Towards the Existential Control of Boolean Networks: A Preliminary Report  
(Extended Abstract)

Soumya Paul\textsuperscript{1}, Jun Pang\textsuperscript{1,2}, and Cui Su\textsuperscript{1}

\textsuperscript{1} Interdisciplinary Centre for Security, Reliability and Trust  
\textsuperscript{2} Faculty of Science, Technology and Communication, University of Luxembourg

\texttt{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 $\mathcal{C}_{BN}$ of vertices of BN, such that the dynamics of BN can reach from a state $s$ in any attractor $A_s \in \mathcal{A}$ to any attractor $A_t \in \mathcal{A}$ by controlling (toggling) a subset of vertices in $\mathcal{C}_{BN}$ 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 $\mathcal{C}_{BN}$. 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 [3]. 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 [6], 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 [6] and also correspond to functional cellular states such as proliferation, apoptosis, differentiation, etc. [4]. 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 [5] 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 [8]. 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. [7] 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.

The BNs used to model real-life biological networks have multiple attractors, the sizes and distribution of which are governed by certain power laws [2]. However in most of the cases only some of these attractors are ‘biologically relevant’, i.e. correspond to meaningful expressions of the GRNs. Thus, focusing 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 \( \mathcal{A}_s \in \mathcal{A} \) and by controlling specific subsets of these variables in a single time-step, the BN can (potentially) reach any desired target attractor \( \mathcal{A}_t \in \mathcal{A} \) 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 $\text{BN} = (x, f)$ where $x = (x_1, x_2, \ldots, x_n)$ such that each $x_i$ is a Boolean variable and $f = (f_1, f_2, \ldots, f_n)$ is a tuple of Boolean functions over $x$. In what follows, $i$ will always range over $N$, unless stated otherwise. A Boolean network $\text{BN} = (x, f)$ may be viewed as a directed graph $\mathcal{G}_{\text{BN}} = (V, E)$, 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 any $i, j \in N$, there is a directed edge from $v_j$ to $v_i$, often denoted as $v_j \rightarrow v_i$, if and only if $f_i$ depends on $x_j$. Thus $V$ is ordered according to the ordering of $x$. For any vertex $v_i \in V$, we let $\text{ind}(v_i) = i$ be the index of $v_i$ in this ordering. For any subset $W$ of $V$, $\text{ind}(W) = \{\text{ind}(v) \mid v \in W\}$. A path from a vertex $v$ to a vertex $v'$ is a (possibly empty) sequence of edges from $v$ to $v'$ in $\mathcal{G}_{\text{BN}}$. For any vertex $v \in V$ we define its set of parents as $\text{par}(v) = \{v' \in V \mid v' \rightarrow v\}$ and for any subset $W$ of $V$, $\text{par}(W) = \{\text{par}(v) \mid v \in W\}$. For the rest of the exposition, we assume that an arbitrary but fixed network $\text{BN}$ of $n$ variables is given to us and $\mathcal{G}_{\text{BN}} = (V, E)$ is its associated directed graph.

A state $s$ of $\text{BN}$ is an element in $\{0, 1\}^n$. Let $S$ be the set of states of $\text{BN}$. For any state $s = (s_1, s_2, \ldots, s_n)$, and for every $i$, the value of $s_i$, often denoted as $s[i]$, represents the value that the variable $x_i$ takes when the $\text{BN}$ is in state $s$. For some $i$, suppose $f_i$ depends on $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. Then $f_i(s)$ will denote the value $f_i(s_{i_1}, s_{i_2}, \ldots, s_{i_k})$. For two states $s, s' \in S$, the Hamming distance between $s$ and $s'$ will be denoted as $\text{hd}(s, s')$ and $\arg(\text{hd}(s, s')) \subseteq N$ will denote the set of indices in which $s$ and $s'$ differ. For a state $s$ and a subset $S' \subseteq S$, the Hamming distance $\text{hd}(s, S')$ is defined as $\text{hd}(s, S') = \min_{s' \in S'} \text{hd}(s, s')$. We let $\arg(\text{hd}(s, S'))$ denote the set of subsets of $N$ such that $I \in \arg(\text{hd}(s, S'))$ if and only if $I$ is a set of indices of the variables that realise $\text{hd}(s, S')$.

