Target Control of Asynchronous Boolean Networks

Cui Su, Jun Pang

Abstract—We study the target control of asynchronous Boolean networks, to identify interventions that can drive the dynamics of a given Boolean network from any initial state to the desired target attractor. Based on the application time, the control can be realised with three types of perturbations, including instantaneous, temporary and permanent perturbations. We develop efficient methods to compute the target control for a given target attractor with these three types of perturbations. We compare our methods with the stable motif-based control method on a variety of real-life biological networks to evaluate their performance. We show that our methods scale well for large Boolean networks and they are able to identify a rich set of solutions with a small number of perturbations.

Index Terms—Boolean networks, network control, attractor, perturbations

1 Introduction

CEll reprogramming has great 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 human body into the desired cells to restore functions of the diseased organs [1], [2], [3]. Cell reprogramming opens up a novel field in cell and tissue engineering and regenerative medicine.

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 cost hinder the efficiency of experimental approaches, which perform brute-force tests of tunable parameters and record corresponding results [4]. This strongly motivates us to turn to mathematical modelling of biological systems, which allows us to identify key genes or pathways that can trigger desired changes using computational methods.

Boolean network, first introduced by Kauffman [5], is a well-established modelling framework for gene regulatory networks and their associated signalling pathways. It has apparent advantages compared to other modelling frameworks [6]. Boolean network 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 Boolean networks, 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 variable. Boolean functions characterise activation or inhibition regulations between molecular species. The dynamics of a Boolean network 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 network dynamics is described as *attractors*, to one of which the system eventually settles down. Attractors are hypothesised to characterise cellular phenotypes [7]. 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 Boolean networks, cell reprogramming is interpreted as a control problem: modifying the parameters of a network to lead its dynamics from the source state towards the 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 [8], [9], [10], are not applicable — they can both overshoot and undershoot the number of control nodes for non-linear networks [11]. For nonlinear systems of ODEs, Fiedler *et al.* [12], [13], [14] proved that the control of a feedback vertex set is sufficient to control the entire network; and Cornelius *et al.* [15] 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 Boolean networks. Several methods based on semi-tensor product (STP) [16], [17], [18], [19], [20], [21], [22], [23] have been proposed to solve different control problems for Boolean control networks (BCNs) under the synchronous updating scheme. For synchronous Boolean networks, Kim *et al.* [24] developed a method to compute a small fraction of nodes, called ‘control kernels’, that can be modulated to govern the

- *C. Su is with the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg. J. Pang is with the Faculty of Science, Technology and Medicine and the Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg.*
  E-mail: firstname.lastname@uni.lu
  Correspondence should be addressed to J. Pang (jun.pang@uni.lu).

.

dynamics of the network; and Moradi *el al.* [25] developed an algorithm guided by forward dynamic programming to solve the control problem. However, all these methods are developed for the synchronous updating scheme of Boolean networks, thus they are not directly applicable to asynchronous Boolean networks.

Previously, we have developed several methods [26], [27], [28], [29], [30] for the *source-target control* of asynchronous Boolean networks: to drive the dynamics of a Boolean network from the source attractor to the target attractor. However, cells in tissues and in culture normally exist as a population of cells, corresponding to different states [31]. This gives rise to the need of *target control* in order to compute a subset of nodes, whose perturbation can drive the network from any initial state to the desired target attractor. The main difference between these two control problems lies in the source state: the source state is a given attractor for the source-target control problem, while the source state can be any state of the Boolean network in the case of target control.

In this paper, we study target control of asynchronous Boolean networks with instantaneous, temporary and permanent perturbations (ITC, TTC and PTC). We aim to find a control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$, such that the instantaneous, temporary or permanent application of C – setting the value of a node, whose index is in $\mathsf{C}_{\mathbb{0}}$ (or $\mathsf{C}_{\mathbb{1}}$) to 0 (or 1) – can drive the network from any initial state s in the state space S to the target attractor A_t. Since the network can take any state $s \in S$ as the source state, there exist a set of possible intermediate states with respect to C and they form a subset S' of S, called *schema*. Instantaneous control should drive the system to states in the strong basin of the target attractor. Thus, we partition the strong basin of the target attractor into a set of disjoint schemata. The support variables of each schema form an instantaneous target control. For temporary and permanent control, due to their extended effects on the network dynamics, all the intermediate states should fall into the weak basin of the target attractor. Therefore, we partition the weak basin of the target attractor into a set of mutually disjoint schemata. Each schema results in a candidate temporary or permanent target control, which will be further optimised and verified.

Clinical applications are highly time-sensitive, controlling more nodes may shorten the period of time for generating sufficient desired cells for therapeutic application [2]. Hence, we parameterise 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 noticing that more perturbations may cause a significant increase in experimental costs, thus the threshold ζ should be considered individually based on specific experimental settings.

Note that in our previous work [32], we have introduced the target control method with temporary perturbations, namely TTC. In this paper, which is an extended and revised version of [32], we further introduce the target control methods with instantaneous and permanent perturbations, i.e. ITC and PTC. We implemented these three target control methods and compared their performance with the stable motif-based control (SMC) [33] on various real-life biological networks. The results show that our methods outperform SMC in terms of the computational time for most of the networks. As for the temporary control, both our method TTC and SMC find a number of valid temporary controls, but our method is able to identify more solutions with fewer perturbations for some networks. Another interesting observation is that the number of required 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 [34].

2 Preliminaries

In this section, we present some preliminary notions of Boolean networks.

2.1 Boolean networks

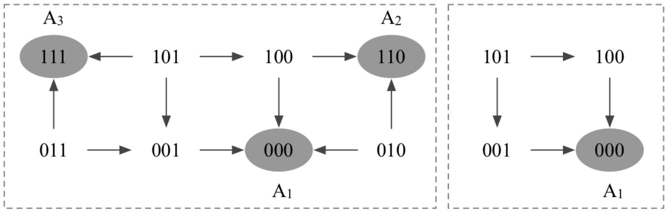
A Boolean network (BN) describes elements of a dynamical system with binary-valued nodes and interactions between elements with Boolean functions.

Definition 1 (Boolean networks)**.** *A Boolean network is a tuple* $\mathsf{BN} = (X, F)$ *where* $X = \{x_1, x_2, \ldots, x_n\}$*, such that* $x_i \in X$ *is a Boolean variable and* $F = \{f_1, f_2, \ldots, f_n\}$ *is a set of Boolean functions over* X*. Each* $f_i : \{x_1, \ldots, x_n\} \to \{0, 1\}$ *is a update function associated with* x_i*.*

The structure of a Boolean network $\mathsf{BN} = (X, F)$ can be viewed as a directed graph $G(V, E)$, called the *dependency graph* of BN, where $V = \{v_1, v_2 \ldots, v_n\}$ is the set of *nodes*. Node $v_i \in V$ corresponds to variable $x_i \in X$. For every $i, j \in \{1, 2, \ldots, n\}$, there is a directed edge from v_j to v_i, if and only if f_i depends on x_j. For the rest of the exposition, we assume an arbitrary but fixed network $\mathsf{BN} = (X, F)$ 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 BN is an element in $\{0, 1\}^n$. Let S be the set of states of BN. 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, denoted as $par(x_i)$. For two states $s, s' \in S$, the *Hamming distance* between s and s' will be denoted as $hd(s, s')$ and $\arg(hd(s, s')) \subseteq \{1, 2, \ldots, n\}$ will denote the set of indices in which s and s' differ. For two subsets $S', S'' \subseteq S$, the Hamming distance between S' and S'' is defined as the minimum of the Hamming distances between all the states in S' and all the states in S''. That is, $hd(S', S'') = \min_{s' \in S', s'' \in S''} hd(s', s'')$. We let $\arg(hd(S', S''))$ denote the set of subsets of $\{1, 2, \ldots, n\}$ such that $I \in \arg(hd(S', S''))$ if and only if I is a set of indices of the variables that realise this Hamming distance.

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



(a) (b)

Fig. 1. (a) Transition system TS and (b) transition system under control $TS|_{\mathsf{C}}$ of Example 2, where $\mathsf{C} = \{x_2 = 0\}$. We omit selfloops for all the states except for state (101) in (a).

Essentially, the application of a control C inhibits the nodes in $\mathsf{C}_{\mathbb{0}}$, i.e., the values of these nodes are set (perturbed) to be 0, and overexpresses the nodes in $\mathsf{C}_{\mathbb{1}}$, i.e., the values of these nodes are set (perturbed) to be 1.

The control can be lifted to a subset of states $S' \subseteq S$. Given a target control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$, $\mathsf{C}(S') = S''$, where $S'' = \{s'' \in S | s'' = \mathsf{C}(s'), s' \in S'\}$. Set S'' includes all the intermediate states with respect to C. Intuitively, sets $\mathsf{C}_{\mathbb{0}}$ and $\mathsf{C}_{\mathbb{1}}$ represent the indices of variables of BN whose values are held fixed to 0 and 1 respectively under the control C. The application of a control C to $\mathsf{BN} = (X, F)$ has the effect of reducing the state space of BN to those which have the values of the variables in $\mathsf{C}_{\mathbb{0}}$ and $\mathsf{C}_{\mathbb{1}}$ set respectively to 0 and 1 and modifying the Boolean functions accordingly. This results in a new Boolean network derived from BN defined as follows.

