CABEAN 2.0: Efficient and Efficacious Control of Asynchronous Boolean Networks

Cui Su[1] and Jun Pang[2,1][0000−0002−4521−4112]

[1] Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, Esch-sur-Alzette, Luxembourg
[2] Faculty of Science, Technology and Medicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg
firstname.lastname@uni.lu

Abstract. We present a new version of the software, CABEAN, integrating six source-target control methods and three target control methods for the reprogramming of asynchronous Boolean networks. The source-target control methods compute the minimal one-step and sequential control strategies that can guide the dynamics of a Boolean network from a source attractor to the desired target attractor with instantaneous, temporary, or permanent perturbations. The target control methods further identify efficacious interventions that can drive the network from any initial state to the desired target attractor with these three types of perturbations. These control methods have been applied to various real-life biological networks to demonstrate their efficacy and efficiency.

Keywords: Boolean networks · cell reprogramming · network control · target control · attractors · software tool.

1 Introduction

Cell reprogramming harnesses the power of somatic cells to treat diseases featured by a deficiency of certain cells or diseased cells [12,44,11]. It reprograms abundant somatic cells to deficient or damaged cells, in order to restore functions of diseased organs in the human body. Therefore, cell reprogramming sheds light on the development of tissue engineering and regenerative medicine.

One of the significant hurdles for the application of cell reprogramming lies in the identification of efficacious intervention targets, whose perturbations can engender desired changes. Experimental approaches select promising combinations of targets, perturb them and monitor if the perturbation triggers desired changes. Such "trial and error" approaches can be costly and require long-time commitment, which render them inefficient [9]. Advances in sequencing techniques and the availability of wealthy data on gene expression profiles promote the shift from experimental approaches to the computational predictions based on mathematical modelling of biological systems. Mathematical models allow us to discover different combinations of targets by providing a broad view of the whole biological system. In this way, we can systematically make predictions

and speed up the development of cell reprogramming, as such predictions are much faster and cheaper than experimental approaches. Moreover, it has great promise to discover novel intervention targets for cell reprogramming.

Several modelling frameworks have been developed for modelling biological systems, and Boolean networks are chosen as the representative of biological networks, thanks to its simplicity and qualitative nature with the ability of dealing with large-scale networks. Boolean networks, first introduced by Kauffman [14], are a well-established modelling framework for gene regulatory networks and their associated signalling pathways. In Boolean networks, molecular species, such as genes and transcription factors, are described as Boolean variables. Each Boolean variable is associated with a Boolean function, which determines the evolution of the variable. Boolean functions characterise activation or inhibition regulations between the molecular species (i.e., nodes in the networks). The states of a Boolean network are binary strings, where every bit of the string represents the state of a molecular species – '0' for inactive (or absent) and '1' for active (or present). The dynamics of a Boolean network is assumed to evolve in discrete time steps, moving from one state to the next, under one of the updating schemes, such as the *synchronous* or *asynchronous* updating scheme. Under the synchronous scheme, all the variables update their values simultaneously at each time step, while under the asynchronous scheme, only one variable is randomly selected to update its value at each time step. The asynchronous updating scheme is considered more realistic than the synchronous one, since it captures the phenomenon that biological processes occur at different classes of time scales [49]. Therefore, we focus on asynchronous Boolean networks. The steady-state behaviour of a Boolean network is described as *attractors*, to one of which the system eventually settles down. Attractors are hypothesised to characterise cellular phenotypes [13].

In the context of Boolean networks, cell reprogramming amounts to driving the network dynamics from a source state to a desired attractor of the network. The realisation of the transition needs to respect certain constraints to ensure the feasibility of the identified perturbations. The original version of the software CABEAN [36] supports six methods [32,33,40,24,38] for the *source-target control* of Boolean networks — to identify one-step or sequential control strategies that guide the network dynamics from a source attractor to a target attractors with instantaneous, temporary or permanent perturbations. The source-target control methods implemented in CABEAN guarantee the minimality of the identified control strategies in terms of the number of required perturbations, even though this makes the control problem sometimes computationally difficult [22,23],

In this paper, we present an extension and a new version of CABEAN with three new appealing methods for *target control* of Boolean networks — to identify a set of nodes, whose perturbations drive the network from *any initial state* to the desired target attractor. The motivation of target control is that cells typically exist as a mixture of different cell types. There is a surge to identify effective perturbations that can reprogram any cells to the desired cell type. Another new functionality of CABEAN 2.0 is that it allows users of the software tool to encode

a phenotype based on the expressions of a subset of nodes (the marker nodes). The phenotype serves as the desired target, which is considered more realistic in biological experiments than specifying a target attractor. CABEAN 2.0 is able to identify the target control strategies towards the desired phenotype, which may correspond to one or more attractors of a given Boolean network.

Structure of the paper. After the introduction, methods for controlling complex networks and Boolean networks, and related software tools are summarised in Section 2. Section 3 contains preliminary notions of asynchronous Boolean networks. Section 4 continues to present the control methods implemented in the tool CABEAN, with a focus on the newly extended functionalities. Evaluation of the three new methods for the target control of Boolean networks on a number of real-life biological networks is presented in Section 5. In Section 6, implementation details of CABEAN are given and an example is also given to illustrate the usage of the new functionalities of the software tool. Section 7 concludes the paper with future developments of the tool.

