On the Number of Driver Nodes for Controlling a Boolean Network to Attractors

Wenpin Hou[1,3], Peiying Ruan[2], Wai-Ki Ching[1,4,5], Tatsuya Akutsu[6,*]

[1]*Department of Mathematics, The University of Hong Kong, Hong Kong*

[2]*Deep Learning Solution Architect, NVIDIA, Tokyo, Japan*

[3]*Department of Computer Science, Johns Hopkins University, Baltimore, U.S.A*

[4]*Hughes Hall, Wollaston Road, Cambridge, U.K.*

[5]*School of Economics and Management,*

Beijing University of Chemical Technology, China

[6]*Bioinformatics Center, Institute of Chemical Research, Kyoto University, Kyoto, Japan* *

(Dated: September 2, 2018)

Abstract

It is known that many driver nodes are required to control complex biological networks. Previous studies imply that $O(N)$ driver nodes are required in both linear complex network and Boolean network models with N nodes if an arbitrary state is specified as the target. In this paper, we mathematically prove under a reasonable assumption that the expected number of driver nodes is only $O(\log_2 N + \log_2 M)$ for controlling Boolean networks if the targets are restricted to attractors, where M is the number of attractors. Since it is expected that M is not very large in many practical networks, this is a significant improvement. This result is based on discovery of novel relationships between control problems on Boolean networks and the coupon collector's problem, a well-known concept in combinatorics. We also provide lower bounds of the number of driver nodes as well as simulation results using artificial and realistic network data, which support our theoretical findings.

Keywords: Boolean networks, controllability, coupon collector's problem, driver nodes

* takutsu@kuicr.kyoto-u.ac.jp

I. Introduction

Boolean network (BN) is a sequential dynamical system composing of a large number of highly interconnected processing nodes the states of which are updated by Boolean functions of other nodes and/or itself [1]. It is simple but very efficient in modeling genetic regulation [2–4], neural networks [5], cancer networks [6], quorum sensing circuits [7], cellular signaling pathways [8], dynamic games [9], computer design [10], and social networks [11]. Each node in a BN takes either 0 or 1, where 0 and 1 mean the node is inactive and active, respectively. Since a BN with N nodes has 2^N possible states, it will eventually reach a previously visited state, thus stay in that state circle, called an attractor.

Among various problems on BNs, control of a BN is particularly important in which the values of a subset of nodes or external signals are manipulated so as to drive the BN to a desired state [9, 12–16]. For example, in disease treatment, one may need to conduct therapeutic intervention that drives the cell state of a patient from a current state to a desired state such as a benign state, and keep this state afterward. Interventions can be additional drugs, hormones, or the embedded genes that can be manipulated as the control variables of the networks [16]. However, it is difficult to give many drugs or to manipulate many genes in practice. Therefore, it is important to select a small subset of nodes from the nodes in a given BN so that the BN is driven to a desired state by manipulating the values of these nodes only. These nodes are called the *driver nodes*.

Extensive studies have been done on control of BNs [15, 17–20], control of complex networks with linear dynamics [21–23], and the chariterization of multiagent controlability from network structures [24, 25]. Recently, semi-tensor product (STP) proposed in [13, 18] has been used widely in various control models of BN [16, 26–31]; the stability and stabilization have also been considered as an important aspect in control of BNs [32–36]. However, it is known that $O(N)$ driver nodes are required to control linear complex networks [21] if an arbitrary state is specified as the target, although a single driver node is enough if the network has a special structure (every node has a self loop) [37]. Another approach has recently been proposed for control of BNs by restricting the target states to attractors (see also Fig. 1). Posing such a restriction is reasonable because target states are stable states in many practical cases, for example, healthy and stable states in disease treatment. Mochizuki et al. showed that the Feedback Vertex Set (FVS) can be a set of driver nodes if the targets are attractors [38], where the relationship between singleton attractors and FVS

was previously found [39, 40]. Zañudo and Albert [41] proposed a method to identify driver nodes for attractors of a BN irrespective of the updating scheme (synchronous/asynchronous) and the number of time steps with stable motifs each of which is defined as a set of nodes that forms a minimal strongly connected component of the network. Concerning the scaling behavior of the minimum control cost of BNs without structural or temporal constraints, however, there is not yet a conclusive results in literature. While simulation results suggest that a small number of driver nodes are enough if the target states are restricted to attractors [38, 41], results from [14, 20] imply that $O(N)$ driver nodes are required if an arbitrary state is specified as the target.

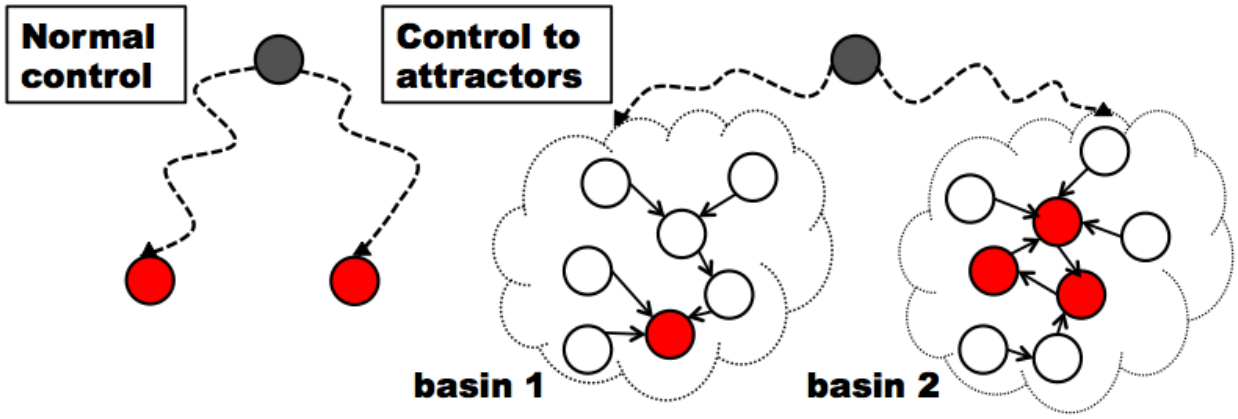


FIG. 1: Comparison of Normal Control and Attractor-based Control. Circles denote states where grey ones are initial states and red ones are targets.

In this paper, we mathematically prove that the expected number of driver nodes is only $\log_2(N) + \log_2(M) + 2$ if the targets are restricted to attractors, under a reasonable assumption. Since it is expected that M is not very large in many practical networks, this is a significant improvement. Even if the number of attractors, M, grows as $O(\exp(\sqrt{N}))$ [42], it is still $O(\sqrt{N})$, which is much smaller than $O(N)$. Our theoretical analyses are based on a discovery of novel relationships between BN control problems and Coupon Collector's Problem, a well-known concept in combinatorics. Based on these relationships, we also prove that the expected number of driver nodes is $\ln(M \ln M) + \ln 3$ (resp., $\log_2(M \ln M) + 2$) if both initial and target states are specified (resp., only a target state is specified). Note that these numbers are very small. For example, the former number is around 8.06 for $M = 200$. Since attractors are often regarded as cell types and the number of cell types in human is often said to be aounnd 200 [43], these results suggests that control of a small

number of genes is enough to modify types of cells. The differences on three bounds show a nontrivial nature of mathematical analyses although all results are based on relationships with BN control problems and Coupon Collector's Problem. Furthermore, we provide lower bounds of the number of driver nodes as well as simulation results supporting our theoretical findings. In particular, we show that the derived upper bounds are close to the lower bounds in the average case, whereas the lower bounds are much higher if we consider the worst case.

