The least number of control nodes selected in this step is denoted as n'_2. The total least control nodes is the sum of both parts, i.e., $n_2 = n'_2 + n''_2$.

Here follows two examples to illustrate the above two routes (see Fig. 3). Figure 3(a) shows the example of Route 1, i.e., the route when the target state $\mathbf{V}^2$ has predecessors (we use $\mathbf{V}$ instead of $\mathbf{v}$). As we assumed, $\mathbf{V}^2$ has m_1 predecessors. The initial state $\mathbf{V}^0$ goes to $\mathbf{X}^1$ by Boolean functions. The case when we need to control is that $\mathbf{X}^1$ cannot go to any of those m_1 predecessors. We can control $\mathbf{X}^1$ to become any state among $\mathbf{a}_1, \ldots, \mathbf{a}_{m_1}$, but finally we will select $\mathbf{X}$ from $\mathbf{a}_1, \ldots, \mathbf{a}_{m_1}$



(a) Example of Route 1

(b) Example of Route 2

Fig. 3. (Color online) Examples of Routes 1 and 2 when $M = 2$. Rectangles, black arrows with solid line, red arrows with dotted line, and red arrows with solid line are used to denote global states, state transition through Boolean functions, possible control between states, and the control we finally execute, respectively.

so that we need the least number of control nodes n_1 to control $\mathbf{X}^1$ to $\mathbf{X}$ (by inverting the states of n_1 nodes). Figure 3(b) shows the example of Route 2, i.e., the route when $\mathbf{V}^2$ does not have predecessors. In this case, at the final time $t = 2$ we should control some node to $\mathbf{V}^2$. According to our assumption, there are m_3 states in a BN that have predecessors, namely, $\mathbf{X}_1^2, \ldots, \mathbf{X}_{m_3}^2$. We can control any of them to become $\mathbf{V}^2$. We shall select $\mathbf{X}^2$ from these states so that the least number of control nodes are needed to control $\mathbf{X}^2$ to $\mathbf{V}^2$ (by inverting the states of n_2'' nodes). Analogously, $\mathbf{X}^2$ is supposed to have m_2 predecessors as we assumed. Since the state is $\mathbf{X}^1$ at $t = 1$, we shall control $\mathbf{X}^1$ to become one of the predecessors of $\mathbf{X}^2$, namely, $\mathbf{b}_1, \ldots, \mathbf{b}_{m_2}$. Then $\mathbf{X}$ will be selected from these states so that the least number of control nodes are needed to control $\mathbf{X}^1$ to be $\mathbf{X}$ (by inverting the states of n_2' nodes).

Proposition 2. *For a random BN of N nodes, in-degree K, if we want to change a state to be one of m states at the same time by control nodes, the minimum number of control nodes is expected to be $(\frac{1}{2})^{Nm} \sum_{i=0}^{N-1} (\sum_{k=i+1}^{N} \binom{N}{k})^m$.*

Proof. If we want to change a state to be another state immediately, nodes on which values are different will be selected as control nodes. The number of control nodes X is a binomially distributed random variable and $X \sim B(N, \frac{1}{2})$.

Note that if we denote the number of control nodes as X_i for the ith predecessor $i \in [1, m], i \in \mathbb{N}$, then $X_1, \ldots, X_m$ are i.i.d. random variables from a discrete distribution with cumulative distribution function $F(x)$ and probability mass function $f(x)$ where

$$F(x) = \sum_{k=0}^{x} \binom{N}{k} \left(\frac{1}{2}\right)^N, \quad f(x) = \binom{N}{x} \left(\frac{1}{2}\right)^N. \tag{3}$$

The first-order statistics $X_{(1)}$ is always the minimum of the samples, namely

$$X_{(1)} = \min\{X_1, \ldots, X_m\}. \tag{4}$$

Here, we give the derivation of $\Pr(X_{(1)} = x)$. Let

$$\begin{aligned} p_1 &= \Pr(X > x) = 1 - F(x), \\ p_2 &= \Pr(X = x) = f(x), \\ p_3 &= \Pr(X < x) = F(x) - f(x). \end{aligned}$$

Note that

$$\begin{aligned} \Pr(X_{(1)} \le x) &= \Pr(\text{there are at most } m-1 \text{ trails greater than } x) \\ &= \sum_{i=0}^{m-1} \binom{m}{i} {p_1}^i (p_2 + p_3)^{m-i}, \end{aligned}$$

and

$$\begin{aligned}\Pr(X_{(1)} < x) &= \Pr(\text{there are at most } m-1 \text{ trails greater than or equal to } x)\\ &= \sum_{i=0}^{m-1} \binom{m}{i} (p_1+p_2)^i (p_3)^{m-i}.\end{aligned}$$

Thus, the probability mass function of $X_{(1)}$ can be computed by

$$\begin{aligned}\Pr(X_{(1)} = x) &= \Pr(X_{(1)} \leq x) - \Pr(X_{(1)} < x)\\ &= \sum_{i=0}^{m-1} \binom{m}{i} ({p_1}^i (p_2+p_3)^{m-i} - (p_1+p_2)^i (p_3)^{m-i})\\ &= \sum_{i=0}^{m=1} \binom{m}{i} ((1-F(x))^i (F(x))^{m-i}\\ &\quad - (1-F(x)+f(x))^i (F(x)-f(x))^{m-i})\\ &= \left(\frac{1}{2}\right)^{Nm} \left(\left(2^N - \sum_{k=0}^{x-1} \binom{N}{k}\right)^m - \left(2^N - \sum_{k=0}^{x} \binom{N}{k}\right)^m\right). \end{aligned}\tag{5}$$

Then, we can obtain the expectation value of $X_{(1)}$ by computing

$$\begin{aligned}E[X_{(1)}] &= \sum_{x=0}^{N} x\Pr(X_{(1)} = x)\\ &= \left(\frac{1}{2}\right)^{Nm} \left(\sum_{x=0}^{N} x \left(\left(2^N - \sum_{k=0}^{x-1} \binom{N}{k}\right)^m - \left(2^N - \sum_{k=0}^{x} \binom{N}{k}\right)^m\right)\right)\\ &= (2^N - C_N^0)^m - \left(2^N - \sum_{k=0}^{1} \binom{N}{k}\right)^m\\ &\quad + 2\left(2^N - \sum_{k=0}^{1} \binom{N}{k}\right)^m - 2\left(2^N - \sum_{k=0}^{2} \binom{N}{k}\right)^m\\ &\quad + 3\left(2^N - \sum_{k=0}^{2} \binom{N}{k}\right)^m - 3\left(2^N - \sum_{k=0}^{3} \binom{N}{k}\right)^m\\ &\qquad \vdots\\ &\quad + N\left(2^N - \sum_{k=0}^{N-1} \binom{N}{k}\right)^m - N\left(2^N - \sum_{k=0}^{N} \binom{N}{k}\right)^m\\ &= \left(\frac{1}{2}\right)^{Nm} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k}\right)^m. \end{aligned}\tag{6}$$

$\square$

Proposition 3. *Suppose that for a random BN of N nodes, and for each node K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible Boolean functions (including constant Boolean functions). Then the minimum number of control nodes for time step $M = 2$ is expected to be at most*

$$P_1 \left(\frac{1}{2}\right)^{Nm_1} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_1} + P_2 \left(\left(\frac{1}{2}\right)^{Nm_2} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_2} + \left(\frac{1}{2}\right)^{Nm_3} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_3} \right),$$

where $P_1 \leq \frac{2}{2^L}$, $P_2 \geq \frac{2^L-2}{2^L}$, $m_1 \geq 2^{L-1}$, $m_2 \geq 2^{L-1}$, $m_3 \leq 2^{N+1-L}$, and $L = \frac{N}{2^{2^K}+1}$.