2 Related Work

Control methods for complex networks. Several important methods have been developed for the control of complex networks [19,27,7,4,10,6,42,47]. However, these methods do not directly apply to Boolean networks. Methods based on the semi-tensor product (STP) have been proposed to solve different control problems for Boolean control networks (BCNs) under the synchronous updating scheme [17,52,21,50,43,3,45,48]. For synchronous Boolean networks, Kim *et al.* [15] developed a method to compute a small fraction of nodes, called "control kernels", whose modulation can govern the dynamics of the network; and Moradi *el al.* [28] developed an algorithm guided by forward dynamic programming to solve the control problem. Lin *et al.* [18] proposed a Max-SAT based automatic test pattern generate to identify faulty genes that cause undesired behaviours of GRNs and to identify the best drug selection for cancer treatment. Their algorithm considers synchronous Boolean networks under a stuck-at fault model. Murrugarra *el al.* [30] proposed a method for identifying intervention targets based on algebraic techniques for synchronous Boolean networks. None of the above-mentioned methods is applicable to asynchronous Boolean networks.

Control methods for asynchronous Boolean networks. Recently, Mandon *et al.* [22,23] proposed several methods to encode all possible control strategies into the transition system for the control of asynchronous Boolean networks. The size of the resulting *perturbed transition graph* grows exponentially in the number of perturbations, which renders these methods inefficient. The algorithm Kali [34] predicts perturbations to reduce the reachability of undesired attractors that are linked to pathological phenotypes for both synchronous and asynchronous Boolean networks. However, Kali can only estimate the attractors and their basins of a Boolean network in an approximate way. Therefore, the predictions of Kali might not be fully accurate. Fontanals *et al.* [8] proposed a

method based on trap space to deal with the temporary target control of asynchronous Boolean networks. This method requires the preservation of the target attractors during control, which could be eased since the control will eventually be released to retrieve the original transition system where the desired attractor is in. The stable motifs-based control method (SMC) [46] predicts a set of transient perturbations that can guide the dynamics from any initial states to the desired target attractor. Based on the functional information of the network, SMC has a substantial improvement in computing the number of perturbations, but it does not guarantee to find the minimal number of perturbations. Details on comparing SMC with the target control methods implemented in CABEAN 2.0 can be found in [37,39]. In general, the target control methods CABEAN 2.0 are more efficient and can produce more and effective control strategies than SMC.

Software tools. A number of software tools have been developed for the analysis of logical models of Biological networks. ACTONETLIB [1] implements a method based on abductive reasoning to identify a minimal set of causal topological actions that cause expected changes at stable states for BCNs. The caspo toolbox [41], CANA [5] and PyBoolNet [16] are all Python packages. In particular, the caspo toolbox [41] provides a work-flow to study logical networks families of three-valued semantics under the synchronous updating scheme. CANA [5] focuses on quantifying redundancy and control of synchronous Boolean networks, PyBoolNet [16] integrates methods for manipulating Boolean networks, such as generation, visualisation, and attractor detection. BoolNet [31] is a powerful R package, which provides functions for reconstruction, generalisation, and attractor identification for synchronous, asynchronous, and probabilistic Boolean networks. Although BoolNet and PyBoolNet can handle asynchronous Boolean networks, neither of them supports the identification of intervention targets for modulating the dynamics.

3 Preliminaries

A *Boolean network* is a tuple $\mathsf{BN} = (X, F)$, where $X = \{x_1, x_2, \ldots, x_n\}$, such that $x_i \in X$ is a Boolean variable and $F = \{f_1, f_2, \ldots, f_n\}$ is a set of Boolean functions over X. A *state* s of BN is an element in $\{0,1\}^n$. Let $S = \{0,1\}^n$ denote the set of all states of BN. For two states $s, s' \in S$, the *Hamming distance* between s and s' will be denoted as $hd(s, s')$ and $\arg(hd(s, s')) \subseteq \{1, 2, \ldots, n\}$ will denote the set of indices in which s and s' differ. These two notions can be lifted to a set of states. For two subsets $S', S'' \subseteq S$, the Hamming distance between S' and S'' is defined as the minimum of the Hamming distances between all the states in S' and all the states in S''. We let $\arg(hd(S', S''))$ denote the set of subsets of $\{1, 2, \ldots, n\}$ such that $I \in \arg(hd(S', S''))$ if and only if I is a set of indices of the variables that realise this Hamming distance.

We assume that a Boolean network $\mathsf{BN} = (X, F)$ evolves in discrete time steps. It starts from an initial state and its state changes in every time step based on the Boolean functions F and the updating schemes. Different updating schemes lead to different dynamics of the network [26,51]. Suppose $s_0 \in S$ is an

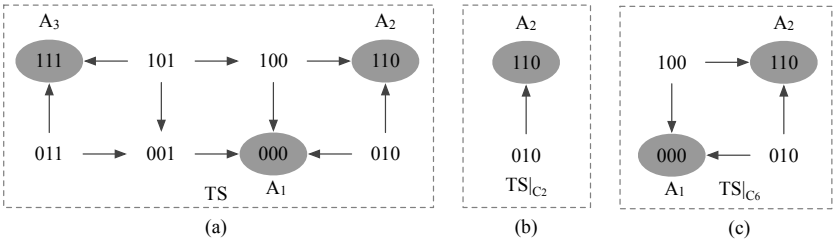

Fig. 1: (a) The transition system for Example 1; (b) the transition system under control $\mathsf{C}_2 = \{x_2 = 1, x_3 = 0\}$ for Example 2; and (c) the transition system under control $\mathsf{C}_6 = \{x_3 = 0\}$ for Example 3.

initial state of BN. We use $\wp(S)$ to denote the power set of S. The *asynchronous evolution* of BN is a function $\xi_{\mathsf{BN}} : \mathbb{N} \to \wp(S)$ such that $\xi_{\mathsf{BN}}(0) = \{s_0\}$ and for every $j \geq 0$, if $s \in \xi_{\mathsf{BN}}(j)$ then $s' \in \xi_{\mathsf{BN}}(j+1)$ is a possible *next state* of s iff either $hd(s, s') = 1$ and there exists i such that $s'[i] = f_i(s) = 1 - s[i]$, or $hd(s, s') = 0$ and there exists i such that $s'[i] = f_i(s) = s[i]$. It is worth noting that the asynchronous dynamics is non-deterministic. At each time step, only one node is randomly selected to update its value based on its Boolean function. A different choice may lead to a different next state $s' \in \xi(j+1)$. The *transition system* of a Boolean network BN, denoted as TS, is a tuple $(S, \rightarrow_{\mathsf{BN}})$, where the vertices are the set of states S and for any two states s and s' there is a directed edge from s to s', denoted $s \to s'$, iff s' is a possible next state of s according to the asynchronous evolution function ξ of BN.

