Target Control of Asynchronous Boolean Networks

Cui Su and 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.

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.

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 dynamics of the network; and Moradi et 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 \( C = (C_0, C_1) \), such that the instantaneous, temporary or permanent application of \( C \) – setting the value of a node, whose index is in \( C_0 \) (or \( C_1 \)) to 0 (or 1) – can drive the network from any initial state \( s \) in the state space \( S \) to the target attractor \( A_s \). 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 \( \zeta \) on the number of perturbations. By increasing \( \zeta \), we can obtain solutions with at most \( \zeta \) perturbations. It is worth noticing that more perturbations may cause a significant increase in experimental costs, thus the threshold \( \zeta \) 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 \( BN = (X, F) \) 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 \( BN = (X, F) \) 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 \( BN = (X, F) \) 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 \( x_{i_1}, \ldots, x_{i_k} \) 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 \( \{1, 2, \ldots, n\} \) 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 $(C_0, C_1)$, where $C_0, C_1 \subseteq \{1, 2, \ldots, n\}$ and $C_0$ and $C_1$ are mutually disjoint (possibly empty) sets of indices of nodes of a Boolean network $BN$. The size of the control $C$ is defined as $|C| = |C_0| + |C_1|$. 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 $i \in C_0$ and $s'[i] = 1$ for $i \in C_1$, and $s'[i] = s[i]$ otherwise. State $s'$ is called the intermediate state w.r.t. $C$.

Essentially, the application of a control $C$ inhibits the nodes in $C_0$, i.e., the values of these nodes are set (perturbed) to be 0, and overexpresses the nodes in $C_1$, 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 $C = (C_0, C_1)$, $C(S') = S''$, where $S'' = \{s'' \in S | s'' = C(s'), s' \in S\}$. Set $S''$ includes all the intermediate states with respect to $C$. Intuitively, sets $C_0$ and $C_1$ 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 $BN = (X, F)$ has the effect of reducing the state space of $BN$ to those which have the values of the variables in $C_0$ and $C_1$ 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 $C = (C_0, C_1)$ be a control and $BN = (X, F)$ be a Boolean network. The Boolean network $BN$ under control $C$, denoted $BN|_C$, is defined as a tuple $BN|_C = (X, F')$, where $X = \{x_1, x_2, \ldots, x_n\}$ and $F' = \{f_1, f_2, \ldots, f_n\}$, such that for all $i \in \{1, 2, \ldots, n\}$:

1. $x_i = 0$ if $i \in C_0$, $x_i = 1$ if $i \in C_1$, and $x_i = x_i$ otherwise;

2. $f_i = 0$ if $i \in C_0$, $f_i = 1$ if $i \in C_1$, and $f_i = f_i$ otherwise.

The state space of $BN|_C$, denoted $S|_C$, is derived by fixing the values of the variables in $C$ to their respective values and is defined as $S|_C = \{s | s[i] = 1$ if $i \in C_1$ and $s[j] = 0$ if $j \in C_0\}$. It is obvious that $S|_C \subseteq S$. For any subset $S'$ of $S$, we let $S'|_C = S' \cap S|_C$.

Example 1. Consider a Boolean network $BN = (X, F)$, 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 $C = (C_0, C_1)$, where $C_0 = \{2\}$, $C_1 = \emptyset$ (i.e., $\{x_3 = 0\}$), $C_0$ and $C_1$ 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 $BN|_C$. We use the generic notion $BN = (X, F)$ to represent either $BN = (X, F)$ or $BN|_C = (X, F')$. We assume that a Boolean network $BN = (X, F)$ 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 $\xi_{BN} : N \rightarrow \psi(S)$ such that $\xi_{BN}(0) = \{s_0\}$ and for every $j \geq 0$, if $s \in \xi_{BN}(j)$ then $s' \in \xi_{BN}(j + 1)$ 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 $(S, R_{BN})$, 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 $\xi$ of $BN$ (i.e. Definition 4).

Similarly, we denote the transition system of a Boolean network under control, $BN|_C$, as $TS|_C$.

