A Dynamics-based Approach for the Target Control of Boolean Networks

Cui Su
Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg

Jun Pang
Faculty of Science, Technology and Medicine &
Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg

ABSTRACT

We study the target control problem of asynchronous Boolean networks, to identify a set of nodes, the perturbation of which can drive the dynamics of the network from any initial state to the desired steady state (or attractor). We are particularly interested in temporary perturbations, which are applied for sufficient time and then released to retrieve the original dynamics. Temporary perturbations have the apparent advantage of averting unforeseen consequences, which might be induced by permanent perturbations. Despite the infamous state-space explosion problem, in this work, we develop an efficient method to compute the temporary target control for a given target attractor of a Boolean network. We apply our method to a number of real-life biological networks and compare its performance with the stable motif-based control method to demonstrate its efficacy and efficiency.

KEYWORDS

Boolean networks, attractors, network control

ACM Reference Format:
Cui Su and Jun Pang. 2020. A Dynamics-based Approach for the Target Control of Boolean Networks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics (BCB '20), September 21–24, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3388440.3412464

1 INTRODUCTION

Cell reprogramming has garnering attention for its therapeutic potential for treating the most devastating diseases characterised by diseased cells or a deficiency of certain cells. It is capable of reprogramming any kind of abundant cells in the body into desired cells to restore functions of the diseased organ [8, 9, 38]. It has shown promising benefits for clinical applications, such as cell and tissue engineering, regenerative medicine and drug discovery.

In their seminal work, Yamanaka *et al.* showed that human somatic cells can be converted to induced pluripotent stem cells (iPSCs) by a cocktail of defined factors [43]. The generated iPSCs have the ability to further propagate and differentiate into many cell types. However, the application of iPSC reprogramming is often restricted, due to that: (1) the generated iPSCs have a risk of cancerous tumour formation [8, 9]; (2) the iPSC reprogramming and differentiation process usually requires long time to produce sufficient cells for application, which leads to a significant experimental cost [9]; and (3) the iPSCs often encounter cell cycle arrest after differentiation, which makes it impossible to expand the number of cells [8]. The limitations of iPSC reprogramming reinforce the need of direct reprogramming, also called transdifferentiation. Direct reprogramming reprograms somatic cells directly into the desired cell type bypassing the pluripotent state. As a consequence, direct reprogramming can not only reduce the risk of tumourigenesis and teratoma formation, but also shorten the period of time for producing enough desired cells for therapeutic application.

A major challenge of cell reprogramming lies in the identification of effective target proteins or genes, the manipulation of which can trigger desired changes. Lengthy time commitment and high costs hinder the efficiency of experimental approaches, which perform brute-force tests of tunable parameters and record corresponding results [41]. This strongly motivates us to turn to mathematical modelling of biological systems, which allows us to identify key proteins or genes that can trigger desired changes using computational methods. *Boolean network (BN)*, first introduced by Kauffman [13], is a well-established modelling framework for gene regulatory networks and their associated signalling pathways. BN has apparent advantages compared to other modelling frameworks [1]. It provides a qualitative description of biological systems and thus evades the parametrisation problem, which often occurs in quantitative models, such as models of ordinary differential equations (ODEs). In BNs, molecular species, such as genes and transcription factors, are described as Boolean variables. Each variable is assigned with a Boolean function, which determines the evolution of the node. Boolean functions characterise activation or inhibition regulations between molecular species. The dynamics of a BN is assumed to evolve in discrete time steps, moving from one state to the next, under one of the updating schemes, such as *synchronous* or *asynchronous*. Under the synchronous scheme, all the nodes update their values simultaneously at each time step, while under the asynchronous scheme, only one node is randomly selected to update its value at each time step. We focus on the asynchronous updating scheme since it can capture the phenomenon that biological processes occur at different time scales. The steady-state behaviour of the dynamics is described as *attractors*. Attractors are hypothesised to characterise cellular phenotypes [12]. Each attractor has a *weak basin* and a *strong basin*. The weak basin contains all the states that can reach this attractor, while the strong basin includes the states that can only reach this attractor and cannot reach any other attractors of the network. In the context of BNs, cell reprogramming

BCB '20, September 21–24, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7964-9/20/09…$15.00
https://doi.org/10.1145/3388440.3412464

is interpreted as a control problem: modifying the parameters of a network to lead its dynamics towards a desired attractor.

Control theories have been employed to modulate the dynamics of complex networks in recent years. Due to the intrinsic non-linearity of biological systems, control methods designed for linear systems, such as structure-based control methods [4, 6, 17], are not applicable – they can both overshoot and undershoot the number of control nodes for non-linear networks [7]. For nonlinear systems of ODEs, Fiedler *et al.* proved that the control of a feedback vertex set is sufficient to control the entire network [25, 46]; and Cornelius *et al.* proposed a simulation-based method to predict instantaneous perturbations that can reprogram a cell from an undesired phenotype to a desired one. However, further study is required to figure out if these two methods can be lifted to control BNs. Several methods based on semi-tensor product (STP) have been proposed to solve different control problems for Boolean control networks (BCNs) under the synchronous updating scheme [16, 19, 42, 44, 47, 48]. For synchronous BNs, Kim *et al.* developed a method to compute a small fraction of nodes, called 'control kernels', that can be modulated to govern the dynamics of the network [14]; and Moradi *et al.* developed an algorithm guided by forward dynamic programming to solve the control problem [26]. However, all these methods are not directly applicable to asynchronous BNs. To tackle this problem, we have developed several decomposition-based methods, which exploit both the structural and dynamical information, to cope with the source-target control with instantaneous, temporary and permanent perturbations [20, 21, 31, 32, 39] and the target control with instantaneous perturbations [2] for asynchronous BNs. In view of the difficulties and expenses in conducting biological experiments, our methods compute the minimal control sets, which can be easily translated for wet-lab validation.

Cells in tissues and in culture normally exist as a population of cells, corresponding to different stable steady states [36]. There is a need of target control methods to compute a subset of nodes, the control of which can drive the system from any initial state to a desired target attractor. The target control method developed in our previous work [2] adopts instantaneous perturbations, that are only applied instantaneously, but at a cost, a rather larger number of control nodes is required than the control with temporary or permanent perturbations [39]. Moreover, it is difficult to guarantee that all the perturbations take effect at the same time in biological experiments. Thus, target control with temporary perturbations is more appealing.