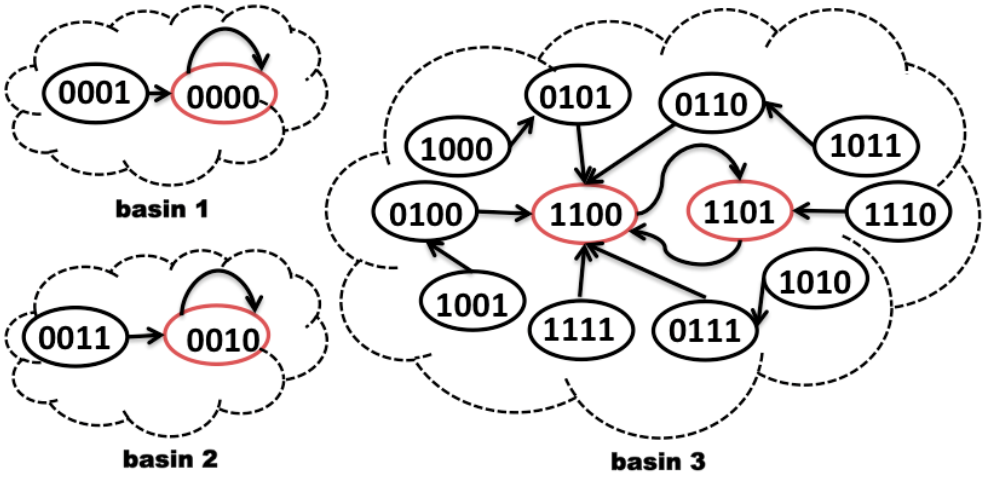
II. Preliminaries

A. Boolean Networks

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 vector $\mathbf{v}(t) = (v_1(t), \ldots, v_N(t))$ denotes the *state* of the BN at time t. The *Boolean function* for node v_i is a logical combination of k_i (called *in-degree*) variables in form of $f_i(v_{i1}, \ldots, v_{ik_i})$ where v_{i1}, ..., v_{ik_i} are k_i input nodes of v_i. Then the state of node v_i at time $t+1$ is $v_i(t+1) = f_i(v_{i1}(t), \ldots, v_{ik_i}(t))$, or equivalently, $v_i(t+1) = f_i(\mathbf{v}(t))$. The state of the (synchronous) BN is determined by $\mathbf{v}(t+1) = \mathbf{f}(\mathbf{v}(t))$. If $\mathbf{v}(t+r) = \mathbf{f}^r(\mathbf{v}(t)) = \mathbf{v}(t)$, then $\{\mathbf{f}^j(\mathbf{v}(t))\}$ $(j = 1, \ldots, r)$ is called an *attractor* of length r. Additionally, the *basin* of an attractor is defined as the set of states leading the system to the attractor. We denote the number of basins (i.e., the number of attractors) by M, and the set of states in the ith basin by A_i. Starting from any initial state, a BN eventually falls into one of its attractors (Fig. 2). For two states $\mathbf{v}_i$ and $\mathbf{v}_j$, $d(\mathbf{v}_i, \mathbf{v}_j)$ denotes the Hamming distance between $\mathbf{v}_i$ and $\mathbf{v}_j$ (i.e., the number of distinct bits).

B. Problem Definitions

We consider the *minimum driver set problem* for a BN [14], which is defined as follows. We are given a BN, an initial state $\mathbf{v}^0$, and a target state $\mathbf{v}^T$, where the target time step may or may not be given. Then, a subset U of V is called a set of *driver nodes* (a driver set, for short) if there exists a sequence of states of U that drives the BN from $\mathbf{v}^0$ to $\mathbf{v}^T$ (if the time step is specified, the BN must take state $\mathbf{v}^T$ at the specified time step). If the number of elements in U is the minimum among such sets, U is called the *minimum driver set*. The task is to find this minimum driver set. Note that there always exists a solution: if we let $U = V$, U is a driver set (although it is not necessarily the minimum). A sequence of states given to U corresponds to a sequence of control signals.



(a)

$$v_1(t+1) = v_2(t)$$
$$v_2(t+1) = v_1(t) \vee v_2(t)$$
$$v_3(t+1) = \overline{v_2}(t) \wedge v_3(t)$$
$$v_4(t+1) = v_1(t) \wedge \overline{v_4}(t)$$

(b)

t	$\mathbf{v_1}$	v_2	v_3	v_4
0	0	0	0	0
1	**1**	0	0	0
2	0	1	0	1
3	1	1	0	0
4	1	1	0	1

(c)

t	v_1	v_2	$\mathbf{v_3}$	v_4
0	0	0	0	0
1	0	0	**1**	0
2	0	0	1	0
3	0	0	1	0

(d)

FIG. 2: Examples of a BN and its control. (a) Basins & state transitions. (b) Boolean transition rules. (c) BN control: state 0000 to basin 3. (d) BN control: state 0000 to basin 2.

In this paper, we focus on the case where *the target state is any state in the specified attractor and the control is given only at time step 1*. Then, it is enough to drive a BN to any state in the target basin since the BN then falls into the target attractor. For example, in Fig. 2(c), the initial state is 0000 and the target state is 1101. If we let $v_1(1) = 1$, the BN reaches 1101 at $t = 4$ and stays in the third attractor. In Fig. 2(d), the initial state is 0000 and the target state is 0010. If we let $v_3(1) = 1$, the BN reaches 0010 at $t = 1$ and stays in the second attractor. In both cases, we need to change the state of 1 node (i.e., 1 bit), where $\{v_1\}$ and $\{v_3\}$ correspond to the minimum sets of driver nodes, respectively. Therefore, the number of driver nodes is 1 in both cases. However, if we are requested to drive the BN to any attractor using the same set of driver nodes, the minimum set of driver nodes is $\{v_1, v_3\}$. In these examples, we assumed that the state at $t = 1$ is determined only by flipping some

bits in a driver node set, for the sake of simplicity. However, all the results in this paper hold if we modify the formulation so that state transition and control are applied at the same time step, where application of a control is limited to step 1.

As mentioned above, a minimum driver set may or may not depend on a target state. Similarly, it may or may not depend on an initial state. Therefore, we consider the following three problems.

Problem 1 [Attractor-dependent Control]

Instance: a BN with N nodes, an initial state, a target basin.

Output: a minimum set of driver nodes with which we can steer the BN to the basin in one control.

Problem 2 [Attractor-independent Control]

Instance: a BN with N nodes, an initial state.

Output: a minimum set of driver nodes with which we can steer the BN to any basin of the BN in one control.

Problem 3 [Anycast Control]

Instance: a BN with N nodes.

Output: a minimum set of driver nodes with which we can steer the BN from any initial state to any basin of the BN in one control.

C. Coupon Collector's Problem

Coupon Collector's Problem (CCP) is a random allocation problem in probability theory describing "collect all coupons and win" game. Assume there are M coupons in an urn, each with a probability $P_i (i = 1, \ldots, M)$ to be collected. In each trail, you can draw one coupon, and then put it back to the urn. CCP describes at least how many trials are needed to have i collections (i different coupons have been collected in history). If $i = M$, then you achieve a full collection.

III. Theoretical Results

A. Upper Bounds via Coupon Collector's Problem

We analyze three problems independently and then compare their results. Our common idea is that the structure of a BN has already been embedded in the generation of the basins of attractors, therefore by considering driving the BN to basins, we take into account the topology and dynamic processes together tactfully. Our theoretical results are based on a

discovery of novel relationships between the three problems and CCP, each in a different way. Basically, we associate each state of a BN to a type of coupon so that states in the same basin have the same type, and then analyze how many bits should be flipped in order to have a specified coupon or all coupons. However, it is difficult to analyze three cases in a unified way and thus we consider three cases separately.

Although we assume in most parts of theoretical analysis that the basins are obtained by a random partition of 2^N states into M sets of the same size, the restriction of the size will be considerably relaxed, to be stated in Corollary 1.

Theorem 1. *Assume that the basins are obtained by a random partition of* 2^N *states into* M *sets of the same size. Then, for* $M \in [3, 1.44^N]$ *with sufficiently large* N*, the expected minimum number of driver nodes in Attractor-dependent Control is bounded above by* $\ln(M \ln M) + \ln 3$.