A *path* ρ from a state s to a state s' is a (possibly empty) sequence of transitions from s to s' in TS. Thus, $\rho = s_0 \to s_1 \to \ldots \to s_k$, where $s_0 = s$ and $s_k = s'$. A path from a state s to a subset S' of S is a path from s to any state $s' \in S'$. For a state $s \in S$, $reach_{TS}(s)$ denotes the set of states s' such that there is a path from s to s' in TS and can be defined as the fixed point of the successor operation which is often denoted as $post^*_{TS}$. An *attractor* A of TS is a minimal non-empty subset of states of S such that for every $s \in A, reach_{TS}(s) = A$.

Attractors are hypothesised to characterise steady-state behaviours of the network. Any state, which is not in an attractor, is a *transient state*. An attractor A of TS is said to be *reachable* from a state s if $reach(s) \cap A \neq \emptyset$. The network starting from any initial state $s_0 \in S$ will eventually end up in one of the attractors of TS and remain there forever unless perturbed externally. Thus, it is easy to observe that any attractor of TS is a *bottom strongly connected component* of TS. Each attractor has a *weak basin* and a *strong basin*. The weak basin of an attractor A with respect to TS is defined as $bas^W_{TS}(A) = \{s \in S \mid reach_{TS}(s) \cap A \neq \emptyset\}$, which equals the fixed point of the predecessor operation on A and is often denoted as $pre^*_{TS}(A)$. The strong basin of A with respect to TS is defined as $bas^S_{TS}(S') = \{s \in S \mid \forall \rho = s_0 \to s_1 \to \ldots \in P_\infty(s), \exists j \geq 0, s_j \in S'\}$.

Intuitively, the weak basin of A includes all the states from which there exists a path to A. It is possible that there also exist paths from a state in the weak basin of A to some other attractor $A' \neq A$ of TS. However, the notion of the strong basin of an attractor does not allow this. Any path from a state in the strong basin of A will eventually reach A and cannot reach any other distinct attractor $A' \neq A$ of TS.

Example 1. To illustrate the notions of Boolean networks, let us consider a Boolean network $\mathsf{BN} = (X, F)$, where $X = \{x_1, x_2, x_3\}$, $F = \{f_1, f_2, f_3\}$, and $f_1 = x_2$, $f_2 = x_1$ and $f_3 = x_2 \wedge x_3$. Its transition system under the asynchronous updating is given in Fig. 1(a) with self-loops omitted. This network consists of three attractors A_1, A_2 and A_3, plotted as grey nodes. For attractor A_1, its strong basin consists of two states (000) and (001). The weak basin of A_1 includes six states, $\{000, 001, 101, 011, 100, 010\}$ and from any of these states, there exists at least one path to A_1.

4 Functionalities

We describe the main functionalities of the software tool CABEAN, implementing six source-target control methods and three target control methods for the reprogramming of asynchronous Boolean networks. First, we define what types of control, in terms of node perturbations, can be applied to Boolean networks. We then proceed with briefly describing the main ideas of the implemented control methods.

Control in Boolean networks. Let BN be a given Boolean network, S be the set of states of BN and $\mathcal{A}$ be the set of attractors of BN. A *control strategy* (*control* for short) C is a tuple $\mathsf{C} = (\mathbb{0}, \mathbb{1})$, where $\mathbb{0}, \mathbb{1} \subseteq \{1, 2, \ldots, n\}$ and $\mathbb{0}$ and $\mathbb{1}$ are mutually disjoint (possibly empty) sets of indices of nodes of a Boolean network BN. The application of a control C inhibits the nodes in $\mathbb{0}$, i.e., the values of these nodes are set (perturbed) to be 0, and overexpresses the nodes in $\mathbb{1}$, i.e., the values of these nodes are set (perturbed) to be 1. Formally, given a state $s \in S$, the *application of a control* C to a state s, denoted $\mathsf{C}(s)$, is defined as the state $s' \in S$ such that $s'[i] = 0$ for $i \in \mathbb{0}$ and $s'[i] = 1$ for $i \in \mathbb{1}$. State s' is called the intermediate state w.r.t. C.

The control can be applied to the network for different periods of time: (a) *instantaneous control* — the control is applied instantaneously (only one time step); (b) *temporary control* — the control is applied for a finite number of time steps and then released; and (c) *permanent control* — the control is applied for all the following time steps, i.e., the parameters are changed for all the following steps. Next, we formulate the problems of *source-target control* and *target control* of Boolean networks.

Source-target control. Given a source attractor A_s and a target attractor A_t of TS, to drive the network from A_s to A_t is called *source-target control*.