Example 2. The transition system $TS$ of $BN$, given in Example 1, is shown in Fig. 1a. 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 $C = \{x_2 = 0\}$ given in Example 1, reshapes the transition system $TS$ of $BN$ in Fig. 1a to the transition system under control $TS|_C$ of $BN|_C$ in Fig. 1b. 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 $TS|_C$ might differ. For this example, only attractor $A_1$ is preserved in $TS|_C$.

2.3 Attractors and Basins

A path $\rho$ 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 $\rho$ 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 $\rho$ if for any $i \geq 0$, there
exists $j \geq i$ such that $s_j = s'$. We assume every infinite path $\rho$ is fair — for any state $s'$ that appears infinitely often in $\rho$, every possible next state $s''$ of $s'$ also appears infinitely often in $\rho$. 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$(s) = A$.

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.

Figs. 2a, 2b and 2c show a singleton attractor, a simple loop, and a complex loop, respectively. For an attractor $A$ of $BN$, we define its weak basin as $bas_{WS}(A) = \{s \in S | reach(s) \cap A \neq \emptyset\}$; the strong basin of $A$ is defined as $bas_{WS}(A) = \{s \in S | reach(s) \cap A \neq \emptyset \}$ and $reach(s) \cap A' = \emptyset$ for any other $A'$ of $BN$, $A' \neq A$. Intuitively, the weak basin of $A$, $bas_{WS}(A)$, 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 $A$, $bas_{WS}(A)$, 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. 1a. For attractor $A_1$, its strong basin, $bas_{WS}(A_1)$, contains two states $\{000, 001\}$; its weak basin, $bas_{WS}(A_1)$, 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 $C_0$ and overexpresses the nodes in $C_1$.

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 $C = (C_0, C_1)$ 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 $C = (C_0, C_1)$ 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 $C = (C_0, C_1)$ 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 $TS|C$).

We define the concept of schema, which is crucial for the development of the target control methods. Given a control $C = (C_0, C_1)$, the possible intermediate states with respect to $C$, denoted $S' = C(S)$, form a schema, defined as follows.

**Definition 8 (Schema).** A subset $S'$ of $S$ is a schema if there exists a triple $M = (C_0, C_1, D)$, where $C_0 \cup C_1 \cup D = \{1, 2, \ldots , n\}$, $C_0, C_1$ and $D$ are mutually disjoint (possibly empty) sets of indices of nodes of $BN$, such that $S\{C_0\} = \{0, 1\}^{|C_0|}$, $S\{C_1\} = \{0\}^{|C_1|}$ and $S\{D\} = \{0, 1\}^{|D|}$. $C_0, C_1$ and $D$ are called off-set, on-set and don’t-care-set of $S'$, respectively. The elements in $C_0 \cup C_1$ are called indices of support variables of $S'$.

Intuitively, for node $x_i, i \in C_0$, it has a value of 0 in any state $s \in S'$; for node $x_i, i \in C_1$, it has a value of 1 in any state $s \in S'$. The projection of $S'$ to the don’t-care-set $D$ contains all combinations of binary strings of $|D|$ bits. Thus, any schema $S'$ is of size $2^{|D|}$. Since the total number of nodes $n = |C_0| + |C_1| + |D|$ is fixed, a larger schema implies more elements in $D$ and fewer elements in $C_0 \cup C_1$.

**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. 1a, its strong basin, $bas_{WS}(A_1) = \{000, 001, 101, 100\}$, can be partitioned into two schemata $\{000, 001, 100, 101\}$ and $\{010, 011\}$, represented as ‘0∗’ and ‘1∗’, 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 (001), the network will spontaneously stabilise to $A_1$.

The control set $\{x_2 = 0\}$ is both an ITC and a PTC. As introduced in Example 2, this control reshapes the
transition system to Fig. 1b, which also means that the control drives the network from any initial state to a state in Fig. 1b. 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 \( s' = C(s) \) 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 \( s \), all possible intermediate states \( s' = C(S) \) 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 \( C = (C_0, C_1) \) is an instantaneous target control from any initial state \( s \in S \) to a target attractor \( A_t \) if \( C(S) \subseteq bas_{TS}^{S}(A_t) \).

