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

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

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

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

KEYWORDS
Boolean networks, attractors, network control

ACM Reference Format:

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

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

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

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

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

In this paper, we develop a target control method with temporary perturbations for asynchronous BNs. Our idea is to find a control \( C = \{0, 1\}\), which is a tuple of two sets, such that the application of \( C \) – setting the value of a node, whose index is in \( 0 \) (or \( 1 \)) to 0 (or 1) – can drive the network from any initial state to the weak basin of the target attractor. We hold the control for sufficient time and let the network evolve to a state in the strong basin of the target attractor. After that, the control can be released and the network will eventually and surely reach the target attractor. Since the network can take any state \( s \in S \) as an initial state, the possible intermediate states form a subset \( S' \) of \( S \), called schema. According to our previous work [39], we know that all the intermediate states should fall into the weak basin of the target attractor. Therefore, we partition the weak basin into a set of mutually disjoint schemata.
Definition 2.3 (Boolean networks under control). The Boolean network $G$ under control $C$, denoted as $G|C$, is defined as a tuple $G|C = (\hat{X}, \hat{F})$, where $\hat{X} = \{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n\}$ 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 $i \not\in C$, $\hat{x}_i = 1$ if $i \in C$, and $\hat{x}_i = x_i$ otherwise;
2. $\hat{f}_i = 0$ if $i \not\in C$, $\hat{f}_i = 1$ if $i \in C$, and $\hat{f}_i = f_i$ otherwise.

The state space of $G|C$, denoted $S|C$, is derived by fixing the values of the variables in $C$ to their respective values and is defined as $S|C = \{s \in S | s|f = 1$ if $i \in C$ and $s|f = 0$ if $i \not\in C\}$. Note that $S|C \subseteq S$. For any subset $S'$ of $S$, we let $S'|C = S' \cap S|C$.

2.2 Dynamics of Boolean networks

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

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

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 $\rho = s_1 \rightarrow s_2 \rightarrow \cdots \rightarrow s_n$. A path from a state $s$ to a subset $S'$ of $S$ is a path from $s$ to any state $s' \in S'$. For a state $s \in S$, $\text{reach}(s)$ denotes the set of states $s$ such that there is a path from $s$ to $s'$ in $TS$.

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

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 $\text{reach}(s) \cap A \neq \emptyset$. The network starting at any initial state $s_0 \in S$ will eventually end up in one of the attractors of $TS$ and remain there forever unless perturbed. Under asynchronous updating scheme, there are singleton attractors and cyclic attractors. Cyclic attractors can be further classified into: (1) a simple loop, in which all the states form a loop and every state appears only once per traversal through the loop; and (2) a complex loop, which has intricate topology and includes several loops. Figure 1 (a), (b) and (c) show a singleton attractor, a simple loop and a complex loop, respectively.

We let $\mathcal{A}$ denote all the attractors of $TS$. For an attractor $A$, $A \in \mathcal{A}$, we define its strong basin as $\text{bas}_{TS}^W(A) = \{s \in S | \text{reach}(s) \cap A \neq \emptyset\}$, the strong basin of $A$ is defined as $\text{bas}_{TS}^S(A) = \{s \in S | \text{reach}(s) \cap A \neq \emptyset$ and $\text{reach}(s) \cap A' = \emptyset$ for any $A' \in \mathcal{A}, A' \neq A\}$. Intuitively, the weak basin of $A$, $\text{bas}_{TS}^W(A)$, contains all the states from which there exists at least one path to $A$, and there may also exist paths from $s$ to other attractors $A'$ ($A' \neq A$) of $TS$. The strong basin of $A$, $\text{bas}_{TS}^S(A)$, consists of all the states from which there only exist paths to $A$.

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

2.3 The control problem

As described in the introduction, many devastating diseases, such as Parkinson’s disease and Alzheimer’s disease, are caused by diseased cells or a deficiency of particular cells. Cell reprogramming can transform abounding somatic cells into the desired cell type. In the context of BNs, this process is, indeed, stirring the dynamics of the network from a source attractor to a desired target attractor. However, cells in culture and in situ are usually not isolated but exist in a population consisting of various cell phenotypes or even transient cell states. Hence, it is important to develop a target control method to identify key nodes that can guide the network towards a desired target attractor from any other distinct steady states or transient states.
This can be defined as a target control problem: given a BN \( G \) and a target attractor \( A_t \), finding a control \( C \), the application of which can drive the network from any source state \( s \in S \) to \( A_t \). When the source state \( s \) is fixed, finding a control \( C \) to drive the network from \( s \) to \( A_t \) is a source-target control problem. Based on the application time of control, we have: (1) temporary control - perturbations are applied for a finite (possibly zero) number of steps and then released; and (2) permanent control - perturbations are applied for all the following steps. When perturbations are applied instantaneously, we call it instantaneous control, which is a special case of temporary control. Temporary control has shown its apparent advantages in reducing the number of perturbations \([39]\), thus in this work, we focus on temporary target control, formally defined as follows.

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

3 RESULTS

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

Lemma 3.1. A control \( C = (0, I) \) is a temporary target control to a target attractor \( A_t \) from any source state \( s \in S \) if \( \text{bas}^S_{TS}(A_t) \cap S|_C \neq \emptyset \) and \( C(S) \subseteq \text{bas}^S_{TS}(A_t) \cap S|_C \).