In this paper, we develop a target control method with temporary perturbations for asynchronous BNs. Our idea is to find a control $C = (\mathbb{0}, \mathbb{1})$, which is a tuple of two sets, such that the application of C – setting the value of a node, whose index is in $\mathbb{0}$ (or $\mathbb{1}$), to 0 (or 1) – can drive the network from any initial state to the weak basin of the target attractor. We hold the control for sufficient time and let the network evolve to a state in the strong basin of the target attractor. After that, the control can be released and the network will eventually and surely reach the target attractor. Since the network can take any state $s \in S$ as an initial state, the possible intermediate states form a subset S' of S, called *schema*. According to our previous work [39], we know that all the intermediate states should fall into the weak basin of the target attractor. Therefore, we partition the weak basin into a set of mutually disjoint schemata. Each schema results in a candidate control, which is further minimised and verified. Clinical applications are highly time-sensitive, controlling more nodes may shorten the period of time for generating sufficient desired cells [9]. Hence, we integrate our method with a threshold ζ on the number of perturbations. By increasing ζ, we can obtain solutions with at most ζ perturbations. It is worth noting that more perturbations may cause a significant increase in the experimental costs, hence, the parameter ζ should be considered individually based on specific experimental settings.

We have implemented our method and compared its performance with the stable motif-based control (SMC) [45] on various real-life biological networks, as both methods focus on temporary target control of asynchronous BNs. The results show that our method outperforms SMC in terms of the computational time for most of the networks. Both methods find a number of temporary controls, but our method is able to identify more controls with fewer perturbations for some networks. Another interesting observation is that the number of perturbations is often quite small compared to the sizes of the networks. This agrees with the empirical findings that the control of few nodes can reprogram biological networks [27].

2 BACKGROUND AND NOTATIONS

In this section, we give preliminary notions of Boolean networks.

2.1 Boolean networks

A Boolean network (BN) describes elements of a dynamical system with binary-valued nodes and interactions between elements with Boolean functions. It is formally defined as:

Definition 2.1 (Boolean networks). A Boolean network is a tuple $G = (X, F)$ where $X = \{x_1, x_2, \ldots, x_n\}$, such that $x_i, x_i \in X$ is a Boolean variable and $F = \{f_1, f_2, \ldots, f_n\}$ is a set of Boolean functions over X.

For the rest of the exposition, we assume an arbitrary but fixed network $G = (X, F)$ of n variables is given to us. For all occurrences of x_i and f_i, we assume x_i and f_i are elements of X and F, respectively. A *state* s of G is an element in $\{0, 1\}^n$. Let S be the set of states of G. For any state $s = (s[1], s[2], \ldots, s[n])$, and for every $i \in \{1, 2, \ldots, n\}$, the value of $s[i]$, represents the value that x_i takes when the network is in state s. For some $i \in \{1, 2, \ldots, n\}$, suppose f_i depends on $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. Then $f_i(s)$ will denote the value $f_i(s[i_1], s[i_2], \ldots, s[i_k])$ and $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ are called *parent nodes* of x_i. For two states $s, s' \in S$, the *Hamming distance* between s and s' is denoted as $hd(s, s')$.

Definition 2.2 (Control). A control C is a tuple $(\mathbb{0}, \mathbb{1})$, where $\mathbb{0}, \mathbb{1} \subseteq \{1, 2, \ldots, n\}$ and $\mathbb{0}$ and $\mathbb{1}$ are mutually disjoint (possibly empty) sets of indices of nodes of a BN G. The size of the control C is defined as $|C| = |\mathbb{0}| + |\mathbb{1}|$. Given a state $s \in S$, the application of C to s, denoted as $C(s)$, is defined as a state $s' \in S$, such that $s'[i] = 0$ for $i \in \mathbb{0}$ and $s'[i] = 1$ for $i \in \mathbb{1}$. s' is called the intermediate state w.r.t. C.

The control can be lifted to a subset of states $S' \subseteq S$. Given a control $C = (\mathbb{0}, \mathbb{1})$, $C(S') = S''$, where $S'' = \{s'' \in S | s'' = C(s'), s' \in S'\}$. S'' includes all the intermediate states with respect to C. The application of C results in a new BN, defined as follows.

Definition 2.3 (Boolean networks under control). The Boolean network G under control C, denoted as $G|_C$, is defined as a tuple $G|_C = (\hat{X}, \hat{F})$, where $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ and $\hat{F} = \{\hat{f}_1, \hat{f}_2, \dots, \hat{f}_n\}$, such that for all $i \in \{1, 2, \dots, n\}$:
(1) $\hat{x}_i = 0$ if $i \in \mathbb{0}$, $\hat{x}_i = 1$ if $i \in \mathbb{1}$, and $\hat{x}_i = x_i$ otherwise;
(2) $\hat{f}_i = 0$ if $i \in \mathbb{0}$, $\hat{f}_i = 1$ if $i \in \mathbb{1}$, and $\hat{f}_i = f_i$ otherwise.

The state space of $G|_C$, denoted $S|_C$, is derived by fixing the values of the variables in C to their respective values and is defined as $S|_C = \{s \in S \mid s[i] = 1 \text{ if } i \in \mathbb{1} \text{ and } s[j] = 0 \text{ if } j \in \mathbb{0}\}$. Note that $S|_C \subseteq S$. For any subset S' of S, we let $S'|_C = S' \cap S|_C$.

2.2 Dynamics of Boolean networks

In this section, we define several notions that can be interpreted on both G and $G|_C$. We use the generic notion $G = (X, F)$ to represent either $G = (X, F)$ or $G|_C = (\hat{X}, \hat{F})$. We assume that a BN $G = (X, F)$ evolves in discrete time steps. It starts in an initial state s_0 and its state changes in every time step according to the update functions F. Different updating schemes lead to different dynamics of the network [23]. In this work, we are interested in the *asynchronous updating scheme* as it allows biological processes to happen at different classes of time scales and thus is more realistic.

The *asynchronous evolution* of G is a function $\xi : \mathbb{N} \rightarrow \wp(S)$ such that $\xi(0) = \{s_0\}$ and for every $j \geq 0$, if $s \in \xi(j)$ then $s' \in \xi(j+1)$ is a possible *next state* of s iff either $hd(s, s') = 1$ and there exists an i such that $s'[i] = f_i(s) = 1 - s[i]$ or $hd(s, s') = 0$ and there exists an i such that $s'[i] = f_i(s) = s[i]$. It is worth noting that the asynchronous dynamics is non-deterministic. At each time step, only one node is randomly selected to update its value and a different choice may lead to a different next state $s' \in \xi(j+1)$. Henceforth, when we talk about the dynamics of G, we mean the asynchronous dynamics. The dynamics of a BN G can be described as a *transition system*, denoted as *TS*. It is defined as a tuple (S, E), where the vertices are the set of states S and for any two states s and s' there is a directed edge from s to s', denoted $s \rightarrow s'$ iff s' is a possible next state of s according to the asynchronous evolution function ξ of G. Similarly, we denote the transition system of a BN under control $G|_C$, as $TS|_C$.

