

Controlling Large Boolean Networks with Temporary and Permanent Perturbations

Cui Su¹, Soumya Paul² and Jun Pang^{1,2}

¹ Interdisciplinary Centre for Security, Reliability and Trust,
University of Luxembourg, Esch-sur-Alzette, Luxembourg

² Faculty of Science, Technology and Communication,
University of Luxembourg, Esch-sur-Alzette, Luxembourg

Abstract. A salient objective of studying gene regulatory networks (GRNs) is to identify potential target genes whose perturbations would lead to effective treatment of diseases. In this paper, we develop two control methods for GRNs in the context of asynchronous Boolean networks. Our methods compute a minimal subset of nodes of a given Boolean network, such that temporary or permanent perturbations of these nodes drive the network from an initial state to a target steady state. The main advantages of our methods include: (1) temporary and permanent perturbations can be feasibly conducted with techniques for genetic modifications in biological experiments; and (2) the minimality of the identified control sets can reduce the cost of experiments to a great extent. We apply our methods to several real-life biological networks *in silico* to show their efficiency in terms of computation time and their efficacy with respect to the number of nodes to be perturbed.

1 Introduction

Cellular reprogramming has opened up an unprecedented opportunity for pathological studies and regenerative medicine. It can rejuvenate somatic cells to pluripotent state, or even convert somatic cells directly to other differentiated cells [1–3]. Yet the identification of potential target genes and reprogramming paths remains a major hurdle in *in vivo* cellular reprogramming [4]. Combinatorial complexity of potential drug targets and the high cost of experimental tasks make an experimental approach [5] infeasible. This reinforces the need for efficient control methods based on mathematical modelling.

Many control methods have been developed in recent years to solve the problem. However, most of them are not applicable to real-life biological networks due to different reasons. First, biological networks have a specific control objective [5, 6]: finding a set of nodes, such that the control of these nodes can drive the system from a steady state to any other steady state. Biologically admissible steady states are observable phenotypes [7] and only the control of these states is meaningful. This rules out the methods based on classical controllability [8]. Second, some modelling frameworks are not suitable for biological networks. For example, linear dynamical networks fail to capture the non-linearity of biological

networks, thus rendering control strategies for such networks inapplicable [8–10]. Lack of biological information prohibits the modelling of biological systems with networks of ordinary differential equations (ODEs) [11]. This further limits the application of control methods based on ODE networks [12, 13]. Compared to the above modelling frameworks, Boolean networks (BNs) are well suited to model discrete and nonlinear dynamical biological systems. In BNs, genes are modelled as binary variables, being either ‘expressed’ or ‘not expressed’ and activation/inhibition regulations between genes are described by Boolean functions. The dynamics of a BN is determined by Boolean functions together with the update mode, either *synchronous* or *asynchronous*. The steady states of biological systems are described as *attractors* in BNs, to one of which the network eventually settles down. Recently, Kim et al. [14] and Zhao et al. [15] developed methods to drive a synchronous BN towards a desired attractor. However, the synchronous update mode is considered less realistic than the asynchronous update mode as only the latter allows for different time-scales of biological processes [16]. For asynchronous BNs, Zañudo et al. [6] developed a promising method to identify attractors and drug targets based on stable motifs. However, this method does not guarantee the minimality of perturbations.

Owing to various shortcomings of the existing control methods, we aim to develop a minimal and realistic control strategy for the control of asynchronous BNs. Given a BN, a source state and a target attractor, our idea is to identify an *exact minimal* set C of nodes of the BN, such that by perturbing the nodes in C , the dynamics of the BN is driven from the source state to the desired target attractor. One key factor to make this strategy realistic is to adopt physically admissible and experimentally feasible perturbations [17]. Rapid development of biomolecular techniques enables us to perturb expressions of nodes for different classes of time periods (instantaneously, temporarily or permanently) in both directions: from ‘expressed’ to ‘not expressed’ and/or from ‘not expressed’ to ‘expressed’ [18]. In [19], we developed such a method for perturbations that are instantaneous and showed that it is well suited for certain biological networks [17]. In this work, we develop methods for identifying a minimal set C of control nodes for asynchronous BNs whose perturbations can be long-term (temporary) or even permanent. The application of control C reshapes the BN to a new one, where the Boolean functions of the nodes in C are fixed to either ON or OFF. Permanent control leads to a permanent shift of the dynamics, whereas, for the temporary control, the perturbations of the identified set C of nodes are maintained for sufficient time until the network reaches a state, from which there only exist paths towards the target in the original BN.

We have implemented our temporary and permanent control methods and evaluated them on a variety of real-life biological networks modelled as BNs. We show that: (1) both temporary and permanent control sets can be efficiently computed on BNs that model real-life GRNs; (2) our methods not only capture the essential genes identified in the literature (e.g, see [20]), but also give other solutions for potential applications; (3) both methods can greatly reduce the number of control nodes compared to the instantaneous control [19]. The control

nodes computed by the two methods form a relatively small set even for large-scale networks. This agrees with the empirical findings that the control of a few nodes is sufficient to control cell fate determination processes [21, 22].

2 Preliminaries

In this section, we give preliminary notions of Boolean networks in Sections 2.1-2.3, and precisely formulate our control problems in Section 2.4.

2.1 Boolean networks

Let $[n]$ denote the set of positive integers $\{1, 2, \dots, n\}$. A Boolean network (BN) describes elements of a dynamical system with binary-valued nodes and interactions between elements with Boolean functions. It is formally defined as:

Definition 1 (Boolean networks). *A Boolean network is a tuple $\text{BN} = (\mathbf{x}, \mathbf{f})$ where $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ such that each $x_i, i \in [n]$ is a Boolean variable and $\mathbf{f} = \{f_1, f_2, \dots, f_n\}$ is a set of Boolean functions over \mathbf{x} .*

A Boolean network $\text{BN} = (\mathbf{x}, \mathbf{f})$ has an associated directed graph $\mathcal{G}_{\text{BN}} = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices or nodes and for every $i, j \in [n]$ 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 that an arbitrary but fixed network $\text{BN} = (\mathbf{x}, \mathbf{f})$ 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 \mathbf{x} and \mathbf{f} resp.

A state \mathbf{s} of BN is an element in $\{0, 1\}^n$. Let \mathbf{S} be the set of states of BN. For any state $\mathbf{s} = (s_1, s_2, \dots, s_n)$, and for every $i \in [n]$, the value of s_i , often denoted as $\mathbf{s}[i]$, represents the value that the variable x_i takes when the BN ‘is in state \mathbf{s} ’. For some $i \in [n]$, suppose f_i depends on $x_{i_1}, x_{i_2}, \dots, x_{i_k}$. Then $f_i(\mathbf{s})$ will denote the value $f_i(\mathbf{s}[i_1], \mathbf{s}[i_2], \dots, \mathbf{s}[i_k])$. For two states $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$, the Hamming distance between \mathbf{s} and \mathbf{s}' will be denoted as $\text{hd}(\mathbf{s}, \mathbf{s}')$ and $\arg \text{hd}(\mathbf{s}, \mathbf{s}') \subseteq [n]$ will denote the set of indices in which \mathbf{s} and \mathbf{s}' differ. It will be convenient to view $\arg \text{hd}(\mathbf{s}, \mathbf{s}')$ as a tuple of two disjoint sets $(\mathbb{1}, \mathbb{0})$, where $\mathbf{s}'[i] = 1$ if $i \in \mathbb{1}$ and $\mathbf{s}'[i] = 0$ if $i \in \mathbb{0}$. For a state \mathbf{s} and a subset $\mathbf{S}' \subseteq \mathbf{S}$, the Hamming distance between \mathbf{s} and \mathbf{S}' is defined as the minimum of the Hamming distances between \mathbf{s} and all the states in \mathbf{S}' , i.e. $\text{hd}(\mathbf{s}, \mathbf{S}') = \min_{\mathbf{s}' \in \mathbf{S}'} \text{hd}(\mathbf{s}, \mathbf{s}')$. Let $\arg \text{hd}(\mathbf{s}, \mathbf{S}')$ denote the set of tuples, such that $(\mathbb{1}, \mathbb{0}) \in \arg \text{hd}(\mathbf{s}, \mathbf{S}')$ if and only if $\mathbb{1} \cup \mathbb{0}$ is a set of indices of the variables that realise the minimum Hamming distance.

