# An FVS-based Approach to Attractor Detection in Asynchronous Boolean Networks*

Trinh Van Giang[1], Tatsuya Akutsu[2], Kunihiko Hiraishi[1]

[1]Japan Advanced Institute of Science and Technology, Japan
[2]Institute for Chemical Research, Kyoto University, Japan

June 28, 2021

*This work has been accepted to IEEE/ACM Transactions on Computational Biology and Bioinformatics.

# Contents

# Contents

# Boolean networks and attractors

- Boolean Networks (BNs) play an increasingly crucial role in modeling and analysis of complex biological systems (e.g., gene regulatory networks [Kauffman, 1993]). They have also been applied to many areas beyond systems biology, such as mathematics, neural networks, social modeling, and robotics (see, e.g., [Valverde et al., 2020]).
- A central aim of BN analysis is to find attractors, which are important long-term behavior of BNs.
  - Attractor detection could provide new insights into systems biology, such as the origins of cancer [Yamanaka, 2009].
  - Attractors also play an important role in the development of new drugs [Tun et al., 2011] and give a starting point for many control approaches for biological systems [Biane and Delaplace, 2018].

# Synchronous vs. Asynchronous

- Synchronous BNs [Garg et al., 2008]:
  - The updating scheme of SBNs is synchronous and deterministic, i.e., all the nodes are updated simultaneously at each time step.
- Asynchronous BNs [Garg et al., 2008]:
  - The updating scheme of ABNs is asynchronous and non-deterministic, i.e., only one node is randomly selected to be updated at each time step.
- ABNs are considered more suitable than SBNs in modeling biological networks [Thomas, 1991, Saadatpour et al., 2010].
  - In biology, the updating process of each component may spend various time from fractions of a second to hours.
  - Moreover, the information on time scales of components is usually lacking.

# Motivations

- Whereas many efficient algorithms and tools (see, e.g., [Garg et al., 2008, Dubrova and Teslenko, 2011, Yuan et al., 2016, He et al., 2018, Yuan et al., 2019]) have been developed for attractor detection in SBNs, few methods (see, e.g., [Garg et al., 2008, Skodawessely and Klemm, 2011, Mizera et al., 2018]) have been proposed for attractor detection in ABNs due to the high complexity of the State Transition Graph (STG) of an ABN.
- Moreover, the efficiency of these few methods is strictly prevented when the ABN becomes large, e.g., the number of nodes is over 100.
- Therefore, it is important and interesting to develop efficient methods that can handle larger networks.

# Contents

# Boolean networks

## Boolean Network (BN)

A BN $\mathcal{N}$ is a 2-tuple $(V, F)$ where $V = \{x_1, x_2, ..., x_n\}$ is a set of $n$ nodes and $F = \{f_1, f_2, ..., f_n\}$ is a set of $n$ Boolean functions. The $i$th node is associated with a Boolean variable $x_i \in \{0, 1\}$ and a Boolean function $f_i : \{0, 1\}^{|IN(f_i)|} \rightarrow \{0, 1\}$, where $IN(f_i)$ denotes the set of input nodes of Boolean function $f_i$. Note that we use $x_i$ to refer to the $i$th node as well as its associated Boolean variable.

In this research, BNs are implicitly considered as general Boolean networks (i.e., there is no restriction on Boolean functions).

# Dynamics of Boolean networks

- A *state* of the BN is given by a vector $x = (x_1, ..., x_n) \in \{0, 1\}^n$.
- At each time step, node $x_i$ can update its value by $x'_i = f_i(x)$, where $x$ is the current state of the BN and $x'_i$ is the next value of $x_i$. For simplicity, we use the notation $f_i(x)$ even if $IN(f_i) \subset V$.
- An *updating scheme* specifies the way the nodes will be updated.
- The BN can transit from a state to a state following its updating scheme. This transition is the *state transition*.
- The dynamics of a BN is captured by a *State Transition Graph* (STG). An STG is a directed graph whose nodes and arcs denote states and state transitions, respectively.

