



On the Full Control of Boolean Networks

Soumya Paul¹(✉), Jun Pang^{1,2}, and Cui Su¹

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

² Faculty of Science, Technology and Communication, University of Luxembourg,
Esch-sur-Alzette, Luxembourg
{soumya.paul, jun.pang, cui.su}@uni.lu

Boolean networks (BNs), introduced by Kauffman [3], is a popular and well-established framework for modelling gene regulatory networks and their associated signalling pathways. The main advantage of this framework is that it is relatively simple and yet able to capture the important dynamical properties of the system under study, thus facilitating the modelling and analysis of large biological networks as a whole.

A Boolean network B is a pair $B = (\mathbf{x}, \mathbf{f})$, where \mathbf{x} is a tuple of n variables $\mathbf{x} = (x_1, x_2, \dots, x_n)$ and \mathbf{f} is a tuple of n Boolean *update functions* $\mathbf{f} = (f_1, f_2, \dots, f_n)$, where for every i , the function f_i , which depends on a subset of the variables in \mathbf{x} , governs the dynamics of x_i in time. BN is called *linear* when the functions \mathbf{f} are linear. It is *non-linear* otherwise. For a BN B with n variables, its *dependency graph* is a directed graph $\mathcal{G}_B = (V, E)$ with a set V of n vertices (or nodes) for the n variables, ordered such that vertex v_i corresponds to the variable x_i . There is a directed edge from vertex v_i to v_j if and only if the function f_j depends on x_i . The *structure* of a BN B refers to the structure of its dependency graph \mathcal{G}_B . The variables of \mathbf{x} take Boolean values. Each such tuple of values gives rise to a *state* of the BN, typically denoted as \mathbf{s} or \mathbf{t} . For a BN with n variables, there can be a total of 2^n possible states, the elements in $\{0, 1\}^n$. The *asynchronous dynamics* of a BN B is assumed to evolve in discrete *time steps* as follows. Suppose B is in state \mathbf{s} in time t . A possible *next state* to \mathbf{s} , i.e., a state in time $(t+1)$, is given by non-deterministically choosing exactly one i and updating the i th component of \mathbf{s} by applying the function f_i and leaving the other components unchanged. This operation results in a directed graph, called the *(state) transition system* (TS) of B , denoted TS_B , whose elements are the states of B and there is a directed edge from a state \mathbf{s} to a state \mathbf{t} if and only if \mathbf{t} is a possible next state to \mathbf{s} .

An *attractor* A of B is a subset of states of B that forms a bottom maximal strongly connected component (SCC) of TS_B . Attractors represent the eventual behaviour or the *steady states* of the system modelled by the BN. In biological context, attractors are hypothesised to characterise cellular phenotypes [3] and also correspond to functional cellular states such as proliferation, apoptosis differentiation etc. [1]. The identification and analysis of the attractors of a BN thus forms an integral part of the study of the corresponding biological network. *Controlling* the network means driving its dynamics from one steady state to another by modifying the parameters of the network which amounts to being able to move it between the different attractors. The *strong basin of attraction* of an attractor A of B , denoted $\text{bas}(A)$, is the subset of states of B such that

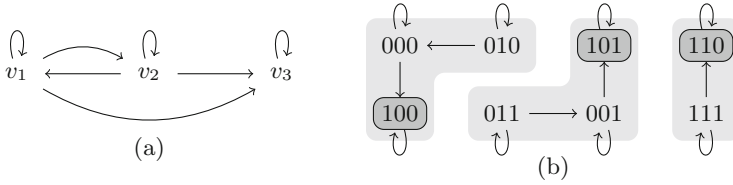


Fig. 1. Running example: the dependency graph of a B and its transition system.

there is a (possibly empty) sequence of edges from every state \mathbf{s} in $\text{bas}(A)$ to a state $\mathbf{t} \in A$ and moreover there is no such sequence from \mathbf{s} to any state $\mathbf{t}' \in A'$ for any other attractor $A' \neq A$ of B . If the current state of B is in $\text{bas}(A)$ for some attractor A , then its dynamics is guaranteed to eventually reach A .