Proof. Since there are two possibilities to achieve $\mathbf{V}^2$ from $\mathbf{V}^0$, we need to divide the problem into two cases.

Case 1. The target state has predecessors. We need to select a global state X from m_1 predecessors by changing which to X^1 we need the least control nodes. By Proposition 2, the minimum number of control nodes is expected to be

$$n_1 = \left(\frac{1}{2}\right)^{Nm_1} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_1}.$$

Case 2. The target state has no predecessors. In this case, the selection and value assignment of control nodes include two steps:

First, select a global state X^2 changing which to $\mathbf{V}^2$ we need the least control nodes among m_3 states having predecessors. The minimum number of control nodes is expected to be

$$n_2'' = \left(\frac{1}{2}\right)^{Nm_3} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_3}.$$

Second, select a global state X among m_2 predecessors of X^2 changing which to X^1 we need the least control nodes. The minimum number of control nodes is expected to be

$$n_2' = \left(\frac{1}{2}\right)^{Nm_2} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_2}.$$

Note that the probabilities of Cases 1 and 2 occur are P_1 and P_2, respectively. Then, overall the minimum number of control nodes is

$$\begin{aligned} n &= P_1 n_1 + P_2(n_2' + n_2'') \\ &= P_1 \left(\frac{1}{2}\right)^{Nm_1} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_1} + P_2 \left(\left(\frac{1}{2}\right)^{Nm_2} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_2} \right. \\ &\quad \left. + \left(\frac{1}{2}\right)^{Nm_3} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m_3} \right). \end{aligned} \tag{7}$$

The range of P_1, P_2, m_1, m_2, and m_3 can be obtained from [3]. □

Proposition 4. *Suppose that for a random BN of N nodes, and for each node K input nodes are randomly selected and then a Boolean function is randomly selected from 2^{2^K} possible Boolean functions (including constant Boolean functions). Assume that m_1 and m_2 equal to the expectation value of number of predecessors and m_3 equals to the expectation value of number of states having predecessors. Then the minimum number of control nodes for time step $M = 2$ is expected to be at most $\sum_{i=0}^{N-1}((\frac{1}{2})^N \sum_{k=i+1}^{N} \binom{N}{k})^{2^{L-1}} + \sum_{i=0}^{N-1}((\frac{1}{2})^N \sum_{k=i+1}^{N} \binom{N}{k})^{2^{N+1-L}}$ where $L = \frac{N}{2^{2^K}+1}$.*

Proof. Let

$$g(m) = \left(\frac{1}{2}\right)^{Nm} \sum_{i=0}^{N-1} \left(\sum_{k=i+1}^{N} \binom{N}{k} \right)^{m}. \tag{8}$$

Then $g(m)$ is decreasing since Eq. (8) can be written as

$$g(m) = \sum_{i=0}^{N-1} \left(\left(\frac{1}{2}\right)^{N} \sum_{k=i+1}^{N} \binom{N}{k} \right)^{m}$$

and

$$0 < \left(\frac{1}{2}\right)^{N} \sum_{k=i+1}^{N} \binom{N}{k} < 1, \quad \text{for } i = 0, \ldots, N-1,\ N \in \mathbb{N}^{+}.$$

Note that in Eq. (7), based on the assumption we have

$$m_1 \cdot m_3 = 2^N \tag{9}$$

and

$$m_2 \cdot m_3 = 2^N. \tag{10}$$

We also know

$$P_1 + P_2 = 1. \tag{11}$$

Substituting Eqs. (9)–(11) to Eq. (7) we obtain

$$\begin{aligned} n &= \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\left(\sum_{k=i+1}^{N}\binom{N}{k}\right)\right)^{m_1} + P_2\sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\left(\sum_{k=i+1}^{N}\binom{N}{k}\right)\right)^{\frac{2^N}{m_1}} \\ &= g(m_1) + P_2 g\left(\frac{2^N}{m_1}\right) \le g(m_1) + g\left(\frac{2^N}{m_1}\right). \end{aligned} \tag{12}$$

Since $g(m_1)$ is decreasing much faster than $g(\frac{2^N}{m_1})$ where $\frac{2^N}{m_1}$ is big enough to make the function value stable at comparatively much smaller values for $m_1 \ge 2^{L-1}$, Eq. (12) achieves its maximum when m_1 equals to its minimum. Thus, we have

$$\begin{aligned} g(m_1) + g\left(\frac{2^N}{m_1}\right) &\le g(2^{L-1}) + g(2^{N+1-L}) \\ &= \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{L-1}} \\ &\quad + \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{N+1-L}}. \end{aligned} \tag{13}$$

That is to say, the minimum number of control nodes is expected to be

$$n \le \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{L-1}} + \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{N+1-L}}. \tag{14}$$

□

Note that the second term of Eq. (14) becomes very small and can be ignored when $N > 3$, and thus we can approximate Eq. (14) by

$$n \le g(m_1) \le g(2^{L-1}) = \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{L-1}}. \tag{15}$$

Therefore, the function

$$u(N,L) = \sum_{i=0}^{N-1}\left(\left(\frac{1}{2}\right)^N\sum_{k=i+1}^{N}\binom{N}{k}\right)^{2^{L-1}} \tag{16}$$

approximately gives an upper bound for the number of control nodes n.

4.2. *Lower bound function of n*

As we have mentioned at the beginning of this section, it is expected that the number of control nodes n approaches to 0 as time step M increases where the number of all nodes N is fixed. However, ILP results show that the number of

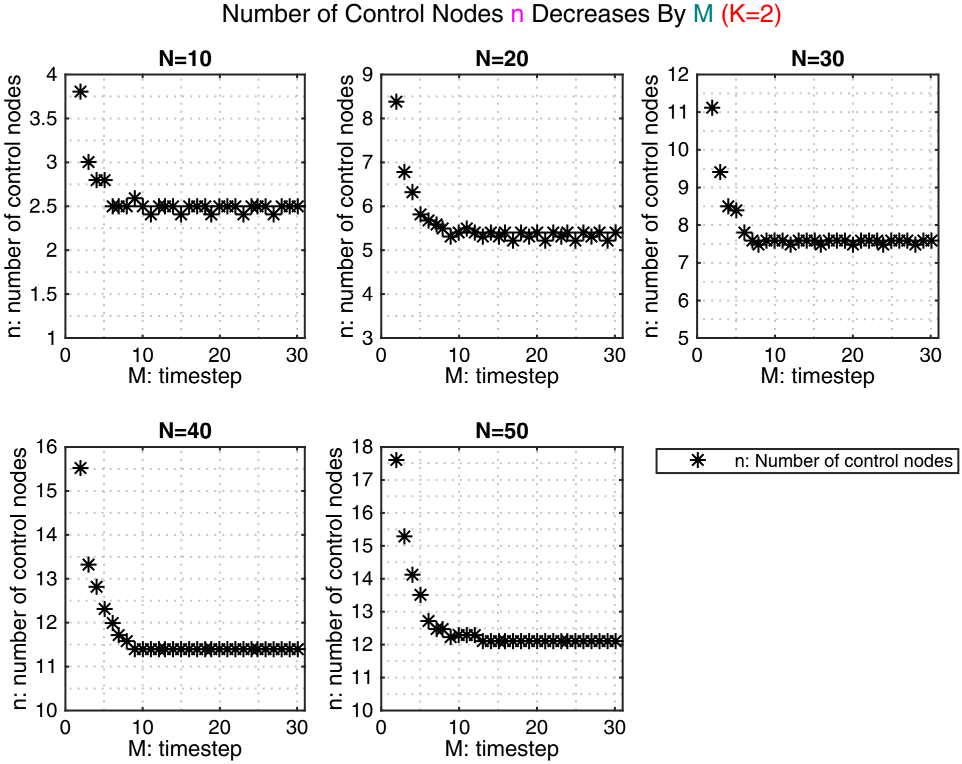


Fig. 4. How n changes by M when $K = 2$.

control nodes n will ultimately approach to a positive constant (See Figs. 4 and 5). Hence, we try to find out the relationship between this constant and the number of all nodes N. In the following, we called this constant as *stable* n.