2.2 Dynamics of Boolean networks

We assume that the Boolean network evolves in discrete time steps. It starts initially in a state \mathbf{s}_0 and its state changes in every time step according to the update functions \mathbf{f} . The updating may happen in various ways [23, 24]. Every such way of updating gives rise to a different dynamics for the network. In this work, we shall be interested primarily in the asynchronous update mode.

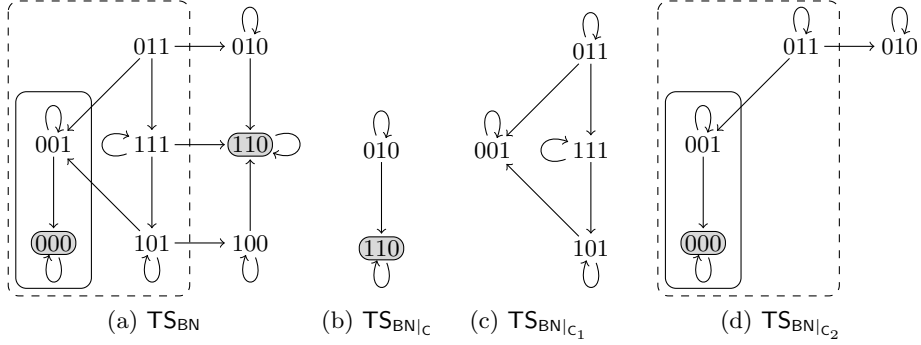


Fig. 1: The transition systems of BN.

Definition 2 (Asynchronous dynamics of Boolean networks). Suppose $\mathbf{s}_0 \in \mathbf{S}$ is an initial state of BN. The asynchronous evolution of BN is a function $\xi_{\text{BN}} : \mathbb{N} \rightarrow \wp(\mathbf{S})$ such that $\xi_{\text{BN}}(0) = \{\mathbf{s}_0\}$ and for every $j \geq 0$, if $\mathbf{s} \in \xi_{\text{BN}}(j)$ then $\mathbf{s}' \in \xi_{\text{BN}}(j+1)$ is a possible next state of \mathbf{s} iff either $\text{hd}(\mathbf{s}, \mathbf{s}') = 1$ and $\mathbf{s}'[i] = f_i(\mathbf{s})$ where $\mathbf{s}'[i] = 1 - \mathbf{s}[i]$ or $\text{hd}(\mathbf{s}, \mathbf{s}') = 0$ and there exists i such that $\mathbf{s}'[i] = f_i(\mathbf{s})$.

Note that the asynchronous dynamics is non-deterministic – the value of exactly one variable is updated in a single time-step whose index is not known in advance. Henceforth, when we talk about the dynamics of BN, we shall mean the asynchronous dynamics as defined above. The dynamics of a Boolean network can be represented as a *state transition graph* or a *transition system (TS)*.

Definition 3 (Transition system of BN). The transition system of BN, denoted as TS_{BN} is a tuple $(\mathbf{S}, \rightarrow_{\text{BN}})$, where the vertices are the set of states \mathbf{S} and for any two states \mathbf{s} and \mathbf{s}' there is a directed edge from \mathbf{s} to \mathbf{s}' , denoted $\mathbf{s} \rightarrow_{\text{BN}} \mathbf{s}'$ iff \mathbf{s}' is a possible next state of \mathbf{s} according to the asynchronous evolution function ξ_{BN} of BN.

Example 1. Consider a network $\text{BN} = (\mathbf{x}, \mathbf{f})$, where $\mathbf{x} = \{x_1, x_2, x_3\}$, $\mathbf{f} = \{f_1, f_2, f_3\}$, and $f_1 = ((\neg x_3) \wedge x_1) \vee x_2$, $f_2 = ((\neg x_3) \wedge x_2) \vee ((\neg x_3) \wedge x_1)$, $f_3 = 0$. The transition system of the network TS_{BN} is given in Fig. 1a.

Definition 4 (Control). A control \mathbf{C} is a tuple $(\mathbf{1}, \mathbf{0})$ where $\mathbf{1}, \mathbf{0} \subseteq [n]$, $\mathbf{1}$ and $\mathbf{0}$ are mutually disjoint (possibly empty) set of indices of variables of BN. The size of control \mathbf{C} is defined as $\text{size}(\mathbf{C}) = |\mathbf{1}| + |\mathbf{0}|$. Given two states $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$, let $\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}'} = (\mathbf{1}, \mathbf{0})$, where $\mathbf{1} = \{i \in [n] \mid \mathbf{s}'[i] = 1 = 1 - \mathbf{s}[i]\}$ and $\mathbf{0} = \{i \in [n] \mid \mathbf{s}'[i] = 0 = 1 - \mathbf{s}[i]\}$.

Intuitively, $\mathbf{1}$ and $\mathbf{0}$ represent the indices of variables of BN whose values are held fixed to 1 and 0 respectively under the control \mathbf{C} . For $\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}'}$, $\mathbf{1} \cup \mathbf{0}$ are the set of indices in which \mathbf{s} and \mathbf{s}' differ, out of which $\mathbf{1}$ and $\mathbf{0}$ are the indices which have a value 1 and 0 in \mathbf{s}' , respectively. The application of a control \mathbf{C} to

$\text{BN} = (\mathbf{x}, \mathbf{f})$ has the effect of reducing the state space of BN to those which have the values of the variables in $\mathbb{1}$ and $\mathbb{0}$ set respectively to 1 and 0 and modifying the update functions accordingly. This results in a new Boolean network derived from BN defined as follows.

Definition 5 (BN under control). *Let $\mathbf{C} = (\mathbb{1}, \mathbb{0})$ be a control. Then the Boolean network BN under control \mathbf{C} , denoted $\text{BN}|_{\mathbf{C}}$, is defined as a tuple $\text{BN}|_{\mathbf{C}} = (\hat{\mathbf{x}}, \hat{\mathbf{f}})$ where the elements of $\hat{\mathbf{x}}$ and $\hat{\mathbf{f}}$ are given as, for all $i \in [n]$:*

- $\hat{x}_i = 1$ if $i \in \mathbb{1}$, $\hat{x}_i = 0$ if $i \in \mathbb{0}$, and $\hat{x}_i = x_i$ otherwise.
- $\hat{f}_i = 1$ if $i \in \mathbb{1}$, $\hat{f}_i = 0$ if $i \in \mathbb{0}$, and $\hat{f}_i = f_i$ otherwise.

The state space of $\text{BN}|_{\mathbf{C}}$, denoted $\mathbf{S}|_{\mathbf{C}}$ is derived by fixing the values of the variables in the set \mathbf{C} to their respective values and is defined as $\mathbf{S}|_{\mathbf{C}} = \{\mathbf{s} \in \mathbf{S} \mid \mathbf{s}[i] = 1, i \in \mathbb{1} \text{ and } \mathbf{s}[j] = 0, j \in \mathbb{0}\}$. Note that $\mathbf{S}|_{\mathbf{C}} \subseteq \mathbf{S}$. For any subset \mathbf{S}' of \mathbf{S} we let $\mathbf{S}'|_{\mathbf{C}} = \mathbf{S}' \cap \mathbf{S}|_{\mathbf{C}}$. The asynchronous dynamics and the transition system of $\text{BN}|_{\mathbf{C}}$ are defined similarly to Definition 2 and Definition 3 by replacing $\text{BN} = (\mathbf{x}, \mathbf{f})$ with $\text{BN}|_{\mathbf{C}} = (\hat{\mathbf{x}}, \hat{\mathbf{f}})$. We omit the definitions to avoid duplications.