We assume that the Boolean network starts initially in a state $s_0$ and its state changes in every discrete time-step according to the update functions $f$. In this work, we shall deal with the asynchronous updating scheme but all our results transfer to the synchronous updating scheme as well. Suppose $s_0 \in S$ is an initial state of $\text{BN}$. The asynchronous evolution of $\text{BN}$ is a function $\xi : N \rightarrow \rho(S)$ such that $\xi(0) = s_0$ and for every $j \geq 0$, if $s \in \xi(j)$ then $s' \in \xi(j + 1)$ if and only if $\text{hd}(s, s') = 1$ and $s'[i] = f_i(s)$ where $i = \arg(\text{hd}(s, s'))$ or $\text{hd}(s, s') = 0$ and there exists $i$ such that $s'[i] = f_i(s)$.

The dynamics of a Boolean network can be represented as a state transition graph or a transition system (TS). The transition system of $\text{BN}$, denoted as $\mathcal{T}_{\text{BN}}$,
is a tuple $(S, \rightarrow)$ where the vertices are the set of states $S$ and for any two states $s$ and $s'$ there is a directed edge from $s$ to $s'$, denoted $s \rightarrow s'$, if and only if either $\text{hd}(s, s') = 1$ and $s'[i] = f_i(s)$ where $i = \text{arg}(\text{hd}(s, s'))$ or $\text{hd}(s, s') = 0$ and there exists $i$ such that $s'[i] = f_i(s)$.

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

An attractor $A$ of $\text{TS}_B$ (or of $B$) is a subset of states of $S$ such that for every $s \in A$, $\text{reach}_{\text{TS}_B}(s) = A$. Any state which is not part of an attractor is a transient state. An attractor $A$ of $B$ is said to be reachable from a state $s$ if $\text{reach}_{\text{TS}_B}(s) \cap A \neq \emptyset$. Attractors represent the stable behaviour of the $B$ according to the dynamics. For an attractor $A$ of $B$, the weak basin or simply the basin of attraction of $A$, denoted $\text{bas}_{\text{TS}_B}(A)$, is a subset of states of $S$ such that $s \in \text{bas}_{\text{TS}_B}(A)$ if $\text{reach}_{\text{TS}_B}(s) \cap A \neq \emptyset$. A control $C$ is a (possibly empty) subset of $N$. For a state $s \in S$, the application of control $C$ to $s$, denoted $C(s)$ is defined as the state $s' \in S$ such that $s'[i] = (1 - s[i])$ if $i \in C$ and $s'[i] = s[i]$, otherwise. Henceforth, we drop the subscripts $\text{TS}$ or $B$ 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 $s$, there ‘exists’ a path from $C(s)$ 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 [11] where after the control, $C(s)$ 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. In particular, here we are interested in the following control problems given a network $B$. Note that for us, the control is applied in a single time step (hence simultaneously) to the state $s$ under consideration.

1. **Minimal existential target control:** Given a state $s \in S$ and a ‘target attractor’ $A_t$ of $B$, it is a control $C_{s \rightarrow A_t}$ such that after the application of $C_{s \rightarrow A_t}(s)$, $B$ can eventually reach $A_t$ and $C_{s \rightarrow A_t}$ is a minimal such subset.

2. **Minimal existential all-pairs control:** Given a set $A = \{A_1, A_2, \ldots, A_p\}$, $p \geq 2$, of attractors of $B$, it is a minimal subset $C_A$ of $N$ such that for any pair $A_i, A_j \in A$ of attractors, there is a state $s \in A_i$ such that $C_{s \rightarrow A_j} \subseteq C_A$.

3. **Minimal existential full control:** $C_B$ is the minimal existential all-pairs control $C_A$ when $A$ is the set of all attractors of $B$.