Here we only consider the case $K = 2$. In this case, there are $2^{2^2} = 16$ Boolean functions to each node with 2 inputs. We shall discuss the following two cases.

Case 1. Boolean functions are only constant Boolean functions.

Constant Boolean functions are 1 or 0 which means the corresponding node is assigned values 1 or 0 always. Recall that our problem is to control a network from a random initial state to the global state $(0, \ldots, 0)$, without loss of generality. In this case, only when a node (assumed as y) is assigned the constant Boolean function 1 does it need to be controlled. Then the probability of one node needs to be controlled is $\frac{1}{2}$. Note that this case happens with probability $\frac{2}{16}$ (two out of 16 Boolean functions are constant) and the BN has N nodes, the expected number of control nodes is

$$n_1 = \frac{2}{16} \times \frac{1}{2} \times N = \frac{N}{16}.$$

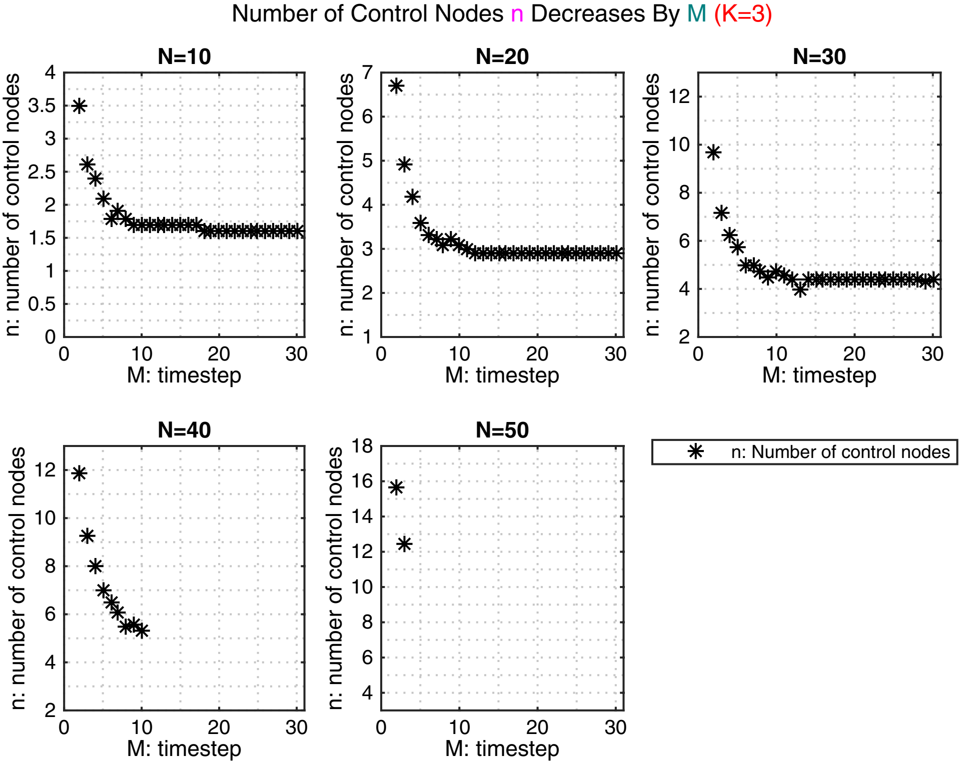


Fig. 5. How n changes by M when $K = 3$.

Case 2. Only non-constant Boolean functions are assigned to nodes.

In this case, there are 14 nonconstant Boolean functions that can be assigned to nodes. Here, we shall consider pairs of nodes because nodes might influence each other when they have same input nodes. We analyze the following two cases.

Subcase 1. Any two nodes have two same input nodes.

Assume nodes y and z, they have two same input nodes x_j and x_k, then totally there are $14^2 = 196$ possible combinations of Boolean functions assigned to y and z because we only consider nonconstant Boolean functions here. All possibilities are shown in the following Tables A.1 and A.2.

As we can see from Tables A.1 and A.2, we need to control either the node y or z in 50 out of all 196 possible combinations. Denote the possibility that a node needs to be controlled as p_1 in this case, then

$$p_1 = \frac{\binom{2}{2}}{\binom{N}{2}} \times \frac{50}{196} = \frac{25}{49N(N-1)},$$

where $\binom{2}{2}/\binom{N}{2}$ is the probability that two nodes have two same input nodes and $\frac{50}{196}$ is the probability that a pair of such nodes with only nonconstant Boolean functions needs to be controlled.

Subcase 2. Any two nodes have only one same input node.

If any two nodes y and z have only one same input node, then only in this circumstances we have to control some node(s): $y(t+1) = \bigcirc \bigvee x_k$ and $z(t+1) = \bar{x}_k \bigvee \bigcirc$ where x_k is the same input node, and $\bigcirc$ can be empty, x_j, $\bar{x}_j (j \neq k)$, or; $y(t+1) = \bigcirc \bigvee \bar{x}_k$ and $z(t+1) = x_k \bigvee \bigcirc$ where x_k is the same input node, and $\bigcirc$ can be empty, x_j, $\bar{x}_j (j \neq k)$.

Analogously, we can learn from Tables A.1 and A.2 that there are $9 + 9 = 18$ combinations of Boolean functions where we need to control either y or z. Denote the possibility that a node needs to be controlled as p_2, we have

$$p_2 = \frac{\binom{2}{1}\binom{N-2}{1}}{\binom{N}{2}} \times \frac{18}{196} = \frac{18(N-2)}{49N(N-1)},$$

where $\frac{\binom{2}{1}\binom{N-2}{1}}{\binom{N}{2}}$ is the probability that two nodes have only one same input node and $\frac{18}{196}$ is the probability that such a pair of nodes with only nonconstant Boolean functions needs to be controlled.

Since there are $\binom{N}{2}$ pairs of nodes and this case occurs with probability $\frac{14}{16}$, the expected number of control nodes is

$$\begin{aligned} n_2 &= (p_1 + p_2) \times \binom{N}{2} \\ &= \left(\frac{25}{49N(N-1)} + \frac{18(N-2)}{49N(N-1)}\right) \times \binom{N}{2} = \frac{18N-11}{98}. \end{aligned} \tag{17}$$

To sum up, the expected number of control nodes needed is at least $n_1 + n_2$, denoted as a new function $l(N)$.

Therefore,

$$l(N) = n_1 + n_2 = \frac{N}{16} + \frac{18N-11}{98}. \tag{18}$$

Since the case that constant functions and nonconstant functions can all be assigned to nodes is not included here, the $l(N)$ can only provide an estimate of the lower bound of minimum number of control nodes (See Table 2).

Table 2. Comparison between lower bound function $l(N)$ and stable n in simulation.

