Towards the Existential Control of Boolean Networks: A Preliminary Report

Soumya Paul[1], Jun Pang[1,2], and Cui Su[1]

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

Abstract. Given a Boolean network BN and a subset $\mathcal{A}$ of attractors of BN, we study the problem of identifying a minimal subset C_{BN} of vertices of BN, such that the dynamics of BN can reach from a state $\mathbf{s}$ in any attractor $A_s \in \mathcal{A}$ to any attractor $A_t \in \mathcal{A}$ by controlling (toggling) a subset of vertices in C_{BN} in a single time step. We describe a method based on the decomposition of the network structure into strongly connected components called 'blocks'. The control subset can be locally computed for each such block and the results then merged to derive the global control subset C_{BN}. This potentially improves the efficiency for many real-life networks that are large but modular and well-structured. We are currently in the process of implementing our method in software.

1 Introduction

Systems biology, with the help of mathematical modelling, has revolutionised the human diseasome research and paved the way towards the development of new therapeutic approaches and personalised medicine. Such therapies target specific proteins within the cellular systems aiming to drive it from a 'diseased' state to a 'healthy' state. However, it has been observed that disease-networks are intrinsically robust against perturbations due to the inherent diversity and redundancy of compensatory signalling pathways [2]. This greatly reduces the efficacy of single-target drugs. Hence, rather than trying to design selective ligands that target individual receptors only, network polypharmacology seeks to modify multiple cellular targets to tackle the compensatory mechanisms and robustness of disease-associated cellular systems. This motivates the question of identifying multiple drug targets using which the network can be 'fully controlled', i.e. driven from any (diseased) state to any desired target (healthy) state. Furthermore, for the feasibility of the synthesis of such drugs, the number of such targets should be minimised. However, biological networks are intrinsically large (number of components, parameters, interactions, etc.) which results in an exponentially increasing number of potential drug target combination making a purely experimental approach quickly infeasible. This reinforces the need of mathematical modelling and computational techniques.

Boolean networks (BNs), first introduced by Kauffman [5], is a popular and well-established framework for modelling gene regulatory networks (GRNs) and

their associated signalling pathways. Its main advantage is that it is simple and yet able to capture the important dynamical properties of the system under study, thus facilitating the modelling of large biological systems as a whole. The states of a BN are tuples of 0s and 1s where each element of the tuple represents the level of activity of a particular protein in the GRN or the signalling pathway it models - 0 for inactive and 1 for active. The BN is assumed to evolve dynamically by moving from one state to the next governed by a Boolean function for each of its components. The steady state behaviour of a BN is given by its subset of states called *attractors* to one of which the dynamics eventually settles down. In biological context, attractors are hypothesised to characterise cellular phenotypes [5] and also correspond to functional cellular states such as proliferation, apoptosis, differentiation, etc. [3]. The *control* of a BN therefore refers to the reprogramming/changing of the parameters of the BN (functions, values of variables, etc.) so that its dynamics eventually reaches a desired attractor or steady state.

The full control of linear networks is a well-studied problem [4] and such control strategies have been proposed over the years. Recent work on network controllability has shown that full controllability and reprogramming of intercellular networks can be achieved by a minimum number of control targets [7]. 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 this regard. Kim et al. [6] developed a method to identify the so-called 'control kernel' which is a minimal set of nodes for fully controlling a biological network. But, their method is based on the construction of the full state transition graph of the network and as such does not scale well for large networks.

In many cases, only some of the attractors of the BNs are 'biologically relevant', i.e. correspond to meaningful expressions of the modelled GRNs. Thus, focussing on only the relevant attractors might help reduce the complexity of the control problem while still being biologically meaningful.