Definition 3 (Boolean networks under control)**.** *Let* $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ *be a control and* $\mathsf{BN} = (X, F)$ *be a Boolean network. The Boolean network* BN *under control* C*, denoted* $\mathsf{BN}|_{\mathsf{C}}$*, is defined as a tuple* $\mathsf{BN}|_{\mathsf{C}} = (\hat{X}, \hat{F})$*, where* $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ *and* $\hat{F} = \{\hat{f}_1, \hat{f}_2, \ldots, \hat{f}_n\}$*, such that for all* $i \in \{1, 2, \ldots, n\}$*:*
(1) $\hat{x}_i = 0$ *if* $i \in \mathsf{C}_{\mathbb{0}}$*,* $\hat{x}_i = 1$ *if* $i \in \mathsf{C}_{\mathbb{1}}$*, and* $\hat{x}_i = x_i$ *otherwise;*
(2) $\hat{f}_i = 0$ *if* $i \in \mathsf{C}_{\mathbb{0}}$*,* $\hat{f}_i = 1$ *if* $i \in \mathsf{C}_{\mathbb{1}}$*, and* $\hat{f}_i = f_i$ *otherwise.*

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

Example 1. *Consider a Boolean network* $\mathsf{BN} = (X, F)$*, 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* $f_3 = x_2 \wedge x_3$*. Given a control* $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$*, where* $\mathsf{C}_{\mathbb{0}} = \{2\}, \mathsf{C}_{\mathbb{1}} = \emptyset$ *(i.e.* $\{x_2 = 0\}$*),* $\mathsf{C}_{\mathbb{0}}$ *and* $\mathsf{C}_{\mathbb{1}}$ *are mutually disjoint. Suppose the network is in state* (111)*, the application of* C *drives the network from* (111) *to state* (101)*. State* (101) *is called the intermediate state w.r.t.* C*.*

2.2 Dynamics of Boolean networks

In this section, we define several notions that can be interpreted on both BN and $\mathsf{BN}|_{\mathsf{C}}$. We use the generic notion $\mathsf{BN} = (X, F)$ to represent either $\mathsf{BN} = (X, F)$ or $\mathsf{BN}|_{\mathsf{C}} = (\hat{X}, \hat{F})$. We assume that a Boolean network $\mathsf{BN} = (X, F)$ evolves in discrete time steps. It starts from an initial state s_0 and its state changes in every time step based on the Boolean functions F and the updating schemes. Different updating schemes lead to different dynamics of the network [35], [36]. 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. We define *asynchronous dynamics* of Boolean networks as follows.

Definition 4 (Asynchronous dynamics of Boolean networks)**.** *Suppose* $s_0 \in S$ *is an initial state of* BN*. The asynchronous evolution of* BN *is a function* $\xi_{\mathsf{BN}} : \mathbb{N} \rightarrow \wp(S)$ *such that* $\xi_{\mathsf{BN}}(0) = \{s_0\}$ *and for every* $j \geq 0$*, if* $s \in \xi_{\mathsf{BN}}(j)$ *then* $s' \in \xi_{\mathsf{BN}}(j+1)$ *is a possible* next state *of* s *iff either* $hd(s, s') = 1$ *and there exists* i *such that* $s'[i] = f_i(s) = 1 - s[i]$*, or* $hd(s, s') = 0$ *and there exists* i *such that* $s'[i] = f_i(s) = s[i]$*.*

It is worth noticing that the asynchronous dynamics is non-deterministic. At each time step, only one node is randomly selected to update its value based on its associated Boolean update function. A different choice may lead to a different next state $s' \in \xi(j+1)$. Henceforth, when we talk about the dynamics of a Boolean network, we shall explicitly mean the asynchronous dynamics. We describe the dynamics of a Boolean network as a *transition system (TS)*, defined as follows.

Definition 5 (Transition system of Boolean networks)**.** *The transition system of a Boolean network* BN*, denoted as* TS*, is a tuple* $(S, \rightarrow_{\mathsf{BN}})$*, 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* BN *(i.e. Definition 4).*

Similarly, we denote the transition system of a Boolean network under control, $\mathsf{BN}|_{\mathsf{C}}$, as $TS|_{\mathsf{C}}$.

Example 2. *The transition system TS of* BN*, given in Example 1, is shown in Fig. 1(a). It depicts the asynchronous dynamics of* BN*. According to Definition 4, the asynchronous dynamics is non-deterministic, thus a state can have more than one out-going edges. For instance, state* (101) *has three out-going edges. The next state of* (101) *can be* (001)*,* (100)*, or* (111)*.*

The application of $\mathsf{C} = \{x_2 = 0\}$ *given in Example 1, reshapes the transition system TS of* BN *in Fig. 1(a) to the transition system under control* $TS|_{\mathsf{C}}$ *of* $\mathsf{BN}|_{\mathsf{C}}$ *in Fig. 1(b). We can see that the control results in a new transition system, where only a subset of states and transitions are preserved. Therefore, the attractors of* TS *and* $TS|_{\mathsf{C}}$ *might differ. For this example, only attractor* A_1 *is preserved in* $TS|_{\mathsf{C}}$*.*

2.3 Attractors and basins

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 $s' \in S'$. An *infinite path* ρ from s, $\rho = s \rightarrow s_1 \rightarrow \ldots$, is a sequence of infinite transitions from s. A state $s' \in S$ appears *infinitely often* in ρ if for any $i \geq 0$, there exists $j \geq i$ such that $s_j = s'$. We assume every infinite path ρ is *fair* – for any state s' that appears infinitely often in ρ, every possible next state s'' of s' also appears infinitely often in ρ. 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 6 (Attractor)**.** *An attractor* A *of* TS *(or of* BN*) is a minimal non-empty subset of states of* S *such that for every state* $s \in A$*,* $reach(s) = A$*.*

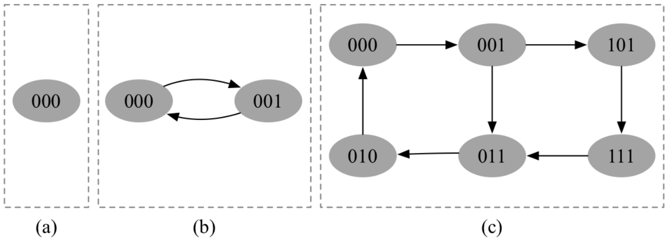


Fig. 2. Different types of attractors of an asynchronous Boolean network: (a) a singleton attractor, (b) a simple loop, and (c) a complex loop. We omit selfloops for all the states.

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 the 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 an intricate topology and includes several loops. Fig. 2 (a), (b) and (c) show a singleton attractor, a simple loop and a complex loop, respectively. For an attractor A of BN, 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 $bas^S_{TS}(A) = \{s \in S \mid reach(s) \cap A \neq \emptyset$ and $reach(s) \cap A' = \emptyset$ for any other A' of BN, $A' \neq A\}$. 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 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.

Example 3. *The network in Example 1 has three attractors $A_1 = \{000\}$, $A_2 = \{110\}$ and $A_3 = \{111\}$, indicated as dark grey nodes in Fig. 1(a). For attractor A_1, its strong basin, $bas^S_{TS}(A_1)$, contains two states $\{000, 001\}$; its weak basin, $bas^W_{TS}(A_1)$, contains six states $\{000, 001, 101, 011, 100, 010\}$.*

3 The target control problems

We have studied the source-target control of Boolean networks [26], [27], [28], [29], [30], [32], [37], to identify control paths that can drive the dynamics of the network from the source attractor to the target attractor. When the source is not given, to identify a subset of nodes, the control of which can stir the dynamics from any state $s \in S$ to the target attractor A_t, is called *target control* of Boolean networks. Target control neglects the values of the nodes in s, the application of a target control C inhibits the nodes in $\mathsf{C}_{\mathbb{0}}$ and overexpresses the nodes in $\mathsf{C}_{\mathbb{1}}$.

For target control, when perturbations are applied instantaneously, temporarily or permanently, we call it *instantaneous target control (ITC)*, *temporary target control (TTC)* or *permanent target control (PTC)*, respectively. Let BN be a given Boolean network, S be the set of states of BN and A_t be the target attractor of BN. We formally define the three target control problems, ITC, TTC and PTC, as follows.

Definition 7 (Target control of Boolean networks)**.**

1) **Instantaneous target control (ITC)***: find a control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ such that the dynamics of* BN *always eventually reaches A_t on the instantaneous application of* C *to any initial state $s, s \in S$.*
2) **Temporary target control (TTC):** *find a control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ such that there exists a $t_0 \geq 0$ such that for all $t \geq t_0$, the dynamics of* BN *always eventually reaches A_t on the application of* C *to any initial state $s, s \in S$ for t steps.*
3) **Permanent target control (PTC):** *find a control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ such that the dynamics of* BN *always eventually reaches A_t on the permanent application of* C *to any initial state $s, s \in S$. (We assume implicitly that A_t is also an attractor of the transition system under control $TS|_{\mathsf{C}}$).*

We define the concept of *schema*, which is crucial for the development of the target control methods. Given a control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$, the possible intermediate states with respect to C, denoted $S' = \mathsf{C}(S)$, form a schema, defined as follows.