Definition 6 (Application and release of control). *Let $\mathbf{s} \in \mathbf{S}$ be a state of BN and let $\mathbf{C} = (\mathbb{1}, \mathbb{0})$ be a control. The instantaneous application or 0-step application of \mathbf{C} to \mathbf{s} results in a state \mathbf{s}' such that $\mathbf{s}'[i] = 1$ for all $i \in \mathbb{1}$, $\mathbf{s}'[j] = 0$ for all $j \in \mathbb{0}$ and $\mathbf{s}'[k] = \mathbf{s}[k]$ otherwise. We will often denote this as $\mathbf{s} \stackrel{\mathbf{C}}{\sim} \mathbf{s}'$ and denote \mathbf{s}' as $\mathbf{C}(\mathbf{s})$. The application of \mathbf{C} to \mathbf{s} for $t \geq 1$ time steps results in a sequence $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_t$, where $\mathbf{s}_0 = \mathbf{C}(\mathbf{s})$ and for every $k \in [t]$, $\mathbf{s}_k \in \xi_{\text{BN}|_{\mathbf{C}}}(\mathbf{s}_{k-1})$. When $t \rightarrow \infty$, we shall call it a permanent application of \mathbf{C} to \mathbf{s} or a permanent control of \mathbf{s} .*

Suppose BN under control \mathbf{C} is in state $\mathbf{s} \in \mathbf{S}|_{\mathbf{C}}$ and has been evolving according to $\xi_{\text{BN}|_{\mathbf{C}}}$. The release of control at \mathbf{s} is performed instantaneously and it restores the update dynamics to ξ_{BN} . We often denote it as $\mathbf{s} \stackrel{\mathbf{C}^{-1}}{\sim} \mathbf{s} \rightarrow_{\text{BN}} \dots$

Thus suppose BN starts evolving from an initial state $\mathbf{s}_0 \in \mathbf{S}$ and after t_1 steps a control \mathbf{C} is applied to it. Suppose the control lasts for t_2 steps and then it is released and then BN goes back to evolving according to its original update dynamics. This will result in a sequence that can be represented as:

$$\underbrace{\mathbf{s}_0 \rightarrow_{\text{BN}} \mathbf{s}_1 \rightarrow_{\text{BN}} \dots \rightarrow_{\text{BN}} \mathbf{s}_{t_1}}_{t_1 \text{ steps}} \stackrel{\mathbf{C}}{\sim} \underbrace{\mathbf{s}'_0 \rightarrow_{\text{BN}|_{\mathbf{C}}} \mathbf{s}'_1 \rightarrow_{\text{BN}|_{\mathbf{C}}} \dots \rightarrow_{\text{BN}|_{\mathbf{C}}} \mathbf{s}'_{t_2}}_{t_2 \text{ steps under control } \mathbf{C}} \stackrel{\mathbf{C}^{-1}}{\sim} \mathbf{s}''_0 \rightarrow_{\text{BN}} \mathbf{s}''_1 \rightarrow_{\text{BN}} \dots$$

Intuitively, on the application of control \mathbf{C} for t_2 steps, the behaviour of BN is given according to $\text{TS}_{\text{BN}|_{\mathbf{C}}}$ for t_2 time steps. After that, when \mathbf{C} is released, the behaviour goes back to TS_{BN} . The release of \mathbf{C} does not change the value of any variable, thus $\mathbf{s}''_0 = \mathbf{s}'_{t_2}$.

Example 2. For Example 1, given a control $\mathbf{C} = (\mathbb{1}, \mathbb{0})$, where $\mathbb{1} = \{2\}$ and $\mathbb{0} = \{3\}$, the transition system $\text{TS}_{\text{BN}|_{\mathbf{C}}}$ of $\text{BN}|_{\mathbf{C}}$ is described in Fig. 1b.

2.3 Attractors and basins

In what follows we shall use the generic notation TS to represent either the full transition system TS_{BN} of BN or the transition system $\text{TS}_{\text{BN}|C}$ of BN under a control C . Similarly we let ' \rightarrow ' stand for either \rightarrow_{BN} or $\rightarrow_{\text{BN}|C}$. We define several notions on TS below which can be interpreted both on TS_{BN} and $\text{TS}_{\text{BN}|C}$. The state space, transitions etc. will correspond either to TS_{BN} or $\text{TS}_{\text{BN}|C}$ (or both) depending on the context.

A path ρ from a state \mathbf{s} to a state \mathbf{s}' is a (possibly empty) sequence of transitions from \mathbf{s} to \mathbf{s}' in TS . Thus, $\rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots \rightarrow \mathbf{s}_k$, where $\mathbf{s}_0 = \mathbf{s}$ and $\mathbf{s}_k = \mathbf{s}'$. A path from a state \mathbf{s} to a subset \mathbf{S}' of \mathbf{S} is a path from \mathbf{s} to any state $\mathbf{s}' \in \mathbf{S}'$. An infinite path ρ from \mathbf{s} is an infinite sequence of transitions from \mathbf{s} . Let $\text{Path}_\infty(\mathbf{s})$ denote the set of infinite paths from \mathbf{s} . Let $\rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots$ be an infinite path from \mathbf{s}_0 . A state $\mathbf{s} \in \mathbf{S}$ appears *infinitely often* in ρ if for every $i \geq 0$ there exists $j \geq i$ such that $\mathbf{s}_j = \mathbf{s}$. \mathbf{s} appears *finitely often* in ρ otherwise.

Definition 7 (Fairness). Let $\mathbf{s}_0 \in \mathbf{S}$. An infinite path $\rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots$ is said to be unfair if for every state \mathbf{s} that occurs infinitely often in ρ , there exists a possible next state \mathbf{s}' of \mathbf{s} which occurs only finitely often in ρ . ρ is said to be fair otherwise.

It is important to impose the restriction of fairness because otherwise, the TS_{BN} would have pseudo-attractors (defined shortly) which would not correspond to any meaningful phenotypes of the GRN being modelled. Fairness ensures that the attractors of TS are exactly the ones that are experimentally observed. Henceforth, we shall assume that the evolution of BN is always fair, and hence consider only fair paths. Therefore, let $\text{Path}_\infty(\mathbf{s})$ denote the set of all infinite fair paths from a state $\mathbf{s} \in \mathbf{S}$.

For any state $\mathbf{s} \in \mathbf{S}$, let $\text{pre}_{\text{TS}}(\mathbf{s}) = \{\mathbf{s}' \in \mathbf{S} \mid \mathbf{s}' \rightarrow \mathbf{s}\}$ and let $\text{post}_{\text{TS}}(\mathbf{s}) = \{\mathbf{s}' \in \mathbf{S} \mid \mathbf{s} \rightarrow \mathbf{s}'\}$ where $\rightarrow \in \{\rightarrow_{\text{BN}}, \rightarrow_{\text{BN}|C}\}$ depending on the context. $\text{pre}_{\text{TS}}(\mathbf{s})$ contains all the states that can reach \mathbf{s} by performing a single transition in TS and $\text{post}_{\text{TS}}(\mathbf{s})$ contains all the states that can be reached from \mathbf{s} by a single transition in TS . $\text{pre}_{\text{TS}}(\mathbf{s})$ and $\text{post}_{\text{TS}}(\mathbf{s})$ are often called the set of *predecessors* and *successors* of \mathbf{s} . Note that, by definition, $\text{hd}(\mathbf{s}, \text{pre}_{\text{TS}}(\mathbf{s})) \leq 1$ and $\text{hd}(\mathbf{s}, \text{post}_{\text{TS}}(\mathbf{s})) \leq 1$. pre_{TS} and post_{TS} can be lifted to a subset \mathbf{S}' of \mathbf{S} as: $\text{pre}_{\text{TS}}(\mathbf{S}') = \bigcup_{\mathbf{s} \in \mathbf{S}'} \text{pre}_{\text{TS}}(\mathbf{s})$ and $\text{post}_{\text{TS}}(\mathbf{S}') = \bigcup_{\mathbf{s} \in \mathbf{S}'} \text{post}_{\text{TS}}(\mathbf{s})$. We define $\text{pre}_{\text{TS}}^{i+1}(\mathbf{S}') = \text{pre}_{\text{TS}}(\text{pre}_{\text{TS}}^i(\mathbf{S}'))$ and $\text{post}_{\text{TS}}^{i+1}(\mathbf{S}') = \text{post}_{\text{TS}}(\text{post}_{\text{TS}}^i(\mathbf{S}'))$ where $\text{pre}_{\text{TS}}^0(\mathbf{S}') = \text{post}_{\text{TS}}^0(\mathbf{S}') = \mathbf{S}'$. For a state $\mathbf{s} \in \mathbf{S}$, $\text{reach}_{\text{TS}}(\mathbf{s})$ denotes the set of states \mathbf{s}' such that there is a path from \mathbf{s} to \mathbf{s}' in TS and can be defined as the fixpoint of the successor operation which is often denoted as $\text{post}_{\text{TS}}^*$. Thus, $\text{reach}_{\text{TS}}(\mathbf{s}) = \text{post}_{\text{TS}}^*(\mathbf{s})$.

Definition 8 (Attractor). An attractor A of TS is a minimal non-empty subset of states of \mathbf{S} such that for every $\mathbf{s} \in A$, $\text{reach}_{\text{TS}}(\mathbf{s}) = A$.

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 \mathbf{s} if $\text{reach}_{\text{TS}}(\mathbf{s}) \cap A \neq \emptyset$. Attractors represent the stable behaviour of the BN according to the dynamics. The network

starting at any initial state $\mathbf{s}_0 \in \mathbf{S}$ will eventually end up in one of the attractors of TS and remain there forever unless perturbed.