Our contributions. In this work, we report the initial results on a method for the control of Boolean networks that exploits both their structural and dynamic properties, as shown inevitable in [1]. More precisely, given a Boolean network BN and a set of 'relevant' attractors $\mathcal{A}$ of BN, the method computes a minimal set of variables (the *minimal control set*), such that starting from an initial attractor $A_s \in \mathcal{A}$ and by controlling specific subsets of these variables in a *single time-step*, the BN can (potentially) reach any desired target attractor $A_t \in \mathcal{A}$ when left to evolve on its own according to its original dynamics. A welcome side-effect of the method is that when $\mathcal{A}$ is the set of all attractors of BN, it gives the minimal set of vertices for fully controlling BN. We use an approach that we have developed for the problem of target control (driving the BN to a given single target attractor) of BNs, based on the decomposition of its network structure into strongly connected components called 'blocks'. Although the method can be applied on the entire BN in one-go, we believe that using the decomposition-based approach can greatly increase its efficiency on large real-life biological

networks whose BN models have well-behaved modular structure. This is work-in-progress and we are currently implementing our method in software to test its effectiveness on various networks.

2 Background and Notations

Let $N = \{1, 2, \ldots, n\}$ where $n \geq 1$. A *Boolean network* is a tuple $\mathsf{BN} = (\mathbf{x}, \mathbf{f})$ where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ such that each x_i is a Boolean variable and $\mathbf{f} = (f_1, f_2, \ldots, f_n)$ is a tuple of Boolean functions over $\mathbf{x}$. In what follows, i will always range over N, unless stated otherwise. A Boolean network $\mathsf{BN} = (\mathbf{x}, \mathbf{f})$ may be viewed as a directed graph $\mathcal{G}_{\mathsf{BN}} = (V, E)$, where $V = \{v_1, v_2 \ldots, v_n\}$ is the set of *vertices* or *nodes* (intuitively, v_i corresponds to the variable x_i for all i) and for every $i, j \in N$, there is a directed edge from v_j to v_i, often denoted as $v_j \to v_i$, if and only if f_i depends on x_j. Thus V is ordered according to the ordering of $\mathbf{x}$. For any vertex $v_i \in V$, we let $\mathsf{ind}(v_i) = i$ be the index of v_i in this ordering. For any subset W of V, $\mathsf{ind}(W) = \{\mathsf{ind}(v) |\ v \in W\}$. A *path* from a vertex v to a vertex v' is a (possibly empty) sequence of edges from v to v' in $\mathcal{G}_{\mathsf{BN}}$. For any vertex $v \in V$ we define its set of *parents* as $\mathsf{par}(v) = \{v' \in V \mid v' \to v\}$ and for any subset W of V, $\mathsf{par}(W) = \{\mathsf{par}(v) \mid v \in W\}$. For the rest of the exposition, we assume that an arbitrary but fixed network BN of n variables is given to us and $\mathcal{G}_{\mathsf{BN}} = (V, E)$ is its associated directed graph.

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 $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, and for every i, 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, suppose f_i depends on $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. Then $f_i(\mathbf{s})$ will denote the value $f_i(\mathbf{s}[i_1], \mathbf{s}[i_2], \ldots, \mathbf{s}[i_k])$. For two states $\mathbf{s}, \mathbf{s}' \in \mathbf{S}$, the *Hamming distance* between $\mathbf{s}$ and $\mathbf{s}'$ will be denoted as $\mathsf{hd}(\mathbf{s}, \mathbf{s}')$ and $\arg(\mathsf{hd}(\mathbf{s}, \mathbf{s}')) \subseteq N$ will denote the set of indices in which $\mathbf{s}$ and $\mathbf{s}'$ differ. 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 $\mathsf{hd}(\mathbf{s}, \mathbf{S}') = \min_{\mathbf{s}' \in \mathbf{S}'} \mathsf{hd}(\mathbf{s}, \mathbf{s}')$. We let $\arg(\mathsf{hd}(\mathbf{s}, \mathbf{S}'))$ denote the set of subsets of N such that $I \in \arg(\mathsf{hd}(\mathbf{s}, \mathbf{S}'))$ if and only if I is a set of indices of the variables that realise $\mathsf{hd}(\mathbf{s}, \mathbf{S}')$.