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

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

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

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

Given a control \( C = (0, I) \), the possible intermediate states with respect to \( C \), denoted \( S' \), form a schema defined as follows.

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

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

Example 3.4. To continue with Example 2.5, let us consider the set \( W_1 = \{000,001,010,011\} \), which is a subset of the weak basin of \( A_1 \). There exists a triple \( M_1 = (0,1,1) \), where \( 0 = (1) \), \( 1 = (1) \), \( 1 = \emptyset \) and \( D = \{2,3\} \), such that \( W_1|_0 = \emptyset \), \( W_1|_I = \emptyset \), \( W_1|_{D} = \{00,01,10,11\} \). Therefore, \( W_1 \) is a schema. Let us denote the value of \( x_i \) in \( 0,1 \) and \( D \), as 0, 1 and * respectively. Then, \( W_1 \) can be represented as 0 **.*.

The notion of schema leads the way to find temporary target control. Each schema \( W_1 \) of the weak basin \( \text{bas}^W_{TS}(A) \) gives a candidate temporary target control \( C_t = (0, I) \) for further optimisation and validation. A larger schema results in a smaller control set. To explore the entire weak basin \( \text{bas}^W_{TS}(A) \), we partition it into a set of mutually disjoint schemata \( W = \{W_1, W_2, \ldots, W_m\} \), where \( W_1 \cup W_2 \cup \ldots \cup W_m = \text{bas}^W_{TS}(A) \). Each \( W_i \in W \) is one of the largest schemata in \( \text{bas}^W_{TS}(A) \) \( \setminus (W_1 \cup \ldots \cup W_{i-1}) \). For \( W_i \), the indices of its support variables in \( 0 \subseteq I \subseteq \emptyset \) form a candidate control \( C_t = (0, I) \). Each candidate control \( C_t \) 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 \( I^m \) are essential for control. For the remaining non-input nodes in \( C_t \), denoted \( C'_t \), we verify its subsets of size \( k \) based on Lemma 3.1 from \( k = 0 \) with an increment of 1, until we find a valid solution.
Algorithm 1 Temporary Target Control

1: procedure Temp_Target_Control(G, A_t)
2: initialise L := ∅ and Ω := ∅ to store valid temporary controls
3: and the chosen ordering of the variables. In BDD, a schema is
4: represented as a cube and each state is the smallest cube, also
called a minterm. To compute the largest schema \( S_0 \) of \( S \) is equivalent
to compute the largest cube of \( S \). The partitioning of the weak basin
into schemata is then transformed into a cube cover problem in BDD.
A different variable ordering may lead to a different partitioning.
Given a fixed ordering, the partitioning remains the same. Even
though finding the best variable ordering is NP-hard, there exist
enough heuristics to find the optimal ordering. In this work, we
compute a partitioning under one variable ordering as provided
by the CUDD package [37] and compute the smallest subsets of
candidate controls that are valid temporary target control sets.

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

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

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

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

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

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

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

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

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

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

**Efficiency.** The last two columns of Table 1 summarise the execution time for computing temporary target control for all the attractors of the networks. We can see that our method is more
Table 1: An overview of the networks and a comparison of the two methods (TTC and SMC). Symbol \(-\) means that the method failed to finish the computation within five hours.

<table>
<thead>
<tr>
<th>Network</th>
<th>#nodes</th>
<th>#edges</th>
<th>Number of attractors</th>
<th>Time (seconds)</th>
<th>Number of control sets</th>
<th>Attractor detection</th>
<th>Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TTC</td>
<td>TTC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>#singleton</td>
<td>#cyclic</td>
<td>#singleton</td>
<td>#quasi</td>
</tr>
<tr>
<td>myeloid</td>
<td>11</td>
<td>30</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.002</td>
</tr>
<tr>
<td>apoptosis</td>
<td>12</td>
<td>26</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>0.004</td>
</tr>
<tr>
<td>cardiac</td>
<td>15</td>
<td>39</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.004</td>
</tr>
<tr>
<td>ERBB</td>
<td>20</td>
<td>52</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.004</td>
</tr>
<tr>
<td>HSPC-MSC</td>
<td>26</td>
<td>81</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0.101</td>
</tr>
<tr>
<td>PC12</td>
<td>33</td>
<td>62</td>
<td>7</td>
<td>0</td>
<td>7</td>
<td>0</td>
<td>0.013</td>
</tr>
<tr>
<td>hematopoiesi</td>
<td>33</td>
<td>88</td>
<td>5</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0.452</td>
</tr>
<tr>
<td>bladder</td>
<td>35</td>
<td>116</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0.735</td>
</tr>
<tr>
<td>MAPK</td>
<td>53</td>
<td>105</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>1.749</td>
</tr>
<tr>
<td>HGF</td>
<td>66</td>
<td>103</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>2.443</td>
</tr>
<tr>
<td>T-diff</td>
<td>68</td>
<td>175</td>
<td>6</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>1.245</td>
</tr>
<tr>
<td>HIV-1</td>
<td>136</td>
<td>321</td>
<td>8</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>28.274</td>
</tr>
</tbody>
</table>

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

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

5 CONCLUSION

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

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

ACKNOWLEDGMENTS

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

REFERENCES