A source-target control C is called *one-step source-target control* if C drives the network from A_s to A_t in one step as shown in Fig. 2a. When the control

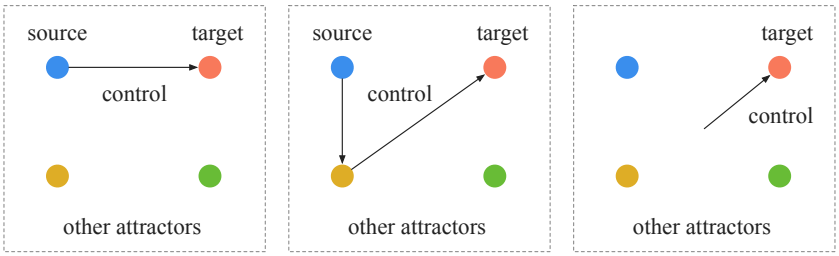

Fig. 2: Schematic illustration for source-target control and target control of Boolean networks: (a) one-step source-target control; (b) sequential source-target control; and (c) target control.

C is the instantaneous, temporary or permanent control, we call it respectively *one-step instantaneous, temporary or permanent control* (OI, OT, or OP). To minimise experimental costs, it is often important to find the minimal solution, denoted as $\mathsf{C}_{\min}^{A_s \to A_t}$, which solves the source-target control of driving the network from A_s to A_t. The minimal OI control [32,33] perturbs the network from s in A_s to a state s' in the strong basin of A_t, denoted $bas_{TS}^{S}(A_t)$, where s and s' are a pair of states that realise the minimal Hamming distance between A_s and $bas_{TS}^{S}(A_t)$.

The minimal OT (temporary) and OP (permanent) controls [40] have extended effects on the network dynamics. The minimal OT control explores the weak basin of A_t, $bas_{TS}^{W}(A_t)$ from the one that has the shortest Hamming distance to A_s. It searches for a state s', such that after applying the corresponding control $\mathsf{C}^{A_s \to s'}$ for a sufficient period of time, BN will surely reach a state s'' in $bas_{TS}^{S}(A_t)$. Once BN is in s'', $\mathsf{C}^{A_s \to s'}$ can be released. To apply $\mathsf{C}^{A_s \to s'}$ will not harm the inevitable reachability of A_t since there only exist paths to A_t from s''. The minimal OP control looks for a state s' in $bas_{TS}^{W}(A_t)$, such that the application of the control $\mathsf{C}^{A_s \to s'}$ preserves A_t and s' can only reach A_t and cannot reach any other attractors.

Besides the one-step source-target control, to identify a sequence of perturbations to drive the network from A_s to A_t in a stepwise manner is called *sequential control*. As shown in Fig. 2b, it is more practical to concentrate on the sequential control through other attractors, called *attractor-based sequential control*: to find a sequence of attractors of TS, i.e. $\Phi = \langle A_1, A_2, \ldots, A_m \rangle$, where $A_1 = A_s$, $A_m = A_t$, $A_i \neq A_j$ for any $i, j \in [1, m]$ and $2 \leq m \leq |\mathcal{A}|$, such that after the application of a sequence of minimal one-step controls $\langle \mathsf{C}_{\min}^{A_1 \to A_2}, \mathsf{C}_{\min}^{A_2 \to A_3}, \ldots, \mathsf{C}_{\min}^{A_{m-1} \to A_m} \rangle$, the network always eventually reaches A_m, i.e. A_t. Similarly, when the control $\mathsf{C}_{\min}^{A_i \to A_j}$, $A_i, A_j \in \Phi$ is the instantaneous, temporary or permanent control, we call it *attractor-based sequential instantaneous, temporary or permanent control* (ASI, AST or ASP), respectively. The ASI, AST and ASP control methods [24,38] are based on the three one-step control methods [32,33,40]. Given a source attrac-

tor A_s, a target attractor A_t and a threshold k of perturbations, the sequential control methods find an intermediate attractor A, such that $\mathsf{C}_{\min}^{A \to A_t}$ needs no more than $k-1$ perturbations. Then, the control paths are extended by taking A as a new target attractor and find a new intermediate attractor A' such that $\mathsf{C}_{\min}^{A' \to A} + \mathsf{C}_{\min}^{A \to A_t} \leq k$. This procedure is repeated until all the sequential control paths with at most k perturbations are found.

The first version of CABEAN (version 1.0) has implemented the above six methods for the minimal one-step and attractor-based source-target control of Boolean networks (see Table 1). To avoid duplication, we refer to [36] for instructions and to [33,40,24,38] for case studies and evaluations of the source-target control methods.

Target control. As illustrated in Fig. 2c, when the source is not given, to identify a subset of nodes, the perturbations of which can stir the dynamics of a Boolean network from any state $s \in S$ to the target attractor A_t, is called *target control* of Boolean networks.

The application of a control C can be lifted to a subset of states $S' \subseteq S$. Given a control $\mathsf{C} = (\mathbb{0}, \mathbb{1})$, $\mathsf{C}(S') = S''$, where $S'' = \{s'' \in S | s'' = \mathsf{C}(s'), s' \in S'\}$. Set S'' includes all the intermediate states with respect to the control C and the subset of states S'. In the case of target control, since we assume a given Boolean network can be in any initial state, we simply focus on applying a control, i.e., a number of node perturbations, on S. When perturbations are applied instantaneously, temporarily or permanently, we call them *instantaneous target control* (ITC), *temporary target control* (TTC) or *permanent target control* (PTC) [37,39], respectively.