A *path* ρ from a state s to a state s' is a (possibly empty) sequence of transitions from s to s' in *TS*, denoted $\rho = s \rightarrow s_1 \rightarrow \ldots \rightarrow s'$. A path from a state s to a subset S' of S is a path from s to any state $s' \in S'$. For a state $s \in S$, $reach(s)$ denotes the set of states s' such that there is a path from s to s' in *TS*.

Definition 2.4 (Attractor). An attractor A of *TS* (or of G) is a minimal non-empty subset of states of S such that for every state $s \in A$, $reach(s) = A$.

Attractors are hypothesised to characterise the steady-state behaviour of the network. Any state which is not part of an attractor is a transient state. An attractor A of *TS* is said to be reachable from a state s if $reach(s) \cap A \neq \emptyset$. The network starting at any initial state $s_0 \in S$ will eventually end up in one of the attractors of *TS* and remain there forever unless perturbed. Under asynchronous updating scheme, there are singleton attractors and cyclic attractors. Cyclic attractors can be further classified into: (1) a simple loop, in which all the states form a loop and every state appears only once per traversal through the loop; and (2) a complex loop, which has intricate topology and includes several loops. Figure 1 (a), (b) and (c) show a singleton attractor, a simple loop and a complex loop, respectively. Let $\mathcal{A}$ denote all the attractors of *TS*. For an attractor A, $A \in \mathcal{A}$, we define its *weak basin* as $bas^W_{TS}(A) = \{s \in S \mid reach(s) \cap A \neq \emptyset\}$; the *strong basin* of A is defined as $bas^S_{TS}(A) = \{s \in S \mid reach(s) \cap A \neq \emptyset \text{ and } reach(s) \cap A' = \emptyset \text{ for any } A' \in \mathcal{A}, A' \neq A\}$. Intuitively, the weak basin of A, $bas^W_{TS}(A)$, contains all the states s from which there exists at least one path to A, and there may also exist paths from s to other attractors A' $(A' \neq A)$ of *TS*. The strong basin of A, $bas^S_{TS}(A)$, consists of all the states from which there only exist paths to A.

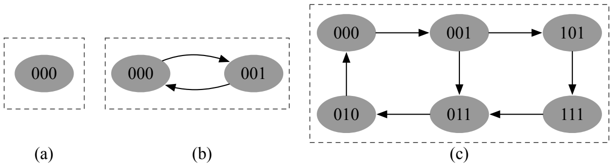

Figure 1: Different types of attractors of an asynchronous BN. We omit selfloops for all the states.

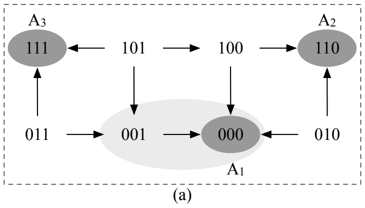

Figure 2: (a) Transition system *TS* and (b) transition system under control $TS|_C$ for Example 2.5. We omit selfloops for all the states except for state 101 in (a).

Example 2.5. Consider a BN $G = (X, F)$, where $X = \{x_1, x_2, x_3\}$, $F = \{f_1, f_2, f_3\}$, and $f_1 = x_2$, $f_2 = x_1$ and $f_3 = x_2 \wedge x_3$. Its transition system *TS* is given in Figure 2 (a). This network has three attractors A_1, A_2 and A_3, indicated as dark grey nodes. For attractor A_1, its strong basin $bas^S_{TS}(A_1) = \{000, 001\}$ is shown as the shaded grey region; its weak basin contains six states, i.e. $bas^W_{TS}(A_1) = \{000, 001, 101, 011, 100, 010\}$. Given a control $C = (\mathbb{0}, \mathbb{1}), \mathbb{0} = \{2\}, \mathbb{1} = \emptyset$ (i.e. $\{x_2 = 0\}$), the transition system under control $TS|_C$ is given in Figure 2 (b). We can see that only attractor A_1 is preserved in $TS|_C$ in Figure 2 (b).

2.3 The control problem

As described in the introduction, many devastating diseases, such as Parkinson's disease and Alzheimer's disease, are caused by diseased cells or a deficiency of particular cells. Cell reprogramming can transform abounding somatic cells into the desired cell type. In the context of BNs, this process is, indeed, stirring the dynamics of the network from a source attractor to a desired target attractor. However, cells in culture and in situ are usually not isolated but exist in a population consisting of various cell phenotypes or even transient cell states. Hence, it is important to develop a target control method to identify key nodes that can guide the network towards a desired target attractor from any other distinct steady states or transient states.

This can be defined as a *target control* problem: given a BN G and a target attractor A_t, finding a control C, the application of which can drive the network from any source state $s \in S$ to A_t. When the source state s is fixed, finding a control C to drive the network from s to A_t, is a *source-target control* problem. Based on the application time of control, we have: (1) *temporary control* - perturbations are applied for a finite (possibly zero) number of steps and then released; and (2) *permanent control* - perturbations are applied for all the following steps. When perturbations are applied instantaneously, we call it *instantaneous control*, which is a special case of temporary control. Temporary control has shown its apparent advantages in reducing the number of perturbations [39], thus in this work, we focus on *temporary target control*, formally defined as follows.

Definition 2.6 (Temporary target control). A temporary target control is a control $C = (\mathbb{O}, \mathbb{1})$, such that there exists a $t_0 > 0$, for all $t > t_0$, the network always reaches the target attractor A_t on the application of C to any source state $s \in S$ for t steps.

3 RESULTS

In this section, we shall develop a method to solve the temporary target control problem. First, we introduce the following lemma, which is crucial for the development of the method.

LEMMA 3.1. *A control* $C = (\mathbb{O}, \mathbb{1})$ *is a temporary target control to a target attractor* A_t *from any source state* $s \in S$ *iff* $bas^S_{TS}(A_t) \cap S|_C \neq \emptyset$ *and* $C(S) \subseteq bas^S_{TS|_C}(bas^S_{TS}(A_t) \cap S|_C)$.