The full control of linear networks is a well-understood problem [2] and a number of control strategies have been developed in the literature. Recent work on network controllability has shown that full controllability and reprogramming of intercellular networks can be achieved by a minimal number of control targets [5]. However, the full control of non-linear networks is apparently more challenging predominantly due to the explosion of the potential search space with the increase in the network size. There has not been a lot of work in the study of the full control of non-linear networks. Recently, Kim et al. [4] developed a method to identify the so-called ‘control kernel’ which is a minimal set of nodes for fully controlling a biological network. However, their method requires the construction of the full state transition graph of the studied network and as such does not scale well for large networks.

In this work, we aim to develop a method for the *full control* of non-linear BNs with asynchronous dynamics, based both on their structural and dynamic properties. The problem is formally defined as: Given a BN B , find a *minimal subset* C of indices of the variables of B such that for any pair of attractors A_s and A_t of B , there exists a state $\mathbf{s} \in A_s$ such that a subset of the variables with indices in C needs to be toggled (controlled) in \mathbf{s} , in a *single step*, so that the system eventually reaches A_t . The problem can be shown to be PSPACE-hard and hence efficient algorithms for dealing with large BNs are highly unlikely. Our method is based on a decomposition-based approach for solving the corresponding minimal *target control* problem [7] which yields efficient results for many large real-life BNs having modular structures. In brief, the method analyses the structure of the BN to identify its maximal strongly connected components and uses them to decompose the vertices of the dependency graph into (possibly overlapping) subsets called *blocks* (see details in [6, 7]). The blocks are sorted topologically and the full control problem is solved locally for each block in the sorted order. The local results are then combined to derive the minimal full control set for the entire network. Due to space-restriction, we describe our method in details on a running example without going into formal notations and proofs.

Consider the three-node asynchronous BN $B = (\mathbf{x}, \mathbf{f})$, where $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{f} = (f_1, f_2, f_3)$, such that $f_1 = \neg x_2 \vee (x_1 \wedge x_2)$, $f_2 = x_1 \wedge x_2$ and $f_3 = x_3 \wedge \neg(x_1 \wedge x_2)$. The dependency graph \mathcal{G}_B and the state transition system TS_B

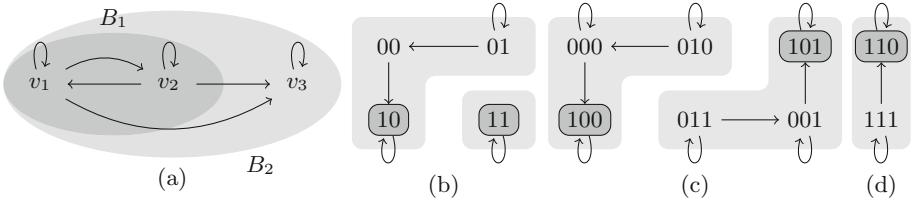


Fig. 2. The local transition systems of the blocks B_1 and B_2 .

are given in Figs. 1(a) and (b), respectively. TS_B has three attractors $A_1 = \{(100)\}$, $A_2 = \{(101)\}$ and $A_3 = \{(110)\}$, shown by dark grey rectangles. Their corresponding strong basins of attractions are shown by enclosing grey regions of a lighter shade.

By definition, we know that for the BN to surely end up in an attractor A it is enough for it to be in any of the states in the strong basin of A . Thus, for example, from attractor A_2 to end up in A_3 one has to control the nodes with indices $\{2, 3\}$ or just $\{2\}$ to enter the strong basin of A_2 . Table 1 notes the indices of the nodes to control for each pair of attractors of B .

To compute the minimal full control, one has to find a minimal subset C of $\{1, 2, 3\}$ such that C is a superset of at least one subset from every cell of Table 1. In this example $C = \{2, 3\}$, but the general problem is NP-hard.

Table 1. Attractor pair control indices.

	A_1	A_2	A_3
A_1		$\{2\}, \{2,3\}$	$\{3\}, \{1,3\}, \{1,2,3\}$
A_2	$\{1\}, \{2\}, \{1,2\}$		$\{2\}, \{2,3\}$
A_3	$\{1,3\}, \{2,3\}, \{1,2,3\}$	$\{2\}, \{2,3\}, \{1,2,3\}$	

We thus take advantage of our decomposition-based approach developed for the efficient computation of minimal target control for well-structured BNs [7], to compute the full control for pairs of attractors in local *blocks* and then merge them to obtain the global full control.