Definition 8 (Schema)**.** *A subset S' of S is a schema if there exists a triple $M = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}}, \mathbb{D})$, where $\mathsf{C}_{\mathbb{0}} \cup \mathsf{C}_{\mathbb{1}} \cup \mathbb{D} = \{1, 2, \ldots, n\}$, $\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}}$ and $\mathbb{D}$ are mutually disjoint (possibly empty) sets of indices of nodes of* BN*, such that $S'|_{\mathsf{C}_{\mathbb{0}}} = \{0\}^{|\mathsf{C}_{\mathbb{0}}|}$, $S'|_{\mathsf{C}_{\mathbb{1}}} = \{1\}^{|\mathsf{C}_{\mathbb{1}}|}$ and $S'|_{\mathbb{D}} = \{0, 1\}^{|\mathbb{D}|}$. $\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}}$ and $\mathbb{D}$ are called off-set, on-set and don't-care-set of S', respectively. The elements in $\mathsf{C}_{\mathbb{0}} \cup \mathsf{C}_{\mathbb{1}}$ are called indices of support variables of S'.*

Intuitively, for node $x_i, i \in \mathsf{C}_{\mathbb{0}}$, it has a value of 0 in any state $s \in S'$; for node $x_i, i \in \mathsf{C}_{\mathbb{1}}$, 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 of $|\mathbb{D}|$ bits. Thus, any schema S' is of size $2^{|\mathbb{D}|}$. Since the total number of nodes $n = |\mathsf{C}_{\mathbb{0}}| + |\mathsf{C}_{\mathbb{1}}| + |\mathbb{D}|$ is fixed, a larger schema implies more elements in $\mathbb{D}$ and fewer elements in $\mathsf{C}_{\mathbb{0}} \cup \mathsf{C}_{\mathbb{1}}$.

Example 4. *Let us denote the values of the nodes in off-set, on-set and don't-care-set as '0', '1' and '$*$', respectively. For attractor A_1 in Fig. 1 (a), its strong basin, $bas^S_{TS}(A_1) = \{000, 001\}$, forms a schema, represented as '$00*$'. The weak basin, $bas^W_{TS}(A_1) = \{000, 001, 010, 011, 101, 100\}$, can be partitioned into two schemata $\{000, 001, 100, 101\}$ and $\{010, 011\}$, represented as '$*0*$' and '$01*$', respectively.*

Suppose A_1 is the target attractor. $\{x_1 = 0, x_2 = 0\}$ is an ITC. The simultaneous inhibition of x_1 and x_2 can drive the network from any state to either state (001) or A_1. If the network reaches A_1, we are done. If the network reaches (001), the network will spontaneously stabilise to A_1.

The control set $\{x_2 = 0\}$ is both a ITC and a PTC. As introduced in Example 2, this control reshapes the transition system to Fig. 1 (b), which also means that the control drives the network from any initial state to a state in Fig. 1 (b). If we use temporary perturbations, they can be released once the network evolves to the strong basin of A_1. Based on the definition of the strong basin, the network will eventually reach A_1. If we apply permanent perturbations, the network will surely reach A_1 as A_1 is the only attractor in the new transition system.

4 Instantaneous target control

An instantaneous control C will surely guide the dynamics of BN from any initial state s to the target attractor A_t if the intermediate state $s' = \mathsf{C}(s)$ is in the strong basin of A_t in

TS. Thus, when the initial state can be any state $s \in S$, to guarantee the inevitable reachability of the target attractor A_t on the instantaneous application of C to any $s \in S$, all possible intermediate states $S' = \mathsf{C}(S)$ must fall in the strong basin of the target attractor A_t. Based on the theorem in [26], we can derive the following corollary.

Corollary 1. *A control* $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ *is an instantaneous target control from any initial state* $s \in S$ *to a target attractor* A_t *iff* $\mathsf{C}(S) \subseteq bas^S_{TS}(A_t)$.

Instantaneous control is only applied instantaneously, thus, its impact on the transition system is transient. If the instantaneous control does not drive the dynamics directly to $bas^S_{TS}(A_t)$ but to any state $s' \in (S \setminus bas^S_{TS}(A_t))$, from s', there exist paths to some other attractor $A, A \neq A_t$, based on the definition of strong basin. This does not ensure the inevitable reachability of the target attractor. Therefore, for an ITC C, its intermediate states $S' = \mathsf{C}(S)$ must form a subset of $bas^S_{TS}(A_t)$. For any possible intermediate state $s' \in S'$, $s'[i] = 0$ for $i \in \mathsf{C}_{\mathbb{0}}$ and $s'[i] = 1$ for $i \in \mathsf{C}_{\mathbb{1}}$, which indicates that $S'|_{\mathsf{C}_{\mathbb{0}}} = \{0\}^{|\mathsf{C}_{\mathbb{0}}|}$ and $S'|_{\mathsf{C}_{\mathbb{1}}} = \{1\}^{|\mathsf{C}_{\mathbb{1}}|}$. Let $\mathbb{D}$ denote the indices of the nodes that are not in $\mathsf{C}_{\mathbb{0}}$ or $\mathsf{C}_{\mathbb{1}}$. Then, $S'|_{\mathbb{D}} = \{0,1\}^{|\mathbb{D}|}$ because the values of the nodes in the initial states $S|_{\mathbb{D}}$ stay unchanged. In another word

Observation 1. *If the initial state can be any state* $s \in S$*, for any control* $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$*, the set of intermediate states* $S' = \mathsf{C}(S)$ *forms a schema.*

The notion of schema sheds light on the computation of ITC. Each schema W_i of the strong basin of the target attractor, $bas^S_{TS}(A_t)$, returns a candidate target control $\mathsf{C}_i = (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$, where $\mathsf{C}_{\mathbb{0}i}$ and $\mathsf{C}_{\mathbb{1}i}$ are the off-set and on-set of W_i. The size of control $|\mathsf{C}_i|$ equals $(n - \log_2 |W_i|)$, therefore, a larger schema results in a smaller control set. Thus, we can partition the strong basin of the target attractor, $bas^S_{TS}(A_t)$, into a set of mutually disjoint schemata $\mathcal{W} = \{W_1, W_2, \ldots, W_m\}$, such that $W_1 \cup W_2 \cup \ldots \cup W_m = bas^S_{TS}(A_t)$. Each $W_i \in \mathcal{W}$ is one of the largest schemata in $bas^S_{TS}(A_t) \setminus (W_1 \cup \ldots \cup W_{i-1})$ and the indices of its support variables in $\mathsf{C}_{\mathbb{0}i}$ and $\mathsf{C}_{\mathbb{1}i}$ form a candidate ITC $\mathsf{C}_i = (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$. In C_i, the specified input nodes can be removed because input nodes do not have any predecessors and the values of the specified input nodes are fixed. For large networks, there may exist many valid control sets. To restrict the number and the size of solutions, we set a threshold ζ on the number of perturbations, keep ζ updated with the minimal size of the computed control sets, and only save the control sets with at most ζ perturbations. Algorithm 1 realises the above idea in pseudocode.

In this way, the computation of ITC is thus reduced to the computation of the strong basin of the target attractor and the computation of schemata. The computation of strong basins can be achieved efficiently with the procedure COMP_STRONG_BASIN, which implements a decomposition-based approach towards the computation of strong basins of Boolean networks (see [26], [30] for details). The computation of schemata is based on BDDs, a symbolic representation of large state space. The size of a BDD is determined by both the set of states being represented and the chosen ordering of the variables. In BDDs, a schema is represented as a *cube* and each state is the smallest cube,

Algorithm 1 Instantaneous target control

```
procedure INSTANTANEOUS_TARGET_CONTROL($\mathsf{BN}, A_t$)
    initialise $\mathcal{L} := \emptyset$ to store computed control sets
    $I, I^s, I^{ns}$ :=COMP_INPUT_NODES($G$) ▷ compute input nodes $I$, specified input nodes $I^s$ and non-specified input nodes $I^{ns}$
    $\mathsf{SB}$ :=COMP_STRONG_BASIN($F, A_t$)
    $\mathcal{W}$ :=COMP_SCHEMATA($\mathsf{SB}$), $m := |\mathcal{W}|$
    $\zeta := n$    ▷ a threshold on the number of perturbations
    for $i = 1 : m$ do    ▷ traverse the set of schemata
        $\mathsf{C}_i$ :=COMP_SUPPORT_VARIABLES($W_i$)    ▷ $\mathsf{C}_i := (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$
        $\mathsf{C}_i := (\mathsf{C}_{\mathbb{0}i} \setminus I^s, \mathsf{C}_{\mathbb{1}i} \setminus I^s)$ ▷ remove specified input nodes
        if $|\mathsf{C}_i| \leq \zeta$ then
            save $\mathsf{C}_i$ to $\mathcal{L}$
            $\zeta := min(|\mathsf{C}_i|, \zeta)$
    return $\mathcal{L}$
```

also called a *minterm*. To compute the largest schema S_i of S is equivalent to the computation of the largest cube of S. The partitioning of the strong basin into schemata is then transformed into a cube cover problem in BDDs. A different variable ordering may lead to a different partitioning. Given a fixed ordering, the partitioning remains the same. Although finding the best variable ordering is NP-hard, there exist efficient heuristics to find the optimal ordering. For our work, we compute a partitioning under one variable ordering as provided by the CUDD package [38] and we call this procedure COMP_SCHEMATA.

5 TEMPORARY TARGET CONTROL

In this section, we develop a method for TTC. First, we introduce the following corollary, which can be derived from the theorem in [27].