N	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150
Stable n	2.3	5.6	8.2	11.8	12.6	15.7	17.6	21.8	23.5	27.4	28.4	32.5	36.1	38.8	39.2
$l(N)$	2.3	4.8	7.3	9.7	12.2	14.7	17.1	19.6	22.0	24.5	27.0	29.4	31.9	34.4	36.8

5. Computational Experiment

5.1. *How n changes by N (fixed M, K)*

We implement the ILP problem with different in-degree upper bound K, and different time steps M. We start with random initial states and the target state is $(0, 0, \ldots, 0)$.

The curve of Eq. (16) is called as *Upper Bound Curve*. Figure A.1 shows the comparison among the upper bound curves and ILP implementation results for the case of $M = 2$ with different K. We can see that Eq. (16) gives a very good bound of the number of control nodes n for the case of $K = 2$.

However, the upper bound curve is less tight when K increases. The reason is as follows. On one hand, note that in Eq. (12) $n \leq g(m_1) + g(\frac{2^N}{m_1})$. When $K \uparrow$ we have $m_1 \downarrow$, $\frac{2^N}{m_1} \uparrow$, then $g(m_1) \uparrow$, $g(\frac{2^N}{m_1}) \downarrow$, and $|\Delta g(\frac{2^N}{m_1})| < |\Delta g(m_1)|$ (because: (1) $m_1 < \frac{2^N}{m_1}$; (2) although $g(m)$ is a discrete function, it has a convex shape as shown in Fig. A.2). Here, Δ is defined as: $\Delta g(m) = g(m) - g(m + \Delta m), m \in \mathbb{R}$. Therefore, the upper bound estimate of n will increase when $K \uparrow$ (for fixed N, $M = 2$). On the other hand, the computer experiment results show that $n \downarrow$ when $K \uparrow$ (for fixed N, $M = 2$). As a result, the upper bound curve is less tight for $K = 3, 4, \ldots$. However, things will not get worse. The upper bound curve is almost the same for $K = 4, 5, \ldots$ because the power 2^{L-1} ($L = \frac{N}{2^{2^K}+1}$) in Eq. (16) almost stays the same (at around 0.5000) when $K \geq 4$. Furthermore, we fit the dots of (N, n) for fixed K and M with curves, as shown in Figs. A.3 and A.4. We find that n approximately follows a linear relationship with N for $K = 2$ and $K = 3$. The slopes of these fitting lines decrease as the number of time steps M increases.

5.2. *How n changes by M (fixed N, K)*

From Figs. A.3 and A.4, we can find that the slopes of those fitting lines decrease as the number of time steps M increases, for fixed K. This result is in accordance with our theoretical analysis in Sec. 4.2. Figures 4 and 5 show the relationship between n and M with a fixed N more specifically. Here we post further explanations of the ILP implementation results. We catch global states at each time t and the corresponding number of control nodes newly selected at that time. We find some rules of the selection of control nodes, e.g., the selection of control nodes finishes in rather early several time steps; and in most of cases some control nodes are selected at final step, which means the target state has no predecessor.

We set the times of repetition as 100. We apply the average of the results of 100 randomly generated networks. We try BNs with 10, 20 nodes. Then we get some conclusions from computer simulation results as follows:

(1) Control nodes are selected at very limited time steps. Take $K = 2$, $N = 10$, $M = 10$ as an example. Control nodes are selected at up to three time step. That is, if we denote n_i as the number of control nodes newly selected at time $t = i$, then

we have only up to three $n_i \neq 0$. Note that this n_i is different from the number of control nodes needed in the ith time step.
(2) The time steps mentioned above are almost early time steps. Take $K = 2$, $N = 10$, $M = 10$ as an example, in 91% iterations $n_3 \sim n_9 = 0$, in 96% iterations $n_4 \sim n_9 = 0$, and in 92% iterations $n_{10} = 0$.
(3) After finishing the selection of control nodes (note that the control nodes selected may be in use for more than one time step), it takes only 3 to 5 more time steps for the Boolean network to achieve the target state. Therefore, the target state can be achieved within around 8 time steps. In other time steps, the networks states are just repetitive (the same or in a loop). This explains why curves stabilize from around $M = 8$ in Figs. 4 and 5.

5.3. *Simulation for realistic BNs*

For further evaluation of our algorithm as well as our analysis on the size of the minimum set of driver nodes, we examined six realistic BNs from literatures [15, 23, 25, 26, 43, 52, 62]. We applied our algorithm to control the above realistic BNs from a random initial state to any one of attractor states. The number (#) of control nodes describes the nodes needed to drive the realistic BN from a random initial state to an attractor state for the case $M = 2$. We took the average over 10 repetitions since the set of driver nodes depends on the initial state. The upper bound function $u(N)$ is from Eq. (16) (note that $L = \frac{N}{2^{2^K}+1}$, and $u(N,L)|_{K=2} < u(N,L)|_{K=3}$, therefore we apply $u(N) = u(N,L)|_{K=2}$ here). The lower bound function is from Eq. (18). The simulation results are shown in Tables 3–8.

As can been seen from Tables 3–8, our algorithm and the analysis in the minimum set of control nodes work very well even for realistic BNs. The upper bound function and lower bound function give very good boundaries for the minimum number of driver nodes. More importantly, the genes most frequently identified as driver nodes (shown in table captions) are biologically meaningful. In flower morphogenesis in arabidopsis thaliana network, genes UFO, WUS and AP1 are most frequently selected. Since UFO is related to ectopic activation, and WUS and AP1 are related to meristem, the obtained control genes seem to have important control

Table 3. Control of realistic BNs: Arabidopsis thaliana.

Attractor state	1	2	3	4	5	6	7	8	9	10	**u(15)**	**l(15)**
Number of control nodes	5.8	6.0	5.9	5.2	5.4	6.3	5.0	5.9	5.8	5.3	**7.6**	**3.6**

Notes: BN model of the control of flower morphogenesis in Arabidobis thaliana from [15]. $N = 15$. There are 10 single attractors. Nodes most frequently identified as driver nodes: v_2 = UFO: when expressed constitutively, Unusual Floral Organs (UFO) causes ectopic activation of AP3 within flowers; v_9 = WUS: WUSCHEL gene (WUS) controls meristem function by direct regulation of cytokinin-inducible response regulators [51]; v_5 = AP1: plays central roles in determining floral meristem identity [47].

Table 4. Control of realistic BNs: T-cell receptor signaling pathway.

Attractor state	1	2	3	4	5	6	7	8
Number of control nodes	14.5	15.8	15.7	14.7	13.9	14.4	15.6	15.4
Attractor state	9	10	11	12	13	14	**u**(**40**)	**l**(**40**)
Number of control nodes	14.9	15.3	15.3	14.7	14.4	14.6	**17.7**	**9.7**

Notes: BN model of the T-cell receptor signaling pathway from [43]. $N = 40$. There are 1 attractor of length 6 and 8 single attractors. Nodes most frequently identified as driver nodes: v_{14} = Gads: GRB2-related adaptor protein (Gads) is related to growth factor receptor signaling, where it couples activated receptors with downstream effector pathways [59]; v_{30} = Calcin: a key signaling enzyme in T-lympocyte activation [20]; v_{39} = NFkB: nuclear factor kappa B (NF-κB) is an inducible and ubiquitously expressed transcription factor for genes involved in cell survival, cell adhesion, inflammation, differentiation, and growth [14].

Table 5. Control of realistic BNs: Mammalian cell cycle.