In this work we shall use ideas from the decomposition-based approach of [11] to compute (2) and (3). We first give the relevant definitions and results.

Let $\text{SCC}$ denote the set of maximal strongly connected components (SCCs) of $\mathcal{G}_B$. A basic block $B$ is a subset of nodes of $B$ such that $B = (S \cup \text{par}(S))$ where $S$ is a maximal SCC of $\mathcal{G}_B$. Let $B$ denote the set of basic blocks of $B$. The union of two or more basic blocks will also be called a block. Using the set of
describe a ‘global’ approach that works on the entire BN and then modify it.

In this section we develop our method for solving control problem (2). We first

3 Results

basic blocks as vertices, we can form a directed graph \( G_B = (B, E_B) \), called the
block graph of BN, where for any pair of basic blocks \( B', B \in B, B' \neq B \), there is
a directed edge from \( B' \) to \( B \) if and only if \( B' \cap B \neq \emptyset \) and for every \( v \in B' \cap B \),
\( \text{par}(v) \cap B = \emptyset \). 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 \( \text{par}(B) \).

A block is called elementary if \( \text{par}(B) = \emptyset \) and non-elementary otherwise.
We shall henceforth assume that BN has \( k \) basic blocks, \( |B| = k \), and \( G_B \) is
topologically sorted as \( \{B_1, B_2, \ldots, B_k\} \). Given how \( G_B \) is constructed, it will
be a directed acyclic graph and hence can always be topologically sorted. Note
that for every \( j : 1 \leq j \leq k \), \( (\bigcup_{i=1}^j B_i) \) is an elementary block. We shall denote
it as \( B_j \) 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 \( G_B \). The ancestor-closure of a basic block \( B \), denoted
\( \text{ac}(B) \) is defined as the union of \( B \) with all its ancestors. Note that \( \text{ac}(B) \) is an
elementary block and so is \( (\text{ac}(B) \setminus B^-) \), denoted as \( \text{ac}(B)^- \).

For a block \( B \) of BN, its state space is \( \{0, 1\}^{|B|} \) and is denoted as \( S_B \). For
any state \( s \in S_B \), where \( s = (s_1, s_2, \ldots, s_n) \), the projection of \( s \) to \( B \), denoted \( s|_B \)
is the tuple obtained from \( s \) by suppressing the values of the variables not in \( B \).
Let \( B_1 \) and \( B_2 \) be two blocks of BN and let \( s_1 \) and \( s_2 \) be states of \( B_1 \) and \( B_2 \),
respectively. \( s_1 \otimes s_2 \) is defined (called crossable) if there exists a state \( s \in S_{B_1 \cup B_2} \)
such that \( s|_{B_1} = s_1 \) and \( s|_{B_2} = s_2 \). \( s_1 \otimes s_2 \) is then defined to be this unique state
\( s \). For any subsets \( S_1 \) and \( S_2 \) of \( S_B \) and \( S_{B_1} \) resp. \( S_1 \otimes S_2 \) is a subset of \( S_{B_1 \cup B_2} \) and
is defined as: \( S_1 \otimes S_2 = \{ s_1 \otimes s_2 \mid s_1 \in S_1, s_2 \in S_2 \) and \( s_1, s_2 \) are crossable \}.
The cross operation can be defined for more than two states \( s_1, s_2, \ldots, s_k \), as
\( s_1 \otimes s_2 \otimes \ldots s_k = ((s_1 \otimes s_2) \otimes \ldots) \otimes s_k \). The cross operation can be similarly
lifted to more than two sets of states.