Corollary 2. *A control* $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ *is a temporary target control to a target attractor* A_t *from any initial state* $s \in S$ *iff* $bas^S_{TS}(A_t) \cap S|_{\mathsf{C}} \neq \emptyset$ *and* $\mathsf{C}(S) \subseteq bas^S_{TS|_{\mathsf{C}}}(bas^S_{TS}(A_t) \cap S|_{\mathsf{C}})$.

Below, we give an intuitive explanation of Corollary 2. We know that the application of a control C results in a new Boolean network $\mathsf{BN}|_{\mathsf{C}}$ and the state space is restricted to $S|_{\mathsf{C}}$. 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 w.r.t. 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 $S|_{\mathsf{C}}$, i.e. $(bas^S_{TS}(A_t) \cap S|_{\mathsf{C}})$, is a non-empty set; otherwise, it is not guaranteed to reach A_t. Furthermore, the condition $\mathsf{C}(S) \subseteq bas^S_{TS|_{\mathsf{C}}}(bas^S_{TS}(A_t) \cap S|_{\mathsf{C}})$ ensures that any possible intermediate state $s' \in \mathsf{C}(S)$ is in the strong basin of the remaining strong basin $(bas^S_{TS}(A_t) \cap S|_{\mathsf{C}})$ in the transition system under control $TS|_{\mathsf{C}}$, 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 TTC.

Algorithm 2 Temporary target control

```
 1: procedure TEMPORARY_TARGET_CONTROL($\mathsf{BN}, A_t$)
 2:     initialise $\mathcal{L} := \emptyset$ and $\Omega := \emptyset$ to store valid and checked control sets, resp.
 3:     $I, I^{ns}$ :=COMP_INPUT_NODES($G$)        ▷ compute input nodes $I$ and non-specified input nodes $I^{ns}$.
 4:     SB :=COMP_STRONG_BASIN($F, A_t$)        ▷ strong basin of $A_t$ in $TS$
 5:     WB :=COMP_WEAK_BASIN($F, A_t$)        ▷ weak basin of $A_t$ in $TS$
 6:     $\mathcal{W}$ :=COMP_SCHEMATA(WB), $m := |\mathcal{W}|$
 7:     generate a vector $\Theta$ of length $m$ and set all the elements to $false$
 8:     $\zeta := n$        ▷ a threshold on the number of perturbations.
 9:     for $i = 1 : m$ do        ▷ traverse the set of schemata
10:         if $\Theta[i] = true$, then continue
11:         $\mathsf{C}_i$ :=COMP_SUPPORT_VARIABLES($W_i$)        ▷ $\mathsf{C}_i := (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$
12:         $\mathsf{C}_i^e := (\mathsf{C}_{\mathbb{0}i} \cap I^{ns}, \mathsf{C}_{\mathbb{1}i} \cap I^{ns})$, $\mathsf{C}_i^r := (\mathsf{C}_{\mathbb{0}i} \setminus I, \mathsf{C}_{\mathbb{1}i} \setminus I)$        ▷ essential control nodes and non-input nodes in $\mathsf{C}_i$
13:         $k := 0$, $isValid := false$
14:         while $isValid = false$ and $k \leq \min(\zeta - |\mathsf{C}_i^e|, |\mathsf{C}_i^r|)$ do
15:             $\mathsf{C}_i^{sub}$ :=COMP_SUBSETS($\mathsf{C}_i^r, k$)        ▷ compute subsets of $\mathsf{C}_i^r$ of size $k$.
16:             for $\mathsf{C}_j^{sub} \in \mathsf{C}_i^{sub}$ do
17:                 $\mathsf{C}_i^j := \mathsf{C}_j^{sub} \cup \mathsf{C}_i^e$, $\Phi := \mathsf{C}_i^j(S)$        ▷ $\Phi$: intermediate states.
18:                 if $\mathsf{C}_i^j \notin \Omega$ then        ▷ $\mathsf{C}_i$ has not been checked.
19:                     $isValid$ :=VERIFY_TTC($F, \mathsf{C}_i^j, \mathsf{SB}, \Phi$)
20:                     add $\mathsf{C}_i^j$ to $\Omega$.
21:                     if $isValid = true$ then
22:                         add $\mathsf{C}_i^j$ to $\mathcal{L}$, $\zeta := \min(\zeta, |\mathsf{C}_i^j|)$
23:                         $\Theta[z] := true$ if $W_z \subseteq \Phi$ for $z \in [i+1, m]$        ▷ if a schema $W_z$ is a subset of $\Phi$, it will be skipped.
24:             if $isValid = false$, then $k := k + 1$
25:     return $\mathcal{L}$
```

Algorithm 3 Verification of temporary target control

```
 1: procedure VERIFY_TTC($F, \mathsf{C}, \mathsf{SB}, \Phi$)
 2:     $isValid := false$
 3:     if $\Phi \subseteq \mathsf{SB}$ then
 4:         $isValid = true$
 5:     else
 6:         $\mathsf{SB}|_\mathsf{C}$ :=COMP_STATE_CONTROL($\mathsf{C}, \mathsf{SB}$)        ▷ compute the remaining strong basin w.r.t. $\mathsf{C}$ in $TS|_\mathsf{C}$
 7:         $F|_\mathsf{C}$ :=COMP_FN_CONTROL($\mathsf{C}, F$)
 8:         $bas^S_{TS|_\mathsf{C}}(\mathsf{SB}|_\mathsf{C})$ :=COMP_STRONG_BASIN($F|_\mathsf{C}, \mathsf{SB}|_\mathsf{C}$)
 9:         if $\Phi \subseteq bas^S_{TS|_\mathsf{C}}(SB|_\mathsf{C})$ then
10:             $isValid = true$
11:     return $isValid$
```

A noteworthy point is that temporary control needs to be released once the network reaches a state in $(bas^S_{TS}(A_t) \cap S|_\mathsf{C})$. On one hand, although Corollary 2 guarantees that partial of the strong basin of A_t in TS is preserved in $TS|_\mathsf{C}$, it does not guarantee the presence of A_t in $TS|_\mathsf{C}$. 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 desired for biologists to estimate the timing based on empirical knowledge and specific experimental settings.

Similar to ITC, the computation of TTC is also based on the concept of schema. Each schema W_i of the weak basin $bas^W_{TS}(A_t)$ gives a candidate TTC, $\mathsf{C}_i = (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$, 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 $\mathcal{W} = \{W_1, W_2, \ldots, W_m\}$, $W_1 \cup W_2 \cup \ldots \cup W_m = bas^W_{TS}(A_t)$. Each $W_i, i \in m$ is one of the largest schemata in $bas^W_{TS}(A_t) \setminus (W_1 \cup \ldots \cup W_{i-1})$. For W_i, the indices of its support variables in $\mathsf{C}_{\mathbb{0}i}$ and $\mathsf{C}_{\mathbb{1}i}$ form a candidate control $\mathsf{C}_i = (\mathsf{C}_{\mathbb{0}i}, \mathsf{C}_{\mathbb{1}i})$. 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_i^r, we verify its subsets of size k based on Corollary 2 from $k = 0$ with an increment of 1, until we find a valid solution.

Algorithm 2 implements the above idea in pseudocode. It takes as inputs the Boolean network $\mathsf{BN} = (X, F)$ 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 weak basin WB and the strong basin SB of A_t of TS are computed using procedures COMP_WEAK_BASIN and COMP_STRONG_BASIN developed in our previous work [26] (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 [38]. 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 and the non-input nodes

Algorithm 4 Verification of permanent target control

```
1: procedure Verify_PTC($F, \mathsf{C}, A_t, \Phi$)
2:     $isValid := false$
3:     if $\Phi|_{\mathsf{C}} = A_t|_{\mathsf{C}}$ then
4:         $F|_{\mathsf{C}}$ :=Comp_Fn_control($\mathsf{C}, F$)
5:         $bas^S_{TS|_{\mathsf{C}}}(A_t)$ :=Comp_Strong_Basin($F|_{\mathsf{C}}, A_t$)
6:         if $\Phi \subseteq bas^S_{TS|_{\mathsf{C}}}(A_t)$ then
7:             $isValid = true$
8:     return $isValid$
```

in C_i constitute a set C^r_i for further optimisation (line 12). We search the subsets of C^r_i starting from size $k = 0$ with an increment of 1 and verify whether the union of a subset C^{sub}_j of C^r_i and the essential nodes C^e_i, namely $\mathsf{C}^j_i = \mathsf{C}^{sub}_j \cup \mathsf{C}^e_i$, is a valid temporary target control using procedure Verify_TTC in Algorithm 3. If C^j_i is valid, save it to $\mathcal{L}$. When all the subsets are traversed or a valid control has been found, we proceed to the next schema W_{i+1}. In the end, all the verified TTCs are returned.

The most time-consuming part of our method lies in the verification process. As shown in Algorithm 3, for each candidate control C, we need to reconstruct the associated transition relations $F|_{\mathsf{C}}$ and compute the strong basin of the remaining strong basin in $TS|_{\mathsf{C}}$, i.e. $bas^S_{TS|_{\mathsf{C}}}(\mathsf{SB}|_{\mathsf{C}})$ (lines 6-8 of Algorithm 3). Even though we have developed an efficient method for the strong basin computation, the computational time of Algorithm 3 still increases when the network size grows. To improve the efficiency, we propose two heuristics: (1) skip a schema W_z (lines 10 and 23 of Algorithm 2) if it is a subset of intermediate states Φ of a pre-validated control C^j_i (line 23 of Algorithm 2); and (2) set a threshold ζ on the number of perturbations, keep ζ updated with the smallest size of valid TTCs C^j_i (line 22 of Algorithm 2) and only compute control sets with at most ζ perturbations.