We assume that the Boolean network starts initially in a state $\mathbf{s}_0$ and its state changes in every discrete time-step according to the update functions $\mathbf{f}$. In this work, we shall deal with the asynchronous updating scheme but all our results transfer to the synchronous updating scheme as well. Suppose $\mathbf{s}_0 \in \mathbf{S}$ is an initial state of BN. The *asynchronous evolution* of BN is a function $\xi : \mathbb{N} \to \wp(\mathbf{S})$ such that $\xi(0) = \mathbf{s}_0$ and for every $j \geq 0$, if $\mathbf{s} \in \xi(j)$ then $\mathbf{s}' \in \xi(j+1)$ if and only if either $\mathsf{hd}(\mathbf{s}, \mathbf{s}') = 1$ and $\mathbf{s}'[i] = f_i(\mathbf{s})$ where $i = \arg(\mathsf{hd}(\mathbf{s}, \mathbf{s}'))$ or $\mathsf{hd}(\mathbf{s}, \mathbf{s}') = 0$ and there exists i such that $\mathbf{s}'[i] = f_i(\mathbf{s})$.

The dynamics of a Boolean network can be represented as a *state transition graph* or a *transition system (TS)*. The *transition system* of BN, denoted as $\mathsf{TS}_{\mathsf{BN}}$ is a tuple $(\mathbf{S}, \to)$ 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 $\mathbf{s} \to \mathbf{s}'$, if and only if either $\mathsf{hd}(\mathbf{s}, \mathbf{s}') = 1$ and $\mathbf{s}'[i] = f_i(\mathbf{s})$ where $i = \arg(\mathsf{hd}(\mathbf{s}, \mathbf{s}'))$ or $\mathsf{hd}(\mathbf{s}, \mathbf{s}') = 0$ and there exists i such that $\mathbf{s}'[i] = f_i(\mathbf{s})$.

For any state $\mathbf{s} \in \mathbf{S}$, $\mathsf{pre}_{\mathsf{TS}}(\mathbf{s}) = \{\mathbf{s}' \in \mathbf{S} \mid \mathbf{s}' \rightarrow \mathbf{s}\}$ contains all the states that can reach $\mathbf{s}$ by performing a single transition in TS. For a subset $\mathbf{S}'$ of $\mathbf{S}$, $\mathsf{pre}_{\mathsf{TS}}(\mathbf{S}') = \bigcup_{\mathbf{s} \in \mathbf{S}'} \mathsf{pre}_{\mathsf{TS}}(\mathbf{s})$. A *path* from a state $\mathbf{s}$ to a state $\mathbf{s}'$ is a (possibly empty) sequence of transitions from $\mathbf{s}$ to $\mathbf{s}'$ in $\mathsf{TS}_{\mathsf{BN}}$. A path from a state $\mathbf{s}$ to a subset $\mathbf{S}'$ of $\mathbf{S}$ is a path from $\mathbf{s}$ to any state $\mathbf{s}' \in \mathbf{S}'$. For a state $\mathbf{s} \in \mathbf{S}$, $\mathsf{reach}_{\mathsf{TS}}(\mathbf{s})$ denotes the set of states $\mathbf{s}'$ such that there is a path from $\mathbf{s}$ to $\mathbf{s}'$ in TS.

An *attractor* A of $\mathsf{TS}_{\mathsf{BN}}$ (or of BN) is a subset of states of $\mathbf{S}$ such that for every $\mathbf{s} \in A$, $\mathsf{reach}_{\mathsf{TS}_{\mathsf{BN}}}(\mathbf{s}) = A$. Any state which is not part of an attractor is a *transient state*. An attractor A of BN is said to be reachable from a state $\mathbf{s}$ if $\mathsf{reach}_{\mathsf{TS}_{\mathsf{BN}}}(\mathbf{s}) \cap A \neq \emptyset$. Attractors represent the stable behaviour of the BN according to the dynamics. For an attractor A of BN, the *weak basin* or simply the *basin* of attraction of A, denoted $\mathsf{bas}_{\mathsf{TS}_{\mathsf{BN}}}(A)$, is a subset of states of $\mathbf{S}$ such that $\mathbf{s} \in \mathsf{bas}_{\mathsf{TS}_{\mathsf{BN}}}(A)$ if $\mathsf{reach}_{\mathsf{TS}_{\mathsf{BN}}}(\mathbf{s}) \cap A \neq \emptyset$. A *control* C is a (possibly empty) subset of N. For a state $\mathbf{s} \in \mathbf{S}$, the *application of control* C to $\mathbf{s}$, denoted $\mathsf{C}(\mathbf{s})$ is defined as the state $\mathbf{s}' \in \mathbf{S}$ such that $\mathbf{s}'[i] = (1 - \mathbf{s}[i])$ if $i \in \mathsf{C}$ and $\mathbf{s}'[i] = \mathbf{s}[i]$, otherwise. Henceforth, we drop the subscripts TS or BN or both when no ambiguity arises.