The TS \( T_{B_B} \) 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 \( \text{ac}(B)^- \), each such attractor realising a different
TS. Thus, if \( A \) is an attractor of \( \text{ac}(B)^- \) then \( T_{B_B} \) realised by \( \text{bas}(A) \) has set of
states \( S \) which the maximum subset of \( S_{\text{ac}(B)} \) such that \( S|_{\text{ac}(B)^-} = \text{bas}(A) \). The
transitions are then defined as usual. The following is a key result, a counterpart
of which was proved in [11], 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** ([11]). \( A \) is an attractor of BN iff there exist attractors \( A_j \) of \( B_j \)
such that \( A_j = A|_{B_j} \) for all \( j : 1 \leq j \leq k \) and \( A = \otimes_j A_j \). Furthermore, \( A|_{\text{ac}(B)} \)
is an attractor of \( \text{ac}(B_j) \) and \( \text{bas}(A) = \otimes_j \text{bas}(A_j) \) 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
Algorithm 1 Computation of the basin of attraction

1: procedure ComputeBasin($f, A$)
2: Bas = A
3: Till $\text{pre}(\text{Bas}) \neq \text{Bas}$ do
4: Bas = $\text{pre}(\text{Bas})$
5: done
6: return Bas
7: end procedure

to exploit the decomposition-based approach of [11]. 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 $s$ 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 $C(s)$ to $A$ in $TS_{BN}$ which means, by definition, that $C(s) \in \text{bas}(A)$. Thus given a set $A$ of attractors of $BN$ to compute $C_A$ it is enough to compute the basins of the attractors in $A$. This can be done, starting from any attractor in $A$, by a repeated application of the $\text{pre}(\cdot)$ operator till a fixed point is reached. The procedure $\text{ComputeBasin}$ described in Algorithm 1 does exactly this.

So, assume that the given set of attractors $A$ 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 $I \subseteq N$, $I \in M_{ij}$ if and only if $I = \arg(\text{hd}(s, s'))$ where $s \in A_i$ and $s' \in \text{bas}(A_j)$. 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 $s \in A_i$ to end up in any of the states of the basin of $A_j$. The minimal all-pairs control $C_A$ is then nothing but a minimal subset of $N$ such that for every $i, j$ there exists $I \in M_{ij}$ such that $I \subseteq C_A$.

We now describe a method to compute the set $C_A$ 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 $L \in M_{ij}$. Let $\ell^*$ 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, $C_A$ 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 $A$ 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 [9, 10]) to compute the set of all attractors $A$ of $BN$, and then invoke the above method.

In general, the problem of computing $C_A$ 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 [11] 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 transition system of each such block. The basin of an attractor in a block can once again be computed using a repeated application of the pre\( (\cdot)\) operator in that block. The details are given in Algorithm 2.

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

1. We decompose BN into basic blocks \( B \), form the block graph \( G_B \) and topologically sort it to obtain an ordering of the blocks as \( B = \{B_1, B_2, \ldots, B_k\} \).
2. Proceeding in the sorted order, for each block \( B_j \) we repeat the steps below:
   (a) Let \( B_j' = (B_j \setminus (\bigcup_{r<j} B_r)) \) and \( I_j = \text{ind}(B_j) \).
   (b) Let \( M^j \) be a \( p \times p \) matrix whose entries are subsets of \( I_j \).
   (c) Note that by Theorem 1, \( A_r|_{ac(B_j)} \) is an attractor of \( B_j \), for every \( r : 1 \leq r \leq p \). For every \( r \), we compute \( \text{bas}(A_r|_{ac(B_j)}) \) using the basin of the parent block of \( B_j \), \( \text{bas}(A_r|_{ac(B_j)}) \), by the application of the procedure Algorithm 2.
   (d) We populate the matrix \( M^j \) as: for every \( I \subseteq I_j \), \( I \in M_{gr}^j \) if and only if \( I = (\arg \text{hd}(s|_{B_j}, s'|_{B_j})) \) for some \( s \in A_{\cup ac(B_j)} \) and \( s' \in \text{bas}(A_r|_{ac(B_j)}) \).
   (e) Let \( L_j \) be the subset lattice of \( I_j \) and let \( \ell_j \) label the elements of \( L_j \) with tuples in \( (I_j \times I_j) \) such that for \( L \in L_j \), \( (q, r) \in \ell_j(L) \) if \( L \in M_{gr}^j \).
   (f) Let \( \ell''_j \) denote the closure of \( \ell_j \) under subsets and let \( C_A \) be any minimal element \( L \) of \( L_j \) 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 \( C_A = \bigcup_{j=1}^p C_A \).