6 Permanent target control

In this section, we develop a method to solve the problem of PTC. We first introduce the following corollary derived from the theorem in [27].

Corollary 3. *A control $\mathsf{C} = (\mathsf{C}_{\mathbb{0}}, \mathsf{C}_{\mathbb{1}})$ is a permanent target control from any initial state $s \in S$ to a target attractor A_t iff A_t is an attractor of $TS|_{\mathsf{C}}$ and $\mathsf{C}(S) \subseteq bas^S_{TS|_{\mathsf{C}}}(A_t)$.*

Different from temporary control, permanent control is applied for all the following time steps. A permanent control need to preserve the target attractor. To guarantee the inevitable reachability of the attractor, all possible intermediate states should fall in the strong basin of the target attractor A_t in the transition system under control, $TS|_{\mathsf{C}}$.

The algorithm for PTC can be derived from Algorithm 2 by replacing procedure Verify_TTC with procedure Verify_PTC in Algorithm 4. To avoid duplication, we only explain procedure Verify_PTC here. This procedure is designed based on Corollary 3. Line 3 verifies whether the target attractor is preserved or not. If the target is preserved, we compute the transition relations under control $F|_{\mathsf{C}}$ and compute the strong basin of A_t in $TS|_{\mathsf{C}}$ (lines 4-5). C is a PTC if the set of intermediate states is a subset of $bas^S_{TS|_{\mathsf{C}}}(A_t)$ (lines 6-7).

	granulocytes	monocytes	megakaryocytes	erythrocytes
ITC	6	7	4	4
TTC	3	3	2	2
PTC	3	3	2	2
SMC	4	4	2	2

TABLE 1
The number of perturbations required by ITC, TTC, PTC and SMC of the myeloid differentiation network.

The most time-consuming step of the three algorithms lies in the basin computation of the target attractor. As introduced in our previous work [26], the basin computation is PSPACE-hard. Even though we developed the decomposition-based method [26] for the basin computation, in the worst case, the complexity can still be exponential in the size of the network. Nevertheless, in practice, our algorithms tend to be quite efficient due to that real-life biological networks have modular structures.

Example 5. *In Example 4, we showed that the strong basin of A_1 of* BN*, $bas^S_{TS}(A_1)$, can be represented as a schema '$00*$'. Based on Algorithm 1, we know that its its support variables form an ITC $\{x_1 = 0, x_2 = 0\}$.*

The weak basin of A_1, $bas^W_{TS}(A_1)$, can be divided into two schemata, represented as '$*0*$' and '$01*$'. '$*0*$' contains more states than '$01*$', which implies that '$*0*$' can potentially give a smaller TTC or PTC. Algorithms for TTC and PTC verify subsets of the control derived from '$*0*$' and '$01*$'. Based on Corollary 2 and Corollary 3, $\{x_2 = 0\}$ is both a TTC and a PTC for A_1. By fixing x_2 to 0, the transition system changes from Fig. 1 (a) to Fig. 1 (b). The network is driven to a state in $TS|_{\mathsf{C}}$ (see Fig. 1 (b)) and will eventually stable in A_1.

7 Evaluation

In this paper, we developed three methods, ITC, TTC and PTC, for the target control of asynchronous Boolean networks. We apply our methods on a variety of biological networks and compare their performance with SMC [33]. SMC predicts a set of temporary perturbations that can guide the dynamics from any initial states to the desired target attractor. It takes into account the functional information of the network (network dynamics) and has high efficiency.

We discuss the results of the myeloid differentiation network and the cardiac gene regulatory network in detail and give an overview of the results of the other networks. Our target control methods are implemented in our software CABEAN [39], which contains a realisation of the decomposition-based detection method [40], [41] for identifying attractors in Boolean networks efficiently.[1] SMC is implemented in Java. All the experiments are performed on a high-performance computing (HPC) platform, which contains CPUs of Intel Xeon Gold 6132 @2.6 GHz.

7.1 Control of the myeloid differentiation network

The myeloid differentiation network is designed to model myeloid differentiation from common myeloid progenitors

[1] These attractor detection methods were originally implemented in the software tool ASSA-PBN [35], [42], [43].

	granulocytes	monocytes
ITC	{GATA2=0 GATA1=0 Fli1=0 EgrNab=0 C/EBPα=1 Gfi1=1}	{GATA2=0 GATA1=0 EgrNab=1 C/EBPα=1 PU1=1 cJun=1 Gfi1=0}

TTC	{C/EBPα=1 PU1=1 cJun=0}
	{EgrNab=0 C/EBPα=1 PU1=1}
	{C/EBPα=1 PU1=1 Gfi1=1}
		{EgrNab=1 C/EBPα=1 PU1=1}
		{C/EBPα=1 PU1=1 Gfi1=0}

PTC	{C/EBPα=1 PU1=1 cJun=0}
	{EgrNab=0 C/EBPα=1 PU1=1}
	{C/EBPα=1 PU1=1 Gfi1=1}
		{EgrNab=1 C/EBPα=1 PU1=1}
		{C/EBPα=1 PU1=1 Gfi1=0}

SMC	{GATA2=0 GATA1=0 C/EBPα=1 EgrNab=0}
	{GATA2=0 GATA1=0 C/EBPα=1 Gfi1=1}
	{GATA2=0 PU1=1 C/EBPα=1 EgrNab=0}
	{GATA2=0 PU1=1 C/EBPα=1 Gfi1=1}
		{GATA2=0 GATA1=0 C/EBPα=1 EgrNab=1}
		{GATA2=0 GATA1=0 C/EBPα=1 Gfi1=0}
		{GATA2=0 PU1=1 EgrNab=1 C/EBPα=1}
		{GATA2=0 PU1=1 Gfi1=0 C/EBPα=1}

TABLE 2
The control sets computed by ITC, TTC, PTC and SMC for granulocytes and monocytes of the myeloid differentiation network.

	megakaryocytes	erythrocytes
ITC	{GATA2=0 GATA1=0 EgrNab=1 C/EBPα=1 PU1=1 cJun=1 Gfi1=0}	{GATA1=1 EKLF=1 Fli1=0 PU1=0}

TTC	{GATA2=1 EKLF=0}
	{GATA1=1 EKLF=0}
	{GATA2=1 Fli1=1}
	{GATA1=1 Fli1=1}
	{Fli1=1 PU1=0 }
		{GATA2=1 EKLF=1}
		{GATA1=1 EKLF=1}
		{GATA2=1 Fli1=0}
		{GATA1=1 Fli1=0}

PTC	{GATA1=1 EKLF=0 }
	{GATA1=1 Fli1=1}
	{Fli1=1 PU1=0 }
		{GATA1=1 EKLF=1 }
		{GATA1=1 Fli1=0 }

SMC	{GATA1=1 EKLF=0}
	{GATA1=1 Fli1=1}
		{GATA1=1 EKLF=1}
		{GATA1=1 Fli1=0}

TABLE 3
The control sets computed by ITC, TTC, PTC and SMC for megakaryocytes and erythrocytes of the myeloid differentiation network.

to granulocytes, monocytes, megakaryocytes and erythrocytes [44]. This network has 11 nodes and 6 attractors, 4 of which correspond to granulocytes, monocytes, megakaryocytes and erythrocytes. We apply ITC, TTC, PTC and SMC to identify interventions to reach the four cell types.

Table 1 gives the number of perturbations required by the four methods. The control with instantaneous perturbations (ITC) requires more perturbations than the control with temporary perturbations (TTC and SMC) and permanent perturbations (PTC) as expected. For granulocytes and monocytes, TTC and PTC find smaller control sets than SMC. For megakaryocytes and erythrocytes, TTC, PTC and SMC require the same number of perturbations.

Table 2 and Table 3 summarise the control sets identified by the four methods. For each attractor, ITC only finds one control set with more perturbations than the other methods. Although SMC finds more control sets than TTC for granulocytes and monocytes as shown in Table 2, SMC requires four perturbations while TTC and PTC need only three perturbations. Since our methods TTC and PTC only compute the results within the threshold, they may identify more solutions if we increase the threshold to four. For megakaryocytes and erythrocytes in Table 3, TTC, PTC and SMC require the same number of perturbations, but TTC provides more control sets than PTC and SMC. For this network, the results of PTC are either identical to TTC or just a subset of the solutions identified by TTC. Potentially, TTC is able to find smaller control sets than PTC, because it does not need to preserve the target attractor during the control. We will demonstrate this point in Section 7.3.

The total execution time of ITC, TTC, PTC and SMC for computing target control of this network are 0.003, 0.026, 0.03 and 8.178 seconds, respectively. We can see that our methods outperform SMC in efficiency.

7.2 Control of the cardiac gene regulatory network

The cardiac gene regulatory network integrates major genes that play essential roles in early cardiac development and FHF/SHF determination [45]. It has 15 nodes. This network consists of three attractors, two of which correspond to FHF and SHF, when the input node, exogenBMP2I, is set to 1 [45]. We apply ITC, TTC, PTC and SMC to identify interventions that can lead the network to FHF and SHF. The results of the four control methods are given in Table 4.