Attractor state	1	2	3	4	5	6	7	8	**u**(**10**)	**l**(**10**)
Number of control nodes	3.8	4.3	3.8	3.5	3.6	3.7	4.0	3.8	**5.4**	**2.3**

Notes: BN model of the control of the mammalian cell cycle from [26]. $N = 10$. There are two attractors: one of length 7 and another one is a single attractor. Nodes most frequently identified as driver nodes: v_4 = E2F: E2F transcription factor can regulate expression of numerous cellular genes controlling proliferation, including porto-oncogenes and genes regulating cell cycle progression [82]; v_5 = CycA: cell cycle regulation of the cyclin A gene promoter is mediated by a variant E2F site [78]; v_8 = UbcH10: UbcH10 gene codes for a protein that belongs to the ubiquitin-conjugating enzyme family [79].

Table 6. Control of realistic BNs: Budding yeast cell cycle.

Attractor state	1	2	3	4	5	6	7	**u**(**12**)	**l**(**12**)
Number of control nodes	4.8	4.4	4.3	4.5	5	4.6	5.3	**6.3**	**2.8**

Notes: BN model of the control of the budding yeast cell cycle regulation from [52]. $N = 12$. There are 7 attractors of length 1. Nodes most frequently identified as driver nodes: v_2 = Cln3: the cell-cycle sequence starts when the cell commits to division by activating Cln3 [52]. v_3 = SBF: when the cell grows large enough, transcription factors including SBF is activated to trigger cyclin genes [52]; v_{11} = Cdc20/cdc14: budding yeast Cdc20 is a target of the spindle checkpoint [35].

Table 7. Control of realistic BNs: T-helper cell differentiation.

Attractor state	1	2	3	**u**(**23**)	**l**(**23**)
Number of control nodes	8.9	9.1	8.8	**11.0**	**5.5**

Notes: BN model of the control of T-helper cell differentiation from [62]. $N = 23$. There are 3 single-point attractors. Nodes most frequently identified as driver nodes: v_{16} = STAT1: one of the firstly identified (Signal Transducer and Activator of Transcription) STAT genes in the interferon (IFN) signal transduction pathways in mammalian cells. Identified as key components linking cytokine signals to transcriptional events in cells [32], STAT proteins possess the ability to transduce signals from the cell membrane to the nucleus to activate gene transcription [57]; v_{15} = JAK1: activation of the Janus kinase/signal transducers and activators of transcription (JAK/STAT) pathway by numerous cytokines and growth factors is one of the most well-studied intracellular signaling networks [42]; v_9 = IL-12: Interleukin-12 (IL-12) is an important immune regulatory cytokine that exerts potent anti-tumor activity and a member of a small family of heterodimeric cytokines [44].

Table 8. Control of realistic BNs: Fission yeast cell cycle regulation.

Attractor state	1	2	3	4	5	6	7	8
Number of control nodes	3.4	3.6	3.2	3.8	4.0	3.7	4.0	3.4
Attractor state	9	10	11	12	13	**u(10)**	**l(10)**	
Number of control nodes	4.2	3.8	3.7	3.7	3.6	**5.4**	**2.3**	

Notes: BN model of the control of the fission yeast cell cycle regulation from [23]. $N = 10$. There are 13 single-point attractors. Nodes most frequently identified as driver nodes: $v_5 =$ Rum1: an inhibitor of Cdc2/Cdc13 [79]; $v_4 =$ Cdc2/Cdc13: in lower eukaryotes, like fission yeast, a single Cdk-cyclin complex, called Cdc2/Cdc13 in fission yeast [69], can drive the whole cell cycle; $v_8 =$ Slp1: an effector of the mad2-dependent spindle checkpoint [39].

roles since morphogenesis should be related to ectopic activation and meristem. In T-cell receptor signaling pathway network, since NF-κB is known to play a central role in various types of cell activity, it is appropriate that NF-κB is included by the control process. In mammalian cell cycle network, since cyclin A is involved in the control of S phase and mitosis in mammalian cells [78] and mediated by E2F, it is likely that both E2F and CycA work as the control genes. As for T-helper cell differentiation network, since the JAK/STAT pathway plays an important role in control of cells *in vivo*, using STAT1 and JAK1 as control nodes is considered to be reasonable.

6. Conclusions

In this paper, we have studied the minimum set of driver nodes for control of BNs. We showed that the problem of computing the minimum driver set is NP-hard. In order to cope with this difficulty, we developed an ILP-based method that works efficiently for medium scale BNs (e.g., BNs with more than a hundred nodes). We also analyzed the average size of the minimum driver set for random BNs with bounded in-degree. Since it is quite hard to perform theoretical analysis for general cases, we focused on the cases where the number of time steps is one and two (i.e., $M = 1$ and $M = 2$). Although the case of $M = 1$ can be analyzed in a straightforward manner, analysis is hard even for the case of $M = 2$. Based on some assumptions, we derived an upper bound of the average number of the minimum driver set for $M = 2$. The derived bound shows a very good agreement with the results of computational experiments using the developed ILP-based method for the case of $K = 2$ (i.e., the maximum in-degree is 2) and a good agreement for the case of $K = 3$. It was observed from the results of computational experiments that the number of driver nodes approaches not to 0 but to some positive constant as the number of time steps increases. In order to explain this phenomena, we gave an estimate of the lower bound of the number of minimum driver set, which showed a good agreement with the result of a computational experiment. Although we could analyze restricted cases only, interesting findings were obtained, which include (almost) the linear increase of the minimum number of driver nodes to the total number of nodes, and the existence of a lower limit of the number of driver nodes against the increase of the number of time steps.

Although we have focused on random BNs in this paper, it is also important to study controllability of realistic BNs. To this end, we further test our algorithm as well as the analysis on the minimum set of driver nodes with 6 realistic BNs. Simulation results have shown that the upper bound and lower bound function derived in Sec. 4 not only works for mathematically constructed BNs but also works very well for realistic BNs, in which cases they gave a very good bounds for the minimum number of control nodes to drive a BN from a random initial state to a target state.

An improved computation method should be developed because the computation time of the proposed ILP-based method increases rapidly as the maximum in-degree increases. As mentioned in Sec. 2, there is no ordering among the selected driver nodes. However, giving a way to prioritize them may be useful in medical applications because it is practically impossible to control many nodes simultaneously. Therefore, prioritization methods should be studied. In this paper, the theoretical analysis of the minimum number of driver nodes was limited up to $M = 2$. Therefore, the theoretical analysis of more general cases, even for the case of $M = 3$, is left as an open problem.

Acknowledgments

TT was partially supported by JSPS, Japan (Grant-in-Aid for Young Scientists (B) 25730005 and Grant-in-Aid for Scientific Research (A) 25250028). TA was partly supported by Grant-in-Aid 26240034 from JSPS, Japan.

Appendix A

Table A.1. Nodes need to be controlled in Boolean functions combinations (1/2).

	Control								
	y								
z	$x_i = 0$	$x_i = 1$	$x_i = x_j$	$x_i = \bar{x}_j$	$x_i = x_k$	$x_i = \bar{x}_k$	$x_i = x_j \bigwedge x_k$	$x_i = x_j \bigwedge \bar{x}_k$	$x_i = \bar{x}_j \bigwedge x_k$
$x_i = 0$			y						
$x_i = 1$	z	z	y & z	z	z	z	z	z	z
$x_i = x_j$			y		y or z				
$x_i = \bar{x}_j$			y	y or z					
$x_i = x_k$			y				y or z		
$x_i = \bar{x}_k$			y			y or z			
$x_i = x_j \bigwedge x_k$			y						
$x_i = x_j \bigwedge \bar{x}_k$			y						
$x_i = \bar{x}_j \bigwedge x_k$			y						
$x_i = \bar{x}_j \bigwedge \bar{x}_k$			y						
$x_i = x_j \bigvee x_k$			y		y or z		y or z		
$x_i = x_j \bigvee \bar{x}_k$	y or z		y		y or z	y or z			
$x_i = \bar{x}_j \bigvee x_k$			y	y or z			y or z		y or z
$x_i = \bar{x}_j \bigvee \bar{x}_k$			y	y or z		y or z		y or z	
$x_i = x_j \bigoplus x_k$			y						
$x_i = x_j \bigoplus \bar{x}_k$			y						

Table A.2. Nodes need to be controlled in Boolean functions combinations (2/2).