The above approach is worked-out in details on a toy example in the next section.

4 A Detailed Example

Consider the four-node Boolean network \( BN = (x, f) \) where \( x = (x_1, x_2, x_3, x_4) \) and \( f = (f_1, f_2, f_3, f_4) \) where \( f_1 = \neg x_2 \lor (x_1 \land x_2), f_2 = x_1 \land x_2, f_3 = x_4 \lor (\neg x_2 \land x_3) \)
Fig. 1: The graph of $\mathcal{BN}$ and its transition system.

and $f_4 = \neg x_3 \land x_4$. The graph of the network $\mathcal{G}_{\mathcal{BN}}$ and its associated transition system $\mathcal{TS}$ is given in Figure 1. Note that every state of $\mathcal{TS}$ also has a self loop (an edge to itself) which we have not shown in the figure to avoid clutter, but is implicit. $\mathcal{TS}$ has three attractors $\mathcal{A} = \{A_1, A_2, A_3\}$ shown by dark grey rectangles, where $A_1 = \{(1000)\}$, $A_2 = \{(1100)\}$ and $A_3 = \{(1010)\}$. Their corresponding basins of attractions are shown by enclosing grey regions of a lighter shade. Note that the states (0110), (0111) and (0101) are in the basins of both the attractors $A_1$ and $A_3$.

Now, suppose the relevant set of attractors given is $\mathcal{A} = \{A_2, A_3\}$. We construct the $2 \times 2$ matrix $M$ as given in Table 1, where the elements of $M_{A_2A_3}$ are the subsets of $\{1, 2, 3, 4\}$ needed to be controlled (toggled) in $A_2$ to end up in one of the states of the basin of $A_3$ and those of $M_{A_3A_2}$ are the subsets $\{1, 2, 3, 4\}$ needed to be controlled in $A_3$ to move to the basin of $A_2$.

|     | $A_2$                           | $A_3$                           |
|-----|--------------------------------|--------------------------------|
| $A_2$| $\{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}, \{1,2,3\}$|                                    |
|     | $\{1,2,4\}, \{1,3,4\}, \{2,3,4\}, \{1,2,3,4\}$                         |
| $A_3$| $\{2\}, \{2,3\}, \{2,4\}, \{2,3,4\}$                               |

Table 1: The $2 \times 2$ matrix $M$.

We next construct the subset lattice $\mathcal{L}$ for $\{1, 2, 3, 4\}$ and label each element $L$ of $\mathcal{L}$ with tuples in $\{(2, 3) \times \{2, 3\}\}$ as: $L$ is labelled with $(i, j)$ if and only if $L$ is an element of $M_{A_iA_j}$ (Figure 2).

Next, we take the subset closure of the labels of the elements of $\mathcal{L}$. That is, we label every element of $\mathcal{L}$ with the label of itself and that of all its subsets. The resulting labelling of $\mathcal{L}$ is shown in Figure 3.

We find that $\{2, 3\}$ and $\{2, 4\}$ are the minimal elements of $\mathcal{L}$ whose label contains both $(2, 3)$ and $(3, 2)$, i.e., all the pairs of the indices of the attractors in $\mathcal{A}$. Hence, we conclude that $C_\mathcal{A}$ is equal to either $\{2, 3\}$ or $\{2, 4\}$.

However, as mentioned in Section 3, the problem of computing the set $C_\mathcal{A}$ from the matrix $M$ is NP-hard in general and the method based on the construction of the subset lattice $\mathcal{L}$ can be clearly exponential in $N$, the number...
Fig. 2: The subset lattice of \{1, 2, 3, 4\} with corresponding labels from \(M\).