# Asynchronous Boolean Networks

- ABNs were first studied by Harvey and Bossomaier [Harvey and Bossomaier, 1997].
- The updating scheme is asynchronous and non-deterministic, since only one randomly selected node is updated at each time step.
- The STG of an ABN has exactly $2^n$ nodes and may have up to $n \times 2^n$ arcs.
- Note that there are some other types of asynchronous Boolean models (but less popular than ABNs), such as Generalized Asynchronous Boolean Networks (GABNs) [Gershenson, 2004], Deterministic Generalized Asynchronous Boolean Networks (DGABNs) [Gershenson et al., 2003], Random Order Asynchronous Boolean Networks (ROABNs) [Saadatpour et al., 2010].

# Attractors

## Attractor [Mizera et al., 2018]

An *attractor* of a BN is a set of states satisfying any state in this set can reach any state in this set and cannot reach any other state that is not in this set.

An attractor of an ABN can be either

- a singleton attractor (or a fixed point) that has only one state;
- or a cyclic attractor that has at least two states and is formed by overlapping one or more cycles of states.

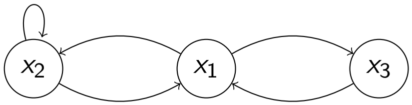
# Interaction graph

The *interaction graph* of a BN $\mathcal{N}$, denoted by $IG(\mathcal{N})$, is the directed graph defined as follows: the set of nodes is $V$ and, for all $x_i, x_j \in V$ (not necessarily distinct), there is an arc $(x_j, x_i)$ if $x_j \in IN(f_i)$ and $x_j$ properly affects the value of $x_i$ (e.g., $f_1 = x_2 \wedge \neg x_2$, $x_2 \in IN(f_1)$ but $x_2$ does not affect the value of $x_1$).

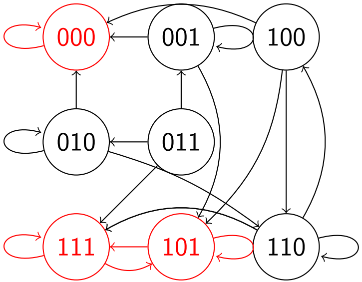
# Example of Boolean networks

## Example 1

Consider an ABN $\mathcal{A}$ of three nodes. Its Boolean functions are given by:
$f_1 = x_2 \vee x_3, \qquad f_2 = x_1 \wedge \neg x_2, \qquad f_3 = x_1.$

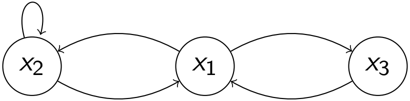



(a) Interaction graph of $\mathcal{A}$.




(b) STG of $\mathcal{A}$.

# Feedback vertex set




- Feedback Vertex Set (FVS) is an important concept in graph theory.
- An FVS of a directed graph is a set of nodes such that removing them makes the graph acyclic.
- For example, the interaction graph of the ABN in Example 1 has three possible FVSs including $\{x_1, x_2\}$, $\{x_2, x_3\}$, $\{x_1, x_2, x_3\}$. $\{x_1, x_2\}$ and $\{x_2, x_3\}$ are two minimum FVSs.

# Contents

# Relations between FVSs and BNs

The following lemmas and theorems do not depend on the updating scheme of the BN.

**Lemma 1**

Let $\mathcal{N}$ be a BN whose interaction graph is acyclic. Then the STG of $\mathcal{N}$ has no cycles.

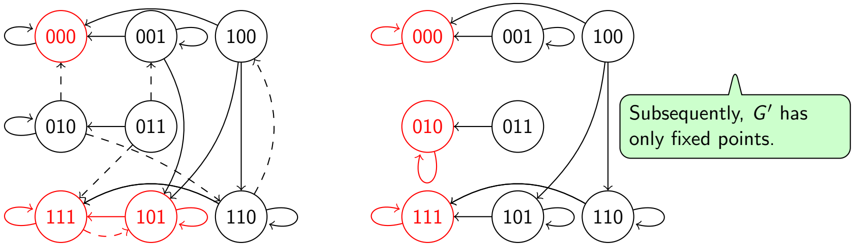
**Lemma 2**

Let $\mathcal{N}$ be a BN and its STG be $G$. Let $U$ be an FVS of $\mathcal{N}$. Then $G$ has no cycles such that the values of the nodes in $U$ do not change through these cycles.