The three target control methods, ITC, TTC and PTC, are based on the computation of the strong basin and the weak basin of the target attractor A_t and the notion of *schema*. A schema is a subset S' of S, such that there exists a triple $M = (\mathbb{0}, \mathbb{1}, \mathbb{D})$, where $\mathbb{0} \cup \mathbb{1} \cup \mathbb{D} = \{1, 2, \ldots, n\}$, $\mathbb{0}, \mathbb{1}$ and $\mathbb{D}$ are mutually disjoint (possibly empty) sets of indices of nodes of BN. The projection of S' onto the set $\mathbb{0}$ is a $|\mathbb{0}|$-bit string of zeros and the projection of S' onto the set $\mathbb{1}$ is a $|\mathbb{1}|$-bit string of ones. Let $\mathbb{D}$ denote the remaining set of nodes (i.e., don't-care-set), $X \setminus (\mathbb{0} \cup \mathbb{1})$. The projection of S' to $\mathbb{D}$ consists of all combinations of binary strings of $|\mathbb{D}|$ bits. Since the total number of nodes $n = |\mathbb{0}| + |\mathbb{1}| + |\mathbb{D}|$ is fixed, a larger schema implies more elements in $\mathbb{D}$ and fewer elements in $\mathbb{0} \cup \mathbb{1}$.

Similar to the instantaneous source-target control (OI), in order to ensure the inevitable reachability of a target attractor A_t, the ITC method drives the network BN from any initial state to the strong basin of A_t, $bas_{TS}^{S}(A_t)$. ITC divides $bas_{TS}^{S}(A_t)$ into a set of schemata and each schema leads to an ITC strategy. Thanks to the sustained effects of temporary and permanent perturbations, TTC and PTC can drive BN to the weak basin of A_t, $bas_{TS}^{W}(A_t)$. In a similar way, they partition $bas_{TS}^{W}(A_t)$ into a set of schemata, which results in a set of potential control strategies. Every subset of the candidate control sets is optimised and verified based on the constraints for temporary and permanent control [37].

Intuitively, in order to guarantee the inevitable reachability of A_t, by the time we release the control, we need to verify (1) whether the network has

	Source-target Control		Target Control
	Minimal One-step Control (OI, OT, OP)	Attractor-based Sequential Control (ASI, AST, ASP)	(ITC, TTC, PTC)
Instantaneous	✓ (version 1.0)	✓ (version 1.0)	✓ (version 2.0)
Temporary	✓ (version 1.0)	✓ (version 1.0)	✓ (version 2.0)
Permanent	✓ (version 1.0)	✓ (version 1.0)	✓ (version 2.0)

Table 1: Control methods integrated in CABEAN.

to reach a state s in the strong basin of A_t, i.e. $bas^S_{TS}(A_t)$, from which there only exist paths to A_t. Furthermore, we need to ensure (2) that any possible intermediate state $s' \in \mathsf{C}(S)$ is in the strong basin of the remaining strong basin (i.e., $bas^S_{TS}(A_t) \cap \mathsf{C}(S)$) in the transition system under control, so that the network will always evolve to the remaining strong basin. Once the network actually reaches the remaining strong basin (i.e., $bas^S_{TS}(A_t) \cap \mathsf{C}(S)$), the control can then be released and the network will evolve spontaneously towards the target attractor A_t.[3]

Next, we elaborate the three target control methods with the network BN given in Example 1.

Example 2. For attractor A_2 of BN given in Example 1, its strong basin contains only one state (110), which is a schema. Thus, the ITC for A_2 is $\mathsf{C}_1 = \{x_1 = 1, x_2 = 1, x_3 = 0\}$. The application of C_1 drives BN from any state directly to A_2. The weak basin of A_2, $bas^W_{TS}(A_2) = \{010, 110, 100, 101\}$, can be divided into two schemata $\{100, 101\}$ and $\{010, 110\}$, represented as '$*10$' and '$10*$'. The two schemata give rise to two candidate control sets: $\mathsf{C}_2 = \{x_2 = 1, x_3 = 0\}$ and $\mathsf{C}_3 = \{x_1 = 1, x_2 = 0\}$. After the step of verifying these two sets, control C_2 is both a TTC and a PTC for A_2, but C_3 is neither a TTC nor a PTC for A_2. The application of C_2 reshapes the transition system (under the control of C_2) to a new one as shown in Fig. 1(b) and BN is driven from any initial state to a state in the transition system in Fig. 1(b), from which the network will eventually stabilise to A_2. Similarly, we know that $\mathsf{C}_4 = \{x_1 = 0, x_2 = 0\}$ is an ITC for A_1 and $\mathsf{C}_5 = \{x_1 = 0\}$ is both a TTC and a PTC for A_1.

CABEAN 2.0 now integrates the three target control methods, and detailed description of the methods are referred to [37,39].

Cell phenotype as control target. Another new and interesting feature of CABEAN 2.0 is that it supports the target control of a desired phenotype. In practice, it is rarely feasible to have *complete observability* of a biological network. *Partial observability* empowers us to distinguish a phenotype based on the expressions of the marker nodes contained in the network.

[3] We refer to [40] for the precise formulation of the verification conditions and the correctness proof.

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

Table 2: An overview of the biological networks.

In general, one phenotype may correspond to one or more attractors in the Boolean network, being a model of the underlying biological network. To meet these practical needs, CABEAN 2.0 implements the new functionality to compute both the source-target and the target control of a desired phenotype (i.e., for all the methods in Table 1). It encodes the phenotype according to the input specification file, groups the attractors associated with the phenotype as a new target, and computes the control strategies for this new target. CABEAN 2.0 merges the attractors of the same phenotype first and then perform the computation for the merged attractor. In this way, it potentially can discover smaller and even new control strategies than solving the control problem by enumerate the attractors contained in the phenotype. One example is given as follows.

Example 3. For BN in Example 1, let us consider the phenotype P, where x_3 has a value of 0. P corresponds to two attractors, A_1 and A_2. That is, $P = \{000, 110\}$. CABEAN explores the strong basin and the weak basin of P to search for solutions of ITC, TTC and PTC. For instance, $\mathsf{C}_6 = \{x_3 = 0\}$ is neither a TTC for A_1 nor a TTC for A_2. But the temporary inhibition of x_3, C_6 transforms the transition system to Fig. 1(c) and it will surely guide BN to P.