With instantaneous, temporary or permanent perturbations, it is guaranteed to reach SHF by the control of the non-initialised input node, exogenCanWntI. To reach FHF, ITC requires seven perturbations, while TTC realises the goal by the control of two nodes, including the input node exogenCanWntI together with one of the nodes in {GATAs, Tbx5, Nkx25, Mesp1, Tbx1, Foxc12}. The results of PTC and SMC are subsets of TTC, which demonstrates the ability of TTC in identifying more novel solutions. With temporary and permanent perturbations, the number of perturbations

	SHF	FHF
ITC	{exogenCanWntI=1}	{canWnt=0 Foxc12=0 Tbx1=0 GATAs=0 Tbx5=1 exogenCanWntI=0 exogenCanWntII=0 }

TTC	{exogenCanWntI=1}	{GATAs=1 exogenCanWntI=0 }
		{Tbx5=1 exogenCanWntI=0}
		{Nkx25=1 exogenCanWntI=0}
		{Mesp1=1 exogenCanWntI=0}
		{Tbx1=1 exogenCanWntI=0}
		{Foxc12=1 exogenCanWntI=0}

PTC	{exogenCanWntI=1}	{GATAs=1 exogenCanWntI=0}
		{Nkx25=1 exogenCanWntI=0}
		{Tbx5=1 exogenCanWntI=0}

SMC	{exogenCanWntI=1}	{GATAs=1 exogenCanWntI=0 }
		{Tbx5=1 exogenCanWntI=0}
		{Nkx25=1 exogenCanWntI=0}

TABLE 4

The control sets computed by ITC, TTC, PTC and SMC for SHF and FHF of the cardiac gene regulatory network.

Network	# nodes	# edges	# singleton attractors	# cyclic attractors
yeast	10	28	12	1
ERBB	20	52	3	0
HSPC-MSC	26	81	2	2
tumour	32	158	9	0
hematopoiesis	33	88	5	0
PC12	33	62	7	0
bladder	35	116	3	1
PSC-bFA	36	237	4	0
co-infection	52	136	30	0
MAPK	53	105	12	0
CREB	64	159	8	0
HGF	66	103	10	0
bortezomib	67	135	5	0
T-diff	68	175	12	0
HIV1	136	327	8	0
CD4+	188	380	6	0
pathway	321	381	3	1

TABLE 5

An overview of the biological networks.

required to reach FHF is reduced from seven to two, which can greatly reduce experimental costs and improve the operability of the experiments.

The total execution time of ITC, TTC, PTC and SMC for computing target control for the three attractors are 0.035, 0.128, 0.148 and 4.540 seconds, respectively. For this network, our methods are more efficient than SMC.

7.3 Control of other biological networks

We apply the four target control methods, ITC, TTC, PTC and SMC, to some other networks introduced below to evaluate their performance. We give an overview of the number of nodes, the number of edges and the number of singleton and cyclic attractors of each network in Table 5. Details on the networks, such as the Boolean functions, are referred to the original works.

- The cell cycle network of the fission yeast is constructed based on known biochemical interactions to recap regulations of the cell cycle of the fission yeast [46].
- 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 [47].
- The HSPC-MSC network of 26 nodes describes intercommunication pathways between hematopoietic stem and progenitor cells (HSPCs) and mesenchymal stromal cells (MSCs) in bone marrow (BM) [48].
- The tumour network is built to study the role of individual mutations or their combinations in the metastatic process [49].
- The network of hematopoietic cell specification covers major transcription factors and signalling pathways for the development of lymphoid and myeloid [50].
- The PC12 cell differentiation network [51] is a comprehensive model for the clarification of the cellular decisions towards proliferation or differentiation. It models the temporal sequence of protein signalling, transcriptional responses as well as the subsequent autocrine feedbacks [51].
- The bladder cancer network of 35 nodes allows one to identify the deregulated pathways and their influence on bladder tumourigenesis [52].
- The model of mouse embryonic stem cells captures the signal-dependent emergence of cell subpopulations under different initial conditions [53].
- The model of immune responses is constructed to study the immune responses against single and co-infections with the respiratory bacterium and the gastrointestinal helminth [54].
- The MAPK network is constructed to study the MAPK responses to different stimuli and their contributions to cell fates [55].
- The CREB network is a complex neuronal network, whose output node is the transcription factor adenosine 3′, 5′-monophosphate (cAMP) response element–binding protein (CREB) [56].
- The model for HGF-induced keratinocyte migration captures the onset and maintenance of hepatocyte growth factor-induced migration of primary human keratinocytes [57].
- The model of bortezomib responses can predict responses to the lower bortezomib exposure and the dose-response curve for bortezomib [58].
- The Th-cell differentiation network models regulatory elements and signalling pathways controlling Th-cell differentiation [59].
- The HIV-1 network models dynamic interactions be-

Network	The minimal number of perturbations			
	ITC	TTC	PTC	SMC
yeast	10	5	5	5
ERBB	10	2	2	2
HSPC-MSC	2	2	2	2
hematopoiesis	5	3	3	*
PC12	12	3	3	3
bladder	14	2	2	4
PSC-bFA	11	1	2	*
co-infection	19	5	5	7
MAPK	24	4	4	5
CREB	3	3	3	*
HGF	22	4	4	*
bortezomib	3	1	1	*
T-diff	20	4	4	4
HIV1	3	3	3	*
CD4+	7	3	3	3
pathway	2	2	2	2

TABLE 6
The minimal number of perturbations required by ITC, TTC, PTC and SMC for several biological networks. Symbol '*' means that the method failed to finish the computation within twelve hours.

tween human immunodeficiency virus type 1 (HIV-1) proteins and human signal-transduction pathways that are essential for activation of CD4+ T lymphocytes [60].

- The CD4$^+$ T-cell network allows us to study downstream effects of CAV1$^{+/+}$, CAV1$^{+/-}$ and CAV1$^{-/-}$ on cell signalling and intracellular networks [61].
- The model of signalling pathways central to macrophage activation integrates four crucial signalling pathways that are triggered early-on in the innate immune response [62].

Efficacy. Table 6 summarises the minimal number of perturbations required by the four methods for one of the attractors of the networks. It is easy to observe that ITC requires more perturbations than TTC, PTC and SMC due to its instantaneous effect. ITC usually needs to control 10 to 20 nodes, whereas TTC, PTC and SMC can achieve the inevitable reachability of the target attractor with at most 7 perturbations. Moreover, it is hard to realise the simultaneous and instantaneous perturbation of a number of nodes, which makes the ITC less practical in application. Thus, TTC, PTC and SMC, which employ temporary or permanent perturbations, are preferable than ITC. For the bladder cancer network and the MAPK network, TTC and PTC identify smaller control sets than SMC. Compared to PTC, TTC has the ability to further reduce the number of perturbations as demonstrated by the model of mouse embryonic stem cells (PSC-bFA) – the number of perturbations required by TTC and PTC are 1 and 2, respectively.

Both TTC and SMC solve the target control problem with temporary perturbations. To further compare these two methods, Fig. 3 shows the number of solutions identified by the two methods. The x-axis lists the names of the networks and the y-axis denotes the number of control sets. Blue bars and grey bars represent the control sets that only appear in the results of TTC and SMC (TTC$\setminus$ (TTC $\cap$ SMC), SMC$\setminus$ (TTC $\cap$ SMC)), respectively. Green bars represent the intersections of the two methods. Equations above the bars ($|\mathsf{C}| = k$) denote the number of nodes contained in the control set, i.e. the number of required perturbations.

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. In Fig. 3, there is no grey bar because the solutions identified by SMC are either also found by TTC and thus belong to (TTC $\cap$ SMC), or require more perturbations than TTC. For the cell cycle network of fission yeast (yeast), the ERBB receptor-regulated G1/S transition protein network (ERBB), the HSPC-MSC network (HSPC-MSC) and the model of signalling pathways (pathway), there are only green bars, which means that the results of TTC and SMC are identical. For the bladder cancer network (bladder), the co-infection network (co-infection) and the MAPK network (MAPK), we can only see blue bars because TTC finds smaller control sets than SMC. SMC failed to finish the computation for several networks within twelve hours, including the network of hematopoietic cell specification (hematopoiesis), the model of mouse embryonic stem cells (PSC-bFA), the CREB network, the model for HGF-induced keratinocyte migration (HGF), the model of bortezomib responses (bortezomib) and the HIV-1 network networks (HIV1). For the PC12 cell differentiation network (PC12), the Th-cell differentiation network (T-diff) and the CD4+ T-cell network (CD4+), although TTC and SMC require the same number of perturbations, our method TTC has the capability to provide more unique solutions, which may give more flexibility for practical applications.

Efficiency. Table 7 summarises the computational time for computing the target control for all the attractors of the networks rather than the selected target attractor. The reason is that SMC computes the control for all the attractors in one-go by generating the stable motif diagram, in which different sequences of stable motifs lead to different attractors. SMC does not support the computation of target control for only one attractor. Hence, for ITC, TTC and PTC, we also take the total computational time for all the attractors of the networks in order to have a fair comparison with SMC.