Instead of presenting a formal proof for Lemma 3.1, we give an intuitive explanation below. Definition 2.3 shows that the application of a control C results in a new BN $G|_C$, whose state space is restricted to $S|_C$. To guarantee the inevitable reachability of A_t, by the time we release the control, the network has to reach a state s in the strong basin of A_t in the original transition system TS, i.e. $bas^S_{TS}(A_t)$, from which there only exist paths to A_t. This requires the remaining strong basin in $S|_C$, i.e. $(bas^S_{TS}(A_t) \cap S|_C)$, is a non-empty set, otherwise, it is not guaranteed to reach A_t. Furthermore, the condition $C(S) \subseteq bas^S_{TS|_C}(bas^S_{TS}(A_t) \cap S|_C)$ ensures any possible intermediate state $s' \in C(S)$ is in the strong basin of the remaining strong basin $(bas^S_{TS}(A_t) \cap S|_C)$ in the transition system under control $TS|_C$, so that the network will always evolve to the remaining strong basin. Once the network reaches the remaining strong basin, the control can be released and the network will evolve spontaneously towards the target attractor A_t. Based on the definition of the weak basin, it is sufficient to search the weak basin $bas^W_{TS}(A_t)$ for temporary target control.

A noteworthy point is that the temporary control needs to be released once the network reaches a state in $(bas^S_{TS}(A_t) \cap S|_C)$. On one hand, Lemma 3.1 guarantees that some states in the strong basin of A_t in TS are preserved in $TS|_C$, while it does not guarantee the presence of A_t in $TS|_C$. In that case, the control C has to be released at one point to recover the original TS, which at the same time retrieves A_t. On the other hand, in clinic, it is preferable to eliminate human interventions to avoid unforeseen consequences. Concerning the timing to release the control, since it is hard to interpret theoretical time steps in diverse biological experiments, it would be more feasible for biologists to estimate the timing based on empirical knowledge and specific experimental settings.

Previously, we have developed efficient decomposition-based algorithms to compute the exact basins of an attractor, which exploit both the structural and dynamical properties of the network [31, 32]. In the algorithm we develop here, we shall use these procedures to compute the weak basin and the strong basin of an attractor and refer them as COMP_WB and COMP_SB, respectively. Next, we define the *projection* of a state $s \in S$ to a subset B of $\{1, 2, \dots, n\}$, which represents the indices of a subset of nodes $X' \subseteq X$ as follows.

Definition 3.2 (Projection). Let $X' = \{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$ be a subset of X and $B = \{i_1, i_2, \dots, i_k\}$ be the set of indices of X'. The projection of a state s to B, is an element of $\{0, 1\}^k$, defined as $s|_B = (s[i_1], s[i_2], \dots, s[i_k])$. The projection is lifted to a subset S' of S as $S'|_B = \{s|_B | s \in S'\}$.

Given a control $C = (\mathbb{O}, \mathbb{1})$, the possible intermediate states with respect to C, denoted $S' = C(S)$, form a *schema* defined as follows.

Definition 3.3 (Schema). A subset S' of S is a schema if there exists a triple $M = (\mathbb{O}, \mathbb{1}, \mathbb{D})$, where $\mathbb{O} \cup \mathbb{1} \cup \mathbb{D} = \{1, 2, \dots, n\}$, $\mathbb{O}, \mathbb{1}$ and $\mathbb{D}$ are mutually disjoint (possibly empty) sets of indices of nodes of G, such that $S'|_{\mathbb{O}} = \{0\}^{|\mathbb{O}|}$, $S'|_{\mathbb{1}} = \{1\}^{|\mathbb{1}|}$ and $S'|_{\mathbb{D}} = \{0, 1\}^{|\mathbb{D}|}$. $\mathbb{O}, \mathbb{1}$ and $\mathbb{D}$ are called off-set, on-set and don't-care-set of S', respectively. The elements in $\mathbb{O} \cup \mathbb{1}$ are called indices of support variables of S'.

Intuitively, for a node x_i, $i \in \mathbb{O}$, it has a value of 0 in any state $s \in S'$; for a node x_i, $i \in \mathbb{1}$, it has a value of 1 in any state $s \in S'$. The projection of S' to the don't-care-set $\mathbb{D}$ contains all combinations of binary strings with $|\mathbb{D}|$ bits. Thus, any schema S' is of size $2^{|\mathbb{D}|}$. Since the total number of nodes $n = |\mathbb{O}| + |\mathbb{1}| + |\mathbb{D}|$ is fixed, a larger schema implies more elements in $\mathbb{D}$ and fewer elements in $\mathbb{O} \cup \mathbb{1}$.

Example 3.4. To continue with Example 2.5, let us consider the set $W_1 = \{000, 001, 010, 011\}$, which is a subset of the weak basin of A_1. There exists a triple $M_1 = (\mathbb{O}_1, \mathbb{1}_1, \mathbb{D}_1)$, where $\mathbb{O}_1 = \{1\}$, $\mathbb{1}_2 = \emptyset$ and $\mathbb{D}_2 = \{2, 3\}$, such that $W_1|_{\mathbb{O}_1} = \{0\}$, $W_1|_{\mathbb{1}_1} = \emptyset$ and $W_1|_{\mathbb{D}_1} = \{00, 01, 10, 11\}$. Therefore, W_1 is a schema. Let us denote the value of x_i, i in $\mathbb{O}_1$, $\mathbb{1}_1$ and $\mathbb{D}_1$, as 0, 1 and $*$, respectively. Then, W_1 can be represented as $0 * *$.

The notion of schema leads the way to find temporary target control. Each schema W_i of the weak basin $bas^W_{TS}(A_t)$ gives a candidate temporary target control $C_i = (\mathbb{O}_i, \mathbb{1}_i)$ for further optimisation and validation. A larger schema results in a smaller control set. To explore the entire weak basin $bas^W_{TS}(A_t)$, we partition it into a set of mutually disjoint schemata $\mathcal{W} = \{W_1, W_2, \dots, W_m\}$, where $W_1 \cup W_2 \cup \dots \cup W_m = bas^W_{TS}(A_t)$. Each $W_i \in \mathcal{W}$ is one of the largest schemata in $bas^W_{TS}(A_t) \setminus (W_1 \cup \dots \cup W_{i-1})$. For W_i, the indices of its support variables in $\mathbb{O}_i$ and $\mathbb{1}_i$ form a candidate control $C_i = (\mathbb{O}_i, \mathbb{1}_i)$. Each candidate control C_i is primarily optimised based on the properties of input nodes. Because input nodes do not have any predecessors, it is reasonable to assume that specified input nodes I^s are redundant control nodes, while non-specified input nodes I^{ns} are essential for control. For the remaining non-input nodes in C_i, denoted C^r_i, we verify its subsets of size k based on Lemma 3.1 from $k = 0$ with an increment of 1, until we find a valid solution.