5 Evaluation

In this section, we present an overview of the evaluation of the target control methods (ITC, TTC and PTC) on a number of real-life biological networks to

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

Table 3: The minimal number of perturbations computed by the control methods ITC, TTC, and PTC for the biological networks.

demonstrate their efficacy and efficiency.[4] In Table 2, we present information on the number of nodes, the number of edges and the number of singleton and cyclic attractors for each of these networks. Further details on the networks are referred to the original works where the Boolean networks were originally presented (see the corresponding references of the networks in [39]) .

Efficacy. Table 3 gives the minimal number of perturbations computed by the target control methods (ITC, TTC and PTC) for one of the attractors of the networks. It is clear to see that ITC always requires more perturbations than the other two methods due to its instantaneous effect of the perturbations. In Table 3, ITC often needs to perturb around 10 to 20 nodes, whereas TTC and PTC can achieve the inevitable reachability of target attractors with at most 5 perturbations. Since it is difficult to realise the instantaneous perturbation of a number of nodes simultaneously, this makes ITC less practical and less attractive for real-life applications. Thus, TTC and PTC, which employ temporary or permanent perturbations, are more preferable than ITC.

Efficiency. Table 4 summarises the computational time for computing the target control strategies for *all* the attractors of the networks. All the experiments are performed on a high-performance computing (HPC) platform, which contains CPUs of Intel Xeon Gold 6132 @2.6 GHz.

From Table 4, we clearly observe that ITC is the most efficient one, but it requires more perturbations (see the above discussion and Table 3). The effi-

[4] We refer to [33,40,24,38] for case studies and evaluations of the source-target control methods implemented in CABEAN 1.0.

Network	Computational time (seconds)		
	ITC	TTC	PTC
yeast	0.028	0.987	0.933
ERBB	0.055	0.117	0.163
HSPC-MSC	0.097	0.101	0.109
hematopoiesis	0.374	139.859	72.793
PC12	0.149	17.653	22.189
bladder	0.302	2.426	7.997
psc-bFA	36.77	3,732.780	9,296.740
co-infection	6,294.290	*	*
MAPK	4.608	22.218	45.504
CREB	7.962	8.277	8.693
HGF	19.925	1,437.290	201.363
bortezomib	15.605	*	*
T-diff	21.581	29,738.500	*
HIV1	302.8	323.666	379.127
CD4+	549.878	1,982.450	21,358.400
pathway	445.251	4,435.590	10,038.600

Table 4: Computational time of the control methods ITC, TTC, and PTC for the biological networks. The symbol '*' means that the corresponding method failed to finish its computation for the network within 12 hours.

ciency of these target control methods are influenced by many factors, including the network size, the network density, the number of attractors and the number of required node perturbations. For instance, for the co-infection network and the model of bortezomib responses, TTC and PTC were able to identify target control efficiently for some of the attractors, but failed for the other attractors of these two networks. One reason is that the target control of those attractors require many perturbations, thus it takes a considerable amount of time to verify all the subsets of the schemata (see Section 4). This can be improved if it is sufficient to provide only some of the solutions instead of all the solutions.

In summary, we conclude that the target control methods (ITC, TTC and PTC) scale well for Boolean networks of a few hundreds of nodes and they are able to identify a rich set of solutions with a small number of node perturbations.

6 Implementation

CABEAN 2.0 implements the six source-target control methods and the three target control methods (see Table 1) in C and C++, based on the CUDD package [35], the model checker MCMAS [20], and the tool ASSA-PBN [26]. There are two main factors resulting in a high efficiency of CABEAN 2.0. First, both the transition system and transition relations of Boolean networks are encoded as binary decision diagrams (BDDs). BDDs are introduced by Bryant [2] to represent Boolean functions, and they have an advantage of memory efficiency to

alleviate the state space explosion. Realisation of the control methods implemented in CABEAN 2.0 depends on the efficiency of BDD operations. Second, the methods are based on the efficient decomposition-based strong basin computation [32,33], which adapts the divide and conquer strategy and thus scales well for large Boolean networks.

In the following, we use the Boolean network BN given in Example 1 to demonstrate the new features of CABEAN 2.0, including the three target control methods and the encoding of a phenotype. For a detailed and complete user guide of CABEAN, we refer to the the website of the software tool.[5]

Model files. CABEAN 2.0 supports two formats for the model file, including the BoolNet and ISPL (Interpreted Systems Programming Language) format of the software MCMAS [20]. Further details on the syntax of the two formats can be found at the website of the tool. Other formats, such as SBML-qual, Petri net, GINsim, can be converted to the BoolNet format using the BioLQM toolkits.[6]

Target control of an attractor. CABEAN 2.0 integrates the decomposition-based attractor detection method [25] to compute attractors of a Boolean network. Prior to the computation of target control strategies, CABEAN 2.0 computes all the exact attractors of the given network and prints them in lexicographic order. Users can then specify the index of an attractor as the desired attractor for the target control of the network.

The target control is computed using the following command line:

./cabean -compositional 2 -control <Control method> -tin <index of the target attractor> <model file>

The option '-compositional 2' indicates that the decomposition-based methods [25,33] are used for attractor detection and the strong basin computation; the option '-control <Control method>' selects one of the target methods, ITC, TTC or PTC; and the option '-tin <index of the target attractor>' sets the index of the target attractor.