Observation 1 Any attractor of TS is a bottom strongly connected component of TS.

Let \mathbf{S}' be a subset of states of \mathbf{S} . We define subsets of states of \mathbf{S} called the weak and strong basins of \mathbf{S}' , denoted as $\text{bas}_{\text{TS}}^W(\mathbf{S}')$ and $\text{bas}_{\text{TS}}^S(\mathbf{S}')$, respectively.

Definition 9 (Basin). Let $\mathbf{S}' \subseteq \mathbf{S}$.

- **Weak basin:** The weak basin of \mathbf{S}' with respect to TS, is defined as $\text{bas}_{\text{TS}}^W(\mathbf{S}') = \{\mathbf{s} \in \mathbf{S} \mid \text{reach}_{\text{TS}}(\mathbf{s}) \cap \mathbf{S}' \neq \emptyset\}$ which equals the fixpoint of the predecessor operation on \mathbf{S}' and is often denoted as $\text{pre}_{\text{TS}}^*(\mathbf{S}')$. Thus, $\text{bas}_{\text{TS}}^W(\mathbf{S}') = \text{pre}_{\text{TS}}^*(\mathbf{S}')$. In other words, since all paths in $\text{Path}_\infty(\mathbf{s})$ are fair,

$$\text{bas}_{\text{TS}}^W(\mathbf{S}') = \{\mathbf{s} \in \mathbf{S} \mid \exists \rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots \in \text{Path}_\infty(\mathbf{s}), \exists j \geq 0, \mathbf{s}_j \in \mathbf{S}'\}$$

- **Strong basin:** The strong basin of \mathbf{S}' with respect to TS, is defined as

$$\text{bas}_{\text{TS}}^S(\mathbf{S}') = \{\mathbf{s} \in \mathbf{S} \mid \forall \rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots \in \text{Path}_\infty(\mathbf{s}), \exists j \geq 0, \mathbf{s}_j \in \mathbf{S}'\}$$

We say that a path $\rho = \mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \dots$ eventually reaches \mathbf{S}' if there exists $j \geq 0$ such that $\mathbf{s}_j \in \mathbf{S}'$. Intuitively, the weak basin of \mathbf{S}' consists of all states from which there is at least one path to \mathbf{S}' , whereas the strong basin of \mathbf{S}' consists of all states from which all paths eventually reach \mathbf{S}' . Clearly thus, $\text{bas}_{\text{TS}}^S(\mathbf{S}') \subseteq \text{bas}_{\text{TS}}^W(\mathbf{S}')$. If \mathbf{S}' is an attractor A (say), $\text{bas}_{\text{TS}}^W(A)$ and $\text{bas}_{\text{TS}}^S(A)$ will also be referred to as weak and strong *basins of attractions* with respect to A . Thus the weak basin of attraction of A is the set of all states \mathbf{s} from which there is a path to A . It is possible that there are paths from \mathbf{s} to some other attractor $A' \neq A$. However, the notion of a strong basin does not allow this. Thus, it is easy to see that,

Observation 2 If $\mathbf{s} \in \text{bas}_{\text{TS}}^S(A)$ then $\mathbf{s} \notin \text{bas}_{\text{TS}}^W(A')$ for any other attractor A' . Therefore, $\text{bas}_{\text{TS}}^S(A) = \text{bas}_{\text{TS}}^W(A) \setminus (\bigcup_{A'} \text{bas}_{\text{TS}}^W(A'))$ where the union is over all attractors $A' \neq A$ of TS.

Note that if \mathbf{S}' is an attractor A , then if ρ eventually reaches A , it gets stuck in A forever. That is, for every $i \geq 0$, $\mathbf{s}_i \in A$ implies $\mathbf{s}_j \in A$ for all $j > i$. This follows directly from Definition 8. We need the notion of strong basin to ensure that every fair sequence under a given update dynamics always reaches the target attractor. The following observation will be crucial for the control algorithms developed in this paper.

Observation 3 Let $\mathbf{s} \in \mathbf{S}$ and $\mathbf{S}' \subseteq \mathbf{S}$. Every path $\rho \in \text{Path}_\infty(\mathbf{s})$

1. possibly eventually reaches \mathbf{S}' if and only if $\mathbf{s} \in \text{bas}_{\text{TS}}^W(\mathbf{S}')$,
2. always eventually reaches \mathbf{S}' if and only if $\mathbf{s} \in \text{bas}_{\text{TS}}^S(\mathbf{S}')$.

Example 3. To continue with the example given in Example 1, TS_{BN} has two attractors $A_1 = \{(110)\}$ and $A_2 = \{(000)\}$ shown by dark grey rectangles in Fig. 1a. The weak basin and the strong basin of A_2 are shown by the dashed and solid rectangles, respectively. The state $\mathbf{s}_1 = \{(011)\}$ is in the weak basin $\text{bas}_{\text{TS}_{\text{BN}}}^W(A_2)$ but not in the strong basin $\text{bas}_{\text{TS}_{\text{BN}}}^S(A_2)$ of A_2 as there exist paths from \mathbf{s}_1 to the other attractor A_1 . Starting from the state \mathbf{s}_1 , BN can reach either A_1 or A_2 eventually. The state $\mathbf{s}_2 = \{(001)\}$ is in the strong basin $\text{bas}_{\text{TS}_{\text{BN}}}^S(A_2)$ of A_2 . Starting from \mathbf{s}_2 , BN always eventually reaches A_2 .

2.4 The control problem

As described in the introduction (Section 1), the attractors of a Boolean network represent the cellular phenotypes. Some of these attractors may be diseased, weak or undesirable while others are healthy and desirable. Curing a disease is thus, in effect, moving the dynamics of the network from an undesired ‘source’ attractor to a desired ‘target’ attractor.

This can be achieved by applying control (as defined in Section 2.2) to the network. There can be various strategies of applying such a control. These can be broadly classified based on the number of parameters of the network controlled at the same time and the amount of time the control is applied. In terms of the number of parameters controlled at the same time, we have: (1) **simultaneous control** – the perturbation is applied to all the parameters at once; and (2) **sequential control** – the perturbation is applied to the required parameters over a sequence of steps. Based on the amount of time that the control is applied, we have: (a) **permanent control** – the control is applied for all the following time steps, i.e., the parameters are changed forever; and (b) **temporary control** – the control is applied for a finite (possibly zero) number of steps and then removed.

In this work we shall be interested in simultaneous control that is applied both temporarily and permanently. Moreover, we aim to compute the exact *minimum* number of parameters needed to be controlled in each case. The control problems that we shall deal with in this work are defined as follows.

Definition 10 (Control problems). *Given a source state $\mathbf{s} \in \mathbf{S}$ and an attractor A of TS_{BN} of BN, a:*

1. **Permanent control:** *is a control $C = (\mathbf{1}, \mathbf{0})$ such that the dynamics of BN always eventually reaches A on the permanent application of C to \mathbf{s} . (Here we assume implicitly that A is also an attractor of the transition system under control $\text{TS}_{\text{BN}|C}$.)*
2. **Temporary control:** *is a control $C = (\mathbf{1}, \mathbf{0})$ 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 on the application of C to \mathbf{s} for t steps.*

In addition, if in each case C is minimal, in the sense that, for every control C' that achieves one of the above objectives, $\text{size}(C) \leq \text{size}(C')$, we call C a minimal permanent (resp. temporary) control. The control problems are then, given a source state $\mathbf{s} \in \mathbf{S}$ and an attractor A of TS_{BN} of BN, find a minimal permanent or temporary control.

Note that the constraint of minimality in the above definition makes the problems non-trivial. Otherwise, one can simply choose a control C such that $C(\mathbf{s}) \in A$ in each case. In [19, 25], we developed a method for the efficient *minimal simultaneous control* of Boolean networks, where given a source state \mathbf{s} and an attractor of TS_{BN} of BN the control C is applied simultaneously and instantaneously to \mathbf{s} so that the dynamics of BN eventually reaches A and C is minimal. Such a control is a special case of the temporary control defined above with $t = 0$. Since the minimal simultaneous control problem of [19, 25] is computationally difficult (PSPACE-hard), the control problems that we study here, defined above, are also computationally difficult (at least PSPACE-hard). Thus, efficient algorithms to solve these problems are highly unlikely. However, we showed in [19, 25], that if the BNs are structurally well-behaved (e.g., the graph of the BN has small strongly connected components (SCCs), with a small number of interdependencies between the SCCs etc.), we can have relatively efficient methods to compute the attractors and basins of such BNs. Since it is known that real-life BNs corresponding to GRNs are reasonably well-behaved, this led us to develop efficient algorithms for computing the weak and strong basins of desired target attractors of such BNs by decomposing the BNs based on the SCCs of their graphs. The algorithms we develop here will crucially use the procedures developed in [19, 25] for the computation of the weak and strong basins of the target attractors, denoted as $\text{COMP_WB}(A)$ and $\text{COMP_SB}(A)$ [19, 25], respectively. These then applied to real-life networks can result in a significant level of efficiency as will be demonstrated later.