Control problems: In this work we shall exclusively deal with the notion of *existential control* in that, after the control C is applied to a state $\mathbf{s}$, there 'exists' a path from $\mathsf{C}(\mathbf{s})$ to the desired target attractor and also perhaps to other non-target attractors. This is different from the notion of *absolute control* dealt with in [10] where after the control, $\mathsf{C}(\mathbf{s})$ is 'guaranteed' to reach the target attractor. Although the techniques applied for the computation of the minimal control are similar in both cases, there are certain fundamental differences. Therefore, here we are interested in the following control problems given a network BN. Note that for us, the control is applied in a single time step (hence simultaneously) to the state $\mathbf{s}$ under consideration.

1. **Minimal existential target control:** Given a state $\mathbf{s} \in \mathbf{S}$ and a 'target attractor' A_t of BN, it is a control $\mathsf{C}_{\mathbf{s} \rightarrow A_t}$ such that after the application of $\mathsf{C}_{\mathbf{s} \rightarrow A_t}(\mathbf{s})$, BN can eventually reach A_t and $\mathsf{C}_{\mathbf{s} \rightarrow A_t}$ is a minimal such subset.
2. **Minimal existential all-pairs control:** Given a set $\mathcal{A} = \{A_1, A_2, \ldots, A_p\}$, $p \geq 2$, of attractors of BN, it is a minimal subset $\mathsf{C}_{\mathcal{A}}$ of N such that for any pair $A_i, A_j \in \mathcal{A}$ of attractors, there is a state $\mathbf{s} \in A_i$, such that $\mathsf{C}_{\mathbf{s} \rightarrow A_j} \subseteq \mathsf{C}_{\mathcal{A}}$.
3. **Minimal existential full control:** C_{BN} is the minimal existential all-pairs control $\mathsf{C}_{\mathcal{A}}$ when $\mathcal{A}$ is the set of all attractors of BN.

In this work we shall use ideas from the decomposition-based approach of [10] to compute (2) and (3). We first give the relevant definitions and results.

Let SCC denote the set of maximal strongly connected components (SCCs) of $\mathcal{G}_{\mathsf{BN}}$. A *basic block* B is a subset of nodes of BN such that $B = (S \cup \mathsf{par}(S))$ where S is a maximal SCC of $\mathcal{G}_{\mathsf{BN}}$. Let $\mathcal{B}$ denote the set of basic blocks of BN. The union of two or more basic blocks will also be called a *block*. Using the set of basic blocks as vertices, we can form a directed graph $\mathcal{G}_{\mathcal{B}} = (\mathcal{B}, E_{\mathcal{B}})$, called the *block graph* of BN, where for any pair of basic blocks $B', B \in \mathcal{B}, B' \neq B$, there is a directed edge from B' to B if and only if $B' \cap B \neq \emptyset$ and for every $v \in B' \cap B$, $\mathsf{par}(v) \cap B = \emptyset$. In such a case, B' is called a *parent* block of B and v is called a *control node* for B. The set of parent blocks of B is denoted as $\mathsf{par}(B)$.

A block is called *elementary* if $\mathsf{par}(B) = \emptyset$ and *non-elementary* otherwise. We shall henceforth assume that BN has k basic blocks, $|\mathcal{B}| = k$, and $\mathcal{G}_{\mathsf{BN}}$ is topologically sorted as $\{B_1, B_2, \ldots, B_k\}$. Given how $\mathcal{G}_{\mathsf{BN}}$ is constructed, it will be a directed acyclic graph and hence can always be topologically sorted. Note that for every $j : 1 \leq j \leq k$, $(\bigcup_{\ell=1}^{j} B_\ell)$ is an elementary block. We shall denote it as $\overline{B}_j$ and let $B_j^- = (B_j \setminus \overline{B}_{j-1})$. For two basic blocks B and B' where B is non-elementary, B' is said to be an *ancestor* of B if there is a path from B' to B in the block graph $\mathcal{G}_{\mathcal{B}}$. The *ancestor-closure* of a basic block B, denoted $\mathsf{ac}(B)$ is defined as the union of B with all its ancestors. Note that $\mathsf{ac}(B)$ is an elementary block and so is $(\mathsf{ac}(B) \setminus B^-)$, denoted as $\mathsf{ac}(B)^-$.