We can see that ITC is the most efficient one, however, it requires more perturbations. TTC and PTC are more efficient than SMC for most of the cases. The efficiency of our methods are influenced by many factors, such as the network size, the network density, the number of attractors and the number of required perturbations. For the co-infection network and the model of bortezomib responses, TTC and PTC are able to identify target control efficiently for some of the attractors, but failed for the other attractors. One conjecture is that the target control of those attractors require many perturbations, such that it takes a considerable amount of time to verify the subsets of the schemata.

SMC failed to finish the computation for several networks within twelve hours, including the hematopoiesis, PSC-bFA, CREB, HGF, bortezomib and HIV1 networks. For the hematopoiesis network and the bortezomib network, SMC failed in the identification of stable motifs, which has been pointed out to be the most time-consuming part of SMC [33]. 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 was 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

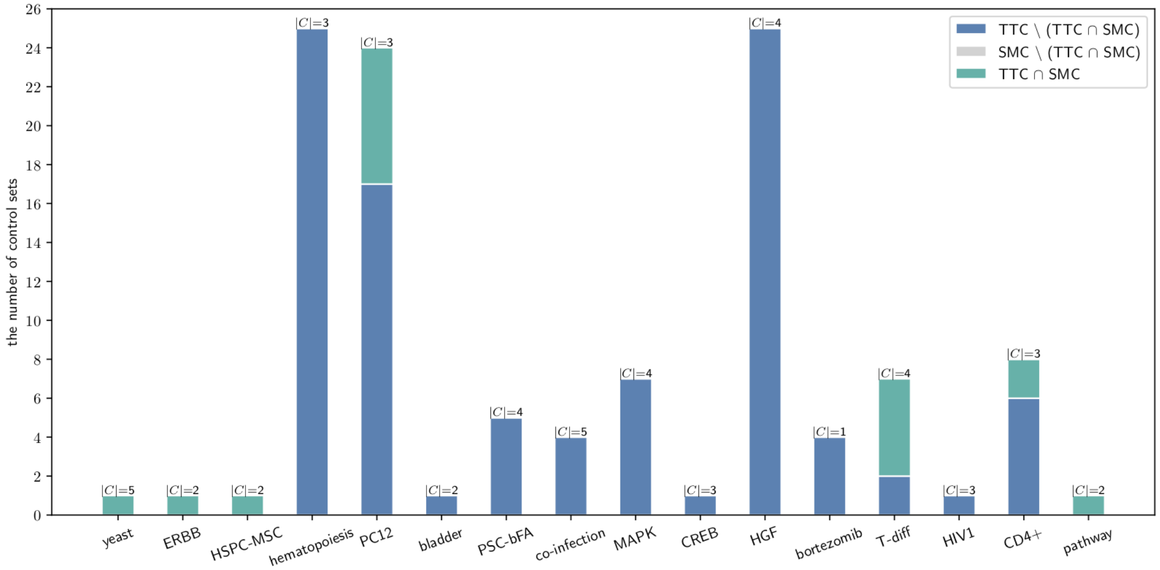


Fig. 3. Comparison of TTC and SMC on the number of solutions.

Network	Time (seconds)			
	ITC	TTC	PTC	SMC
yeast	0.028	0.987	0.933	10.837
ERBB	0.055	0.117	0.163	6.400
HSPC-MSC	0.097	0.101	0.109	11.393
hematopoiesis	0.374	139.859	72.793	*
PC12	0.149	17.653	22.189	234.513
bladder	0.302	2.426	7.997	36.277
psc-bFA	36.77	3732.78	9296.740	*
co-infection	6294.29	*	*	15097.511
MAPK	4.608	22.218	45.504	395.014
CREB	7.962	8.277	8.693	*
HGF	19.925	1437.29	201.363	*
bortezomib	15.605	*	*	*
T-diff	21.581	29738.5	*	353.473
HIV1	302.8	323.666	379.127	*
CD4+	549.878	1982.45	21358.400	27.836
pathway	445.251	4435.59	10038.600	42.180

TABLE 7
Computational time of ITC, TTC, PTC and SMC for several biological networks. Symbol '*' means that the method failed to finish the computation within twelve hours.

failed to construct the expanded network representation for the CREB, PSC-BFA and HIV-1 networks because some of their Boolean functions depend on many parent nodes ($k \geq 10$). Detailed discussion on the complexity of SMC can be found in [33].

From the above analysis, we can recommend TTC to compute the target control sets for networks of small and medium sizes (e.g., less than 100 nodes). For larger networks, SMC generally outperforms TTC. TTC works well on networks which contain relatively small SCCs, while SMC is more suitable for networks that have fewer cycles/SCCs in their expanded network.

8 Conclusion

In this paper, we have developed three methods for the target control of asynchronous Boolean networks with instantaneous, temporary and permanent perturbations. We compared their performance with SMC [33] on various real-life biological networks. The results showed that ITC requires more perturbations than TTC, PTC and SMC as it uses instantaneous perturbations. Both TTC and SMC solve the target control problem with temporary perturbations and potentially they may require fewer perturbations than PTC. Moreover, from Table 6 and Figure 3, our method TTC has the potential to identify more solutions with even fewer perturbations when compared to SMC. This requires further statistical analysis on a large number of Boolean networks in the future.

Regarding the computational time, our methods are quite efficient and scale well for large networks. SMC explores both structures and Boolean functions of Boolean networks, and is potentially more scalable for large networks as demonstrated by the CD4+ T-cell network and the pathway network in Table 7. In contrast, our methods are essentially based on the dynamics of the networks, and they will suffer the state space explosion problem for networks of several hundreds of nodes. We believe that our methods and SMC complement each other. In the near future, we aim to improve our methods by simultaneously exploring network structure and dynamics to achieve more efficient computational methods for the control of large biological networks. We will also apply our methods for the analysis of real-life biological networks with the aim of identifying potential drug targets for effective disease treatments.

Acknowledgments

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

References

[1] D. Srivastava and N. DeWitt, "In vivo cellular reprogramming: the next generation," *Cell*, vol. 166, no. 6, pp. 1386–1396, 2016.

[2] A. Grath and G. Dai, "Direct cell reprogramming for tissue engineering and regenerative medicine," *Journal of Biological Engineering*, vol. 13, no. 1, p. 14, 2019.

[3] M. S. Goligorsky, "New trends in regenerative medicine: reprogramming and reconditioning," *Journal of the American Society of Nephrology*, vol. 30, no. 11, pp. 2047–2051, 2019.

[4] L.-Z. Wang, R.-Q. Su, Z.-G. Huang, X. Wang, W.-X. Wang, C. Grebogi, and Y.-C. Lai, "A geometrical approach to control and controllability of nonlinear dynamical networks," *Nature Communications*, vol. 7, no. 1, pp. 1–11, 2016.

[5] S. Kauffman, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, pp. 177–178, 1969.

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

[7] S. Huang, "Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation," *Pharmacogenomics*, vol. 2, no. 3, pp. 203–222, 2001.

[8] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, pp. 167–173, 2011.

[9] J. Gao, Y.-Y. Liu, R. M. D'Souza, and A.-L. Barabási, "Target control of complex networks," *Nature Communications*, vol. 5, p. 5415, 2014.

[10] E. Czeizler, C. Gratie, W. K. Chiu, K. Kanhaiya, and I. Petre, "Target controllability of linear networks," in *Proc. 14th International Conference on Computational Methods in Systems Biology*, ser. LNCS, vol. 9859. Springer, 2016, pp. 67–81.

[11] A. J. Gates and L. M. Rocha, "Control of complex networks requires both structure and dynamics," *Scientific Reports*, vol. 6, no. 1, pp. 1–11, 2016.

[12] A. Mochizuki, B. Fiedler, G. Kurosawa, and D. Saito, "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*, vol. 335, pp. 130–146, 2013.

[13] B. Fiedler, A. Mochizuki, G. Kurosawa, and D. Saito, "Dynamics and control at feedback vertex sets. I: Informative and determining nodes in regulatory networks," *Journal of Dynamics and Differential Equations*, vol. 25, no. 3, pp. 563–604, 2013.

[14] J. G. T. Zañudo, G. Yang, and R. Albert, "Structure-based control of complex networks with nonlinear dynamics," *Proceedings of the National Academy of Sciences*, vol. 114, no. 28, pp. 7234–7239, 2017.

[15] S. P. Cornelius, W. L. Kath, and A. E. Motter, "Realistic control of network dynamics," *Nature Communications*, vol. 4, no. 1, pp. 1–9, 2013.

[16] J. Liang, H. Chen, and J. Lam, "An improved criterion for controllability of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 62, no. 11, pp. 6012–6018, 2017.

[17] Q. Zhu, Y. Liu, J. Lu, and J. Cao, "Further results on the controllability of Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 440–442, 2018.

[18] J. Lu, J. Zhong, D. W. Ho, Y. Tang, and J. Cao, "On controllability of delayed Boolean control networks," *SIAM Journal on Control and Optimization*, vol. 54, no. 2, pp. 475–494, 2016.

[19] J. Zhong, Y. Liu, K. I. Kou, L. Sun, and J. Cao, "On the ensemble controllability of Boolean control networks using STP method," *Applied Mathematics and Computation*, vol. 358, pp. 51–62, 2019.

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

[21] H. Chen, J. Liang, and Z. Wang, "Pinning controllability of autonomous Boolean control networks," *Science China Information Sciences*, vol. 59, no. 7, p. 070107, 2016.