Suppose we want to compute TTC for A_1 and A_2 for BN in Example 1. The outputs of CABEAN 2.0 are given below. The results show that the network will stabilise in A_1 with the temporary inhibition of A_1 and it will settle down to A_2 from any initial state with the temporary control of {x2=1 x3=0}. These results are consistent with Example 2.

```
**********************************************
TARGET ATTRACTOR #1
**********************************************
Control set 1: x1=0
**********************************************
TARGET ATTRACTOR #2
**********************************************
Control set 1: x2=1 x3=0
```

[5] https://satoss.uni.lu/software/CABEAN/

[6] BioLQM is available at http://colomoto.org/biolqm/.

Target control of a phenotype. The target control of a phenotype can be computed using the following command line:

./cabean -compositional 2 -control <Control method> -tmarker <specification file> <model file>

Instead of setting the index of a target attractor, we need to specify a phenotype with an input file. For example, the phenotype P in Example 3 is specified as the following:

```
nodes, value
x3, 0
```

The computed TTC of P is given below. The line 'TARGET ATTRACTOR #1, #2' indicates that attractors A_1 and A_2 agree with the phenotype P. The results show that temporary inhibition of x_1, x_2 or x_3 guarantees the inevitable reachability of P. In Example 3, we have explained the case of $\mathsf{C}_6 = \{x_3 = 0\}$.

```
*********************************************
TARGET ATTRACTOR #1 #2
*********************************************
Control set 1: x1=0
Control set 2: x2=0
Control set 3: x3=0
```

7 Conclusion

Motivated by the important and appealing application of cell reprogramming in biology, recent years have seen a rapid development of a number of computational methods for the control of gene regulatory networks modelled as Boolean networks. In this paper, we have presented a new release of CABEAN (version 2.0) that integrates three new target control methods for asynchronous Boolean networks. These methods identify a set of nodes, whose instantaneous, temporary or permanent perturbations can drive a Boolean network from any initial state to a desired attractor or a phenotype.

CABEAN 2.0 has assembled a variety of control methods (see Table 1) that manipulate the dynamics of a Boolean network in different ways. All these methods focus on identifying which nodes in the Boolean network to be perturbed in order to drive the network's dynamics into a desired attractor. A node perturbation corresponds to the removal or blocking of a particular gene in a gene regulatory network. However, in complex diseases it is more common that several subtle changes affect interactions between genes [29]. This suggests perturbations at the edge level, i.e., targeting selected interactions between genes. Currently, new methods are under development for the control of Boolean networks with edgetic perturbations, which will be eventually integrated into CABEAN as well.

In future, we also plan to provide a graphical user interface (GUI) for the convenience of users.

Acknowledgements. This work was supported by the project SEC-PBN funded by University of Luxembourg and the ANR-FNR project AlgoReCell (grant No. INTER/ANR/15/11191283).

References

1. Biane, C., Delaplace, F.: Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**(5), 1574–1585 (2018)
2. Bryant, R.E.: Symbolic verification of mos circuits. In: Proc. the 1985 Chapel Hill Conference on Very Large Scale Integration. pp. 419–438. Computer Science Press (1985)
3. Chen, H., Liang, J., Wang, Z.: Pinning controllability of autonomous Boolean control networks. Science China Information Sciences **59**(7), 070107 (2016)
4. Cornelius, S.P., Kath, W.L., Motter, A.E.: Realistic control of network dynamics. Nature Communications **4**(1), 1–9 (2013)
5. Correia, R.B., Gates, A.J., Wang, X., Rocha, L.M.: CANA: a python package for quantifying control and canalization in Boolean networks. Frontiers in Physiology **9**, 1046 (2018)
6. Czeizler, E., Gratie, C., Chiu, W.K., Kanhaiya, K., Petre, I.: Target controllability of linear networks. In: Proc. 14th International Conference on Computational Methods in Systems Biology. LNCS, vol. 9859, pp. 67–81. Springer (2016)
7. Fiedler, B., Mochizuki, A., Kurosawa, G., Saito, D.: Dynamics and control at feedback vertex sets. I: Informative and determining nodes in regulatory networks. Journal of Dynamics and Differential Equations **25**(3), 563–604 (2013)
8. Fontanals, L.C., Tonello, E., Siebert, H.: Control strategy identification via trap spaces in Boolean networks. In: Proc. 18th International Conference on Computational Methods in Systems Biology. LNCS, vol. 12314, pp. 159–175. Springer (2020)
9. Gam, R., Sung, M., Prasad Pandurangan, A.: Experimental and computational approaches to direct cell reprogramming: Recent advancement and future challenges. Cells **8**(10), 1189 (2019)
10. Gao, J., Liu, Y.Y., D'Souza, R.M., Barabási, A.L.: Target control of complex networks. Nature Communications **5**, 5415 (2014)
11. Grath, A., Dai, G.: Direct cell reprogramming for tissue engineering and regenerative medicine. Journal of Biological Engineering **13**(1), 14 (2019)
12. Gurdon, J.B.: The developmental capacity of nuclei taken from intestinal epithelium cells of feeding tadpoles. Development **10**(4), 622–640 (1962)
13. Huang, S.: Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. Pharmacogenomics **2**(3), 203–222 (2001)
14. Kauffman, S.: Homeostasis and differentiation in random genetic control networks. Nature **224**, 177–178 (1969)
15. Kim, J., Park, S.M., Cho, K.H.: Discovery of a kernel for controlling biomolecular regulatory networks. Scientific Reports **3**, 2223 (2013)
16. Klarner, H., Streck, A., Siebert, H.: PyBoolNet: a python package for the generation, analysis and visualization of Boolean networks. Bioinformatics **33**(5), 770–772 (2017)