# Relations between FVSs and BNs (cont.)

## Theorem 1

Let $\mathcal{N}$ be a BN and its STG be $G$. Let $U = \{x_{i_1}, ..., x_{i_k}\}$ be an FVS of $\mathcal{N}$. Let $B = \{b_{i_1}, ..., b_{i_k}\}$ be the set of retained values corresponding to the nodes of $U$. $G'$ is the graph obtained by removing all arcs $(x, x')$ from $G$ where $\bigvee_{j=1}^{k}(x_{i_j} \leftrightarrow b_{i_j} \wedge x'_{i_j} \leftrightarrow 1 - b_{i_j})$ holds. This means an arc $(x, x')$ will be removed if it changes at least one node $x_{i_j} \in U$ from $b_{i_j}$ to $1 - b_{i_j}$. Then $G'$ has no cycles.



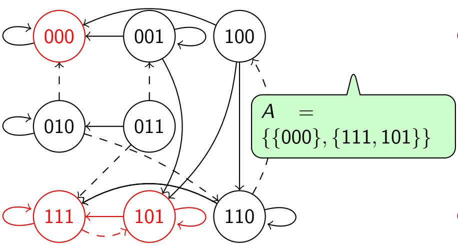

(a) $G$

(b) $G'$ with $b_1 = 0, b_2 = 1$

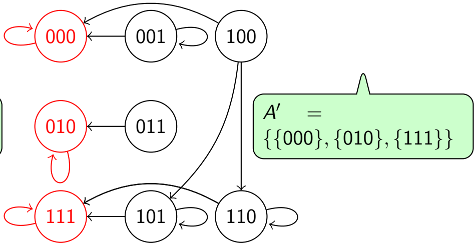
# Relations between FVSs and BNs (cont.)

## Theorem 2

Let $\mathcal{N}$ be a BN and its STG be $G$. $G'$ is the graph obtained by removing arcs from $G$. Let $A$ and $A'$ be the sets of attractors of $G$ and $G'$, respectively. Then, the exists a mapping $m : A \to A'$ with $m(att) \subseteq att$ for all $att \in A$ and $m(att_1) \neq m(att_2)$ for all $att_1, att_2 \in A, att_1 \neq att_2$.




(a) $G$




(b) $G'$ with $b_1 = 0, b_2 = 1$

By removing arcs from the STG $G$, any attractor of $\mathcal{N}$ does not disappear; it may only be transformed to a new attractor in $G'$.

# Contents

# The proposed method

- From the above relations, we propose an FVS-based method (named as FVS-ABN) for finding all attractors (fixed points and cyclic attractors) of an ABN.
- This method includes many steps. We first show the general approach of our method. Then, we show each step in detail.

# The general approach

- Based on the relations between an FVS of the interaction graph of an ABN and the dynamics of the ABN, FVS-ABN systematically removes arcs in the STG of the ABN to get a candidate set of states that covers all attractors of the ABN.

# The general approach

- Based on the relations between an FVS of the interaction graph of an ABN and the dynamics of the ABN, FVS-ABN systematically removes arcs in the STG of the ABN to get a candidate set of states that covers all attractors of the ABN.
- Then, FVS-ABN uses reachability analysis on the ABN to filter out this set.

# The general approach

- Based on the relations between an FVS of the interaction graph of an ABN and the dynamics of the ABN, FVS-ABN systematically removes arcs in the STG of the ABN to get a candidate set of states that covers all attractors of the ABN.
- Then, FVS-ABN uses reachability analysis on the ABN to filter out this set.
- The obtained result is a set of states such that there exists a one-to-one correspondence between the set of states and the set of attractors. This set is sufficient because starting from a state in an attractor, we can enumerate all other states in the attractor by listing all states reachable from this state [Garg et al., 2008].

# The general approach

- Based on the relations between an FVS of the interaction graph of an ABN and the dynamics of the ABN, FVS-ABN systematically removes arcs in the STG of the ABN to get a candidate set of states that covers all attractors of the ABN.
- Then, FVS-ABN uses reachability analysis on the ABN to filter out this set.
- The obtained result is a set of states such that there exists a one-to-one correspondence between the set of states and the set of attractors. This set is sufficient because starting from a state in an attractor, we can enumerate all other states in the attractor by listing all states reachable from this state [Garg et al., 2008].
- We here omit the formal proof for the correctness of FVS-ABN. Please see the details of the proof in Theorem 3 of the paper.