3 Results

In this section, we develop algorithms to solve the control problems described in Definition 10. These algorithms are based on the key observation made in Observation 3. Indeed, given a source state \mathbf{s} and a target attractor A of TS_{BN} of BN, if after the application of a control C to \mathbf{s} , the resulting state $C(\mathbf{s})$ lies in the strong basin of A w.r.t. the transition system under control, $\text{TS}_{\text{BN}|C}$, then the dynamics will always eventually reach A . One needs to be careful though as the attractors and their structure in $\text{TS}_{\text{BN}|C}$ might be different from TS_{BN} . However, we note that the application of C to TS does not create any additional edges except for self loops.

Lemma 1. *Let C be a control. If $\mathbf{s} \rightarrow_{\text{BN}|C} \mathbf{s}'$ is in $\text{TS}_{\text{BN}|C}$ then $\mathbf{s} \neq \mathbf{s}'$ implies $\mathbf{s} \rightarrow_{\text{BN}} \mathbf{s}'$ is in TS_{BN} .*

3.1 Permanent control

We now develop an algorithm to solve the problem of permanent control. For the sake of simplicity, we use $\text{bas}_{\text{BN}}^{S(W)}(\cdot)$ and $\text{bas}_{\text{BN}|C}^{S(W)}(\cdot)$ to represent $\text{bas}_{\text{TS}_{\text{BN}}}^{S(W)}(\cdot)$ and $\text{bas}_{\text{TS}_{\text{BN}|C}}^{S(W)}(\cdot)$, respectively. The following proposition will be useful.

Proposition 1. *Let C be a control and A be an attractor of TS_{BN} such that A is also an attractor of $\text{TS}_{\text{BN}|C}$. For any $\mathbf{s} \in \mathbf{S}$, if $\mathbf{s} \in \text{bas}_{\text{BN}|C}^W(A)$ then $\mathbf{s} \in \text{bas}_{\text{BN}}^W(A)$.*

The converse of Proposition 1 may not hold as shown by the following example.

Example 4. Let $C_1 = \{\mathbb{1}_1, \mathbb{0}_1\}$ with $\mathbb{1}_1 = \{3\}$ and $\mathbb{0}_1 = \emptyset$ and $C_2 = \{\mathbb{1}_2, \mathbb{0}_2\}$ with $\mathbb{1}_2 = \emptyset$ and $\mathbb{0}_2 = \{1\}$ be two controls of the BN of Example 1. The original transition system TS_{BN} and the two transition systems under control $\text{TS}_{\text{BN}|C_1}$ and $\text{TS}_{\text{BN}|C_2}$ are given in Fig. 1. The application of C_1 fixes \mathbf{f}_3 to 1 and neither of the attractors is preserved in $\text{TS}_{\text{BN}|C_1}$. The application of C_2 fixes \mathbf{f}_1 to 0. In $\text{TS}_{\text{BN}|C_2}$, the attractor $A_2 = \{(000)\}$ is the attractor of TS_{BN} and $\text{TS}_{\text{BN}|C_2}$. The state $\mathbf{s} = \{(111)\}$ is in $\text{bas}_{\text{BN}}^W(A_2)$ but not in $\text{bas}_{\text{BN}|C_2}^W(A_2)$.

The intuition for the algorithm for the problem of permanent control that we shall develop in this section is as follows. Suppose $\mathbf{s} \in \mathbf{S}$ is an initial state and A is the target attractor of TS_{BN} that we want the dynamics of BN to always eventually reach. The following is a straightforward corollary of Observation 3.

Corollary 1. *A control C is a permanent control from \mathbf{s} to A iff A is an attractor of $\text{TS}_{\text{BN}|C}$ and $C(\mathbf{s}) \in \text{bas}_{\text{BN}|C}^S(A)$.*

Thus, we want to find a control C such that the condition $C(\mathbf{s}) \in \text{bas}_{\text{BN}|C}^S(A)$ is satisfied. Now, since we want the control C to be minimal, we proceed as follows. We start with a state $\mathbf{s}' \in \text{bas}_{\text{BN}}^W(A)$ that has the minimal Hamming distance with \mathbf{s} . We first check if A is an attractor of $\text{TS}_{\text{BN}|C^{\mathbf{s} \rightarrow \mathbf{s}'}}$, since otherwise, $C^{\mathbf{s} \rightarrow \mathbf{s}'}$ cannot be a permanent control (by definition). If A is indeed an attractor of $\text{TS}_{\text{BN}|C^{\mathbf{s} \rightarrow \mathbf{s}'}}$, we check if $\mathbf{s}' \in \text{bas}_{\text{BN}|C^{\mathbf{s} \rightarrow \mathbf{s}'}}^S(A)$. If so, we are done. Otherwise, we remove \mathbf{s}' from $\text{bas}_{\text{BN}}^W(A)$ and select a state \mathbf{s}'' from $(\text{bas}_{\text{BN}}^W(A) \setminus \{\mathbf{s}'\})$ having the minimal Hamming distance with \mathbf{s} . We repeat the same procedure this time with $C^{\mathbf{s} \rightarrow \mathbf{s}''}$. We iterate till we find a state $\mathbf{s}^* \in \text{bas}_{\text{BN}}^W(A)$ such that $\mathbf{s}^* \in \text{bas}_{\text{BN}|C^{\mathbf{s} \rightarrow \mathbf{s}^*}}^S(A)$.

The procedure described above, in the worst case, explores all possible states in $\text{bas}_{\text{TS}_{\text{BN}}}^W(A)$. By Proposition 1 we know that for any control C , $\text{bas}_{\text{BN}|C}^W(A) \subseteq \text{bas}_{\text{BN}}^W(A)$. Thus, it is enough to explore only the states in $\text{bas}_{\text{BN}}^W(A)$ and it will eventually find the required control. $C^{\mathbf{s} \rightarrow \mathbf{s}^*}$ is then the required minimal permanent control. Algorithm 1 describes this procedure in pseudo-code.

3.2 Temporary control

The algorithm for computing a minimal temporary control is slightly more involved than that for computing a minimal permanent control developed in Section 3.1. We first prove the following proposition with the help of Lemma 1.

Proposition 2. *Let A be an attractor of TS_{BN} and C be any control.*

1. *For any state $\mathbf{s} \in \text{bas}_{\text{BN}}^S(A)$, $\text{reach}_{\text{TS}_{\text{BN}}}(\mathbf{s}) \subseteq \text{bas}_{\text{BN}}^S(A)$.*
2. *For any state $\mathbf{s} \in (\text{bas}_{\text{BN}}^S(A)|_C)$, $\text{reach}_{\text{TS}_{\text{BN}|C}}(\mathbf{s}) \subseteq (\text{bas}_{\text{BN}}^S(A)|_C)$.*

Algorithm 1 Minimal permanent control

```

1: procedure COMP_PERM_CONTROL( $\mathcal{G}, \mathbf{f}, \mathbf{s}, A$ ) %  $\mathbf{s}$ : the source state;  $A$ : the target
2:    $\text{WB} := \text{COMP\_WB}(A, \mathbf{f})$  % the weak basin of  $A$  in  $\text{TS}_{\text{BN}}$  [19, 25]
3:    $\text{isMin} := \text{false}$ 
4:   while  $\text{isMin} = \text{false}$  and  $\text{WB} \neq \emptyset$  do
5:      $\mathbf{C} \in \arg \text{hd}(\mathbf{s}, \text{WB}), \mathbf{s}' := \mathbf{C}(\mathbf{s})$  % a possible minimal control from  $\mathbf{s}$  to  $A$ 
6:     if  $\{\mathbf{s}'\}|_{\mathbf{C}} = A|_{\mathbf{C}}$  then %  $A$  is preserved in  $\text{TS}_{\text{BN}|_{\mathbf{C}}}$ 
7:        $\mathbf{f}_{\mathbf{C}} := \text{COMP\_FN\_CONTR}(\mathbf{f}, \mathbf{C})$  %  $\mathbf{f}$  in  $\text{BN}|_{\mathbf{C}}$  (see Algorithm 2)
8:        $\text{SB}_{\mathbf{C}} := \text{COMP\_SB}(A, \mathbf{f}_{\mathbf{C}})$  % the strong basin of  $A$  in  $\text{TS}_{\text{BN}|_{\mathbf{C}}}$  [19, 25]
9:       if  $\mathbf{s}' \in \text{SB}_{\mathbf{C}}$  then
10:         $\text{isMin} := \text{true}$ 
11:       if  $\text{isMin} = \text{false}$  then
12:          $\text{WB} := \text{WB} \setminus \{\mathbf{s}'\}$ 
13:   return  $\mathbf{C}$ 