17. Liang, J., Chen, H., Lam, J.: An improved criterion for controllability of Boolean control networks. IEEE Transactions on Automatic Control **62**(11), 6012–6018 (2017)
18. Lin, P.C.K., Khatri, S.P.: Application of Max-SAT-based ATPG to optimal cancer therapy design. BMC Genomics **13**(S6), S5 (2012)
19. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. Nature **473**, 167–173 (2011)
20. Lomuscio, A., Qu, H., Raimondi, F.: MCMAS: An open-source model checker for the verification of multi-agent systems. International Journal on Software Tools for Technology Transfer **19**(1), 9–30 (2017)
21. Lu, J., Zhong, J., Ho, D.W., Tang, Y., Cao, J.: On controllability of delayed Boolean control networks. SIAM Journal on Control and Optimization **54**(2), 475–494 (2016)
22. Mandon, H., Haar, S., Paulevé, L.: Relationship between the reprogramming determinants of Boolean networks and their interaction graph. In: Proc. 5th International Workshop on Hybrid Systems Biology. LNCS, vol. 9957, pp. 113–127. Springer (2016)
23. Mandon, H., Haar, S., Paulevé, L.: Temporal reprogramming of Boolean networks. In: Proc. 15th International Conference on Computational Methods in Systems Biology. LNCS, vol. 10545, pp. 179–195. Springer (2017)
24. 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, vol. 11773, pp. 3–19. Springer (2019)
25. Mizera, A., Pang, J., Qu, H., Yuan, Q.: Taming asynchrony for attractor detection in large Boolean networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics **16**(1), 31–42 (2019)
26. 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), 1203–1216 (2018)
27. 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 of Theoretical Biology **335**, 130–146 (2013)
28. Moradi, M., Goliaei, S., Foroughmand-Araabi, M.H.: A Boolean network control algorithm guided by forward dynamic programming. PLOS One **14**(5), e0215449 (2019)
29. Mosca, R., Tenorio-Laranga, J., Olivella, R., Alcalde, V., Céol, A., Soler-López, M., Aloy, P.: dSysMap: exploring the edgetic role of disease mutations. Nature Methods **12**(3), 167–168 (2015)
30. Murrugarra, D., Veliz-Cuba, A., Aguilar, B., Laubenbacher, R.: Identification of control targets in Boolean molecular network models via computational algebra. BMC Systems Biology **10**(1), 94 (2016)
31. Müssel, C., Hopfensitz, M., Kestler, H.A.: BoolNet-an R package for generation, reconstruction and analysis of Boolean networks. Bioinformatics **26**(10), 1378–1380 (2010)
32. 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. pp. 11–20. ACM Press (2018)

33. 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 **17**(6), 1932–1945 (2020)
34. Poret, A., Guziolowski, C.: Therapeutic target discovery using Boolean network attractors: improvements of kali. Royal Society Open Science **5**(2), 171852 (2018)
35. Somenzi, F.: CUDD: CU Decision diagram package (release 2.5.1). http://vlsi.colorado.edu/ fabio/CUDD/ (2015)
36. Su, C., Pang, J.: CABEAN: a software for the control of asynchronous Boolean networks. Bioinformatics **37**(6), 879–881 (2020)
37. Su, C., Pang, J.: A dynamics-based approach for the target control of Boolean networks. In: Proc. 11th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. pp. 50:1–50:8. ACM Press (2020)
38. Su, C., Pang, J.: Sequential temporary and permanent control of Boolean networks. In: Proc. 18th International Conference on Computational Methods in Systems Biology. LNCS, vol. 12314, pp. 234–251. Springer (2020)
39. Su, C., Pang, J.: Target control of asynchronous Boolean networks. arXiv preprint arXiv:2101.00644 (2021)
40. Su, C., Paul, S., Pang, J.: Controlling large Boolean networks with temporary and permanent perturbations. In: Proc. 23rd International Symposium on Formal Methods. LNCS, vol. 11800, pp. 707–724. Springer (2019)
41. Videla, S., Saez-Rodriguez, J., Guziolowski, C., Siegel, A.: caspo: a toolbox for automated reasoning on the response of logical signaling networks families. Bioinformatics **33**(6), 947–950 (2017)
42. 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**(1), 1–11 (2016)
43. Wu, Y., Sun, X.M., Zhao, X., Shen, T.: Optimal control of Boolean control networks with average cost: A policy iteration approach. Automatica **100**, 378–387 (2019)
44. Yamanaka, S.: Strategies and new developments in the generation of patient-specific pluripotent stem cells. Cell Stem Cell **1**(1), 39–49 (2007)
45. Yue, J., Yan, Y., Chen, Z., Jin, X.: Identification of predictors of Boolean networks from observed attractor states. Mathematical Methods in the Applied Sciences **42**(11), 3848–3864 (2019)
46. Zañudo, J.G.T., Albert, R.: Cell fate reprogramming by control of intracellular network dynamics. PLOS Computational Biology **11**(4), e1004193 (2015)
47. 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), 7234–7239 (2017)
48. Zhao, Y., Kim, J., Filippone, M.: Aggregation algorithm towards large-scale Boolean network analysis. IEEE Transactions on Automatic Control **58**(8), 1976–1985 (2013)
49. Zheng, D., Yang, G., Li, X., Wang, Z., Liu, F., He, L.: An efficient algorithm for computing attractors of synchronous and asynchronous Boolean networks. PLOS One **8**(4), e60593 (2013)
50. Zhong, J., Liu, Y., Kou, K.I., Sun, L., Cao, J.: On the ensemble controllability of Boolean control networks using STP method. Applied Mathematics and Computation **358**, 51–62 (2019)
51. Zhu, P., Han, J.: Asynchronous stochastic Boolean networks as gene network models. Journal of Computational Biology **21**(10), 771–783 (2014)
52. Zhu, Q., Liu, Y., Lu, J., Cao, J.: Further results on the controllability of Boolean control networks. IEEE Transactions on Automatic Control **64**(1), 440–442 (2018)