# The description of FVS-ABN

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

```
Find an FVS U = {x_{i_1}, ..., x_{i_k}} of IG(A)
Choose a set B = {b_{i_1}, ..., b_{i_k}} of retained values corresponding to the nodes of U
Let G be the STG of A
Let G' be the STG obtained by removing all arcs (x, x') from G where
    ⋁_{j=1}^{k} (x_{i_j} ↔ b_{i_j} ∧ x'_{i_j} ↔ 1 − b_{i_j}) holds
F_fix ← the set of fixed points of G
F ← the set of fixed points of G'
F ← F \ F_fix
Perform Preprocessing SSF to shrink the set F
F ← F \ F_fix
A ← F_fix
while F ≠ ∅ do
    Remove a state s from F
    if UnfReach(A, s, A ∪ F) = false then
        A ← A ∪ {s}
    end if
end while
return A
```

Using an FVS of the ABN to get the candidate set $F$. By Theorems 1 and 2, $F$ covers all attractors of the ABN.

# The description of FVS-ABN (cont.)

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

```
Find an FVS U = {x_{i_1}, ..., x_{i_k}} of IG(A)
Choose a set B = {b_{i_1}, ..., b_{i_k}} of retained values corresponding to the nodes of U
Let G be the STG of A
Let G' be the STG obtained by removing all arcs (x, x') from G where
  ⋁_{j=1}^{k} (x_{i_j} ↔ b_{i_j} ∧ x'_{i_j} ↔ 1 − b_{i_j}) holds
F_fix ← the set of fixed points of G
F ← the set of fixed points of G'
F ← F \ F_fix
Perform Preprocessing SSF to shrink the set F
F ← F \ F_fix
A ← F_fix
while F ≠ ∅ do
    Remove a state s from F
    if UnfReach(A, s, A ∪ F) = false then
        A ← A ∪ {s}
    end if
end while
return A
```

Preprocess $F$ to get a smaller candidate set but the correctness of FVS-ABN is preserved.

# The description of FVS-ABN (cont.)

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

```
Find an FVS U = {x_{i_1}, ..., x_{i_k}} of IG(A)
Choose a set B = {b_{i_1}, ..., b_{i_k}} of retained values corresponding to the nodes of U
Let G be the STG of A
Let G' be the STG obtained by removing all arcs (x, x') from G where
    \bigvee_{j=1}^{k} (x_{i_j} <-> b_{i_j} ∧ x'_{i_j} <-> 1 - b_{i_j}) holds
F_fix <- the set of fixed points of G
F <- the set of fixed points of G'
F <- F \ F_fix
Perform Preprocessing SSF to shrink the set F
F <- F \ F_fix
A <- F_fix
while F ≠ ∅ do
    Remove a state s from F
    if UnfReach(A, s, A ∪ F) = false then
        A <- A ∪ {s}
    end if
end while
return A
```

- $F$ also contains $F_{fix}$. Hence, we can exclude $F_{fix}$ from $F$ before performing Preprocessing SSF.

- After finishing Preprocessing SSF, $F$ may contain some fixed points in $F_{fix}$, so FVS-ABN again excludes $F_{fix}$ from $F$.

# The description of FVS-ABN (cont.)

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

```
Find an FVS U = {x_{i_1}, ..., x_{i_k}} of IG(A)
Choose a set B = {b_{i_1}, ..., b_{i_k}} of retained values corresponding to the nodes of U
Let G be the STG of A
Let G' be the STG obtained by removing all arcs (x, x') from G where
  \bigvee_{j=1}^{k} (x_{i_j} <-> b_{i_j} ∧ x'_{i_j} <-> 1 - b_{i_j}) holds
F_fix <- the set of fixed points of G
F <- the set of fixed points of G'
F <- F \ F_fix
Perform Preprocessing SSF to shrink the set F
F <- F \ F_fix
A <- F_fix
while F ≠ ∅ do
    Remove a state s from F
    if UnfReach(A, s, A ∪ F) = false then
        A <- A ∪ {s}
    end if
end while
return A
```

The filtering process using reachability analysis on ABNs.
- If $s$ reaches any state in $A \cup F$ in $G$, then FVS-ABN skips $s$.
- Otherwise, $s$ corresponds to an attractor and is added to $A$.