Algorithm 1 Temporary Target Control

```
 1: procedure TEMP_TARGET_CONTROL($G, A_t$)
 2:     initialise $\mathcal{L} := \emptyset$ and $\Omega := \emptyset$ to store valid temporary control sets and the checked control sets, respectively.
 3:     $I, I^{ns}$ :=COMP_INPUT_NODES($G$)
 4:     $SB$ :=COMP_SB($F, A_t$)                 //strong basin of $A_t$ in $TS$
 5:     $WB$ :=COMP_WB($F, A_t$)                 //weak basin of $A_t$ in $TS$
 6:     $\mathcal{W}$ :=COMP_SCHEMATA($WB$), $m := |\mathcal{W}|$
 7:     generate a vector $\Theta$ of length $m$, set all its elements to $false$
 8:     $\zeta := n$          //the threshold on the number of perturbations
 9:     for $i = 1 : m$ do                         //traverse the schemata
10:         if $\Theta[i] = true$, then continue
11:         $C_i$ :=COMP_SUPPORT_VARIABLES($W_i$) // $C_i := (\mathbb{0}_i, \mathbb{1}_i)$
12:         $C_i^e := (\mathbb{0}_i \cap I^{ns}, \mathbb{1}_i \cap I^{ns})$, $C_i^r := (\mathbb{0}_i \setminus I, \mathbb{1}_i \setminus I)$
13:         $k := 0$, $isValid := false$
14:         while $isValid = false$ and $k \leq \min(\zeta - |C_i^e|, |C_i^r|)$ do
15:             $C_i^{sub}$ :=COMP_SUBSETS($C_i^r, k$) //compute all subsets of $C_i^r$ of size $k$
16:             for $C_j^{sub} \in C_i^{sub}$ do
17:                 $C_i^j := C_j^{sub} \cup C_i^e$, $\Phi := C_i^j(S)$
18:                 if $C_i^j \notin \Omega$ then        //$C_i^j$ has not been checked.
19:                     $isValid$ :=VERIFY_TTC($F, C_i^j, SB, \Phi$)
20:                     add $C_i^j$ to $\Omega$.
21:                     if $isValid = true$ then
22:                         add $C_i^j$ to $\mathcal{L}$, $\zeta := \min(\zeta, |C_i^j|)$
23:                         $\Theta[z] := true$ if $W_z \subseteq \Phi$ for $z \in [i+1, m]$
24:                     end if
25:                 end if
26:             end for
27:             if $isValid = false$, then $k := k + 1$
28:         end while
29:     end for
30:     return $\mathcal{L}$
31: end procedure

32: procedure VERIFY_TTC($F, C, SB, \Phi$)
33:     $isValid := false$
34:     if $\Phi \subseteq SB$ then
35:         $isValid := true$
36:     else
37:         $SB|_C$ :=COMP_STATE_CONTROL($C, SB$)      //compute the remaining strong basin w.r.t. $C$ in $TS|_C$
38:         $F|_C$ :=COMP_FN_CONTROL($C, F$)
39:         $bas^S_{TS|_C}(SB|_C)$ :=COMP_SB($F|_C, SB|_C$)
40:         if $\Phi \subseteq bas^S_{TS|_C}(SB|_C)$ then
41:             $isValid := true$
42:         end if
43:     end if
44:     return $isValid$
45: end procedure
```

To improve the efficiency of our method, we use binary decision diagram (BDD) as a symbolic representation of large state space. The size of a BDD is determined by the set of states being represented and the chosen ordering of the variables. In BDD, a schema is represented as a *cube* and each state is the smallest cube, also called a *minterm*. To compute the largest schema S_i of S is equivalent to compute the largest cube of S. The partitioning of the weak basin into schemata is then transformed into a cube cover problem in BDD. A different variable ordering may lead to a different partitioning. Given a fixed ordering, the partitioning remains the same. Even though finding the best variable ordering is NP-hard, there exist efficient heuristics to find the optimal ordering. In this work, we compute a partitioning under one variable ordering as provided by the CUDD package [37] and compute the smallest subsets of candidate controls that are valid temporary target control sets.

Algorithm 1 implements the idea in pseudo-code. It takes as inputs the BN $G = (X, F)$ and the target attractor A_t. It first initialises two vectors $\mathcal{L}$ and Ω to store valid controls and the checked controls, respectively. (We use Ω to avoid duplicate control validations.) Then, it computes input nodes I and the non-specified input nodes I^{ns}, $I^{ns} \subseteq I$ (line 3). The strong basin *SB* and the weak basin *WB* of A_t of *TS* are computed using procedures Comp_SB and Comp_WB developed in [31, 32] (lines 4-5). The weak basin *WB* is then partitioned into m mutually disjoint schemata with procedure Comp_schemata. Realisation of this procedure relies on the function to compute the largest cube provided by the CUDD package [37]. For each schema W_i, the indices of its support variables computed by procedure Comp_support_variables form a candidate control C_i (line 11). The essential control nodes C_i^e of C_i consist of the non-specified input nodes. The non-input nodes in C_i constitute a set C_i^r for further optimisation (line 12). We search for the minimal subsets of C_i^r starting from size $k = 0$ to the threshold $(\min(\zeta - |C_i^e|, |C_i^r|))$ with an increment of 1 and verify whether the union of a subset C_j^{sub} of C_i^r and the essential nodes C_i^e, namely $C_i^j = C_j^{sub} \cup C_i^e$, is a valid temporary target control using the procedure Verify_TTC in Algorithm 1. If C_i^j is valid, save it to $\mathcal{L}$. When all the subsets have been traversed or a valid control has been found, we proceed to the next schema W_{i+1}. In the end, all the verified temporary target controls are returned.

The most time-consuming part of our method lies in the verification process. As shown in the procedure Verify_TTC in Algorithm 1, for each candidate control C, we need to reconstruct the associated transition relations $F|_C$ and compute the strong basin of the remaining strong basin in $TS|_C$, i.e. $bas^S_{TS|_C}(SB|_C)$ (lines 38 and 39). Although we have developed an efficient method for the strong basin computation, the computational time still increases when the network size grows. To improve the efficiency of our algorithm, we further propose two heuristics: (1) skip a schema (line 10) if it is a subset of intermediate states Φ of a pre-validated control C_i^j (line 23); and (2) set a threshold ζ on the number of perturbations, keep ζ updated with the smallest size of valid temporary target control C_i^j (line 22) and only compute control sets with at most ζ perturbations.

Algorithm 1 can be adapted to solve target control problem with instantaneous perturbations by focusing on the schemata of the strong basin of A_t. In this way, we don't need to use procedure Verify_TTC for additional verification and the indices of support variables of each schema form an instantaneous control.

4 EVALUATION

Our temporary target control method, described in Algorithm 1, is implemented in the tool ASSA-PBN [23, 24] based on the model checker MCMAS [18] to encode BNs into the efficient data structure BDD. All the experiments are performed on a high-performance computing (HPC) platform, which contains CPUs of Intel Xeon Gold 6132 @2.6 GHz.