Proof. We formulate the Attractor-dependent Control problem as CCP. Let $\mathbf{v}^0$ be the initial state. Without loss of generality, we assume $\mathbf{v}^0$ is a vector of bits whose values are all 0. Let $k = \max_{j\in\{1,\ldots,M\}} \min_{\mathbf{v}\in A_j} d(\mathbf{v}^0, \mathbf{v})$. It means that there exists at least one state with at most k bits value 1 in each basin and thus this k gives the minimum number of driver nodes in Attractor-dependent Control.

We sort the N-bit states in the following order (i.e., increasing order of the number of 1s):

$$
\begin{array}{c}
00000 \\
00001 \\
00010 \\
\vdots \\
10000 \\
00011 \\
00101 \\
\vdots
\end{array}
$$

$$\vdots$$

$$00111$$

$$\vdots$$

Then, each vector is considered as a coupon belonging to one of M types where the type is essentially differentiating which basin that state belongs to. We draw coupons in the above order. Suppose that after drawing the hth coupon, we have all M types. Then, the number of 1s in the hth coupon gives k.

It is well-known that the expected number of trials to have a full collection of all M types of coupons is

$$MH_M = M(\ln M + \gamma + \frac{1+o(1)}{2M})$$

if all types are being collected equally likely [44]. Here H_M is the Mth harmonic number, $\gamma \approx 0.5772156649$ is the Euler-Mascheroni constant. When $M \geq 2$, we have

$$MH_M < M(\ln M + 1) < 3M \ln M.$$

Note that the probability of drawing a coupon of a new type is increasing actually because we draw coupons by following the above list without replacement. As a result, the number of trials needed to have a full collection is actually smaller than MH_M, which further indicates that we are deriving an upper bound.

Here we recall that there exist M basins. Although one basin is specified in Attractor-dependent Control, one of the elements of this basin must be included in the set of states with at most k 1s if the size of a driver set is k (since we are assuming that $\mathbf{v}^0$ is the zero vector). Furthermore, in order to have at least one state from any specified basin, $3M \ln M$ states are needed in the average case. Since the number of states with at most k 1s is

$$1 + N + \binom{N}{2} + \cdots + \binom{N}{k},$$

we have a required driver set if

$$1 + N + \binom{N}{2} + \cdots + \binom{N}{k} \geq 3M \ln M \tag{1}$$

holds. Of course, the expected number of driver nodes does not exactly correspond to the expected number of coupons. However, since the probability that the number of trials needed

to cover all coupons is larger than $3M \ln M$ is quite small [45], it is enough that InEq. (1) is satisfied.

Since $\binom{N}{k} \leq (\frac{N}{k})^k$ holds, InEq. (1) is satisfied if the following is satisfied

$$(\frac{N}{k})^k \geq 3M \ln M. \tag{2}$$

By taking logarithm of both sides of this inequality, we have

$$k(\ln N - \ln k) \geq \ln(3M \ln M).$$

Here, we assume $k \leq \frac{N}{e}$, where e is the base of the natural logarithm. Then, we have

$$k(\ln N - \ln k) \geq k(\ln N - (\ln N - 1)) = k.$$

Therefore, InEq. (2) (and thus InEq. (1)) is satisfied if

$$k \geq \ln(3M \ln M) = \ln(M \ln M) + \ln 3$$

is satisfied.

In order to ensure the existence of such k,

$$\ln(3M \ln M) \leq \frac{N}{e}$$

must be satisfied because we assumed $k \leq \frac{N}{e}$. Suppose that $M < \alpha^N$ holds for some $\alpha < e^{(1/e)}$. Then, we have

$$3M \ln M \ < \ 3\alpha^N \ln(\alpha^N) \ = \ (3N \ln \alpha)\alpha^N.$$

Since $(3N \ln \alpha)\alpha^N < (e^{(1/e)})^N$ holds for sufficiently large N for any positive constant $\alpha < e^{(1/e)}$,

$$\ln(3M \ln M) \ < \ \ln\left((e^{(1/e)})^N\right) \ = \ \frac{N}{e}$$

holds for sufficiently large N for any positive constant $\alpha < e^{(1/e)}$. Therefore, the theorem follows from $1.44 < e^{(1/e)}$. □

Theorem 2. *Assume that the basins are obtained by a random partition of 2^N states into M sets of the same size. Then the expected minimum number of driver nodes in Attractor-independent Control is bounded above by $\log_2(M \ln M) + 2$.*

Proof. As in the proof of Theorem 1, we formulate the Attractor-independent Control problem as CCP. Since we need to use a fixed set of driver nodes for all attractors, we consider the different ordering of the N-bit states. We sort these states in the increasing order of their values (see also Fig. 3):

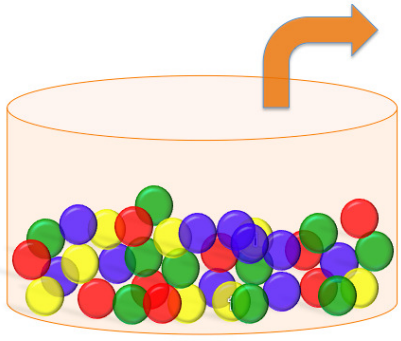
$$\begin{array}{c} 00000 \\ 00001 \\ 00010 \\ 00011 \\ 00100 \\ 00101 \\ \vdots \end{array}$$

Again, each vector is considered as a coupon belonging to one of M types. We draw coupons in the above order. Let τ be the number of states needed to collect all M labels. We can see from the ordering that the first $N - \lceil \log_2(\tau) \rceil$ bits of all vectors until τ are all 0, which means that it is enough to control the last $\lceil \log_2(\tau) \rceil$ bits. As in the proof of Theorem 1, the expected number of τ is upper bounded by $2M \ln M$. Since $\mathbb{E}[\log_2(X)] \leq \log_2(\mathbb{E}[X])$ holds from Jensen's inequality, we have

$$\begin{aligned} \mathbb{E}[\lceil \log_2(\tau) \rceil] &\leq \mathbb{E}[\log_2(\tau) + 1] \\ &\leq \log_2(\mathbb{E}[\tau]) + 1 \\ &\leq \log_2(2M \ln M) + 1 \\ &= \log_2(M \ln M) + 2. \end{aligned}$$

□

Theorem 3. *Assume that the basins are obtained by a random partition of 2^N states into M sets of the same size. Then, for $M < 1.587^N$, the expected minimum number of driver nodes in Anycast Control is bounded above by $\lceil \log_2(N) + \log_2(M) + 2 \rceil$ for sufficiently large N.*



i-th Draw	Draw Result	State	Basin
1		000000	3
2		000001	1
3		000010	3
4		000011	4
5		000100	1
6		000101	3
7		000110	4
8		000111	1
9		001000	4
10		001001	2
⋮			

FIG. 3: Attractor-independent Control formulated as CCP: $M = 4$ as an example. Suppose all 2^N states are balls in an urn (left). Colors yellow, red, blue, green of the balls are labels showing the ball belongs to basin 1,2,3,4, respectively. In each trial, we draw a ball, record its label, and then put it back to the urn. Draw results are shown in the table (right). It shows that in the tenth draw, it is the first time we have drawn balls from all 4 basins in history. Given $\mathbf{v}^0$, we can drive $\mathbf{v}^0$ to any of these 10 states in one control by the first $\lceil \log_2(10) \rceil$ bits of $\mathbf{v}^0$. Notice that $\lceil \log_2(10) \rceil < \log_2(M \ln M) + 2$ with $M = 4$ which is consistent with Theorem 2.