# Constituent steps of FVS-ABN

- We here analyze the problems involving the constituent steps of FVS-ABN and then give possible solutions for them.
- A possible solution may be a new algorithm or an existing technique.

# Finding an FVS

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$

The problem of finding a minimum FVS is NP-complete [Johnson and Garey, 1979]. We propose a greedy algorithm for finding an (not necessarily minimum) FVS. This algorithm relies on Strongly Connected Components (SCCs). Please see its details in Algorithm 1 of the paper.

$G$ where

Perform Preprocessing SSF to shrink the set $F$
$F \leftarrow F \backslash F_{fix}$
$A \leftarrow F_{fix}$
**while** $F \neq \emptyset$ **do**
  Remove a state $s$ from $F$
  **if** $UnfReach(\mathcal{A}, s, A \cup F) = false$ **then**
    $A \leftarrow A \cup \{s\}$
  **end if**
**end while**
**return** $A$

# Choosing the retained set

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$

Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$

Let $G$ be the STG of $\mathcal{A}$

L … $G$ where

\

$F$

$F$

$F$

P

$F \leftarrow F \backslash F_{fix}$

$A \leftarrow F_{fix}$

**while** $F \neq \emptyset$ **do**

  Remove a state $s$ from $F$

  **if** $UnfReach(\mathcal{A}, s, A \cup F) = false$ **then**

    $A \leftarrow A \cup \{s\}$

  **end if**

**end while**

**return** $A$

We should choose the retained set $B$ such that the candidate set $F$ is as small as possible. The problem of choosing an optimal $B$ is very hard. Hence, we propose a heuristic method for this problem. Please see its details in Algorithm 3 of the paper.

# Computing fixed points of the reduced STG

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$
Let $G$ be the STG of $\mathcal{A}$
Let $G'$ be the STG obtained by removing all arcs $(x, x')$ from $G$ where $\bigvee_{j=1}^{k}(x_{i_j} \leftrightarrow b_{i_j} \wedge x'_{i_j} \leftrightarrow 1 - b_{i_j})$ holds
$F_{fix} \leftarrow$ the set of fixed points of $G$
$F \leftarrow$ the set of fixed points of $G'$
$F \leftarrow$
Perf
$F \leftarrow$
$A \leftarrow$
**whil**
    R
    **if** $UnfReach(\mathcal{A}, s, A \cup F) = false$ **then**
        $A \leftarrow A \cup \{s\}$
    **end if**
**end while**
**return** $A$

The set $F$ of fixed points of the reduced STG can be characterized by a propositional formula. Hence, we can use some techniques (such as, BDD or SAT) to compute $F$. In addition, we can also use these techniques for computing the set $F_{fix}$ of fixed points of the original STG.

# Preprocessing SSF

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$
Let $G$ be the STG of $\mathcal{A}$
Let $G'$ be the STG obtained by removing all arcs $(x, x')$ from $G$ where $\bigvee_{j=1}^{k}(x_{i_j} \leftrightarrow b_{i_j} \wedge x'_{i_j} \leftrightarrow 1 - b_{i_j})$ holds
$F_{fix} \leftarrow$ the set of fixed points of $G$
$F \leftarrow$ the set of fixed points of $G'$
$F \leftarrow F \setminus F_{fix}$
Perform Preprocessing SSF to shrink the set $F$
$F \leftarrow F \setminus F_{fi}$ [partially hidden]

In each interaction of Preprocessing SSF, FVS-ABN randomly chooses a node $x_i$ and replaces $F$ by its forward image set by updating only $x_i$ (say $F'$).
- Preprocessing SSF may be useful because it is possible that $|F'| < |F|$ leading to the final obtained set may be much smaller than the original set.
- Preprocessing SSF preserves the correctness of FVS-ABN because the final obtained set still covers all cyclic attractors of the ABN (i.e., $A \cup F$ covers all attractors of the ABN).

**return** $A$

# Reachability analysis

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

Checking the reachability in ABNs is the key task in FVS-ABN. We propose a new algorithm called *UnfReach* that relies on Petri net unfoldings and a preprocessing step called Preprocessing BCN as follows.