Fig. 3: The subset lattice of \{1, 2, 3, 4\} with labels closed under subsets.
of variables of the BN. We can take advantage of the decomposition-based approach [9–11] on certain well-structured modular BNs by splitting it into ‘blocks’ and performing the computations locally on the blocks. This can improve the efficiency for many real-life biological networks whose BN models have such modular structures. We described the approach towards the end of Section 3. We now demonstrate it on our example BN.

Towards that, first note that BN has two SCCs $S_1 = \{v_1, v_2\}$ and $S_2 = \{v_3, v_4\}$. No node of $S_1$ has a parent node that does not belong to $S_1$. Hence $S_1$ forms the elementary block $B_1 = S_1$. Next, the parent of the node $v_3$ of $S_2$ is $v_2$. Hence $S_2$ forms the non-elementary block $B_2 = \{v_2, v_3, v_4\}$ where $B_1$ is its parent block and $v_2$ is its control node. We have $\hat{B}_2 = (B_2 \setminus B_1) = \{v_3, v_4\}$. The block structure of BN is shown in Figure 4.

Now, $A_2|_{B_1} = A_2^1 = \{(11)\}$ and $A_3|_{B_1} = A_3^1 = \{(10)\}$ and by Theorem 1 we know that both $A_2^1$ and $A_3^1$ are attractors of $B_1$. $\text{bas}(A_2^1)$ and $\text{bas}(A_3^1)$ are shown in Figure 5(a) where we again drop the self-loops (transitions to itself) present in all the states. The attractors $A_2^1$ and $A_3^1$ are shown in dark grey rectangles and their corresponding basins of attractions, $\text{bas}(A_2^1)$ and $\text{bas}(A_3^1)$ are shown in enclosing lighter grey regions.

Now, $\text{ac}(B_2) = B_1 \cup B_2 = V$ and $\hat{B}_2 = B_2 \setminus B_1 = \{v_2, v_3\}$. Hence $A_2|_{\text{ac}(B_2)} = A_2^2 = A_2$ and $A_3|_{\text{ac}(B_2)} = A_3^2 = A_3$ are attractors of $\text{ac}(B_2)$ by Theorem 1. We then compute $\text{bas}(A_2^2)$ and $\text{bas}(A_3^2)$. The attractors $A_2^2$ and $A_3^2$ are shown in
the lattices is \(4 + 4 = 8\). Thus, we believe that for many well-structured real-life BNs the decomposition-based approach for computing the minimal (all-pairs and full) control might be more efficient than a global approach. This was already shown by us in [11] for the case of target control and we are currently extending.

We next look at the basins of the attractors \(A_1^1\) and \(A_3^1\) of block \(B_1\) to compute the \(2 \times 2\) matrix \(M^1\), shown in Table 2a, recording the control needed to move to the basin of attractor \(A_1^1\) from attractor \(A_3^1\) and vice-versa. The elements of \(M^1\) are subsets of \(\text{ind}(B_1) = \{1, 2\}\), the indices of the vertices in \(B_1\). We similarly look at the basins of attractors of \(A_1^2\) and \(A_3^2\) to compute the \(2 \times 2\) matrix \(M^2\) shown in Table 2b. In \(M^2\) we record the control needed to move \(A_1^2\) to the basin of \(A_3^2\) and vice-versa. The elements of \(M^2\) are subsets of \(\text{ind}(B_2) = \{3, 4\}\), the indices of the vertices in \((B_2 \setminus B_1)\).

![Table 2: The control matrices for the blocks \(B_1\) and \(B_2\).](image)