```

Algorithm 2 Helper functions

```

1: procedure COMP_FN_CONTR( $\mathbf{f}, \mathbf{C}$ ) %  $\mathbf{C} = (1, 0)$ 
2:    $\mathbf{f}_{\mathbf{C}} := \mathbf{f}$ 
3:   for  $i \in 1$  do
4:      $\mathbf{f}_{\mathbf{C}}[i] := 1$ 
5:   for  $i \in 0$  do
6:      $\mathbf{f}_{\mathbf{C}}[i] := 0$ 
7:   return  $\mathbf{f}_{\mathbf{C}}$ 
8:
9:
10: procedure COMP_STATE_CONTR( $\mathbf{S}, \mathbf{C}$ )
11:    $\mathbf{S}|_{\mathbf{C}} := \mathbf{S}$ 
12:   for  $\mathbf{s} \in \mathbf{S}$  do
13:     for  $i \in 1$  do
14:       if  $\mathbf{s}[i] \neq 1$  then
15:          $\mathbf{S}|_{\mathbf{C}} := \mathbf{S}|_{\mathbf{C}} \setminus \{\mathbf{s}\}$ 
16:     for  $j \in 0$  do
17:       if  $\mathbf{s}[j] \neq 0$  then
18:          $\mathbf{S}|_{\mathbf{C}} := \mathbf{S}|_{\mathbf{C}} \setminus \{\mathbf{s}\}$ 
19:   return  $\mathbf{S}_{\mathbf{C}}$ 

```

Using Observation 3 and Proposition 2, we can prove the following theorem.

Theorem 4. *Let $\mathbf{s} \in \mathbf{S}$ be a source state and A be the target attractor of TS_{BN} . A control \mathbf{C} is a temporary control from \mathbf{s} to A if and only if $\text{bas}_{\text{BN}}^{\mathbf{S}}(A)|_{\mathbf{C}} \neq \emptyset$ and $\mathbf{C}(\mathbf{s}) \in \text{bas}_{\text{BN}|_{\mathbf{C}}}^{\mathbf{S}}(\text{bas}_{\text{BN}}^{\mathbf{S}}(A)|_{\mathbf{C}})$.*

Theorem 4 forms the basis for our algorithm for computing the temporary control \mathbf{C} , given a source state \mathbf{s} and a target attractor A . Intuitively, on the application of \mathbf{C} , we want the dynamics to move to a state $\mathbf{C}(\mathbf{s})$ which is in the strong basin (w.r.t the restricted transition system $\text{TS}_{\text{BN}|_{\mathbf{C}}}$) of the strong basin (w.r.t the original transition system TS_{BN}) of the target attractor A , restricted to states in $\mathbf{S}|_{\mathbf{C}}$. Then if we hold the control \mathbf{C} for long enough, the dynamics will eventually reach the strong basin of A in TS_{BN} . By Proposition 2, we know that once in the strong basin, the dynamics cannot escape it. This means that finally when the control \mathbf{C} is released, the dynamics is in the strong basin of A and hence will eventually reach A which is the target.

How can we ensure that we indeed compute a temporary control \mathbf{C} that is minimal? Once again, we proceed as before. We start with a state $\mathbf{s}' \in$

Algorithm 3 Minimal temporary control

```

1: procedure COMP_TEMP_CONTROL( $\mathcal{G}, \mathbf{f}, \mathbf{s}, A$ ) %  $\mathbf{s}$ : the source state;  $A$ : the target
2:    $\text{WB} := \text{COMP\_WB}(A, \mathbf{f})$  % the weak basin of  $A$  in  $\text{TS}_{\text{BN}}$ 
3:    $\text{SB} := \text{COMP\_SB}(A, \mathbf{f})$  % the strong basin of  $A$  in  $\text{TS}_{\text{BN}}$ 
4:    $\text{isMin} := \text{false}$ 
5:   while  $\text{isMin} = \text{false}$  and  $\text{WB} \neq \emptyset$  do
6:      $\mathbf{C} \in \arg \text{hd}(\mathbf{s}, \text{WB}), \mathbf{s}' = \mathbf{C}(\mathbf{s})$  % a possible minimal control from  $\mathbf{s}$  to  $A$ 
7:     if  $\mathbf{s}' \in \text{SB}$  then
8:        $\text{isMin} := \text{true}$  % instantaneous perturbation
9:     else
10:       $\mathbf{f}_{\mathbf{C}} := \text{COMP\_FN\_CONTR}(\mathbf{f}, \mathbf{C})$  %  $\mathbf{f}$  in  $\text{BN}|_{\mathbf{C}}$  (see Algorithm 2)
11:       $\text{SB}|_{\mathbf{C}} := \text{COMP\_STATE\_CONTR}(\text{SB}, \mathbf{C})$  %  $\text{SB}$  in  $\text{TS}_{\text{BN}|_{\mathbf{C}}}$  (see Algorithm 2)
12:      if  $\text{SB}|_{\mathbf{C}} \neq \emptyset$  then
13:         $\text{bas}_{\text{BN}|_{\mathbf{C}}}^{\mathbf{S}}(\text{SB}|_{\mathbf{C}}) := \text{COMP\_SB}(\text{SB}|_{\mathbf{C}}, \mathbf{f}_{\mathbf{C}})$ 
14:        if  $\mathbf{s}' \in \text{bas}_{\text{BN}|_{\mathbf{C}}}^{\mathbf{S}}(\text{SB}|_{\mathbf{C}})$  then
15:           $\text{isMin} := \text{true}$ 
16:        if  $\text{isMin} = \text{false}$  then
17:           $\text{WB} := \text{WB} \setminus \{\mathbf{s}'\}$ 
18:   return  $\mathbf{C}$ 

```

$\text{bas}_{\text{BN}}^{\mathbf{W}}(A)$ that has the minimal Hamming distance with \mathbf{s} . We check if $\mathbf{s}' \in \text{bas}_{\text{BN}|_{\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}'}}}^{\mathbf{S}}(\text{bas}_{\text{BN}}^{\mathbf{S}}(A)|_{\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}'}})$. If so, we are done. Otherwise, we remove \mathbf{s}' from $\text{bas}_{\text{BN}}^{\mathbf{W}}(A)$ and select a state \mathbf{s}'' from $(\text{bas}_{\text{BN}}^{\mathbf{W}}(A) \setminus \{\mathbf{s}'\})$ having the minimal Hamming distance with \mathbf{s} . We repeat the same procedure this time with $\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}''}$. We iterate till we find a state $\mathbf{s}^* \in \text{bas}_{\text{BN}}^{\mathbf{W}}(A)$ such that $\mathbf{s}^* \in \text{bas}_{\text{BN}|_{\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}^*}}}^{\mathbf{S}}(\text{bas}_{\text{BN}}^{\mathbf{S}}(A)|_{\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}^*}})$. Once again, by Proposition 1 we know that for any control \mathbf{C} , $\text{bas}_{\text{BN}|_{\mathbf{C}}}^{\mathbf{W}}(A) \subseteq \text{bas}_{\text{BN}}^{\mathbf{W}}(A)$. Thus, it is enough to explore only the states in $\text{bas}_{\text{BN}}^{\mathbf{W}}(A)$ and it will eventually find the required control. $\mathbf{C}^{\mathbf{s} \rightarrow \mathbf{s}^*}$ is then the required minimal temporary control. Algorithm 3 describes this procedure in pseudo-code.

Notice that the amount of time for the application of \mathbf{C} depends on the specific system and the detailed perturbations. Biologists can determine when to release the control case by case based on experimental settings. As long as the control is released in finite steps, holding it longer will not affect its effectiveness.