In the above example \mathcal{G}_B has two maximal SCCs $S_1 = \{v_1, v_2\}$ and $S_2 = \{v_3\}$. Each such component S_j generates a *block* by including all the vertices from which there are incoming edges into S_j . Thus \mathcal{G}_B has two blocks $B_1 = \{v_1, v_2\}$ and $B_2 = \{v_1, v_2, v_3\}$ as shown in Fig. 2(a). The vertex in B_2 depends on (has incoming edges from) vertices in B_1 whereas, B_1 has no such dependency on other blocks. Hence, they can be topologically sorted as $\{B_1, B_2\}$. In fact, the dependency relation between the blocks will always be acyclic and hence the blocks can be topologically sorted [7]. We compute the transition system TS_{B_1} of B_1 (Fig. 2(b)) and find that it has 2 attractors $A_1^1 = \{(10)\}$ and $A_1^2 = \{(11)\}$.

Table 2. Control indices for the blocks.

	A_1^1	A_1^2
A_1^1		{2}
A_1^2	{1}, {2}, {1,2}	

(a)

	A_2^1	A_2^2	A_2^3
A_2^1		{3}	\emptyset , {3}
A_2^2	{3}		\emptyset , {3}
A_2^3	\emptyset	{3}	

(b)

The two basins of attractions $\text{bas}(A_1^1)$ and $\text{bas}(A_1^2)$ are used to compute two transition systems of B_2 (Figs. 2(c) and (d), respectively). The first has two attractors $A_2^1 = \{(100)\}$ and $A_2^2 = \{(101)\}$ and the second has one attractor $A_2^3 = \{(110)\}$.

It holds, as was shown in [7], that $A_1 = A_1^1 \otimes A_1^2, A_2 = A_1^1 \otimes A_2^2$ and $A_3 = A_1^2 \otimes A_2^3$ are the only attractors of the global B (where \otimes is a combination operation on boolean tuples defined in [7]). Thus we can work with the transition systems of B_1 and B_2 separately to compute the minimal full control of B . For that, we construct Tables 2(a) and (b) similar to Table 1 listing the sets of indices to be controlled to move between attractors of B_1 and B_2 , respectively. Here an entry of \emptyset means that it is possible to move between the corresponding attractors without controlling any index. For B_2 we need only consider the indices of the vertices in $B_2 \setminus B_1$. From Table 2(a), $C^1 = \{2\}$ and from Table 2(b), $C^2 = \{3\}$. Combining, $C = C^1 \cup C^2 = \{2, 3\}$.

For the general case, suppose there are k blocks that are topologically sorted as $\{B_1, B_2, \dots, B_k\}$. Then the matrix for every block B_i will involve indices of the vertices in $B_i \setminus (\bigcup_{j < i} B_j)$. The rest of the procedure is similar to the 2-block case as described here. We note that for certain BNs, the minimal global control for moving to a target attractor A_t computed by combining the minimal local control for the blocks might move the BN to a state which is not in the strong basin of attraction of A_t . We deal with this issue by augmenting the procedure to systematically rule out such problematic cases. Currently, we are implementing our approach in software and evaluating it on real-life biological networks modelled as BNs.

References

1. Huang, S.: Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. *Pharmacogenomics* **2**(3), 203–222 (2001)
2. Kalman, R.E.: Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math.* **1**(2), 152–192 (1963)
3. Kauffman, S.: Homeostasis and differentiation in random genetic control networks. *Nature* **224**, 177–178 (1969)
4. Kim, J., Park, S.M., Cho, K.H.: Discovery of a kernel for controlling biomolecular regulatory networks. *Sci. Rep.* **3**, 2223 (2013)
5. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. *Nature* **473**, 167–173 (2011)

6. Mizera, A., Pang, J., Qu, H., Yuan, Q.: Taming asynchrony for attractor detection in large Boolean networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* (2018, in press)
7. Paul, S., Su, C., Pang, J., Mizera, A.: A decomposition-based approach towards the control of Boolean networks. In: *Proceedings of ACM-BCB 2018*. ACM (2018, in press)