- *UnfReach* uses Mole [Schwoon and Romer, 2016] to build on the fly the finite complete prefix of the encoded 1-safe Petri net of the ABN [Chatain et al., 2014].
- Based constant nodes of the ABN, Preprocessing BCN excludes from $A \cup F$ the states that cannot be reachable from $s$. If the excluded set is empty, then we do not need to build the finite prefix. A constant node is the node that retains its value once it is set to a specific value (e.g., if $f_1 = x_1 \vee x_2$, then $x_1$ is a constant node.)

es of $U$

```
    Remove a state s from F
    if UnfReach(A, s, A ∪ F) = false then
        A ← A ∪ {s}
    end if
end while
return A
```

# Contents

# Implementation

- We have implemented the proposed method in a JAVA tool called FVS-ABN.
- An executable file of FVS-ABN and some examples of real biological networks are available at: https://sites.google.com/site/trinhgiangjaist/.

# Experiments

- We conduct experiments to compare the performance among FVS-ABN, genYsis [Garg et al., 2008], and CABEAN [Mizera et al., 2018]. genYsis and CABEAN are two exact and famous tools for finding attractors of ABNs.
- All the experiments were run on a virtual machine whose environment is CPU: Intel(R) Xeon(R) Silver 4116 4x2.10GHz, Memory: 24 GB, CentOS 7 64 bit.
- We applied the three algorithms to two sets of Boolean networks.
  - BN models of real biological networks obtained from the literature [Helikar et al., 2012].
  - Random BNs generated with Bool Net R package [Hopfensitz et al., 2013].
- Hereafter, we shall show briefly the observations obtained from the experimental results. The detailed analysis can be found in the paper.

# Results on real biological networks

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\lvert U \rvert$ | $\lvert A \rvert$ | $\lvert F \rvert$ | $\lvert F_1 \rvert$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |

- $n$: the number of nodes
- $|U|$: the size of the used FVS
- $|A|$: the number of attractors
- $|F|$: the size of the candidate set $F$ before Preprocessing SSF
- $|F_1|$: the size of the candidate set $F$ after Preprocessing SSF
- time: the running time in seconds
- -: the case of not obtaining the result within the time limit (10 hours)

# Results on real biological networks (cont.)

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\|U\|$ | $\|A\|$ | $\|F\|$ | $\|F_1\|$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |
| Treatment_of_Castration_Resistant | 42 | 14 | 16384 | 0 | 0 | 0.13 | 18.18 | 0.73 |
| GuardCellAbscisicAcidSignaling | 44 | 8 | 28 | 32 | 15 | 1.33 | 7.90 | 0.83 |
| InflammatoryBowelDisease | 47 | 22 | 1 | 960 | 1 | 2.47 | - | - |
| Stomatal_Opening_Model | 49 | 13 | 48 | 243 | 14 | 10.99 | 31.22 | 2.38 |
| Differentiation_of_T_lymphocytes | 50 | 18 | 2050 | 5581 | 0 | 627.76 | - | 89.75 |
| Senescence | 51 | 12 | 17 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Drosophila | 52 | 14 | 128 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| MAPK | 53 | 10 | 18 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| B_bronchiseptica_T_retortaeformis | 53 | 15 | 30 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| TcellLGL | 60 | 23 | 142 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| TLGLSurvival | 61 | 25 | 318 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| PC12CellDifferentiation | 62 | 3 | 3 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| ButanolProduction | 66 | 18 | 8192 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| HumanMyelomaCells | 67 | 14 | 83 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| HGF_Signaling_in_Keratinocytes | 68 | 10 | 72 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] |
| Colitis_associated_colon_cancer | 70 | 13 | 10 | 100 | 14 | 391.05 | - | - |
| Bcell | 72 | 19 | 72 | 934 | 69 | 22.59 | 8702.80 | 29.84 |
| YeastApoptosis | 73 | 17 | 8448 | 4352 | 4352 | 75.32 | 45.85 | 1.16 |
| IL_6_Signalling | 86 | 21 | 32768 | 20480 | 4096 | 297.51 | - | - |
| T_Cell_Receptor_Signaling | 101 | 10 | 128 | 72 | 24 | 5.27 | 3596.65 | - |

The size of the FVS obtained by the proposed greedy algorithm is much smaller than $n$. This observation confirms that the greedy algorithm is good enough.