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 \( bas_{TS}^{S}(A_t) \) but to any state \( s' \in (S \backslash bas_{TS}^{S}(A_t)) \), 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 \( C_t \) its intermediate states \( s' = C(S) \) must form a subset of \( bas_{TS}^{S}(A_t) \). For any possible intermediate state \( s' \in S' \), \( s'[i] = 0 \) for \( i \in C_0 \) and \( s'[i] = 1 \) for \( i \in C_1 \), which indicates that \( S'_t[c_0] = (0)_{c_0} \) and \( S'_t[c_1] = (1)_{c_1} \). Let \( D \) denote the indices of the nodes that are not in \( C_0 \) or \( C_1 \). Then, \( S'_t[D] = (0, 1)^{|D|} \) because the values of the nodes in the initial states \( S'_t[D] \) stay unchanged. In another word

**Observation 1.** If the initial state can be any state \( s \in S \), for any control \( C = (C_0, C_1) \), the set of intermediate states \( S' = C(S) \) forms a schema.

The notion of schema sheds light on the computation of ITC. Each schema \( W_t \) of the strong basin of the target attractor, \( bas_{TS}^{S}(A_t) \), returns a candidate target control \( C = (C_0, C_1) \), where \( C_0 \) and \( C_1 \) are the off-set and on-set of \( W_t \). The size of control \( |C| \) equals \((n - \log_2|W_t|)\), therefore, a larger schema results in a smaller control set. Thus, we can partition the strong basin of the target attractor, \( bas_{TS}^{S}(A_t) \), into a set of mutually disjoint schemata \( W = \{W_1, W_2, \ldots, W_m\} \), such that \( W_1 \cup W_2 \cup \ldots \cup W_m = bas_{TS}^{S}(A_t) \). Each \( W_t \in W \) is one of the largest schemata in \( bas_{TS}^{S}(A_t) \) \( \setminus (W_1 \cup \ldots \cup W_{t-1}) \) and the indices of its support variables in \( C_0 \) and \( C_1 \) form a candidate ITC \( C_t = (C_0, C_1) \). In \( C_t \), 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 \( \zeta \) on the number of perturbations, keep \( \zeta \) updated with the minimal size of the computed control sets, and only save the control sets with at most \( \zeta \) perturbations. Algorithm 1 realises the above idea in pseudocode.

**Algorithm 1.** Instantaneous Target Control

1: procedure \textsc{Instantaneous Target Control} (BN, \( A_t \))
2: initialise \( \mathcal{L} := \emptyset \) to store computed control sets
3: \( I, I^* \): \textsc{Comp_input_nodes}(\( G \)) \( \triangleright \) compute input nodes \( I \), specified input nodes \( I^* \) and non-specified input nodes \( I^{**} \)
4: \( \mathcal{S}B := \text{Comp_Strong_Basin}(F, A_t) \)
5: \( \mathcal{W} := \text{Comp_schema}(\mathcal{S}B), m = |\mathcal{W}| \)
6: \( \zeta := n \triangleright \) a threshold on the number of perturbations
7: for \( i = 1 : m \) do \( \triangleright \) traverse the set of schemata
8: \( C_i := \text{Comp_support_variables}(W_i) \triangleright C_i := (C_{0i}, C_{1i}) \)
9: \( C_i := (C_{0i} \setminus I^* \cup I_{\text{INSTANTANEOUS}}) \triangleright \) remove specified input nodes
10: if \( |C_i| \leq \zeta \) then
11: \( \text{save } C_i \text{ to } \mathcal{L} \)
12: \( \zeta := \min(|C_i|, \zeta) \)
13: return \( \mathcal{L} \)

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 \textsc{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, also called a minterm. To compute the largest schema \( S_t \) 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 \textsc{Comp_Schema}.

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 \( C = (C_0, C_1) \) is a temporary target control to a target attractor \( A_t \) from any initial state \( s \in S \) if \( bas_{TS}^{S}(A_t) \cap S_c \neq \emptyset \) and \( C(S) \subseteq bas_{TS}^{S}(bas_{TS}^{S}(A_t) \cap S_c) \).