Proof. We divide N bits to the first H bits and the remaining L bits, where the last L bits are used for control ($H + L = N$). For any bit vector (i.e., state of a BN) $\mathbf{v}$ of size N, $\mathbf{v}_H$ and $\mathbf{v}_L$ denote bit vectors consisting of the first H bits and the last L bits, respectively. Namely, $\mathbf{v} = \mathbf{v}_H \mathbf{v}_L$. Although this partition is similar to that of the proof of Theorem 2, we cannot assume that the first H bits are all 0 since we are considering arbitrary initial states.

If every basin A_i contains states whose first H bits cover all 2^H bit patterns (i.e., $(\forall A_i)(|\{\mathbf{v}_H | \mathbf{v} \in A_i\}| = 2^H)$), we can drive any initial state to any basin. Therefore, we determine H so that this property (denoted by property (#)) holds with high probability.

Let T be the number of trials needed to collect all M coupons. It is known that $T > \beta M \ln M$ holds with probability at most $M^{-\beta+1}$, if each type of coupon is selected uniformly at random (see Section 3.6 of [44]).

We relate Anycast Control with CCP. Different from the previous problems. we regard the first H bits of each state as a type of coupon. If each basin contains all 2^H type coupons, (#) holds.

Suppose that $|A_i| \geq 3 \cdot 2^H \ln(2^H)$ holds, where H and L are determined so that this condition is satisfied. We regard the first $3 \cdot 2^H \ln(2^H)$ vectors of each A_i as coupons. Then,

the probability that there is a missed coupon in a basin A_i is at most $(2^H)^{-\beta+1} = 2^{-2H}$, where $\beta = 3$. Since there exist M basins, the probability that (#) does not hold is at most

$$\frac{M}{2^{2H}}.$$

Hereafter, we assume $M < 2^{\alpha N}$ and $H > (1-\beta)N$ (i.e., $L < \beta N$), where $\alpha, \beta > 0$ are to be determined later. Then, we have

$$\frac{M}{2^{2H}} < \frac{2^{\alpha N}}{2^{2(1-\beta)N}}.$$

If (#) is not satisfied, we use all N bits as driver nodes. However, contribution of such a factor to the expected number is less than

$$\frac{N2^{\alpha N}}{2^{2(1-\beta)N}}.$$

Therefore, we focus on the case where (#) holds.

In order to satisfy the assumption of $|A_i| \geq 3 \cdot 2^H \ln(2^H)$, the following inequality must be satisfied:

$$3 \cdot 2^H \ln(2^H) \leq \frac{2^N}{M}$$

since $|A_i| = 2^N/M$ holds. By taking logarithm of both sides, we have

$$\begin{aligned}
N &\geq \log_2(M) + \log_2(3) + H + \log_2(H) + \log_2(\ln(2)), \\
N &\geq \log_2(M) + \log_2(3) + N - L + \log_2(N-L) + \log_2(\ln(2)), \\
L &\geq \log_2(N-L) + \log_2(M) + \delta,
\end{aligned}$$

where $\delta = \log_2(3) + \log_2(\ln(2)) < 2$. The last inequality holds if the following holds:

$$L \geq \log_2(N) + \log_2(M) + \delta.$$

Therefore, the number of driver nodes is bounded above by

$$\lceil \log_2(N) + \log_2(M) + \delta \rceil + \frac{N2^{\alpha N}}{2^{2(1-\beta)N}} \leq \lceil \log_2(N) + \log_2(M) + 2 \rceil$$

for sufficiently large N if the following are satisfied

$$\begin{aligned}
\alpha &< 2(1-\beta), \\
\alpha &< \beta,
\end{aligned}$$

where $\alpha < \beta$ comes from $\lceil \log_2(N) + \log_2(M) + 2 \rceil < \beta N$. By solving $2(1-\beta) = \beta$, we have $\beta = 2/3$ and $\alpha < 2/3$. Since $2^{2/3} > 1.587$, we have the theorem. □

Here we note that $\ln(M \ln M) + \ln 3 < \log_2(M \ln M) + 2 < \lceil \log_2(N) + \log_2(M) + 2 \rceil$ holds for $M \leq 2^N$. This is consistent with the difficulties of the three problems.

Although we assumed in the above that all basins have the same size, this restriction can be considerably relaxed by adding a constant factor. We show this result only for Anycast Control since it is most general.

Corollary 1. *Assume that the basins are obtained by a random partition of 2^N states into M sets where each basin has size at least $2^N/(cM)$ ($c \geq 1$). Then, for $M < 1.587^N$, the expected minimum number of driver nodes in Anycast Control is bounded above by $\lceil \log_2(N) + \log_2(M) + 2 + \log_2(c) \rceil$ for sufficiently large N.*

Proof. In the proof of Theorem 3, we assumed that the size of each basin is $\frac{2^N}{M}$ and thus we had a condition of

$$3 \cdot 2^H \ln(2^H) \leq \frac{2^N}{M}.$$

Here, we replace $\frac{2^N}{M}$ with $\frac{2^N}{cM}$ where $c \geq 1$. Then, the following condition must be satisfied:

$$3 \cdot 2^H \ln(2^H) \leq \frac{2^N}{cM}.$$

As in the proof of Theorem 3, we have

$$L \geq \lceil \log_2(N) + \log_2(M) + \delta + \log_2(c) \rceil$$

from which the corollary follows. □

B. Lower Bounds

We can also show lower bounds of the number of driver nodes for all three problems.

For Attractor-independent Control and Anycast Control, $\log_2(M)$ is a trivial lower bound because at least $\log_2(M)$ bits are required to differentiate M basins.

Proposition 1. *For any BN with M attractors, the number of driver nodes in Attractor-independent Control and Anycast Control is bounded below by $\log_2(M)$.*

For Attractor-dependent Control, we can change driver sets depending on target attractors. Therefore, we need another analysis method.

Proposition 2. *For any BN with M attractors, the number of driver nodes in Attractor-dependent Control is bounded below by $\log_N(M)$.*

Proof. Let $\mathbf{v}^0$ be the initial state. Without loss of generality, we assume $\mathbf{v}^0$ is a vector of bits whose values are all 0. Let $k = \max_{j\in\{1,\dots,M\}} \min_{\mathbf{v}\in A_j} d(\mathbf{v}^0, \mathbf{v})$. It means that there exists at least one state with at most k bits value 1 in each basin. Then the number of such states is bounded above by

$$1 + N + \binom{N}{2} + \cdots + \binom{N}{k} < N^k.$$

Since the number of states can be produced by driver nodes must be larger than the number of basins to make it possible to drive $\mathbf{v}^0$ to all basins, we have

$$N^k \geq M$$

or equivalently

$$k \geq \log_N(M).$$

□

These results suggest that the bounds shown in Theorems 1, 2, and 3 cannot be significantly improved. Recall that it is assumed in all three problems that control is given only at $t = 1$.

In the above, we considered the expected number of driver nodes. If we consider the worst case, a much larger number of driver nodes are required as shown below.

Proposition 3. *Suppose that N is even. Then, the minimum number of driver nodes in Attractor-dependent Control is lower bounded by $(N/2) + 1$ in the worst case even if $M = 2$ and two basins are of equal size.*

Proof. We partition the set of 2^N states into two basins A_1 and A_2 such that A_1 consists of states each of which has at most $N/2$ 1s and consequently A_2 consists of states each of which has at least $N/2 + 1$ 1s. Suppose that the initial state consists of N 0s (i.e., every bit is 0) and the target basin is A_2. Then, the minimum Hamming distance between the initial state and the states in A_2 is $(N/2) + 1$. Therefore, the proposition holds. □

Since Attractor-dependent Control is the easiest among the three problems, this worst case lower bound holds also for Attractor-independent Control and Anycast Control. It is to be noted that the number of driver nodes is trivially upper bounded by N. Therefore, this proposition justifies the assumption of considering the average case.