Control							
	y						
z	$x_i = \bar{x}_j \bigwedge \bar{x}_k$	$x_i = x_j \bigvee x_k$	$x_i = x_j \bigvee \bar{x}_k$	$x_i = \bar{x}_j \bigvee x_k$	$x_i = \bar{x}_j \bigvee \bar{x}_k$	$x_i = x_j \bigoplus x_k$	$x_i = x_j \bigoplus \bar{x}_k$
$x_i = 0$							
$x_i = 1$	z	z	z	z	z	z	z
$x_i = x_j$				y or z	y or z		
$x_i = \bar{x}_j$		y or z	y or z				
$x_i = x_k$			y or z		y or z		
$x_i = \bar{x}_k$		y or z		y or z			
$x_i = x_j \bigwedge x_k$					y or z		
$x_i = x_j \bigwedge \bar{x}_k$				y or z			
$x_i = \bar{x}_j \bigwedge x_k$			y or z				
$x_i = \bar{x}_j \bigwedge \bar{x}_k$		y or z					
$x_i = x_j \bigvee x_k$	y or z		y or z	y or z	y or z		y or z
$x_i = x_j \bigvee \bar{x}_k$		y or z		y or z	y or z	y or z	
$x_i = \bar{x}_j \bigvee x_k$			y or z		y or z	y or z	
$x_i = \bar{x}_j \bigvee \bar{x}_k$		y or z	y or z	y or z			y or z
$x_i = x_j \bigoplus x_k$			y or z	y or z			y or z
$x_i = x_j \bigoplus \bar{x}_k$					y or z	y or z	

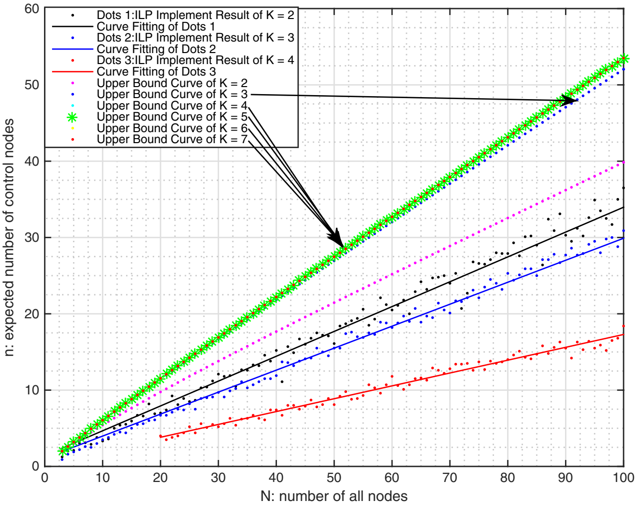


Fig. A.1. Comparison of ILP implementation results and upper bound function Eq. (16) when $M = 2$ with different K.

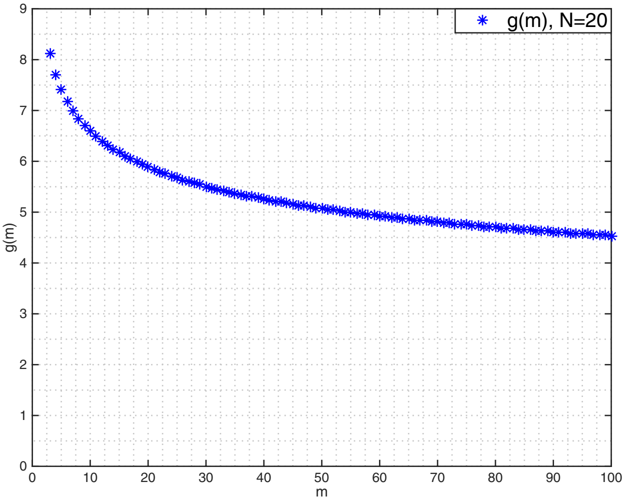


Fig. A.2. Plot of function $g(m)$ for $N = 20$.

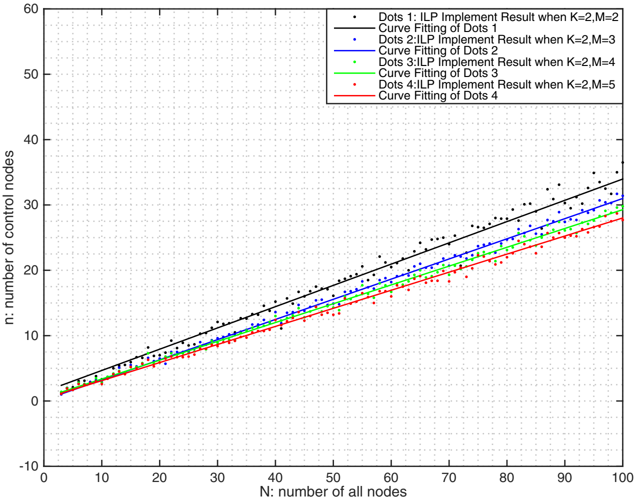


Fig. A.3. Plots of ILP results when $K = 2$.

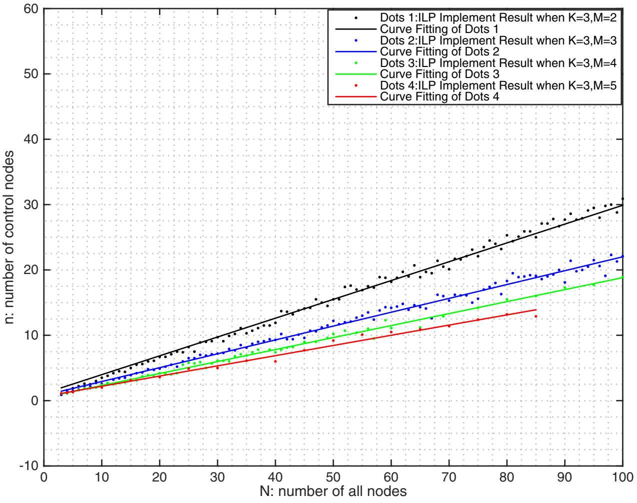


Fig. A.4. Plots of ILP results when $K = 3$.

References

[1] Abul, O., Alhajj, R. and Polat, F., Markov decision processes based optimal control policies for probabilistic boolean networks, in *Proc. Forth IEEE Symp. Bioinformatics and Bioengineering* (*BIBE 2004*) (2004), pp. 337–344.

[2] Akutsu, T., Hayashida, M., Ching, W.-K. and Ng, M. K., Control of boolean networks: Hardness results and algorithms for tree structured networks, *J. Theor. Biol.* **244** (2007) 670–679.

[3] Akutsu, T., Hayashida, M., Zhang, S.-Q., Ching, W.-K. and Ng, M. K., Analyses and algorithms for predecessor and control problems for boolean networks of bounded indegree, *IPSJ Trans. Bioinform.* **1** (2008) 23–24.

[4] Akutsu, T., Kuhara, S., Maruyama, O. and Miyano, S., A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions, *Genome Inform.* **9** (1998) 151–160.

[5] Akutsu, T., Miyano, S. and Kuhara, S., Inferring qualitative relations in genetic networks and metabolic pathways, *Bioinformatics* **16** (2000) 727–734.