Below, we give an intuitive explanation of Corollary 2. We know that the application of a control \( C \) results in a new Boolean network \( BN_{C} \) and the 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 \( A_t \), w.r.t. the original transition system \( TS \), i.e., \( bas_{TS}^{S}(A_t) \), from which there only exist paths to \( A_t \). This requires the remaining strong basin in \( S_c \), i.e., \( (bas_{TS}^{S}(A_t) \cap S_c) \), is a non-empty set; otherwise, it is not
guaranteed to reach \( A_i \). Furthermore, the condition \( C(S) \subseteq bas^S_{TS}(A_i) \cap S(C) \) ensures that any possible intermediate state \( s' \in C(S) \) is in the strong basin of the remaining strong basin \( bas^S_{TS}(A_i) \cap S(C) \) 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 \( A_i \). Based on the definition of the weak basin, it is sufficient to search the weak basin \( bas^W_{TS}(A_i) \) for TTC.

Algorithm 2. Temporary Target Control

1: procedure TEMPORARY_TARGET_CONTROL (BN, \( A_i \))
2: initialise \( \mathcal{L} := \emptyset \) and \( \Omega := \emptyset \) to store valid and checked control sets, resp.
3: \( I, I^* := \text{COMP\_INPUT\_NODES}(G) \) \( \triangleright \) compute input nodes \( I \) and non-specified input nodes \( I^* \).
4: \( \text{SB} := \text{COMP\_STRONG\_BASIN}(F, A_i) \)
5: \( \text{WB} := \text{COMP\_WEAK\_BASIN}(F, A_i) \)
6: \( \mathcal{W} := \text{COMP\_SCHEMATAS}([W]), m := |\mathcal{W}| \)
7: generate a vector \( \Theta \) of length \( m \) and set all the elements to false
8: \( \xi := n \)
9: for \( i = 1 : m \) do
10: if \( \Theta[i] = true \), then continue
11: \( C_i := \text{COMP\_SUPPORT\_VARIABLES}(W_i) \)
12: \( C_i' := (C_i \cap I^*) \cap C_i \cap \text{strong basin of } A_i \)
13: \( k := 0, \text{isValid} := false \)
14: while \( \text{isValid} = false \) and \( k \leq \min(\xi - |C_i'|, |C'_i|) \) do
15: \( C_{iub} := \text{COMP\_SUBSETS}(C_i', k) \)
16: for \( C_{iub} \in C_{iub} \) do
17: \( C_i := C_{iub} \cup C_i', \Phi := C_i(S) \)
18: if \( C_i' \in \Omega \), then
19: \( \text{isValid} := \text{VERIFY\_TTC}(F, C_i, \text{SB}, \Phi) \)
20: add \( C_i' \) to \( \Omega \).
21: if \( \text{isValid} = true \), then
22: add \( C_i' \) to \( \mathcal{L} \), \( \xi := \min(\xi, |C'_i|) \)
23: \( \Theta[z] := true \) if \( W_i \subseteq \Phi \) for \( z \in [i + 1, m] \)
24: if \( \text{isValid} = false \), then \( k := k + 1 \)
25: return \( \mathcal{L} \)

Algorithm 3. Verification of Temporary Target Control

1: procedure VERIFY\_TTC (F, C, SB, \( \Phi \))
2: \( \text{isValid} := false \)
3: if \( \Phi \not\subseteq \text{SB} \) then
4: \( \text{isValid} := true \)
5: else
6: \( \text{SB}_C := \text{COMP\_STATE\_CONTROL}(C, \text{SB}) \triangleright \) compute the remaining strong basin w.r.t. \( C \) in \( TS|_C \)
7: \( F|_C := \text{COMP\_FN\_CONTROL}(C, F) \)
8: \( bas^S_{TS}(\text{SB}_C) := \text{COMP\_STRONG\_BASIN}(F|_C, \text{SB}_C) \)
9: \( \text{if } \Phi \subseteq bas^S_{TS}(\text{SB}_C) \) then
10: \( \text{isValid} := true \)
11: return \( \text{isValid} \)