Constraints on the control sets. Constraints encoding practical requirements can eliminate perturbations of certain nodes, for instance essential genes for cell survival [26]. We implement the constraints by slightly modifying our algorithms as follows. Let $\mathbf{R}_1, \mathbf{R}_0$ be two sets of indices of nodes, where the state of a node with index $i \in \mathbf{R}_1(\mathbf{R}_0)$ cannot be perturbed to 0 (1). The above constraints can be realised by removing the states $\{\mathbf{s}' \in \text{bas}_{\text{BN}}^{\mathbf{W}}(A) \mid \text{for } i \in \mathbf{R}_1, \mathbf{s}'[i] = 0 \text{ or for } j \in \mathbf{R}_0, \mathbf{s}'[j] = 1\}$ from $\text{bas}_{\text{BN}}^{\mathbf{W}}(A)$ before the main loop of Algorithms 1 and 3.

network	# nodes	# edges	#A	range of C			time (seconds)		
				C _I	C _T	C _P	T _I	T _T	T _P
myeloid	11	30	6	1-5	1-3	1-3	0.015	0.059	0.056
cardiac	15	39	6	1-9	1-4	1-4	0.233	0.885	0.842
ERBB	20	52	3	1-9	1-3	1-3	0.054	0.179	0.251
tumour	32	158	9	1-10	1-4	1-4	6.726	35.207	34.065
PC12	33	62	7	1-11	1-4	1-4	0.394	2.150	2.634
hematopoietic	33	88	5	1-13	1-3	1-3	1.749	11.356	16.080
MAPK_r3	53	105	20	1-19	1-5	1-5	112.429	213.111	230.871
HGF	66	103	18	1-31	1-5	1-5	234.373	441.541	417.897
bortezomib	67	135	5	1-21	1-3	1-4	46.062	145.111	106.268
CD4 ⁺	188	380	12	1-5	1-4	1-4	8536.420	15930.900	16007.500

Table 1: An overview of the networks and the evaluation results. I, T and P stand for the instantaneous, temporary and permanent controls, respectively.

	A ₁			A ₂			A ₃			A ₄			A ₅			A ₆		
	C _I	C _T	C _P	C _I	C _T	C _P	C _I	C _T	C _P	C _I	C _T	C _P	C _I	C _T	C _P	C _I	C _T	C _P
A ₁	–	–	–	1	1	1	1	1	1	2	2	2	1	1	1	2	2	2
A ₂	2	1	1	–	–	–	4	2	3	1	1	1	1	1	1	2	2	2
A ₃	1	1	1	2	2	2	–	–	–	1	1	1	2	2	2	1	1	1
A ₄	4	2	2	1	1	1	3	1	2	–	–	–	2	2	2	1	1	1
A ₅	8	3	3	6	2	2	9	4	4	6	3	3	–	–	–	1	1	1
A ₆	8	4	4	7	3	3	6	2	2	4	2	2	1	1	1	–	–	–

Table 2: The number of perturbations computed for the cardiac network.

4 Case Studies

To demonstrate the efficacy and efficiency of our control methods, we apply our minimal temporary and permanent controls (Algorithms 1 and 3) to 10 biological networks [20, 27–33, 29, 34]. The results are compared with the minimal instantaneous control method developed in [19]. These three algorithms are implemented as part of the software ASSA-PBN [23]. All the experiments are performed on a computer with a CPU of Intel Core i7 @3.1 GHz and 8 GB of DDR3 RAM. An overview of the networks is given in Table 1.

Efficiency. We compute a minimal control C with the three methods for each pair of source and target attractors. The total execution time is summarised in the last three columns of Table 1. We do not give detailed time costs for each pair due to the page limit. The temporary and permanent controls have similar performance in terms of efficiency. Both of them are less efficient than the instantaneous control as it may take several iterations to compute a valid control set (see Algorithms 1 and 3). Despite that, both methods are still very efficient. For instance, for the CD4⁺ T-cell network, the computation time for each case is in the range of (38-212), (84-581) and (63-718) seconds for the instantaneous, temporary and permanent controls, respectively.

Efficacy. The number of perturbations are summarised in Table 1. By extending the application time of control, the number of perturbations can be greatly

	Megakaryocyte	Erythrocyte	Granulocyte	Monocyte
Megakaryocyte	–	{EKL <u>F</u> } { <u>Fli1</u> }	{C/EBP α , PU1, Gfi1} {C/EBP α , Gfi1, <u>GATA1</u> }	{EgrNab, C/EBP α , <u>GATA1</u> } {EgrNab, C/EBP α , PU1}
Erythrocyte	{EKL <u>F</u> } { <u>Fli1</u> }	–	{C/EBP α , PU1, Gfi1} {C/EBP α , Gfi1, <u>GATA1</u> }	{EgrNab, C/EBP α , <u>GATA1</u> } {EgrNab, C/EBP α , PU1}
Granulocyte	{GATA2, <u>Fli1</u> } {GATA1, <u>Fli1</u> } { <u>Fli1</u> , <u>PU1</u> }	{GATA2, EKL <u>F</u> } {GATA1, EKL <u>F</u> }	–	{EgrNab} { <u>Gfi1</u> }
Monocyte	{GATA2, <u>Fli1</u> } {GATA1, <u>Fli1</u> } { <u>Fli1</u> , <u>PU1</u> }	{GATA2, EKL <u>F</u> } {GATA1, EKL <u>F</u> }	{ <u>PU1</u> } {Gfi1}, { <u>cJun</u> }, {EgrNab}	–

Table 3: The control sets for the myeloid differentiation network computed by the temporary and permanent controls. The sets in grey are only required by the temporary control. Underlined genes are switched to OFF, otherwise to ON.

reduced, which in turn reduces the cost of biological experiments. Especially for the model of bortezomib responses, the number of perturbations can be reduced from 21 to 3 and 4 by the temporary and permanent controls, respectively.

Table 2 summarises the number of perturbations for the cardiac network. The first column and the first row represent the source and target attractors, respectively. For most cases, the temporary and permanent controls require the same number of perturbations, while the temporary control can further reduce it for a few cases (e.g., $A_2 \rightarrow A_3$ and $A_4 \rightarrow A_3$). In general, the instantaneous control needs to control more nodes to guarantee the reachability.

We take the myeloid differentiation network as an example to compare our results with the perturbations found in [20]. Four of the six attractors of this network correspond to megakaryocytes, erythrocytes, granulocytes or monocytes. Table 3 gives the minimal control sets computed by our temporary and permanent controls for these four attractors. In general, our results are consistent with the conclusions of [20].

1. Reprogramming of EKLF or Fli1 can achieve a conversion between erythrocytes and megakaryocytes [20].
2. Simultaneous perturbations of C/EBP α and PU1 can convert the network from MegE lineage (megakaryocytes and erythrocytes) to GM lineage (granulocytes and monocytes) [20], but to reach a specific state (granulocyte or monocyte), one more gene is required. Besides, our methods also identified other paths to realise the reprogramming.
3. We also spotted the pivotal role of GATA1 and GATA2 in the transdifferentiation from GM lineage to MegE lineage [20]. However, the over-expression of GATA2 can only be applied with temporary perturbations, since the permanent over-expression of GATA2 leads to absence of MegE lineage.

5 Discussion and Future Work

In this paper, we have developed the temporary and permanent control methods to identify a minimal set of nodes, such that the temporary or permanent

perturbations of these nodes guide the network from a source state to a desired target attractor. Together with the instantaneous control [19], we have been working on bridging the gap between computational control methods and practical reprogramming of GRNs from three perspectives.

First, we have explored three kinds of perturbations: instantaneous, temporary and permanent perturbations. All of them are feasible to conduct in biological experiments. Besides that, each kind of perturbations has its own merits and demerits. (1) Permanent perturbations have a permanent influence on the dynamics of the system and thus are more invasive than instantaneous and temporary perturbations. (2) Temporary and permanent perturbations alter the dynamics of networks either for sufficient time or permanently, thus less number of perturbations are required to achieve the goal compared to instantaneous perturbations. Indeed, there is no universal standard of good perturbations for different biological networks. We provide control methods for different kinds of perturbations, so that biologists can choose suitable strategies to deal with different biological networks.

Second, considering the expensive cost and other difficulties in performing biological experiments, practical constraints are encoded to make our control strategies more realistic and applicable. Two problems commonly arise in biological experiments. One is that some genes are essential for cell survival and thus cannot be turned off. The other is that some genes are harder or more expensive to perturb. For instance, GATA1 and GATA2 are part of a family of transcription factors that may have different functions but may have similar structural features to be recognised by the ‘perturbation tool’. Hence, we adapt our methods to avoid (1) perturbing certain nodes from ‘expressed’ to ‘not expressed’ and/or (2) perturbing certain nodes from ‘not expressed’ to ‘expressed’.