For a block B of BN, its state space is $\{0,1\}^{|B|}$ and is denoted as $\mathbf{S}_B$. For any state $\mathbf{s} \in \mathbf{S}$, where $\mathbf{s} = (s_1, s_2, \ldots, s_n)$, the projection of $\mathbf{s}$ to B, denoted $\mathbf{s}|_B$ is the tuple obtained from $\mathbf{s}$ by suppressing the values of the variables not in B. Let B_1 and B_2 be two blocks of BN and let $\mathbf{s}_1$ and $\mathbf{s}_2$ be states of B_1 and B_2, respectively. $\mathbf{s}_1 \otimes \mathbf{s}_2$ is defined (called *crossable*) if there exists a state $\mathbf{s} \in \mathbf{S}_{B_1 \cup B_2}$ such that $\mathbf{s}|_{B_1} = \mathbf{s}_1$ and $\mathbf{s}|_{B_2} = \mathbf{s}_2$. $\mathbf{s}_1 \otimes \mathbf{s}_2$ is then defined to be this unique state $\mathbf{s}$. For any subsets $\mathbf{S}_1$ and $\mathbf{S}_2$ of $\mathbf{S}_{B_1}$ and $\mathbf{S}_{B_2}$ resp. $\mathbf{S}_1 \otimes \mathbf{S}_2$ is a subset of $\mathbf{S}_{B_1 \cup B_2}$ and is defined as: $\mathbf{S}_1 \otimes \mathbf{S}_2 = \{\mathbf{s}_1 \otimes \mathbf{s}_2 \mid \mathbf{s}_1 \in \mathbf{S}_1, \mathbf{s}_2 \in \mathbf{S}_2 \text{ and } \mathbf{s}_1, \mathbf{s}_2 \text{ are crossable}\}$. The cross operation can be defined for more than two states $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$, as $\mathbf{s}_1 \otimes \mathbf{s}_2 \otimes \ldots \mathbf{s}_k = (((\mathbf{s}_1 \otimes \mathbf{s}_2) \otimes \ldots) \otimes \mathbf{s}_k)$. The cross operation can be similarly lifted to more than two sets of states.

The TS TS_B of an elementary block B of BN is defined similarly to the TS of BN, which can indeed be done as the update functions do not depend on vertices outside B. The attractors, basin of attractions, etc. of such a TS is also defined similarly. The TSs of a non-elementary basic block B are 'realised' by the basins of attractions of the attractors of $\mathsf{ac}(B)^-$, each such attractor realising a different TS. Thus, if A is an attractor of $\mathsf{ac}(B)^-$ then TS_B realised by $\mathsf{bas}(A)$ has set of states $\mathbf{S}$ which the maximum subset of $\mathbf{S}_{\mathsf{ac}(B)}$ such that $\mathbf{S}|_{\mathsf{ac}(B)^-} = \mathsf{bas}(A)$. The transitions are then defined as usual. The following is a key result, a counterpart of which was proved in [10], saying that the 'global' attractors of BN and their basins can be computed by first computing the 'local' attractors and basins of the basic blocks and then merging them using the cross operation.

Theorem 1 ([10]). *A is an attractor of BN iff there exist attractors A_j of B_j where $A_j = A|_{B_j}$ for all $j : 1 \leq j \leq k$ and $A = \otimes_j A_j$. Furthermore, $A|_{\mathsf{ac}(B_j)}$ is an attractor of $\mathsf{ac}(B_j)$ and $\mathsf{bas}(A) = \otimes_j \mathsf{bas}(A_j)$ w.r.t. their TSs.*

3 Results

In this section we develop our method for solving control problem (2). We first describe a 'global' approach that works on the entire BN and then modify it to exploit the decomposition-based approach of [10]. For simplicity, we assume that every attractor of BN is a single state with a self loop. The methods can be generalised for the case where an attractor can comprise of two or more states.