As discussed in the introduction, both our method (TTC) and the stable motif-based control (SMC) [45] focus on temporary target control of asynchronous BNs. We apply our method on several real-life biological networks and compare its performance with SMC. Here we give a brief description on the networks. An overview of the networks can be found in Table 1.

- The myeloid differentiation network is designed to model myeloid differentiation from common myeloid progenitors to four cell types, including megakaryocytes, erythrocytes, granulocytes and monocytes [15].
- The apoptosis network consists of necessary pro-apoptotic and anti-apoptotic pathways to capture decision-making on cell survival or apoptosis [40].
- The cardiac gene regulatory network integrates major genes that play important roles in early cardiac development and FHF/SHF determination [11].
- The ERBB receptor-regulated G1/S transition protein network combines ERBB signalling with G1/S transition of the mammalian cell cycle to identify new targets for breast cancer treatment [34].
- The HSPC-MSC network describes intercommunication pathways between hematopoietic stem and progenitor cells and mesenchymal stromal cells (MSCs) in bone marrow (BM) [5].
- The PC12 cell network models temporal sequence of protein signalling, transcriptional responses and subsequent autocrine feedbacks [29].
- The network of hematopoietic cell specification covers major transcription factors and signalling pathways for lymphoid and myeloid development [3].
- The bladder cancer network allows us to identify deregulated pathways and their influence on bladder tumourigenesis [33].
- The MAPK network is constructed to study MAPK responses to different stimuli and their contributions to cell fates [10].
- The model of HGF-induced keratinocyte migration captures the onset and maintenance of hepatocyte growth factor-induced migration of primary human keratinocytes [35].
- The Th-cell differentiation network models regulatory elements and signalling pathways controlling Th-cell differentiation [28].
- The HIV-1 network models dynamic interactions between human immunodeficiency virus type 1 (HIV-1) proteins and human signal-transduction pathways that are essential for activation of CD4+ T lymphocytes [30].

Attractors of the networks. Before the computation of target control, attractors are identified with our decomposition-based attractor detection method [22] and SMC, respectively. Our method identifies all the exact attractors (the number of states and the structures for both singleton and cyclic attractors) introduced in Section 2.2, while SMC identifies exact singleton attractors and quasi-attractors, which correspond to cyclic attractors. A quasi-attractor can be considered as a superset of an attractor: the values of oscillate nodes in the corresponding attractor are not specified in a quasi-attractor. Columns 2-3 of Table 1 give the number of nodes and edges of the networks. Columns 4-5 and 6-7 of Table 1 show the number of attractors computed by the two methods. Most of the attractors identified by the two methods are the same except for the cyclic attractor of the apoptosis network (marked in bold in Table 1). SMC identifies its quasi-attractor, which consists of 64 states, while the corresponding cyclic attractor has 56 states. Columns 8 and 9 of Table 1 show the execution time for attractor detection. We can see that our attractor detection method is more efficient than SMC.

Efficacy. We compute temporary target control for each attractor of the networks with TTC and SMC. Since neither of the methods guarantees the minimal control, they may find control sets of different sizes for one attractor. For comparison, we only consider the smallest control sets.

Figure 3 (*a*) shows the number of smallest control sets for the myeloid differentiation network. The blue bars and grey bars represent the control sets that only appear in the results of TTC and SMC, respectively. The green bars denote the intersection of the two methods. The equation above each bar $|C| = k$ describes the size of the control sets. For attractors A_1, A_5 and A_6, TTC identifies smaller control sets than SMC. Taking A_5 as an example, the minimal number of perturbations required by TTC and SMC is 3 and 4, respectively. Since we only consider the smallest controls, SMC identifies zero control set of size 3, thus we can only see the blue bar for this case. For other attractors (A_2, A_3 and A_4) of the myeloid network in Figure 3 (*a*) and all the attractors of the cardiac network in Figure 3 (*b*), two methods require the same number of perturbations, but our method has the potential to identify more solutions than SMC. For the other networks listed in Table 1, we summarise the number of control sets for one of the attractors in Figure 3 (*c*). It shows that our method is able to identify smaller control sets than SMC for the bladder cancer network (SMC failed to compute results for hematopoiesis, HGF and HIV-1 networks). Our method also has the capability to provide more solutions, which may give more flexibility for clinical applications. Another interesting observation is that even for large networks, the number of perturbations is relatively small.

Now we use the myeloid differentiation network as an example to show the consistency of our results with biological conclusions in [15]. This network consists of six attractors, four of which correspond to erythrocytes, megakaryocytes, monocytes and granulocytes. To realise the conversion to granulocytes (A_5 in Figure 3 (*a*)) from any initial state, TTC needs to perturb C/EBPα, PU.1, together with one of the nodes in {cJun, EgrNab, Gfi1}. It has been verified that coordinated overexpression of C/EBPα and PU.1 is required for the convergence to GM lineage (granulocytes and monocytes) [15]. One more control node in {cJun, EgrNab, Gfi1} helps to further distinguish granulocytes from monocytes.

Efficiency. The last two columns of Table 1 summarise the execution time for computing temporary target control for all the attractors of the networks. We can see that our method is more

Network	#nodes	#edges	Number of attractors				Time (seconds)			
			TTC		SMC		Attractor detection		Control	
			#singleton	#cyclic	#singleton	#quasi	TTC	SMC	TTC	SMC
myeloid	11	30	6	0	6	0	0.002	7.100	0.025	7.710
apoptosis	12	26	2	**1**	2	**1**	0.004	2.423	0.010	2.679
cardiac	15	39	6	0	6	0	0.004	10.710	0.200	10.279
ERBB	20	52	3	0	3	0	0.004	6.400	0.105	5.788
HSPC-MSC	26	81	2	2	2	2	0.101	33.910	0.099	11.433
PC12	33	62	7	0	7	0	0.013	84.904	14.953	191.299
hematopoiesis	33	88	5	0	-	-	0.452	-	97.773	-
bladder	35	116	3	1	3	1	0.735	25.662	2.181	34.035
MAPK	53	105	2	0	2	0	1.749	6.461	7.980	86.073
HGF	66	103	2	0	2	0	2.443	20.694	58.727	-
T-diff	68	175	6	0	6	0	1.245	13.475	18.790	14.103
HIV-1	136	321	8	0	-	-	28.274	-	270.617	-

Table 1: An overview of the networks and a comparison of the two methods (TTC and SMC). Symbol '-' means that the method failed to finish the computation within five hours.