[6] Akutsu, T., Zhao, Y., Hayashida, M. and Tamura, T., Integer programming-based approach to attractor detection and control of boolean networks, *IEICE Trans. Inform. Syst.* **E95-D** (2012) 2960–2970.

[7] Albert, R. and Barabási, A. L., Dynamics of complex systems: Scaling laws for the period of boolean networks, *Phys. Rev. Lett.* **84** (2000) 5660–5663.

[8] Albert, R. and Othmer, H. G., The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster, *J. Theor. Biol.* **223** (2003) 1–18.

[9] Albert, R. and Thakar, J., Boolean modeling: A logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions, *WIREs Syst. Biol. Med.* **6** (2014) 353–369.

[10] Aldana, M., Boolean dynamcis of networks with scale-free topology, *Physica D* **185** (2003) 45–66.

[11] Amaral, L. A. N., Guilera, A. D., Moreira, A. A., Goldberger, A. L. and Lipsitz, L. A., Emergence of complex dynamics in a simple model of signaling networks, *Proc. Natl. Acad. Sci. USA* **101** (2004) 15551–15555.

[12] Azuma, S. and Imura, J., Synthesis of optimal controllers for piecewise affine systems with sampled-data switching, *Automatica* **42** (2006) 697–710.

[13] Barabási, A. L. and Oltval, Z. N., Network biology: Understanding the cell's functional organization, *Nat. Rev. Genet.* **5** (2004) 101–113.

[14] Bharti, A. C. and Aggarwal, B. B., Nuclear factor-kappa b and cancer: Its role in prevention and therapy, *Biochem. Pharmacol.* **64** (2002) 883–888.

[15] Chaos, A., Aldana, M., Espinosa-Soto, C., de Leon, B., Arroyo, A. G. and Alvares-Buylla, E. R. From genes to flower patterns and evolution: Dynamic models of gene regulatory networks, *J. Plant Growth Regul.* **25** (2006) 278–289.

[16] Chen, H., Li, X. and Sun, J., Stabilization, controllability and optimal control of boolean networks with impulsive effects and state constraints, *IEEE Trans. Autom. Control* **60** (2015) 806–811.

[17] Cheng, D. and Qi, H., Controllability and observability of boolean control networks, *Automatica* **45** (2009) 1659–1667.

[18] Cheng, D., Qi, H. and Li, Z., *Analysis and Control of Boolean Networks*: *A Semi-Tensor Product Approach* (Springer, New York, 2011).

[19] Cheng, D. and Zhao, Y., Identification of boolean control networks, *Automatica* **47** (2011) 702–710.

[20] Clipstone, N. A. and Crabtree, G. R., Identification of calcineurin as a key signalling enzyme in t-lymphocyte activation, *Lett. Nat.* **357** (1992) 695–697.

[21] Coppersmith, S. N., Complexity of the predecessor problem in kauffman networks, *Phys. Rev. E* **75** (2007) 051108.

[22] Datta, A., Choudhary, A., Bittner, M. L. and Dougherty, E. R., External control in markovian genetic regulatory networks, *Mach. Learn.* **52** (2003) 169–191.

[23] Davidich, M. I. and Bornholdt, S., Boolean network model predicts cell cycle sequence of fission yeast, *PLoS One* **3** (2008) e1672.

[24] Drossel, B., Mihaljev, T. and Greil, F., Number and length of attractors in a critical kauffman model with connectivity one, *Phys. Rev. Lett.* **94** (2005) 088701.

[25] Dubrova, E., Teslenko, M. and Martinelli, A., Kauffman networks: Analysis and applications, *Proc. Int. Conf. Computer-Aided Design* (*ICCAD'2005*) (2005), pp. 479–484.

[26] Fauré, A., Naldi, A., Chaouiya, C. and Thieffry, D., Dynamical analysis of a generic boolean model for the control of the mammalian cell cycle, *Bioinformatics* **22** (2006) e124–e131.

[27] Feng, J., Yao, J. and Cui, P., Singular boolean networks: Semi-tensor product approach, *Sci. China Inf. Sci.* **56** (2013) 112203.

[28] Fornasini, E. and Valcher, M. E., Optimal control of boolean control networks, *IEEE Trans. Autom. Control* **59** (2014) 1258–1270.

[29] Gao, J., Liu, Y.-Y., D'Souza, R. M. and Barabási, A.-L., Target control of complex networks, *Nat. Commun.* **5** (2014) 5415.

[30] Garg, A., DeMicheli, G., Xenarios, L. and Mendoza, L., An efficient method for dynamic analysis of gene regulatory networks and *in silico* gene perturbation experiments, *Lect. Notes Comput. Sci.* **4453** (2007) 62–76.

[31] Giacomantonio, C. E. and Goodhill, G. J., A boolean model of the gene regulatory network underlying mammalian cortical area development, *PLoS Comput. Biol.* **6** (2010) e1000936.

[32] Haura, E. B., Turkson, J. and Jove, R., Mechanisms of disease: Insights into the emerging role of signal transducers and activators of transcription in cancer, *Nat. Clin. Practice Oncol.* **2** (2005) 315–324.

[33] Hayashida, M., Tamura, T., Akutsu, T., Zhang, S.-Q. and Ching, W.-K., Algorithms and complexity analyses for control of singleton attractors in boolean networks, *EURASIP J. Bioinform. Syst. Biol.* **2008** (2008) 521407.

[34] Hopcroft, J. E. and Karp, R. M., An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs, *SIAM J. Comput.* **2** (1973) 225–231.

[35] Hwang, L. H., Lau, L. F., Smith, D. L., Mistrot, C. A., Hardwick, K. G., Hwang, E. S., Amon, A. and Murray, A. W., Budding yeast cdc20: A target of the spindle checkpoint, *Science* **279** (1998) 1041–1044.

[36] Karlof, J. K., *Integer Programming*: *Theory and Practice* (CRC Press, 2006).

[37] Kauffman, S., Homeostasis and differentiation in random genetic control networks, *Nature* **224** (1969) 177–178.

[38] Kauffman, S., *The Origins of Order*: *Self-Organization and Selectionin Evolution* (Oxford University Press, New York, 1993).

[39] Kim, S. H., Lin, D. P., Matsumoto, S., Kitazono, A. and Matsumoto, T., Fission yeast slp1: An effector of the mad2-dependent spindle checkpoint, *Science* **279** (1998) 1045–1047.

[40] Kitano, H., Computational systems biology, *Nature* **420** (2002) 206–210.

[41] Kitano, H., Cancer as a robust system: Implications for anticancer therapy, *Nat. Rev. Cancer* **4** (2004) 227–235.

[42] Kiu, H. and Nicholson, S. E., Biology and significance of the jak/stat signalling pathways, *Growth Factors* **30** (2012) 88–106.

[43] Klamt, S., Saez-Rodriguez, J., Lindquist, J. A., Simeoni, L. and Giles, E. D., A methodology for the structural and functional analysis of signaling and regulatory networks, *BMC Bioinform.* **7** (2006) 56.

[44] Klinke, D. J., A multiscale systems perspective on cancer, immunotherapy, and interleukin-12, *Mol. Cancer* **9** (2010) 1.

[45] Kobayashi, K. and Hiraishi, K., An integer programming approach to optimal control problems in context-sensitive probabilistic boolean networks, *Automatica* **47** (2011) 1260–1264.

[46] Kobayashi, K. and Hiraishi, K., Optimal control of probabilistic boolean networks using polynomial optimization, *IEICE Trans. Fundam.* **E95-A** (2012) 1512–1517.