C. Max-min Analysis Method

We can analyze the minimum number of nodes in Attractor-independent Control without using CCP and can get a more accurate bound if we adopt an assumption that each basin is an *independently selected multi-set* of size $2^N/M$ whose elements are randomly selected from $\{0,1\}^N$. It follows from this assumption that $d_j \sim B(N, \frac{1}{2}), \forall j = 1, \ldots, M$, where d_j is the Hamming distance between a randomly selected initial state $\mathbf{v}^0$ to a state in the jth basin, and $B(N,p)$ denotes the binomial distribution for N trials with success probability p. To be shown later, this assumption is supported by computer experiments, and the resulting bound better explains the results of computational experiments.

Theorem 4. *The expected minimum number of driver nodes of Attractor-dependent Control is* $N - \sum_{x=0}^{N-1} \left(1 - \left(\sum_{k=x+1}^{N} \binom{N}{k} (\frac{1}{2})^N\right)^m\right)^M$, *if each basin is an independently selected multi-set of size* $2^N/M$ *whose elements are randomly selected from* $\{0,1\}^N$.

Proof. Recall that $d_j \sim B(N, \frac{1}{2})$. By assumption, $|A_j| = \frac{2^N}{M}$, denoted as m. Then $d(\mathbf{v}^0, \mathbf{v})$ $\forall \mathbf{v} \in A_j$ is m times repetitive binomial trials. Let the results of all trials be $d_{j_1}, d_{j_2}, \ldots, d_{j_m}$ and the order statistics be

$$d_{j_{(1)}} \leq d_{j_{(2)}} \leq \ldots d_{j_{(m)}}.$$

The first order statistics $d_{j_{(1)}} = \min_{\mathbf{v} \in A_j} d(\mathbf{v}^0, \mathbf{v})$. Its cumulative distribution function (CDF) is given by

$$\begin{aligned} F_{j_{(1)}}(x) = \mathbb{P}\{d_{j_{(1)}} \leq x\} &= 1 - \mathbb{P}\{d_{j_{(1)}} > x\} = 1 - [1 - F(x)]^m \\ &= 1 - \left(\sum_{k=x+1}^{N} \binom{N}{k} (\tfrac{1}{2})^N\right)^m. \end{aligned}$$

Therefore, for d_j where $j = 1, \ldots, M$, we have $d_{1(1)}, \ldots, d_{M(1)}$. For convenience, we denote these random variables as $\xi_1, \ldots, \xi_M$. Their order statistics is given by

$$\xi_{(1)} \leq \xi_{(2)} \leq \xi_{(M)}.$$

Then the last order statistic $\xi_{(M)} = \max_j \min_{\mathbf{v} \in A_j} d(\mathbf{v}^0, \mathbf{v})$. Its CDF is given by

$$\begin{aligned} F_{\xi_{(M)}}(x) = \mathbb{P}\{\xi_{(M)} \leq x\} = \mathbb{P}\{\forall_i \xi_{(i)} \leq x\} &= (F_{j_{(1)}}(x))^M \\ &= \left(1 - \left(\sum_{k=x+1}^{N} \binom{N}{k} (\tfrac{1}{2})^N\right)^m\right)^M. \end{aligned}$$

Note that we can calculate the expected value of a non-negative random variable X by using its CDF $F(X)$,i.e., $\mathbb{E}[X] = \sum_{x=0}^{\infty}(1 - F(x))$ because

$$\begin{aligned}\mathbb{E}[X] &= \textstyle\sum_{k=0}^{\infty} k\mathbb{P}(X = k) = \sum_{t=1}^{\infty}\sum_{k=t}^{\infty}\mathbb{P}(X = k) \\ &= \textstyle\sum_{t=0}^{\infty}\mathbb{P}(X > t) = \sum_{t=0}^{\infty}(1 - \mathbb{P}(X \leq t)).\end{aligned}$$

Therefore, we can further give the expected number of $\max_j \min_{\mathbf{v}\in A_j} d(\mathbf{u}, \mathbf{v})$ as

$$\begin{aligned}\mathbb{E}[\xi_{(M)}] &= \sum_{x=0}^{N-1}\left(1 - \left(1 - \left(\sum_{k=x+1}^{N}\binom{N}{k}(\frac{1}{2})^N\right)^m\right)^M\right) \\ &= N - \sum_{x=0}^{N-1}\left(1 - \left(\sum_{k=x+1}^{N}\binom{N}{k}(\frac{1}{2})^N\right)^m\right)^M. \end{aligned} \tag{3}$$

□

D. Non-Uniform Basin Size Distributions

We can extend our result on Attractor-independent Control for arbitrary basin size distributions by utilizing a known result on CCP with arbitrary probability distributions [46]. It has been proved in [46] that the expected number of trials C_M needed to have a full collection of M different coupons is

$$\mathbb{E}[C_M] = \int_0^{\infty}(1 - \prod_{i=1}^{M}(1 - e^{-P_i t}))dt \quad \text{(integral form)} \tag{4}$$

or

$$\begin{aligned}\mathbb{E}[C_M] &= \sum_{q=0}^{j-1}(-1)^{j-1-q}\binom{M-q-1}{M-j}\sum_{|\mathbf{J}|=q}\frac{1}{1-P_{\mathbf{J}}} \\ &\text{with} \quad P_{\mathbf{J}} = \sum_{j\in\mathbf{J}} P_j \quad \text{(combinatorial form)}\end{aligned} \tag{5}$$

where P_i is the probability of collecting the ith coupon, $\mathbf{J} \in \Omega$, and $\Omega = \{1, 2, \ldots, M\}$. By using this result, we have:

Theorem 5. *The expected minimum number of driver nodes in Attractor-independent Control for an arbitrary basin distribution is bounded above by* $\log_2(\mathbb{E}[C_M]) + 1$.

Proof. We associate the minimum of driver nodes in Attractor-independent Control to CCP as in the proof of Theorem 2, with letting $P_i = |A_i|/2^N$. Then, in the same way, the expected minimum number of driver nodes needed to drive the BN to any one basin is bounded above by $\log_2(\mathbb{E}[C_M]) + 1$. □

Since "+1" factor in $\log_2(\mathbb{E}[C_M])+1$ is rather an overestimation, we use $\log_2(\mathbb{E}[C_M])$ in computational experiments.

To be discussed in Section IV, we empirically find that the distribution of the basin size follows a power-law. Therefore, we consider the case where the basin size follows a power-law.

It has been shown in Example 4.4 in [47] that if the probability of picking the ith coupon follows a power-law i^{-b} $(b>0)$, the expected number of trials needed for a full collection is

$$\begin{aligned} \mathbb{E}[C_M] \sim & M\ln M/(1-b) && \text{for} \quad 0<b<1, \\ \mathbb{E}[C_M] \sim & M\ln^2 M && \text{for} \quad b=1, \\ \mathbb{E}[C_M] \sim & \zeta(b)M^b\ln M && \text{for} \quad b>1 \end{aligned} \tag{6}$$

by letting r in [47] equal to 1, where $\zeta(\cdot)$ denotes the Riemann zeta function.

It has been shown that any process generating Zipf rank distribution would also have a power-law probability density function (see Appendix 2 in [48]) as outlined below. The relationship between these two is that if $\mathbf{X}$ follows a power-law $\mathbb{P}[\mathbf{X}=x]\sim x^{-a}$ then it would also have the rth ranked variable X_r satisfying $\mathbb{E}[X_r]\sim Cr^{-b}$ where $a=1+(1/b)$. Therefore, we can apply Eq. (6) to the case when the basin size follows power-law distribution $\mathbb{P}[\mathbf{X}=x]=Cx^{-a}$.

Then, we can apply the analysis method used in the proof of Theorem 2 and obtain the following upper bounds:

$$\begin{aligned} & O(\log_2(M\ln M/(1-b))) && \text{for} \quad b>2, \\ & O(\log_2(M\ln^2 M)) && \text{for} \quad b=2, \\ & O(\log_2(\zeta(b)M^b\ln M)) && \text{for} \quad 1\le b<2, \end{aligned} \tag{7}$$

where $b=1/(a-1)$.