A noteworthy point is that temporary control needs to be released once the network reaches a state in \( (bas^S_{TS}(A_i) \cap S(C)) \). On the other hand, although Corollary 2 guarantees that partial of the strong basin of \( A_i \) in \( TS \) is preserved in \( TS|_C \), it does not guarantee the presence of \( A_i \) 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 \( A_i \). 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 \( bas^W_{TS}(A_i) \) gives a candidate TTC, \( C_i := (C_0, C_i) \), for further optimisation and validation. A larger schema results in a smaller control set. To explore the entire weak basin \( bas^W_{TS}(A_i) \), we partition it into a set of mutually disjoint schemata \( \mathcal{W} = \{W_1, W_2, \ldots, W_m\} \). Each \( W_i, i \in m \) is one of the largest schemata in \( bas^W_{TS}(A_i) \) \( \setminus (W_1 \cup W_2 \cup \ldots \cup W_{i-1}) \). For \( W_i \), the indices of its support variables in \( C_0 \) and \( C_i \) form a candidate control \( C_i := (C_0, C_i) \). 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 \( I^* \) 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 \( C'_i \), we verify its subsets of size \( k \) based on Corollary 2 from \( k = 0 \) with an increment of \( I \), until we find a valid solution.

Algorithm 2 implements the above idea in pseudocode. It takes as inputs the Boolean network \( BN = (X, F) \) and the target attractor \( A_i \). It first initialises two vectors \( \mathcal{L} \) and \( \Omega \) to store valid controls and the checked controls, respectively. (We use \( \Omega \) to avoid duplicate control validations.) Then, it computes input nodes \( I \) and the non-specified input nodes \( I^{ns} \). Then, \( I^{ns} \subseteq I \) (line 3). The weak basin \( WB \) and the strong basin \( SB \) of \( A_i \) of \( TS \) are computed using procedures \( \text{COMP\_WEAK\_BASIN} \) and \( \text{COMP\_STRONG\_BASIN} \) developed in our
previous work [26] (lines 4-5). The weak basin \( WB \) is then partitioned into \( m \) mutually disjoint schemata with procedure \texttt{COMP_SCHEMA}. 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 \texttt{COMP_SUPPORT_VARIABLES} form a candidate control \( C_i \) (line 11). The essential control nodes \( C'_i \) of \( C_i \) consist of the non-specified input nodes and the non-input nodes in \( C_i \) constitute a set \( C'_i \) for further optimisation (line 12). We search the subsets of \( C'_i \) starting from size \( k = 0 \) with an increment of 1 and verify whether the union of a subset \( C'_i \) and the essential nodes \( C'_j \), namely \( C'_i = C'_i \cup C'_j \), is a valid temporary target control using procedure \texttt{VERIFY_TTC} in Algorithm 3. If \( C'_i \) is valid, save it to \( L \). 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 \( C_i \), we need to reconstruct the associated transition relations \( F|_C \) and compute the strong basin of the remaining strong basin in \( TSI|_C \), i.e., \( \text{bas}_{TSI|_C}(SB|_C) \) (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 \( W_i \) (lines 10 and 23 of Algorithm 2) if it is a subset of intermediate states \( \Phi \) of a pre-validated control \( C_i \) (line 23 of Algorithm 2); and
2. set a threshold \( \zeta \) on the number of perturbations, keep \( \zeta \) updated with the smallest size of valid TTCs \( C'_i \) (line 22 of Algorithm 2) and only compute control sets with at most \( \zeta \) 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 \( C = (C_0, C_1) \) is a permanent target control from any initial state \( s \in S \) to a target attractor \( A_t \) if \( A_t \) is an attractor of \( TSI|_C \) and \( C(S) \subseteq \text{bas}_{TSI|_C}(A_t) \).

**Algorithm 4.** Verification of Permanent Target Control

\begin{verbatim}
1: procedure Verify_PTC (F, C, A, \Phi)  
2: isValid := false  
3: if \( \Phi|_C = A_t|_C \) then  
4: \( F|_C := \text{COMP_FN_CONTROL}(C, F) \)  
5: \( \text{bas}_{TSI|_C}(A_t) := \text{COMP_STRONG_BASIN}(F|_C, A_t) \)  
6: if \( C \subseteq \text{bas}_{TSI|_C}(A_t) \) then  
7: isValid = true  
8: return isValid
\end{verbatim}

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, \( TSI|_C \).

The algorithm for PTC can be derived from Algorithm 2 by replacing procedure \texttt{VERIFY_TTC} with procedure \texttt{VERIFY_PTC} in Algorithm 4. To avoid duplication, we only explain procedure \texttt{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 \( F|_C \) and compute the strong basin of \( A_t \) in \( TSI|_C \) (lines 4-5). \( C \) is a PTC if the set of intermediate states is a subset of \( \text{bas}_{TSI|_C}(A_t) \) (lines 6-7).