First, note that given a state $\mathbf{s}$ and an attractor A, for BN to potentially end up in A after the application of a control C, it is necessary and sufficient

that there is a path from $\mathsf{C}(\mathbf{s})$ to A in $\mathsf{TS}_{\mathsf{BN}}$ which means, by definition, that $\mathsf{C}(\mathbf{s}) \in \mathsf{bas}(A)$. Thus given a set $\mathcal{A}$ of attractors of BN to compute $\mathsf{C}_{\mathcal{A}}$ it is enough to compute the basins of the attractors in $\mathcal{A}$. This can be done by a repeated application of the $\mathsf{pre}(\cdot)$ operator on an attractor till a fixed point is reached. See [9] for a detailed description of this fixed point procedure.

So, assume that the given set of attractors $\mathcal{A}$ is sorted as $\{A_1, A_2, \ldots, A_p\}$. We then construct a $p \times p$ matrix M whose entries are subsets of N and are defined as: for every $I \subseteq N$, $I \in \mathsf{M}_{ij}$ if and only if $I = \arg(\mathsf{hd}(\mathbf{s}, \mathbf{s}'))$ where $\mathbf{s} \in A_i$ and $\mathbf{s}' \in \mathsf{bas}(A_j)$. That is, for every pair of attractors A_i and A_j the entries of M_{ij} record the indices of the variables that need to be toggled in state $\mathbf{s} \in A_i$ to end up in any of the states of the basin of A_j. The minimal all-pairs control $\mathsf{C}_{\mathcal{A}}$ is then nothing but a minimal subset of N such that for every i, j there exists $I \in \mathsf{M}_{ij}$ such that $I \subseteq \mathsf{C}_{\mathcal{A}}$.

We now describe a method to compute the set $\mathsf{C}_{\mathcal{A}}$ based on the power-set lattice of N, denoted by $\mathcal{L}$. Let $\ell : \mathcal{L} \to \wp(N \times N)$ be a labelling function that labels the elements of $\mathcal{L}$ with tuples in $(N \times N)$ defined as follows. For any element L of $\mathcal{L}$, $(i, j) \in \ell(L)$ iff $L \in \mathsf{M}_{ij}$. Let ℓ^* denote the closure of the labelling function of $\mathcal{L}$ under subsets, defined as: for every element L of $\mathcal{L}$, $\ell^*(L) = \bigcup_{L' \subseteq L} \ell(L')$. Finally, $\mathsf{C}_{\mathcal{A}}$ is any minimal element L of $\mathcal{L}$ such that $\ell^*(L) = (\{1, 2, \ldots, p\} \times \{1, 2, \ldots, p\}) \setminus \{(i, i) \mid i \in \{1, 2, \ldots, p\}\}$. Control problem (3) is a special case of (2) where $\mathcal{A}$ is the set of all attractors of BN. For solving (3), given a BN as input, we can first apply any of the methods available in the literature (e.g., see [8]) to compute the set of all attractors $\mathcal{A}$ of BN, and then invoke the above method.

In general, the problem of computing $\mathsf{C}_{\mathcal{A}}$ given the matrix M is NP-hard. Moreover, given a BN and an attractor A as input, the problem of computation of the strong basin of A is PSPACE hard. Hence, the control problem (2) is at least PSPACE-hard and so unlikely to have efficient algorithms for the general case. However, in [10] we show that using a decomposition-based approach we can improve the efficiency for many modular well-structured networks. We now describe a similar approach for solving control problem (2) [and hence (3)].

The method is iterative where instead of computing the basin of attractions of the given attractors for the entire BN in one-go, we decompose the BN into blocks, as described in the previous section, and compute the basins and also the minimal control w.r.t the basins of each such block. The basin of an attractor in a block can once again be computed using a repeated application of the $\mathsf{pre}(\cdot)$ operator in that block. The details are given in [9].