# Results on real biological networks (cont.)

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\lvert U\rvert$ | $\lvert A\rvert$ | $\lvert F\rvert$ | $\lvert F_1\rvert$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |
| Treatment_of_Castration_Resistant | 42 | 14 | 16384 | 0 | 0 | 0.13 | 18.18 | 0.73 |
| GuardCellAbscisicAcidSignaling | 44 | 8 | 28 | 32 | 15 | 1.33 | 7.90 | 0.83 |
| InflammatoryBowelDisease | 47 | 22 | 1 | 960 | 1 | 2.47 | - | - |
| Stomatal_Opening_Model | 49 | 13 | 48 | 243 | 14 | 10.99 | 31.22 | 2.38 |
| Differentiation_of_T_lymphocytes | 50 | 18 | 2050 | 5581 | 0 | 627.76 | - | 89.75 |
| Senescence | 51 | 12 | 17 | 84 | 2 | 9.93 | 18.05 | 3.00 |
| | | | 128 | 84 | 0 | 4.88 | - | 1984.40 |
| | | | 18 | 226 | 6 | 8.15 | - | - |
| B_bronc | | | 30 | 298 | 0 | 15.61 | 3556.85 | 440.16 |
| | | | 142 | 11156 | 108 | 55.56 | 21198.63 | 916.23 |
| | | | 318 | 18276 | 260 | 174.66 | - | - |
| PC | | | 3 | 0 | 0 | 0.20 | 5.01 | 0.59 |
| | | | 8192 | 12416 | 6144 | 324.22 | - | - |
| H | | | 83 | 558 | 0 | 47.00 | 12983.39 | - |
| HGF_Signaling_in_Keratinocytes | 68 | 10 | 72 | 256 | 0 | 3.79 | 1200.04 | 8.75 |
| Colitis_associated_colon_cancer | 70 | 13 | 10 | 100 | 14 | 391.05 | - | - |
| Bcell | 72 | 19 | 72 | 934 | 69 | 22.59 | 8702.80 | 29.84 |
| YeastApoptosis | 73 | 17 | 8448 | 4352 | 4352 | 75.32 | 45.85 | 1.16 |
| IL_6_Signalling | 86 | 21 | 32768 | 20480 | 4096 | 297.51 | - | - |
| T_Cell_Receptor_Signaling | 101 | 10 | 128 | 72 | 24 | 5.27 | 3596.65 | - |

In most networks, $|F_1|$ is much smaller than $|F|$. This observation is evidence for the usefulness of Preprocessing SSF.

# Results on real biological networks (cont.)

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\lvert U\rvert$ | $\lvert A\rvert$ | $\lvert F\rvert$ | $\lvert F_1\rvert$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |
| Treatment_of_Castration_Resistant | 42 | 14 | 16384 | 0 | 0 | 0.13 | 18.18 | 0.73 |
| GuardCellAbscisicAcidSignaling | 44 | 8 | 28 | 32 | 15 | 1.33 | 7.90 | 0.83 |
| InflammatoryBowelDisease | 47 | 22 | 1 | 960 | 1 | 2.47 | - | - |
| Stomatal_Opening_Model | 49 | 13 | 48 | 243 | 14 | 10.99 | 31.22 | 2.38 |
| Differentiation_of_T_lymphocytes | 50 | 18 | 2050 | 5581 | 0 | 627.76 | - | 89.75 |
| Senescence | 51 | 12 | 17 | 84 | 2 | 9.93 | 18.05 | 3.00 |
| Drosophila | 52 | 14 | 128 | 84 | 0 | 4.88 | - | 1984.40 |
| MAPK | 53 | 10 | 18 | 226 | 6 | 8.15 | - | - |
| B_bronchisepti[illegible] | [illegible] | [illegible] | [illegible] | [illegible]8 | 0 | 15.61 | 3556.85 | 440.16 |
| T[illegible] | [illegible] | [illegible] | [illegible] | [illegible]6 | 108 | 55.56 | 21198.63 | 916.23 |
| TL[illegible] | [illegible] | [illegible] | [illegible] | [illegible]6 | 260 | 174.66 | - | - |
| PC12CellDifferentiation | 62 | 5 | 5 | 0 | 0 | 0.20 | 5.01 | 0.59 |
| ButanolProduction | 66 | 18 | 8192 | 12416 | 6144 | 324.22 | - | - |
| HumanMyelomaCells | 67 | 14 | 83 | 558 | 0 | 47.00 | 12983.39 | - |
| HGF_Signaling_in_Keratinocytes | 68 | 10 | 72 | 256 | 0 | 3.79 | 1200.04 | 8.75 |
| Colitis_associated_colon_cancer | 70 | 13 | 10 | 100 | 14 | 391.05 | - | - |
| Bcell | 72 | 19 | 72 | 934 | 69 | 22.59 | 8702.80 | 29.84 |
| YeastApoptosis | 73 | 17 | 8448 | 4352 | 4352 | 75.32 | 45.85 | 1.16 |
| IL_6_Signalling | 86 | 21 | 32768 | 20480 | 4096 | 297.51 | - | - |
| T_Cell_Receptor_Signaling | 101 | 10 | 128 | 72 | 24 | 5.27 | 3596.65 | - |