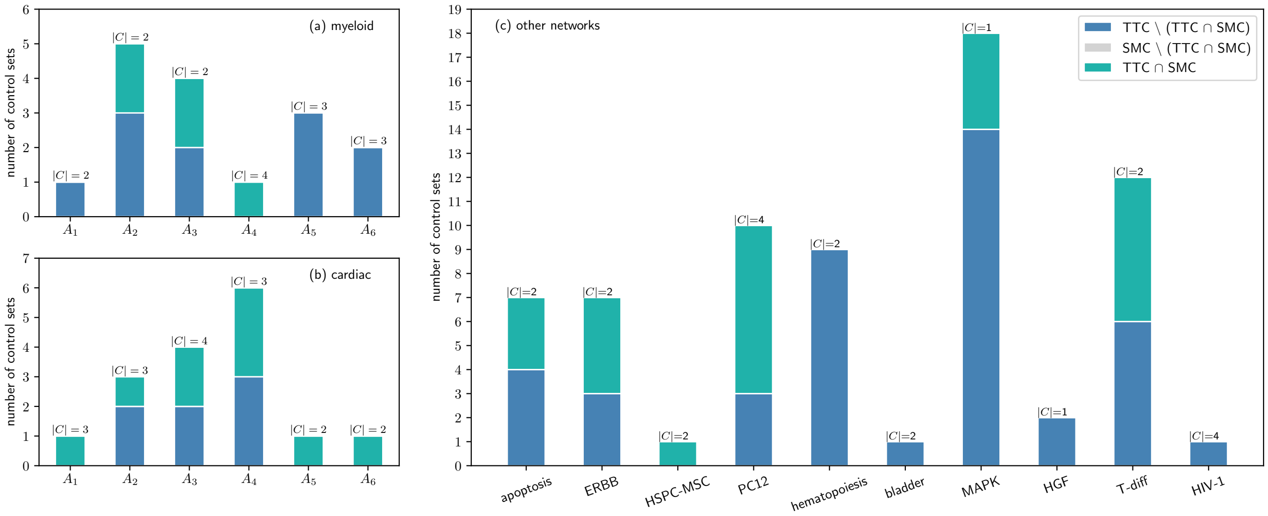

Figure 3: An overview of the control results. The blue bars and grey bars represent the control sets that only appear in the results of TTC and SMC, respectively. The green bars denote the intersection of the two methods.

efficient than SMC for most of the cases. SMC failed to finish the computation for three networks (hematopoiesis, HGF, and HIV-1) within five hours. For the hematopoiesis network, SMC failed in the identification of stable motifs, which has been pointed out to be the most time-consuming part of SMC [45]. The reason could be that the number of cycles and/or SCCs in its expanded network is computationally intractable. For the HGF-induced keratinocyte migration network, SMC is blocked in the optimisation of stable motifs due to that this network has 19 stable motifs and most of the stable motifs contain more than 16 nodes. SMC failed to construct the expanded network representation for the HIV-1 network because some of its Boolean functions depend on many parent nodes ($k \geq 10$). Detailed discussion on the complexity of SMC can be found in [45]. The efficiency of our method is influenced by not only the network size, but also the number of attractors and the number of required perturbations. The results show that our method is quite efficient and scales well for large networks.

5 CONCLUSION

In this work, we have developed a temporary target control method for asynchronous BNs to identify a set of nodes, the temporary perturbation of which can drive the network from any initial state to the desired target attractor. We have evaluated our method on a variety of real-life biological networks to demonstrate its efficacy and efficiency.

We compared our method with SMC, a promising method to solve the same control problem. SMC explores both structures and Boolean functions of BNs, and is potentially more scalable for large networks. In contrast, our method is essentially based on the dynamics of the networks, and it will suffer the state space explosion problem for networks of several hundreds of nodes. We believe that these two methods complement each other well. In the near future, we aim to find a way to combine the strengths of both methods by simultaneously exploring network structure and dynamics to achieve more efficient computational methods for the control of large biological networks.

ACKNOWLEDGMENTS

This work was partially supported by the project SEC-PBN funded by University of Luxembourg and the ANR-FNR project AlgoReCell (INTER/ANR/15/11191283).

REFERENCES

[1] T. Akutsu. 2018. *Algorithms for Analysis, Inference, and Control of Boolean Networks.* World Scientific.

[2] A. Baudin, S. Paul, C. Su, and J. Pang. 2019. Controlling large Boolean networks with single-step perturbations. *Bioinformatics* 35, 14 (2019), i558–i567.

[3] S. Collombet, C. van Oevelen, J.L.S. Ortega, W. Abou-Jaoudé, B. Di Stefano, M. Thomas-Chollier, T. Graf, and D. Thieffry. 2017. Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proceedings of the National Academy of Sciences* 114, 23 (2017), 5792–5799.

[4] E. Czeizler, C. Gratie, W.K. Chiu, K. Kanhaiya, and I. Petre. 2016. Target controllability of linear networks. In *Proc. 14th International Conference on Computational Methods in Systems Biology (LNCS, Vol. 9859)*. Springer, 67–81.

[5] J. Enciso, H. Mayani, L. Mendoza, and R. Pelayo. 2016. Modeling the pro-inflammatory tumor microenvironment in acute lymphoblastic leukemia predicts a breakdown of hematopoietic-mesenchymal communication networks. *Frontiers in Physiology* 7 (2016), 349.

[6] J. Gao, Y.Y. Liu, R.M. D'souza, and A.L. Barabási. 2014. Target control of complex networks. *Nature Communications* 5 (2014), 5415.

[7] A.J. Gates and L.M. Rocha. 2016. Control of complex networks requires both structure and dynamics. *Scientific Reports* 6, 1 (2016), 1–11.

[8] M.S. Goligorsky. 2019. New trends in regenerative medicine: reprogramming and reconditioning. *Journal of the American Society of Nephrology* (2019), ASN–2019070722.

[9] A. Grath and G. Dai. 2019. Direct cell reprogramming for tissue engineering and regenerative medicine. *Journal of Biological Engineering* 13, 1 (2019), 14.

[10] L. Grieco, L. Calzone, I. Bernard-Pierrot, F. Radvanyi, B. Kahn-Perles, and D. Thieffry. 2013. Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLOS Computational Biology* 9, 10 (2013), e1003286.

[11] F. Herrmann, A. Groß, D. Zhou, H.A. Kestler, and M. Kühl. 2012. A Boolean model of the cardiac gene regulatory network determining first and second heart field identity. *PLOS ONE* 7 (10 2012), 1–10.

[12] S. Huang. 2001. Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. *Pharmacogenomics* 2, 3 (2001), 203–222.

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

[14] J. Kim, S.M. Park, and K.H. Cho. 2013. Discovery of a kernel for controlling biomolecular regulatory networks. *Scientific Reports* 3 (2013), 2223.