We next construct the subset lattices \(\mathcal{L}_1\) and \(\mathcal{L}_2\) corresponding to \(M^1\) and \(M^2\), respectively. The elements of \(\mathcal{L}_1\) and \(\mathcal{L}_2\) are subsets of \(\text{ind}(B_1) = \{1, 2\}\) and those of \(\mathcal{L}_2\) are subsets of \(\text{ind}(B_2) = \{3, 4\}\). Each element \(L\) of \(\mathcal{L}_1\) is labelled with tuples in \((\{2, 3\} \times \{2, 3\})\) [corresponding to pairs of attractors in \(\{A_1^2, A_1^4\}\)] as: \(L\) is labelled with \((i, j)\) if and only if \(L\) is an element of \(M^1_{A_iA_j}\). Similarly, each element \(L\) of \(\mathcal{L}_2\) is labelled with tuples in \((\{2, 3\} \times \{2, 3\})\) [corresponding to pairs of attractors in \(\{A_2^2, A_2^4\}\)] as: \(L\) is labelled with \((i, j)\) if and only if \(L\) is an element of \(M^2_{A_iA_j}\). The lattices \(\mathcal{L}_1\) and \(\mathcal{L}_2\) with their labels are shown in Figure 6.

We then take the closure of the labels of the elements of \(\mathcal{L}_1\) and \(\mathcal{L}_2\) under subsets. That is, we label every element with the label of itself and that of all its subsets. The resulting labelling of \(\mathcal{L}_1\) and \(\mathcal{L}_2\) is shown in Figure 7. We find that in \(\mathcal{L}_1\), \(C^1 = \{2\}\) is the minimal element of \(\mathcal{L}_1\) whose label contains both \(\{2, 3\}\) and \(\{3, 2\}\), i.e., all the pairs of the indices of the attractors \(\{A_1^2, A_1^4\}\). Similarly, in \(\mathcal{L}_2\), \(C^2\) is either \(\{3\}\) or \(\{4\}\). Each is the minimal element of \(\mathcal{L}_2\) whose label contains both the pairs \(\{2, 3\}\) and \(\{3, 2\}\), i.e., all the pairs of the indices of the attractors \(\{A_2^2, A_2^4\}\). Combining, we have that \(C_A = C_{BN} = C^1 \cup C^2\) is either \(\{2, 3\}\) or \(\{2, 4\}\).

Note that the lattice for the full TS of \(BN\), \(L\) had \(2^4 = 16\) elements. On the other hand, for the decomposition-based approach we computed two smaller lattices \(\mathcal{L}_1\) and \(\mathcal{L}_2\) each of which has \(2^2 = 4\) elements and hence the total size of the lattices is \(4 + 4 = 8\). Thus, we believe that for many well-structured real-life BNs the decomposition-based approach for computing the minimal (all-pairs and full) control might be more efficient than a global approach. This was already shown by us in [11] for the case of target control and we are currently extending.
Fig. 6: The subset lattices for the blocks.

Fig. 7: The subset lattices for the blocks with labels closed under subsets.

our implementation to (all-pairs and full) control to test in efficiency on various real-life BNs for biological systems.

5 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 [11]. 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. Greil, F., Bassler, K.E.: Attractor period distribution for critical Boolean networks. ArXiv e-prints (2009)
3. Hopkins, A.L.: Network pharmacology: the next paradigm in drug discovery. Nature Chemical Biology 4, 682–690 (2008)
4. Huang, S.: Genomics, complexity and drug discovery: insights from Boolean network models of cellular regulation. Pharmacogenomics 2(3), 203–222 (2001)
5. Kalman, R.E.: Mathematical description of linear dynamical systems. Journal of the Society for Industrial and Applied Mathematics 1(2), 152–192 (1963)
6. Kauffman, S.: Homeostasis and differentiation in random genetic control networks. Nature 224, 177–178 (1969)
7. Kim, J., Park, S.M., Cho, K.H.: Discovery of a kernel for controlling biomolecular regulatory networks. Scientific Reports 3(2223) (2013)
8. Liu, Y.Y., Slotine, J.J., Barabási, A.L.: Controllability of complex networks. Nature 473, 167–173 (2011)
9. Mizera, A., Pang, J., Qu, H., Yuan, Q.: A new decomposition method for attractor detection in large synchronous Boolean networks. In: Proc. 3rd International Symposium on Dependable Software Engineering: Theories, Tools, and Applications. LNCS, vol. 10606, pp. 232–249. Springer (2017)
10. 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 (Special issue of APBC’18) (2018)
11. 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 (2018), in press