IV. Computational Experiments

A. Data

The BNs we used for simulation comprise of random BNs and realistic BNs. Random BNs are generated by using C and R programming languages, including random NK-BNs (BNs with N nodes and the in-degree is bounded by K) and random scale-free BNs with γ in different values. There are four realistic BNs constructed from real-valued gene measurements: *yeastTS net* is a BN with two attractors constructed from real-valued time series

data of four preselected genes from the yeast cell cycle [49]; *cellcycle net* is a BN with 10 genes and two attractors constructed from mammalian cell cycle network introduced by [3]; *budding net* is a BN model of the control of the budding yeast cell cycle regulation from [50] with 12 genes and seven attractors; *flower net* is a BN model of the control of flower morphogenesis in Arabidobis thaliana from [51] with 15 genes and 10 attractors.

B. State and Basin Size Distributions

In Theorem 4, we assumed that the distance between an initial state and the states in each basin follows a binomial distribution: $d_j \sim B(N, \frac{1}{2}), \forall j = 1, \ldots, M$. It is supported by computer experiments as follows. Firstly, we generated random BNs and identified all basins by applying depth-first-search. Next, for a randomly selected initial state $\mathbf{v}^0$, we calculated the Hamming distance between $\mathbf{v}^0$ and all other states. Interestingly, the distribution of $d(\mathbf{v}^0, \mathbf{v})$ for $\mathbf{v} \in A_j$ with a fixed j is found to be binomially distributed with $p \approx \frac{1}{2}$ (Fig.4). Therefore, we hypothesize that $d_j \sim B(N, \frac{1}{2})$. This empirical finding also supports the assumption of random partition in Theorems 1, 2, and 3.

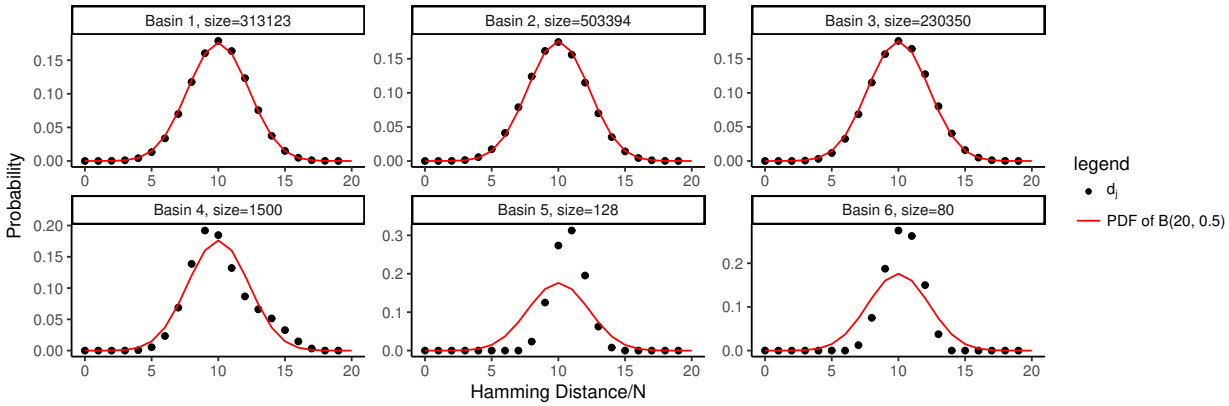


FIG. 4: Distribution of d_j in a BN of $N = 20$ and $M = 6$ as an example. In each basin, the larger the basin is, the more the d_j follows binomial distribution with $p = \frac{1}{2}$.

We also found empirically that the basin size follows a power-law distribution. We examined both random BNs with $N = 20$ and $K = 3$ (maximum in-degree) and scale-free BNs of $N = 20$ and $\gamma = 2.5$ (in-degree distribution). The results are shown in Fig. 5, from which we can see that the basin size follows a power-law. Berdahl *et al.* had similar findings on NK-BNs [52]. Specifically, we investigated the log-log plots of the distribution of basin size and found that the slopes for those BNs with maximum in-degree (bound = 3) are -1s (error tolerance 0.05), but for those of scale-free BNs are more diverse in $[-3, -1]$ with more than

60% around -2 (error tolerance 0.05). Recall that a power law distribution has a probability distribution function of $\mathbb{P}[X = x] \sim x^{-a}$ where $-a$ is the slope of the corresponding log-log plot because $\log(y) = \log(C) - a\log(x)$ for a power law $y = Cx^{-a}$. Therefore, we can make an estimation suggestion that the expected number of driver nodes needed to drive the BN to all basins can be estimated as $O(\log_2(\zeta(b)M^b \ln M))$ and $O(\log_2(M\ln^2 M))$ for BNs with the maximum in-degree K ($K \leq 3$) and for those scale-free BNs, respectively.

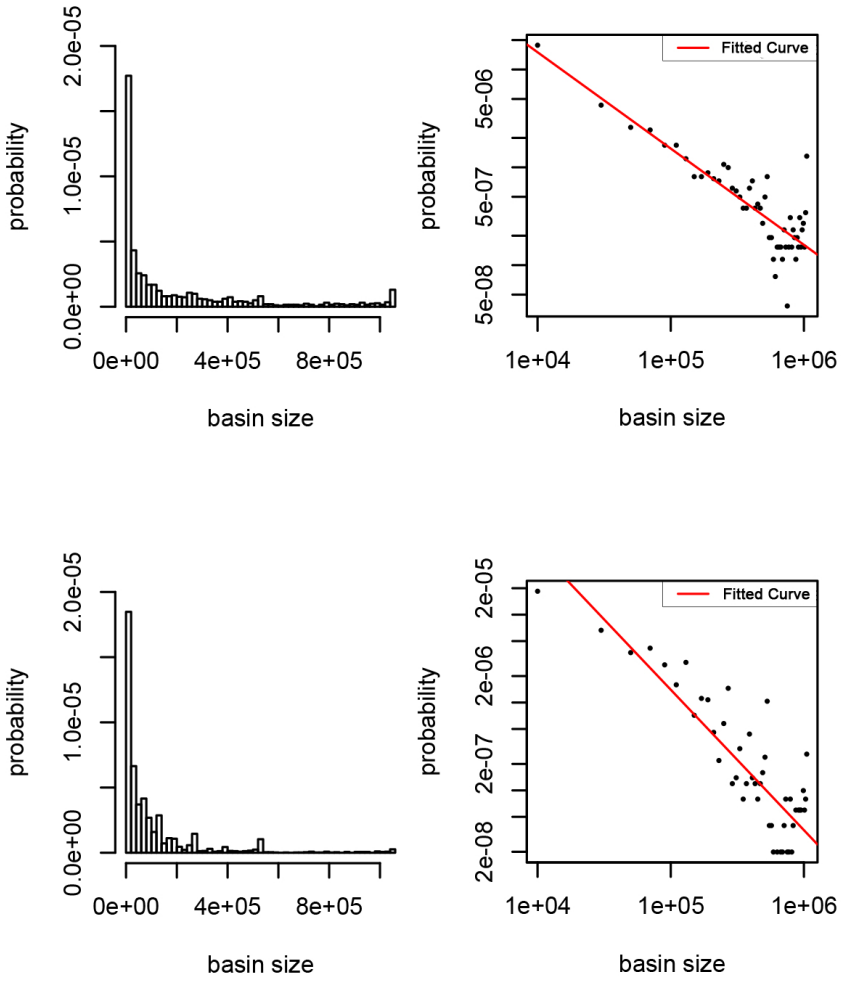


FIG. 5: Distribution of basin size (BNs of $N = 20$ as examples). First row: on BNs with $N = 20$ and $K = 3$ (max in-degree). Second row: on scale-free BNs with $N = 20$ and $\gamma = 2.5$. Left: Linear scale histogram of the distribution of basin size. Right: log-log scale plot of the distribution of basin size with red lines denoting fitting lines (slope: -0.98(1st row), - 1.60(2nd row)). Data are obtained from 250 random BNs of each type.