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 \( A_t \) of \( BN, \text{bas}_{TSI}(A_t) \), can be represented as a schema ‘00+’. Based on Algorithm 1, we know that its support variables form an ITC \( \{x_1 = 0, x_2 = 0\} \).

The weak basin of \( A_t, \text{bas}_{TSI}(A_t) \), can be divided into two schemata, represented as ‘01+w’ and ‘01+s’. ‘01+w’ contains more states than ‘01+s’, which implies that ‘01+w’ potentially give a smaller TTC or PTC. Algorithms for TTC and PTC verify subsets of the control derived from ‘01+w’ and ‘01+s’. Based on Corollary 2 and Corollary 3, \( \{x_2 = 0\} \) is both a TTC and a PTC for \( A_t \). By fixing \( x_2 = 0 \), the transition system changes from Figs. 1a to 1b. The network is driven to a state in \( TSI|_C \) (see Fig. 1b) and will eventually stable in \( A_t \).

### 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 CAEBAN [39], which contains a realisation of the decomposition-based detection method [40], [41] for identifying attractors in Boolean networks efficiently. 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 to granulocytes, monocytes, megakaryocytes and erythrocytes [44]. This network has 11 nodes and 6 attractors, 4 of 1. These attractor detection methods were originally implemented in the software tool ASSA-PBN [35], [42], [43].
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.

Tables 2 and 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 provide 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 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].

## Table 4

|       | SHF                                                                 |                                                                 | FHF                                                                 |
|-------|----------------------------------------------------------------------|-----------------------------------------------------------------|------------------------------------------------------------------|
| ITC   | [exogenCanWntI=1]                                                   | [canWnt=0 Foxc12=0 Tbx1=0 GATAs=0 & Tbx5=1                      |
|       |                                                                    | exogenCanWntI=0 exogenCanWntII=0]                               | [exogenCanWntI=0]                                                 |
| TTC   | [exogenCanWntI=1]                                                   | {GATAs=1 & exogenCanWntII=0}                                     | {Tbx5=1 & exogenCanWntII=0}                                       |
|       |                                                                    | {Nxk25=1 & exogenCanWntII=0}                                     | {Mesp1=1 & exogenCanWntII=0}                                      |
|       |                                                                    | {Tbx1=1 & exogenCanWntII=0}                                       | {Foxc12=1 & exogenCanWntII=0}                                     |
| PTC   | {exogenCanWntI=1}                                                   | {GATAs=1 & exogenCanWntII=0}                                     | {Tbx5=1 & exogenCanWntII=0}                                       |
|       |                                                                    | {Nxk25=1 & exogenCanWntII=0}                                     | {Mesp1=1 & exogenCanWntII=0}                                      |
|       |                                                                    | {Tbx1=1 & exogenCanWntII=0}                                       | {Foxc12=1 & exogenCanWntII=0}                                     |
| SMC   | {exogenCanWntI=1}                                                   | {GATAs=1 & exogenCanWntII=0}                                     | {Tbx5=1 & exogenCanWntII=0}                                       |
|       |                                                                    | {Nxk25=1 & exogenCanWntII=0}                                     | {Mesp1=1 & exogenCanWntII=0}                                      |
|       |                                                                    | {Tbx1=1 & exogenCanWntII=0}                                       | {Foxc12=1 & exogenCanWntII=0}                                     |

## Table 5

| 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                   |
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 adenine 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 between 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, PTC and SMC 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 \ (TTC ∩ SMC), SMC \ (TTC ∩ SMC)), respectively. Green bars represent the intersections of the two methods. Equations above the bars (|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 ∩ SMC), or require more perturbations

---

**TABLE 6**

| 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       |
| hematopoeisis | 5     | 3       | 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       |

Symbol '*' means that the method failed to finish the computation within twelve hours.

---

**TABLE 7**

| Network     | ITC (seconds) | TTC (seconds) | PTC (seconds) | SMC (seconds) |
|-------------|---------------|---------------|---------------|---------------|
| 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        |
| hematopoeisis | 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     | *             |
| pathway     | 445.251       | 4435.59       | 10038.600     | 42.180        |

Symbol '*' means that the method failed to finish the computation within twelve hours.
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 is 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 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 Fig. 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.