[15] J. Krumsiek, C. Marr, T. Schroeder, and F.J. Theis. 2011. Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PLOS ONE* 6, 8 (2011), e22649.

[16] J. Liang, H. Chen, and J. Lam. 2017. An improved criterion for controllability of Boolean control networks. *IEEE Trans. Automat. Control* 62, 11 (2017), 6012–6018.

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

[18] A. Lomuscio, H. Qu, and F. Raimondi. 2017. MCMAS: An open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer* 19, 1 (2017), 9–30.

[19] J. Lu, J. Zhong, D.W. Ho, Y. Tang, and J. Cao. 2016. On controllability of delayed Boolean control networks. *SIAM Journal on Control and Optimization* 54, 2 (2016), 475–494.

[20] H. Mandon, C. Su, S. Haar, J. Pang, and L. Paulevé. 2019. Sequential reprogramming of Boolean networks made practical. In *Proc. 17th International Conference on Computational Methods in Systems Biology (LNCS, Vol. 11773)*. Springer, 3–19.

[21] H. Mandon, C. Su, J. Pang, S. Paul, S. Haar, and L. Paulevé. 2019. Algorithms for the sequential reprogramming of Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16, 5 (2019), 1610–1619.

[22] A. Mizera, J. Pang, H. Qu, and Q. Yuan. 2019. Taming asynchrony for attractor detection in large Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 16, 1 (2019), 31–42.

[23] A. Mizera, J. Pang, C. Su, and Q. Yuan. 2018. ASSA-PBN: A toolbox for probabilistic Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 15, 4 (2018), 1203–1216.

[24] A. Mizera, J. Pang, and Q. Yuan. 2015. ASSA-PBN: a tool for approximate steady-state analysis of large probabilistic Boolean networks. In *Proc. 13th International Symposium on Automated Technology for Verification and Analysis (LNCS, Vol. 9364).* Springer, 214–220.

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

[26] M. Moradi, S. Goliaei, and M.H. Foroughmand-Araabi. 2019. A Boolean network control algorithm guided by forward dynamic programming. *PLOS ONE* 14, 5 (2019), e0215449.

[27] F.J. Müller and A. Schuppert. 2011. Few inputs can reprogram biological networks. *Nature* 478, 7369 (2011), E4.

[28] A. Naldi, J. Carneiro, C. Chaouiya, and D. Thieffry. 2010. Diversity and plasticity of Th cell types predicted from regulatory network modelling. *PLOS Computational Biology* 6, 9 (2010), e1000912.

[29] B. Offermann, S. Knauer, A. Singh, M. L. Fernández-Cachón, M. Klose, S. Kowar, H. Busch, and M. Boerries. 2016. Boolean modeling reveals the necessity of transcriptional regulation for bistability in PC12 cell differentiation. *Frontiers in Genetics* 7 (2016), 44.

[30] O.J. Oyeyemi, O. Davies, D.L. Robertson, and J.M. Schwartz. 2014. A logical model of HIV-1 interactions with the T-cell activation signalling pathway. *Bioinformatics* 31, 7 (2014), 1075–1083.

[31] S. Paul, C. Su, J. Pang, and A. Mizera. 2018. A decomposition-based approach towards the control of Boolean networks. In *Proc. 9th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics.* ACM Press, 11–20.

[32] S. Paul, C. Su, J. Pang, and A. Mizera. 2019. An efficient approach towards the source-target control of Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2019). accepted.

[33] E. Remy, S. Rebouissou, C. Chaouiya, A. Zinovyev, F. Radvanyi, and L. Calzone. 2015. A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis. *Cancer Research* 75, 19 (2015), 4042–4052.

[34] Ö. Sahin, Ho. Fröhlich, C. Löbke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, et al. 2009. Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Systems Biology* 3, 1 (2009), 1.

[35] A. Singh, J. M. Nascimento, S. Kowar, H. Busch, and M. Boerries. 2012. Boolean approach to signalling pathway modelling in HGF-induced keratinocyte migration. *Bioinformatics* 28, 18 (2012), 495–501.

[36] A. del Sol and N.J. Buckley. 2014. Concise review: A population shift view of cellular reprogramming. *Stem Cells* 32, 6 (2014), 1367–1372.

[37] F. Somenzi. 2015. CUDD: CU Decision Diagram Package - Release 2.5.1. http://vlsi.colorado.edu/~fabio/CUDD/.

[38] D. Srivastava and N. DeWitt. 2016. In vivo cellular reprogramming: the next generation. *Cell* 166, 6 (2016), 1386–1396.

[39] C. Su, S. Paul, and J. Pang. 2019. Controlling large Boolean networks with temporary and permanent perturbations. In *Proc. 23rd International Symposium on Formal Methods (LNCS, Vol. 11800)*. Springer-Verlag, 707–724.

[40] L. Tournier and M. Chaves. 2009. Uncovering operational interactions in genetic networks using asynchronous Boolean dynamics. *Journal of theoretical biology* 260, 2 (2009), 196–209.

[41] L.Z. Wang, R. Q. Su, Z. G. Huang, X. Wang, W. X. Wang, C. Grebogi, and Y. C. Lai. 2016. A geometrical approach to control and controllability of nonlinear dynamical networks. *Nature Communications* 7, 1 (2016), 1–11.

[42] Y. Wu, X.M. Sun, X. Zhao, and T. Shen. 2019. Optimal control of Boolean control networks with average cost: A policy iteration approach. *Automatica* 100 (2019), 378–387.

[43] S. Yamanaka. 2007. Strategies and new developments in the generation of patient-specific pluripotent stem cells. *Cell Stem Cell* 1, 1 (2007), 39–49.

[44] J. Yue, Y. Yan, Z. Chen, and X. Jin. 2019. Identification of predictors of Boolean networks from observed attractor states. *Mathematical Methods in the Applied Sciences* 42, 11 (2019), 3848–3864.

[45] J.G.T. Zañudo and R. Albert. 2015. Cell fate reprogramming by control of intracellular network dynamics. *PLOS Computational Biology* 11, 4 (2015), e1004193.

[46] J.G.T. Zañudo, G. Yang, and R. Albert. 2017. Structure-based control of complex networks with nonlinear dynamics. *Proceedings of the National Academy of Sciences* 114, 28 (2017), 7234–7239.

[47] Y. Zhao, J. Kim, and M. Filippone. 2013. Aggregation algorithm towards large-scale Boolean network analysis. *IEEE Trans. Automat. Control* 58, 8 (2013), 1976–1985.

[48] J. Zhong, Y. Liu, K.I. Kou, L. Sun, and J. Cao. 2019. On the ensemble controllability of Boolean control networks using STP method. *Appl. Math. Comput.* 358 (2019), 51–62.