C. Results on the Number of Driver Nodes

Fig. 6 visualizes our key results that Attractor-dependent Control, Attractor-dependent Control, and Anycast Control are in the order of increasing the minimum control cost (i.e., the minimum number of driver nodes). Additionally, regarding the Attractor-dependent Control, we generated random BNs for 2,000 times and compared the $\xi_{(M)}$ of each simulated BN with the expectation given by Eq. (3). We also analyzed 4 realistic BNs and calculated the corresponding value of $\xi_{(M)}$. Results show that Eq. (3) gives reliable results of the minimum number of driver nodes for both random BNs and realistic BNs (see Fig. 6), illustrating the feasibility of taking the minimum set of driver nodes for the control of gene regulatory networks. Although the results of simulated BNs are a bit larger than the expectation given in Eq. (3) due to the constraints on in-degree bound, the error is small and tolerable. Meanwhile, the upper bounds given in Theorems 1, 2, 3 and the lower bound of $\log_N(M)$ serve as good boundaries for Eq. (3).

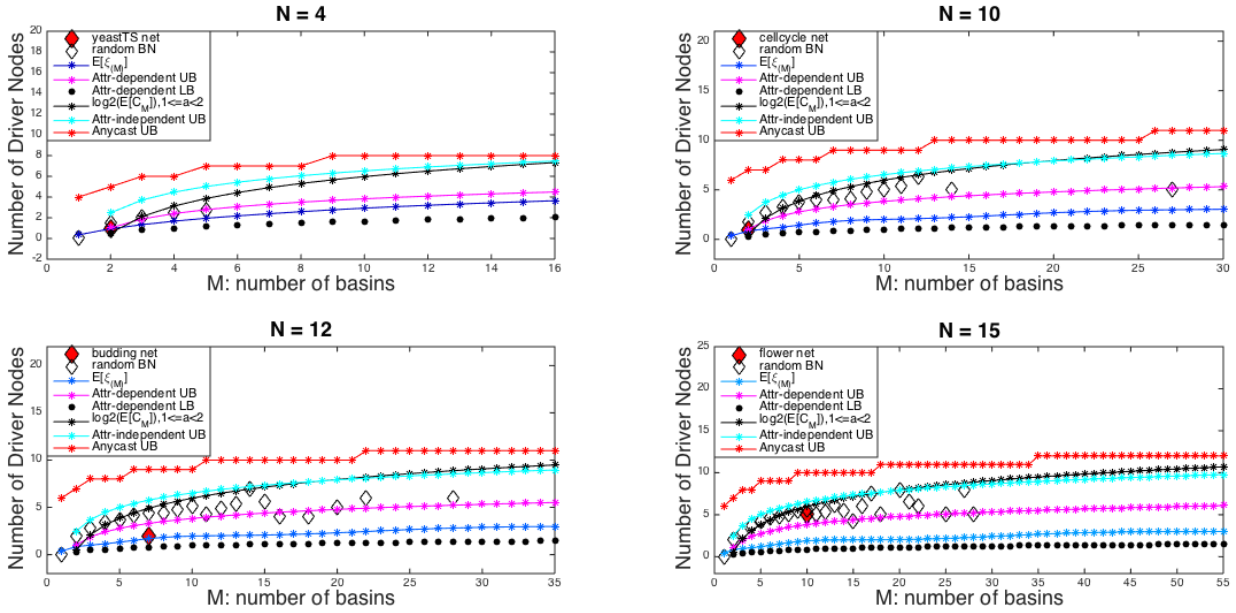


FIG. 6: Comparison of the number of driver nodes between simulation and theoretical results. "attr", "UB", "LB" are short forms of "attractor", "upper bound", "lower bound", respectively.

Meanwhile, it shows that CCP is a very good model to formulate this problem. Firstly, we generated random BNs. Next, we drove the BN from a random initial state to all basins with the least number of bits (denoted as d_i in the ith simulation) for 1000 simulations. The value of d_i was obtained as follows. For $d_i = 1, \ldots, N$, we checked all combinations of

d_i bits in $\mathbf{v}^0$ to see if we could drive the BN from $\mathbf{v}^0$ to all basins with these d_i bits. We returned d_i if it succeeded, otherwise we increased d_i by 1. Then we compared the average, $\frac{1}{1000}\sum_{i=1}^{1000} d_i$, with $\log_2(\mathbb{E}[C_M])$. As shown in Fig. 7, $\log_2(\mathbb{E}[C_M])$ provides a very good prediction on the number of driver nodes (error ≤ 1), indicating that CCP is a very good model to formulate this problem. Note that although N is small, we are considering all 2^N states in each simulation, which we think is enough to elucidate the comparison.

Recall that we give the formula of $\mathbb{E}[\xi_{(M)}]$. Fig. 6 shows that $\log_2(\mathbb{E}[C_M])$ provides a good upper bound of $\mathbb{E}[\xi_{(M)}]$. The number of driver nodes calculated by $\log_2(\mathbb{E}[C_M])$ is greater than that by $\mathbb{E}[\xi_{(M)}]$ because the bits selected in the Attractor-independent Control have to be specified (fixed) to drive the BN to all basins but it is not in the Attractor-dependent Control.

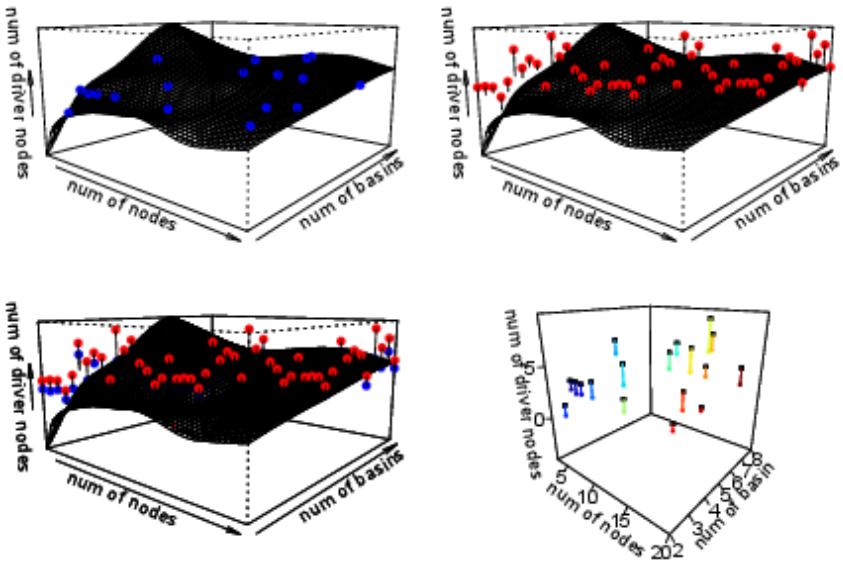


FIG. 7: Number of driver nodes obtained by random BNs and that by CCP. First row: dot plots and fitting surfaces of simulation results (left), Eq. (4) (right). Bottom left: overlap of figures in the 1st row. Bottom right: connected by colorful segments, black squares and colorful dots denote the results obtained from simulated BNs and Eq. (4), respectively.

V. Conclusion

In this paper, we theoretically showed under a reasonable assumption that only a small number of driver nodes are required for controlling Boolean networks if the targets are restricted to attractors. This result explains previous empirical findings [38, 41] and suggests that control of biological networks might not be so difficult if the targets are steady states.

We also performed computational experiments using artificial networks and realistic biological networks for verifying our theoretical findings. In addition, we pioneered the idea of formulating the minimum control cost-related problem to Coupon Collector's Problem, which might be useful for further studies on the minimum driver set problem for other mathematical models of biological networks. The Max-min Analysis Method is utilized to get a more accurate bound in Attractor-independent Control problem, while the difficulties of applying such method in another two problems lie in the complexity of their asymptotic forms.