[22] J. Yue, Y. Yan, Z. Chen, and X. Jin, "Identification of predictors of Boolean networks from observed attractor states," *Mathematical Methods in the Applied Sciences*, vol. 42, no. 11, pp. 3848–3864, 2019.

[23] Y. Zhao, J. Kim, and M. Filippone, "Aggregation algorithm towards large-scale Boolean network analysis," *IEEE Transactions on Automatic Control*, vol. 58, no. 8, pp. 1976–1985, 2013.

[24] J. Kim, S.-M. Park, and K.-H. Cho, "Discovery of a kernel for controlling biomolecular regulatory networks," *Scientific Reports*, vol. 3, p. 2223, 2013.

[25] M. Moradi, S. Goliaei, and M.-H. Foroughmand-Araabi, "A Boolean network control algorithm guided by forward dynamic programming," *PLOS ONE*, vol. 14, no. 5, p. e0215449, 2019.

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

[27] C. Su, S. Paul, and J. Pang, "Controlling large Boolean networks with temporary and permanent perturbations," in *Proc. 23rd International Symposium on Formal Methods*, ser. LNCS, vol. 11800. Springer, 2019, pp. 707–724.

[28] H. Mandon, C. Su, J. Pang, S. Paul, S. Haar, and L. Paulevé, "Algorithms for the sequential reprogramming of Boolean networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 5, pp. 1610–1619, 2019.

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

[30] S. Paul, C. Su, J. Pang, and A. Mizera, "An efficient approach towards the source-target control of Boolean networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 17, no. 6, pp. 1932–1945, 2020.

[31] A. del Sol and N. J. Buckley, "Concise review: A population shift view of cellular reprogramming," *Stem Cells*, vol. 32, no. 6, pp. 1367–1372, 2014.

[32] C. Su and J. Pang, "A dynamics-based approach for the target control of Boolean networks." in *Proc. 11th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM Press, 2020, pp. 50:1–50:8.

[33] J. G. Zañudo and R. Albert, "Cell fate reprogramming by control of intracellular network dynamics," *PLOS Computational Biology*, vol. 11, no. 4, p. e1004193, 2015.

[34] F.-J. Müller and A. Schuppert, "Few inputs can reprogram biological networks," *Nature*, vol. 478, no. 7369, p. E4, 2011.

[35] A. Mizera, J. Pang, C. Su, and Q. Yuan, "ASSA-PBN: A toolbox for probabilistic Boolean networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 4, pp. 1203–1216, 2018.

[36] P. Zhu and J. Han, "Asynchronous stochastic Boolean networks as gene network models," *Journal of Computational Biology*, vol. 21, no. 10, pp. 771–783, 2014.

[37] C. Su and J. Pang, "Sequential temporary and permanent control of Boolean networks." in *Proc. 18th International Conference on Computational Methods in Systems Biology*, ser. LNCS, vol. 12314. Springer, 2020, pp. 234–251.

[38] F. Somenzi, "CUDD: CU decision diagram package - release 2.5.1," http://vlsi.colorado.edu/ fabio/CUDD/, 2015.

[39] C. Su and J. Pang, "CABEAN: a software for the control of asynchronous Boolean networks," *Bioinformatics*, vol. 37, no. 6, pp. 879–881, 2021.

[40] A. Mizera, J. Pang, H. Qu, and Q. Yuan, "Taming asynchrony for attractor detection in large Boolean networks," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 31–42, 2019.

[41] Q. Yuan, A. Mizera, J. Pang, and H. Qu, "A new decomposition-based method for detecting attractors in synchronous boolean networks," *Science of Computer Programming*, vol. 180, pp. 18–35, 2019.

[42] A. Mizera, J. Pang, and Q. Yuan, "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*, ser. LNCS, vol. 9364. Springer, 2015, pp. 214–220.

[43] A. Mizera, J. Pang, H. Qu, and Q. Yuan, "ASSA-PBN 3.0: Analysing context-sensitive probabilistic Boolean networks," in *Proc. 16th International Conference on Computational Methods in Systems Biology*, ser. LNCS, vol. 11095. Springer, 2018, pp. 313–317.

[44] J. Krumsiek, C. Marr, T. Schroeder, and F. J. Theis, "Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network," *PLOS ONE*, vol. 6, no. 8, p. e22649, 2011.

[45] F. Herrmann, A. Groß, D. Zhou, H. A. Kestler, and M. Kühl, "A Boolean model of the cardiac gene regulatory network determining first and second heart field identity," *PLOS ONE*, vol. 7, no. 10, p. e46798, 2012.

[46] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PLOS ONE*, vol. 3, no. 2, p. e1672, 2008.

[47] Ö. Sahin, H. Fröhlich, C. Löbke, U. Korf, S. Burmester, M. Majety, J. Mattern, I. Schupp, C. Chaouiya, D. Thieffry, A. Poustka, S. Wiemann, T. Beissbarth, and D. Arlt, "Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance," *BMC Systems Biology*, vol. 3, no. 1, p. 1, 2009.

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

[49] D. P. Cohen, L. Martignetti, S. Robine, E. Barillot, A. Zinovyev, and L. Calzone, "Mathematical modelling of molecular pathways enabling tumour cell invasion and migration," *PLOS Computational Biology*, vol. 11, no. 11, p. e1004571, 2015.

[50] S. Collombet, C. van Oevelen, J. L. S. Ortega, W. Abou-Jaoudé, B. Di Stefano, M. Thomas-Chollier, T. Graf, and D. Thieffry, "Logical modeling of lymphoid and myeloid cell specification and transdifferentiation," *Proceedings of the National Academy of Sciences*, vol. 114, no. 23, pp. 5792–5799, 2017.

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

[52] E. Remy, S. Rebouissou, C. Chaouiya, A. Zinovyev, F. Radvanyi, and L. Calzone, "A modeling approach to explain mutually exclusive and co-occurring genetic alterations in bladder tumorigenesis," *Cancer Research*, vol. 75, no. 19, pp. 4042–4052, 2015.

[53] A. Yachie-Kinoshita, K. Onishi, J. Ostblom, M. A. Langley, E. Posfai, J. Rossant, and P. W. Zandstra, "Modeling signaling-dependent pluripotency with boolean logic to predict cell fate transitions," *Molecular Systems Biology*, vol. 14, no. 1, p. e7952, 2018.

[54] J. Thakar, A. K. Pathak, L. Murphy, R. Albert, and I. M. Cattadori, "Network model of immune responses reveals key effectors to single and co-infection dynamics by a respiratory bacterium and a gastrointestinal helminth," *PLOS Computational Biology*, vol. 8, no. 1, p. e1002345, 2012.

[55] L. Grieco, L. Calzone, I. Bernard-Pierrot, F. Radvanyi, B. Kahn-Perles, and D. Thieffry, "Integrative modelling of the influence of MAPK network on cancer cell fate decision," *PLOS Computational Biology*, vol. 9, no. 10, p. e1003286, 2013.

[56] A. Abdi, M. B. Tahoori, and E. S. Emamian, "Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways," *Science Signaling*, vol. 1, no. 42, pp. ra10–ra10, 2008.

[57] A. Singh, J. M. Nascimento, S. Kowar, H. Busch, and M. Boerries, "Boolean approach to signalling pathway modelling in HGF-induced keratinocyte migration," *Bioinformatics*, vol. 28, no. 18, pp. 495–501, 2012.

[58] V. L. Chudasama, M. A. Ovacik, D. R. Abernethy, and D. E. Mager, "Logic-based and cellular pharmacodynamic modeling of bortezomib responses in U266 human myeloma cells," *Journal of Pharmacology and Experimental Therapeutics*, vol. 354, no. 3, pp. 448–458, 2015.

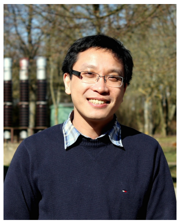
[59] A. Naldi, J. Carneiro, C. Chaouiya, and D. Thieffry, "Diversity and plasticity of th cell types predicted from regulatory network modelling," *PLOS Computational Biology*, vol. 6, no. 9, p. e1000912, 2010.

[60] O. J. Oyeyemi, O. Davies, D. L. Robertson, and J.-M. Schwartz, "A logical model of HIV-1 interactions with the T-cell activation signalling pathway," *Bioinformatics*, vol. 31, no. 7, pp. 1075–1083, 2014.

[61] B. D. Conroy, T. A. Herek, T. D. Shew, M. Latner, J. J. Larson, L. Allen, P. H. Davis, T. Helikar, and C. E. Cutucache, "Design, assessment, and in vivo evaluation of a computational model illustrating the role of CAV1 in CD4+ T-lymphocytes," *Frontiers in Immunology*, vol. 5, p. 599, 2014.

[62] S. Raza, K. A. Robertson, P. A. Lacaze, D. Page, A. J. Enright, P. Ghazal, and T. C. Freeman, "A logic-based diagram of signalling pathways central to macrophage activation," *BMC Systems Biology*, vol. 2, no. 1, p. 36, 2008.

Dr. Cui Su received her MSc degrees in Systems Engineering from Yanshan University in 2016 and her PhD in Computer Science from University of Luxembourg in 2020. Her research interests lie in network control and computational systems biology.



Dr. Jun Pang received his PhD in Computer Science from Vrije Universiteit Amsterdam in 2004. Currently, he is a senior researcher in the Department of Computer Science at the University of Luxembourg. His research interests include formal methods, security and privacy, machine learning, complex networks, and computational systems biology.