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,” J. Biol. Eng., vol. 13, no. 1, 2019, Art. no. 14.
[3] M. S. Goligorsky, “New trends in regenerative medicine: reprogramming and reconditioning,” J. Amer. Soc. Nephrol., vol. 30, no. 11, pp. 2047–2051, 2019.
[4] L.-Z. Wang et al., “A geometrical approach to control and controllability of nonlinear dynamical networks,” Nat. Commun., 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. Singapore: Singapore: World Scientific, 2018.
[7] S. Huang, “Genomics, complexity and drug discovery: Insights from Boolean network models of cellular regulation,” Pharmacogeonomics, 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,” Nat. Commun., vol. 5, 2014, Art. no. 5415.
[10] E. Czeizler, C. Gratie, W. K. Chiu, K. Kanhaiya, and I. Petre, “Target controllability of linear networks,” in Proc. 14th Int. Conf. Comput. Methods Syst. Biol., 2016, vol. 9859, pp. 67–81.
[11] A. J. Gates and L. M. Rocha, “Control of complex networks requires both structure and dynamics,” Sci. Rep., vol. 6, no. 1, pp. 1–11, 2016.
[12] M. Moradi, S. Goliaei, and M.-H. Foroughmand-Araabi, “A Boolean networks framework for the control of complex networks,” in Proc. 23rd Int. Symp. Fuzzy Systems, 2019, vol. 11800, pp. 707–724.
[13] A. Mandon, C. Su, S. Paul, S. Haar, and L. Paulèvè, “Algorithms for the sequential reprogramming of Boolean networks,” IEEE/ACM Trans. Computat. Biol. Bioinformatics., vol. 16, no. 5, pp. 1610–1619, Sep./Oct. 2019.
[14] H. Mandon, C. Su, S. Haar, J. Pang, and L. Paulèvè, “Sequential reprogramming of Boolean networks made practical,” in Proc. 17th Int. Conf. Computat. Methods Syst. Biol., 2019, pp. 3–19.
[15] M. Moradi, S. Paul, J. Su, and A. Mizera, “An efficient approach towards the target control of Boolean networks,” IEEE/ACM Trans. Computat. Biol. Bioinformatics., vol. 17, no. 6, pp. 1932–1945, Nov./Dec. 2020.
[16] 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.
[17] J. Yue, Y. Yan, and R. Albert, “Cell fate reprogramming by control of intracellular network dynamics,” PLoS Computat. Biol., vol. 11, no. 4, 2015, Art. no. e1004193.
[18] F.-J. Müller and A. Schuppert, “Few inputs can reprogram biological networks,” Nature, vol. 478, pp. 7369, 2011, Art. no. E4.
[19] A. Mizera, J. Pang, C. Su, and Q. Yuan, “ASSA-PBN: A toolbox for probabilistic Boolean networks,” IEEE/ACM Trans. Computat. Biol. Bioinformatics., vol. 15, no. 4, pp. 1203–1216, Jul./Aug. 2018.
[20] P. Zhu and J. Han, “Asynchronous stochastic Boolean networks as gene network models,” J. Computat. Biol., vol. 21, no. 10, pp. 771–783, 2014.
[21] C. Su and J. Pang, “Sequential and permanent control of Boolean networks,” in Proc. 19th Int. Conf. Computat. Methods Syst. Biol., 2020, pp. 234–251.
[22] F. Somensi, “CUDD: CU decision diagram package - release 2.5.1,” 2015. [Online]. Available: http://vlsi.colorado.edu/~fabio/CUDD/2015.
[23] C. Su and J. Pang, “CABEAN: A software for the control of asynchronous Boolean networks,” Bioinformatics, vol. 37, no. 6, pp. 779–811, 2021.
[24] A. Mizera, J. Pang, H. Qu, and Q. Yuan, “Taming attractors for asynchronous constraint detection in large Boolean networks,” IEEE/ACM Trans. Computat. Biol. Bioinformatics., vol. 16, no. 1, pp. 31–42, Jan./Feb. 2019.
[25] Q. Yuan, A. Mizera, J. Pang, and H. Qu, “A new decomposition-based method for detecting attractors in synchronous Boolean networks,” Sci. Computat. Programm., vol. 188, pp. 18–35, 2019.
SU AND PANG: TARGET CONTROL OF ASYNCHRONOUS BOOLEAN NETWORKS

A. Mizera, J. Pang, and Q. Yuan, “ASSA-PBN: A tool for approximate steady-state analysis of large probabilistic Boolean networks,” in Proc. 13th Int. Symp. Automat. Technol. Verification Anal., 2015, vol. 9364, pp. 214–220.