We focused on the cases in which control is applied only at $t = 1$. This assumption is reasonable since giving heavy controls (e.g., change of gene expression values) at many time steps is not feasible. However, if we allow control operations at multiple time steps, the obtained bounds might be improved. Such an improvement is left as an open problem. We have assumed that the structure of a BN has already been embedded in the generation of the basins of attractors, However, clarifying the relationship between the structure of a BN and the distribution of basins is also important but difficult, and thus is left as future work.

---

[1] S. Kauffman, Nature **224**, 177 (1969).

[2] A. Saadatpour, R.-S. Wang, A. Liao, X. Liu, T. P. Loughran, I. Albert, and R. Albert, PLoS Comput. Biol. **7**, e1002267 (2011).

[3] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, Bioinformatics **22**, e124 (2006).

[4] R. Albert and H. G. Othmer, J. Theor. Biol. **223**, 1 (2003).

[5] L. P. Wang, E. E. Pichler, and J. Ross, Proc. Natl. Acad. Sci. USA **87**, 9467 (1990).

[6] S. Srihari, V. Raman, H. W. Leong, and M. A. Ragan, IEEE/ACM Transactions on Computatinal Biology and Bioinformatics **6**, 83 (2014), arXiv:arXiv:1310.3528v1.

[7] S. E. Dallidis and I. G. Karafyllidis, IEEE Trans. Nanobiosci. **13**, 343 (2014).

[8] R. Albert and R. Wang, Methods Enzymol. **467**, 281 (2009).

[9] Y. Zhao, Z. Li, and D. Cheng, IEEE Transactions on Automatic Control **56**, 1766 (2011).

[10] J. F. Lynch, Random Struct. Algor. **6**, 239 (1995).

[11] D. G. Green, T. G. Leishman, and S. Sadedin, Proceedings of CI-ALife'07 , 402 (2007).

[12] T. Akutsu, Y. Zhao, M. Hayashida, and T. Tamura, IEICE Transactions on Information and

Systems **E95-D**, 2960 (2012).

[13] D. Cheng and H. Qi, IEEE Trans. Neural Networks **21**, 584 (2010).

[14] W. Hou, T. Tamura, W.-K. Ching, and T. Akutsu, Advances in Complex Systems **19** (2016), 10.1142/S0219525916500065.

[15] K. Kobayashi and K. Hiraishi, IEICE Transactions **96-A**, 532 (2013).

[16] Y. Zhao, B. K. Ghosh, and D. Cheng, IEEE Transactions on Neural Networks and Learning Systems **27**, 1527 (2016).

[17] T. Akutsu, M. Hayashida, W. K. Ching, and M. K. Ng, J. Theor. Biol. **244**, 670 (2007).

[18] D. Cheng and Y. Zhao, Automatica **47**, 702 (2011).

[19] C. J. Langmead and S. K. Jha, Journal of Bioinformatics and Computational Biology **7**, 323 (2009).

[20] R. Li, M. Yang, and T. Chu, Chaos **25**, 23104 (2015).

[21] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, Nature **473**, 167 (2011).

[22] Y. Y. Liu and A. L. Barabási, Review of Modern Physics **88**, 35006 (2016).

[23] J. C. Nacher and T. Akutsu, Methods **102**, 57 (2016).

[24] Z. Ji and H. Yu, IEEE Transactions on Cybernetics **47**, 1471 (2017).

[25] Z. Ji, H. Lin, and H. Yu, IEEE Transactions on Automatic Control **60**, 781 (2015).

[26] R. Li and T. Chu, IEEE Trans Neural Netw Learn Syst. **23**, 840 (2012).

[27] Y. Liu, L. Sun, J. Lu, and J. Liang, IEEE Trans Neural Netw Learn Syst. **27**, 1991 (2016).

[28] Y. Wang, C. Zhang, and Z. Liu, Automatica **48**, 1227 (2012).

[29] Z. Liu and Y. Wang, Automatica **48**, 1839 (2012).

[30] J. Lu, J. Zhong, C. Huang, and J. Cao, IEEE Trans. Autom. Control **61**, 1658 (2016).

[31] J. Lu, J. Zhong, D. W. C. Ho, Y. Tang, and J. Cao, SIAM J. Control Optim. **54**, 475 (2016).

[32] F. Li, IEEE Trans. Neural Netw. Learn. Syst. **27**, 1585 (2016).

[33] R. Yang and Y. Wang, Automatica **49**, 390 (2013).

[34] J. Suo and J. Sun, Automatica **51**, 302 (2015).

[35] G. Wei, Z. Wang, H. Shu, and J. Fang, Syst. Control Lett. **56**, 623 (2007).

[36] Y. Liu, Neural Computation **28**, 778 (2016).

[37] N. Cowan, E. Chastain, D. Vilhena, J. Freudenberg, and C. Bergstrom, PLoS ONE **7**, e38398 (2012).

[38] A. Mochizuki, B. Fiedler, G. Kurosawa, and D. Saito, J. Theor. Biol. **335**, 130 (2013).

[39] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, Genome Informatics **9**, 151 (1998).

[40] J. Aracena, Bulletin of Mathematical Biology **70**, 1398 (2008).

[41] J. G. T. Zañudo and R. Albert, PLoS Comput. Biol. **11**, e1004193 (2015).

[42] B. Drossel, T. Mihaljev, and F. Greil, Phys. Rev. Lett. **94**, 88701 (2005).

[43] X. Fu, F. He, Y. Li, A. Shahveranov, and P. H. Hutchins, Cell Regeneration **6**, 1 (2017).

[44] R. Motwani and P. Raghavan, *Randomized algorithms* (Cambridge University Press, 1995).

[45] M. Mitzenmacher and E. Upfal, *Probability and computing*, 2nd ed. (2017).

[46] P. Flajolet, D. Gardy, and L. Thimonier, Discrete Applied Mathematics **39**, 207 (1992).

[47] A. V. Doumas and V. G. Papanicolaou, Electron. J. Probab. **18**, 1 (2012).

[48] L. A. Adamic, "Zipf, power-laws, and Pareto – a ranking tutorial," (2002).

[49] P. T. Spellman, G. Sherlock, M. Q. Zhang, R. Vishwanath, K. Anders, M. B. Eisen, P. O. Brown, and B. Futcher, Mol. Biol. Cell **9**, 3273 (1998).

[50] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, Proc. Natl. Acad. Sci. USA **101**, 4781 (2004).

[51] A. Chaos, M. Aldana, C. Espinosa-Soto, B. G. P. de Leon, A.G.Arroyo, and E.R.Alvares-Buylla, Journal of Plant Growth Regulation **25**, 278 (2006).

[52] A. Berdahl, A. Shreim, V. Sood, M. Paczuski, and J. Davidsen, New J. Physics **11**, 43024 (2009).

[53] F. Greil, Frontiers in Plant Science **3**, 1 (2012).

[54] R. Haghighi and H. Namazi, Math. Probl. Eng. **2015** (2015), 10.1155/2015/192307.

[55] X.-F. Zhang, L. Ou-Yang, Y. Zhu, M.-Y. Wu, and D.-Q. Dai, BMC Bioinform. **16**, 146 (2015).

[56] T. Akutsu, M. Hayashida, S. Q. Zhang, W. K. Ching, and and M. K. Ng, IPSJ Transactions on Bioinformatics **1**, 23 (2008).

[57] M. Aldana, Physica D **185**, 45 (2003).

[58] C. Lin, IEEE Transactions on Automatic Control **AC-19**, 201 (1974).

[59] C. E. Giacomantonio and G. J. Goodhill, PLoS Comput. Biol. **6**, e1000936 (2010).

[60] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, Nat. Commun. **4**, 2447 (2013).