Suppose we are given a BN and a set of attractors $\mathcal{A}$ sorted as $\{A_1, A_2, \ldots, A_p\}$ as input. We proceed in the following steps:

1. We decompose BN into basic blocks $\mathcal{B}$, form the block graph $\mathcal{G}_{\mathcal{B}}$ and topologically sort it to obtain an ordering of the blocks as $\mathcal{B} = \{B_1, B_2, \ldots, B_k\}$.
2. Proceeding in the sorted order, for each block B_j we repeat the steps below:
   (a) Let $\hat{B}_j = (B_j \setminus (\bigcup_{r<j} B_r))$ and $I_j = \mathsf{ind}(\hat{B}_j)$.
   (b) Let M^j be a $p \times p$ matrix whose entries are subsets of I_j.

(c) Note that by Theorem 1, $A_r|_{\mathsf{ac}(B_j)}$ is an attractor of B_j, for every $r : 1 \leq r \leq p$. For every r, we compute $\mathsf{bas}(A_r|_{\mathsf{ac}(B_j)})$.
(d) We populate the matrix M^j as: for every $I \subseteq I_j$, $I \in \mathsf{M}^j_{qr}$ if and only if $I = (\arg(\mathsf{hd}(\mathbf{s}|_{\hat{B}_j}, \mathbf{s}'|_{\hat{B}_j})))$ for some $\mathbf{s} \in A_q|_{\mathsf{ac}(B_j)}$ and $\mathbf{s}' \in \mathsf{bas}(A_r|_{\mathsf{ac}(B_j)})$.
(e) Let $\mathcal{L}_j$ be the subset lattice of I_j and let ℓ_j label the elements of $\mathcal{L}_j$ with tuples in $(I_j \times I_j)$ such that for $L \in \mathcal{L}_j$, $(q, r) \in \ell_j(L)$ iff $L \in \mathsf{M}^j_{qr}$.
(f) Let ℓ^*_j denote the closure of ℓ_j under subsets and let $\mathsf{C}^j_{\mathcal{A}}$ be any minimal element L of $\mathcal{L}_j$ such that $\ell^*(L) = ((\{1, 2, \ldots, p\} \times \{1, 2, \ldots, p\}) \setminus \{(i, i) \mid i \in \{1, 2, \ldots, p\}\})$

3. Finally we let $\mathsf{C}_{\mathcal{A}} = \bigcup_{j=1}^{k} \mathsf{C}^j_{\mathcal{A}}$.

The above approach is worked-out in details on a toy example in [9].

4 Conclusion

In this report, we describe work-in-progress on the development of a procedure for the computation of a minimal subset of nodes required for the existential control of a given BN. Our procedure can be applied on the entire BN in one-go or on the 'blocks' of the BN locally and then later combined to derive the global control, whereby taking advantage of the decomposition-based approach towards the problem of target control of BNs that we have developed in [10]. We are currently implementing our procedure in software to test its efficacy and efficiency on various real-life and random BNs. We believe that our decomposition-based approach has great potential to efficiently solve the control problem for large real-life biological networks modelled as BNs that are modular and well-structured.

References

1. Gates, A.J., Rocha, L.M.: Control of complex networks requires both structure and dynamics. Scientific Reports **6**(24456) (2016)
2. Hopkins, A.L.: Network pharmacology: the next paradigm in drug discovery. Nature Chemical Biology **4**, 682–690 (2008)
3. Huang, S.: Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. Pharmacogenomics **2**(3), 203–222 (2001)
4. Kalman, R.E.: Mathematical description of linear dynamical systems. Journal of the Society for Industrial and Applied Mathematics **1**(2), 152–192 (1963)
5. Kauffman, S.: Homeostasis and differentiation in random genetic control networks. Nature **224**, 177–178 (1969)
6. Kim, J., Park, S.M., Cho, K.H.: Discovery of a kernel for controlling biomolecular regulatory networks. Scientific Reports **3**(2223) (2013)
7. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. Nature **473**, 167–173 (2011)
8. Mizera, A., Pang, J., Qu, H., Yuan, Q.: Taming asynchrony for attractor detection in large Boolean networks. IEEE/ACM TCBB (2018), in press
9. Paul, S., Pang, J., Su, C.: Towards the existential control of boolean networks (extended abstract). Tech. rep., UL (2018), http://arxiv.org/abs/1806.10927
10. Paul, S., Su, C., Pang, J., Mizera, A.: A decomposition-based approach towards the control of Boolean networks. In: Proc. ACM-BCB 2018. ACM (2018), in press