A. Mizera, J. Pang, H. Qu, and Q. Yuan, “ASSA-PBN 3.0: Analysing context-sensitive probabilistic Boolean networks,” in Proc. 16th Int. Conf. Comput. Methods Syst. Biol., 2018, vol. 11095, pp. 313–317.

J. Krumsieck, 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, 2011, Art. no. e22649.

F. Herrmann, A. Groß, D. Zhou, H. A. Kestler, and M. Kühle, “A Boolean model of the cardiac gene regulatory network determining first and second heart field identity,” PLoS One, vol. 7, no. 10, 2012, Art. no. e1672.

M. I. Davidich and S. Bornholdt, “Boolean network model predicts cell cycle sequence of fission yeast,” PLoS One, vol. 3, no. 2, 2008, Art. no. e1672.

O. Sahin et al., “Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance,” BMC Syst. Biol., vol. 3, no. 1, 2009, Art. no. 1.

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,” Front. Physiol., vol. 7, 2016, Art. no. 349.

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 Comput. Biol., vol. 11, no. 11, 2015, Art. no. e1004871.

S. Collombet et al., “Logical modeling of lymphoid and myeloid cell specification and transdifferentiation,” Proc. Nat. Acad. Sci. USA, vol. 114, no. 23, pp. 5792–5799, 2017.

B. Offermann et al., “Boolean modeling reveals the necessity of transcriptional regulation for bistability in PC12 cell differentiation,” Front. Genet., vol. 7, 2016, Art. no. 44.

E. Remy, S. Rebourcassou, 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 Res., vol. 75, no. 19, pp. 4042–4052, 2015.

A. Yachie-Kinoshita et al., “Modeling signaling-dependent pluripotency with boolean logic to predict cell fate transitions,” Mol. Syst. Biol., vol. 14, no. 1, 2018, Art. no. e7952.

J. Thakar, A. K. Pathak, L. Murphy, R. Albert, and I. M. Cattadori, “Network model of immune response reveals key effectors to single and co-infection dynamics by a respiratory bacterium and a gastrointestinal helminth,” PLoS Comput. Biol., vol. 8, no. 1, 2012, Art. no. e1002345.

L. Greico, 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 Comput. Biol., vol. 9, no. 10, 2013, Art. no. e1003286.

A. Abdi, M. B. Tahoori, and E. S. Emamian, “Fault diagnosis engineering of digital circuits can identify vulnerable molecules in complex cellular pathways,” Sci. Signaling, vol. 1, no. 42, pp. ra10–ra10, 2008.

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.

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,” J. Pharmacol. Experimental Ther., vol. 354, no. 3, pp. 448–458, 2015.

A. Naldi, J. Carneiro, C. Chaouiya, and D. Thieffry, “Diversity and plasticity of TH cell types predicted from regulatory network modelling,” PLoS Comput. Biol., vol. 6, no. 9, 2010, Art. no. e1000912.

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.

B. D. Conroy et al., “Design, assessment, and in vivo evaluation of a computational model illustrating the role of CAV1 in CD4+ T-lymphocytes,” Front. Immunol., vol. 5, 2014, Art. no. 599.

S. Raza et al., “A logic-based diagram of signalling pathways central to macrophage activation,” BMC Syst. Biol., vol. 2, no. 1, 2008, Art. no. 36.

Cui Su received the MSc degree in systems engineering from Yanshan University in 2016 and the PhD degree in computer science from the University of Luxembourg in 2020. Her research interests include network control and computational systems biology.

Jun Pang received the PhD degree in computer science from Vrije Universiteit Amsterdam in 2004. He is currently a senior researcher with the Department of Computer Science, University of Luxembourg. His research interests include formal methods, security and privacy, machine learning, complex networks, and computational systems biology.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.