[47] Kyozuka, J., Harcourt, R., Peacock, W. and Dennis, E., Eucalyptus has functional equivalents of the arabidopsis ap1 gene, *Plant Mol. Biol.* **35** (1997) 573–584.

[48] Lähdesmäki, H., Shmulevich, I. and Yli-Harja, O., On learning gene regulatory networks under the boolean network model, *Mach. Learn.* **52** (2003) 147–167.

[49] Langmead, C. J. and Jha, S. K., Symbolic approaches for finding control strategies in boolean networks, *J. Bioinform. Comput. Biol.* **7** (2009) 323–338.

[50] Laschov, D. and Margaliot, M., A maximum principle for single-input boolean control networks, *IEEE Trans. Autom. Control* **56** (2011) 913–917.

[51] Leibfried, A., To, J. P., Busch, W., Stehling, S., Kehle, A., Demar, M., Kieber, J. J. and Lohmann, J. U., Wuschel controls meristem function by direct regulation of cytokinin-inducible response regulators, *Nature* **438** (2005) 1172–1175.

[52] Li, F., Long, T., Lu, Y., Ouyang, Q. and Tang, C., The yeast cell-cycle network is robustly designed, *Proc. Natl. Acad. Sci.* **101** (2004) 4781–4786.

[53] Li, F. and Sun, J., Controllability of boolean control networks with time delays in states, *Automatica* **47** (2011) 603–607.

[54] Li, F. and Sun, J., Controllability and optimal control of a temporal boolean network, *Neural Netw.* **34** (2012) 10–17.

[55] Li, H., Wang, Y. and Liu, Z., A semi-tensor product approach to pseudo-boolean functions with application to boolean control networks, *Asian J. Control* **16** (2014) 1073–1081.

[56] Liang, S., Fuhrman, S. and Somogyi, R., Reveal, a general reverse engineering algorithm for inference of genetic network architectures, *Proc. Pac. Symp. Biocomput.* **3** (1998), pp. 18–29.

[57] Lim, C. P. and Cao, X., Structure, function, and regulation of stat proteins, *Molecular Biosystems* **2** (2006) 536–550.

[58] Lin, C., Structural controllability, *IEEE Trans. Autom. Control* **AC-19** (1974) 201–208.

[59] Liu, S. K., Smith, C. A., Arnold, R., Kiefer, F. and McGlade, C. J., The adaptor protein gads (grb2-related adaptor downstream of shc) is implicated in coupling hemopoietic progenitor kinase-1 to the activated tcr, *J. Immunol.* **165** (2000) 1417–1426.

[60] Liu, Y., Chen, H., Lu, J. and Wu, B., Controllability of probabilistic boolean control networks based on transition probability matrices, *Automatica* **52** (2015) 340–345.

[61] Liu, Y., Slotine, J. J. and Barabási, A. L., Controllability of complex networks, *Nature* **473** (2011) 167–173.

[62] Mendoza, L. and Xenarios, I., A method for the generation of standardized qualitative dynamical systems of regulatory networks, *J. Theor. Biol. Med. Model.* **3** (2006) 3–13.

[63] Mochizuki, A., Fiedler, B., Kurosawa, G. and Saito, D., Dynamics and control at feedback vertex sets. ii: A faithful monitor to determine the diversity of molecular activities in regulatory networks, *J. Theor. Biol.* **335** (2013) 130–146.

[64] Nacher, J. C. and Akutsu, T., Dominating scale-free networks with variable scaling exponent: Heterogeneous networks are not difficult to control, *New J. Phys.* **14** (2015) 073005.

[65] Nacher, J. C. and Akutsu, T., Structurally robust control of complex networks, *Phys. Rev. E* **91** (2015) 012826.

[66] Nepusz, T. and Vicsek, T., Controlling edge dynamics in complex networks, *Nature Physics* **8** (2012) 568–573.

[67] Ng, M. K., Zhang, S., Ching, W.-K. and Akutsu, T., A control model for markovian genetic regulatory networks, in *Transactions on Computational Systems Biology V*, Lecture Notes in Bioinformatics, Vol. 4070 (Springer, Heidelberg, 2006), pp. 36–48.

[68] Nie, S., Wang, X., Zhang, H., Li, Q. and Wang, B., Robustness of controllability for networks based on edge-attack, *PLoS One* **9** (2014) e89066.

[69] Novák, B. and Tyson, J. J., Modelling the controls of the eukaryotic cell cycle, *Biochem. Soc. Trans.* **31** (2003) 1526–1529.

[70] Zañudo, J. G. T. and Albert, R., Cell fate reprogramming by control of intracellular network dynamics, *PLoS Comput. Biol.* **11** (2015) e1004193.

[71] Pal, R., Datta, A., Bittner, M. L. and Dougherty, E. R., Intervention in context-sensitive probabilistic boolean networks, *Bioinform.* **21** (2005) 1211–1218.

[72] Papadimitriou, C. H. and Steiglitz, K., *Combinatorial Optimization*: *Algorithms and Complexity* (Courier Corporation, 1982).

[73] Pu, C., Pei, W. and Michaelson, A., Robustness analysis of network controllability, *Physica A* **391** (2012) 4420–4425.

[74] Ruths, J. and Ruths, D., Robustness of network controllability under edge removal, *Complex Networks IV Proc. 4th Workshop on Complex Networks CompleNet 2013*, Vol. 476 (2013), pp. 185–193.

[75] Saadatpour, A., Wang, R.-S., Liao, A., Liu, X., Loughran, T. P., Albert, I. and Albert, R., Dynamical and structural analysis of a t cell survival network identifiers novel candidate therapeutic targets for large granular lymphocyte leukemia, *PLoS Comput. Biol.* **7** (2011) e1002267.

[76] Samuelsson, B. and Troein, C., Superpolynomial growth in the number of attractors in kauffman networks, *Phys. Rev. Lett.* **90** (2003) 098701.

[77] Schrijver, A., *Theory of Linear and Integer Programming* (Wiley, 1998).

[78] Schulze, A., Zerfass, K., Spitkovsky, D., Middendorp, S., Berges, J., Helin, K., Jansen-Dürr, P. and Henglein, B., Cell cycle regulation of the cyclin a gene promoter is mediated by a variant e2f site, *Proc. Natl. Acad. Sci.* **92** (1995) 11264–11268.

[79] Sveiczer, A., Csikasz-Nagy, A., Gyorffy, B., Tyson, J. J. and Novak, B., Modeling the fission yeast cell cycle: Quantized cycle times in wee1-cdc25δ mutant cells, *Proc. Natl. Acad. Sci.* **97** (2000) 7865–7870.

[80] Takahashi, K., Tanabe, K., Ohnuki, M., Narita, M., Ichisaka, T., Tomoda, K. and Yamanaka, S., Induction of pluripotent stem cells from adult human fibroblasts by defined factors, *Cell* **131** (2007) 861–872.

[81] Veliz-Cuba, A., Aguilar, B., Hinkelmann, F. and Laubenbacher, R., Steady state analysis of boolean molecular network models via model reduction and computational algebra, *BMC Bioinform.* **15** (2014) 221.

[82] Yang, X. and Sladek, T., Overexpression of the e2f-1 transcription factor gene mediates cell transformation, *Gene Expression* **4** (1994) 195–204.

[83] Zhang, S.-Q., Hayashida, M., Akutsu, T., Ching, W.-K. and Ng, M. K., Algorithms for finding small attractors in boolean networks, *EURASIP J. Bioinform. Syst. Biol.* **2007** (2007) 20180.

[84] Zhao, Y., Li, Z. and Cheng, D., Optimal control of logical control networks, *IEEE Trans. Autom. Control* **56** (2011) 1766–1776.