Third, so far we have focused on identifying a minimal set of perturbations to fulfil the control purpose with 100% success rate. Apart from that, given an upper bound of perturbations, our methods can compute all the control sets within the upper bound efficiently for different kinds of perturbations. Incorporated with practical constraints, our methods can compute a rich set of restricted solutions, which will be beneficial to biological applications.

Currently, we are working a sequential control method, where other attractors (observable biological phenotypes) can act as intermediates [35]. We want to drive the network from a source state to a target attractor through intermediate attractors by applying a sequence of instantaneous or temporary or permanent perturbations. Such a sequential method can provide more potential reprogramming solutions and may further reduce the number of required perturbations. Other than that, we plan to extend our work to the control of probabilistic Boolean networks (PBNs) [36, 37] and explore if and how to adapt the instantaneous, temporary and permanent control strategies to such networks.

Acknowledgements. The work was partially supported by the research project SEC-PBN funded by the University of Luxembourg and the ANR-FNR project AlgoReCell (INTER/ANR/15/11191283). We also want to thank Loïc Paulevé for discussions.

References

1. Takahashi, K.: Cellular reprogramming. *Cold Spring Harbor Perspectives in Biology* **6**(2) (2014) a018606
2. Sol, A.d., Buckley, N.: Concise review: A population shift view of cellular reprogramming. *Stem Cells* **32**(6) (2014) 1367–1372
3. Graf, T., Enver, T.: Forcing cells to change lineages. *Nature* **462**(7273) (2009) 587–594
4. Srivastava, D., DeWitt, N.: In vivo cellular reprogramming: the next generation. *Cell* **166**(6) (2016) 1386–1396
5. Wang, L.Z., Su, R.Q., Huang, Z.G., Wang, X., Wang, W.X., Grebogi, C., Lai, Y.C.: A geometrical approach to control and controllability of nonlinear dynamical networks. *Nature Communications* **7** (2016) 11323
6. Zañudo, J.G.T., Albert, R.: Cell fate reprogramming by control of intracellular network dynamics. *PLoS Computational Biology* **11**(4) (2015) e1004193
7. Kauffman, S.A.: Homeostasis and differentiation in random genetic control networks. *Nature* **224** (1969) 177–178
8. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. *Nature* **473** (2011) 167–73
9. Gao, J., Liu, Y.Y., D’Souza, R.M., Barabási, A.L.: Target control of complex networks. *Nature Communications* **5** (2014) 5415
10. Czeizler, E., Gratie, C., Chiu, W.K., Kanhaiya, K., Petre, I.: Target controllability of linear networks. In: *Proc. 14th Conference on Computational Methods in Systems Biology*. Volume 9859 of LNCS., Springer (2016) 67–81
11. Bornholdt, S.: Less is more in modeling large genetic networks. *Science* **310**(5747) (2005) 449–451
12. Mochizuki, A., Fiedler, B., Kurosawa, G., Saito, D.: Dynamics and control at feedback vertex sets. II: A faithful monitor to determine the diversity of molecular activities in regulatory networks. *Journal Theoretical Biology* **335** (2013) 130–146
13. Zañudo, J.G.T., Yang, G., Albert, R.: Structure-based control of complex networks with nonlinear dynamics. *Proceedings of the National Academy of Sciences* **114**(28) (2017) 7234–7239
14. Kim, J., Park, S., Cho, K.: Discovery of a kernel for controlling biomolecular regulatory networks. *Scientific Reports* **3**(2223) (2013)
15. Zhao, Y., Kim, J., Filippone, M.: Aggregation algorithm towards large-scale Boolean network analysis. *IEEE Transactions on Automatic Control* **58**(8) (2013) 1976–1985
16. Papin, J.A., Hunter, T., Palsson, B.O., Subramaniam, S.: Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology* **6**(2) (2005) 99
17. Cornelius, S.P., Kath, W.L., Motter, A.E.: Realistic control of network dynamics. *Nature Communications* **4**(1942) (2013)
18. Germini, D., Tsfasman, T., Zakharova, V.V., Sjakste, N., Lipinski, M., Vassetzky, Y.: A comparison of techniques to evaluate the effectiveness of genome editing. *Trends in Biotechnology* **36**(2) (2018) 147–159
19. Paul, S., Su, C., Pang, J., Mizera, A.: A decomposition-based approach towards the control of Boolean networks. In: *Proc. 9th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, ACM Press (2018) 11–20
20. Krumsiek, J., Marr, C., Schroeder, T., Theis, F.J.: Hierarchical differentiation of myeloid progenitors is encoded in the transcription factor network. *PLOS ONE* **6**(8) (2011) e22649

21. Müller, F., Schuppert, A.: Few inputs can reprogram biological networks. *Nature* **478**(7369) (2011) E4
22. Takahashi, K., Yamanaka, S.: Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell* **126**(4) (2006) 663–676
23. Mizera, A., Pang, J., Su, C., Yuan, Q.: ASSA-PBN: A toolbox for probabilistic Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **15**(4) (2018) 1203–1216
24. Zhu, P., Han, J.: Asynchronous stochastic Boolean networks as gene network models. *Journal of Computational Biology* **21**(10) (2014) 771–783
25. Paul, S., Su, C., Pang, J., Mizera, A.: An efficient approach towards the source-target control of Boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2019) accepted.
26. Zhang, R., Lin, Y.: Deg 5.0, a database of essential genes in both prokaryotes and eukaryotes. *Nucleic Acids Research* **37**(Database issue) (2008) D455–D458
27. Herrmann, F., Gro, A., Zhou, D., Kestler, H.A., Köhl, M.: A Boolean model of the cardiac gene regulatory network determining first and second heart field identity. *PLOS ONE* **7** (10 2012) 1–10
28. Sahin, Ö., Fröhlich, H., Löbke, C., Korf, U., Burmester, S., Majety, M., Mattern, J., Schupp, I., Chaouiya, C., Thieffry, D., et al.: Modeling ERBB receptor-regulated G1/S transition to find novel targets for de novo trastuzumab resistance. *BMC Systems Biology* **3**(1) (2009) 1
29. Cohen, D.P.A., Martignetti, L., Robine, S., Barillot, E., Zinovyev, A., Calzone, L.: Mathematical modelling of molecular pathways enabling tumour cell invasion and migration. *PLoS Computational Biology* **11**(11) (2015) e1004571
30. Offermann, B., Knauer, S., Singh, A., Fernández-Cachón, M.L., Klose, M., Kowar, S., Busch, H., Boerries, M.: Boolean modeling reveals the necessity of transcriptional regulation for bistability in PC12 cell differentiation. *Frontiers in Genetics* **7** (2016) 44
31. Collombet, S., van Oevelen, C., Ortega, J., Abou-Jaoudé, W., Di Stefano, B., Thomas-Chollier, M., Graf, T., Thieffry, D.: Logical modeling of lymphoid and myeloid cell specification and transdifferentiation. *Proceedings of the National Academy of Sciences* **114**(23) (2017) 5792–5799
32. Grieco, L., Calzone, L., Bernard-Pierrot, I., Radvanyi, F., Kahn-Perles, B., Thieffry, D.: Integrative modelling of the influence of MAPK network on cancer cell fate decision. *PLoS Computational Biology* **9**(10) (2013) e1003286
33. Singh, A., Nascimento, J.M., Kowar, S., Busch, H., Boerries, M.: Boolean approach to signalling pathway modelling in HGF-induced keratinocyte migration. *Bioinformatics* **28**(18) (2012) 495–501
34. Conroy, B.D., Herek, T.A., Shew, T.D., Latner, M., Larson, J.J., Allen, L., Davis, P.H., Helikar, T., Cutucache, C.E.: Design, assessment, and in vivo evaluation of a computational model illustrating the role of CAV1 in CD4+ T-lymphocytes. *Frontiers in Immunology* **5** (2014) 599
35. Mandon, H., Su, C., Haar, S., Pang, J., Paulevé, L.: Sequential reprogramming of Boolean networks made practical. In: *Proc. 17th International Conference on Computational Methods in Systems Biology*. LNCS, Springer-Verlag (2019) accepted.
36. Shmulevich, I., Dougherty, E.R.: *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. SIAM Press (2010)
37. Trairatphisan, P., Mizera, A., Pang, J., Tantar, A.A., Schneider, J., Sauter, T.: Recent development and biomedical applications of probabilistic Boolean networks. *Cell Communication and Signaling* **11** (2013) 46