FVS-ABN outperforms genYsis in most networks.

# Results on real biological networks (cont.)

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\|U\|$ | $\|A\|$ | $\|F\|$ | $\|F_1\|$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |
| Treatment_of_Castration_Resistant | 42 | 14 | 16384 | 0 | 0 | 0.13 | 18.18 | 0.73 |
| GuardCellAbscisicAcidSignaling | 44 | 8 | 28 | 32 | 15 | 1.33 | 7.90 | 0.83 |
| InflammatoryBowelDisease | 47 | 22 | 1 | 960 | 1 | 2.47 | - | - |
| Stomatal_Opening_Model | 49 | 13 | 48 | 243 | 14 | 10.99 | 31.22 | 2.38 |
| Differentiation_of_T_lymphocytes | 50 | 18 | 2050 | 5581 | 0 | 627.76 | - | 89.75 |
| Senescence | 51 | 12 | 17 | 84 | 2 | 9.93 | 18.05 | 3.00 |
| Drosophila | 52 | 14 | 128 | 84 | 0 | 4.88 | - | 1984.40 |
| MAPK | 53 | 10 | 18 | 226 | 6 | 8.15 | - | - |
| B_bronchisepti[illegible] | [illegible] | [illegible] | [illegible] | [illegible]8 | 0 | 15.61 | 3556.85 | 440.16 |
| T[illegible] | [illegible] | [illegible] | [illegible] | [illegible]6 | 108 | 55.56 | 21198.63 | 916.23 |
| TL[illegible] | [illegible] | [illegible] | [illegible] | [illegible]6 | 260 | 174.66 | - | - |
| PC12Cel[illegible] | [illegible] | [illegible] | [illegible] | 0 | 0 | 0.20 | 5.01 | 0.59 |
| ButanolProduction | 66 | 18 | 8192 | 12416 | 6144 | 324.22 | - | - |
| HumanMyelomaCells | 67 | 14 | 83 | 558 | 0 | 47.00 | 12983.39 | - |
| HGF_Signaling_in_Keratinocytes | 68 | 10 | 72 | 256 | 0 | 3.79 | 1200.04 | 8.75 |
| Colitis_associated_colon_cancer | 70 | 13 | 10 | 100 | 14 | 391.05 | - | - |
| Bcell | 72 | 19 | 72 | 934 | 69 | 22.59 | 8702.80 | 29.84 |
| YeastApoptosis | 73 | 17 | 8448 | 4352 | 4352 | 75.32 | 45.85 | 1.16 |
| IL_6_Signalling | 86 | 21 | 32768 | 20480 | 4096 | 297.51 | - | - |
| T_Cell_Receptor_Signaling | 101 | 10 | 128 | 72 | 24 | 5.27 | 3596.65 | - |

FVS-ABN outperforms CABEAN in most networks.

# Results on real biological networks (cont.)

| | | | | FVS-ABN | | | genYsis | CABEAN |
|---|---|---|---|---|---|---|---|---|
| name | $n$ | $\|U\|$ | $\|A\|$ | $\|F\|$ | $\|F_1\|$ | time | time | time |
| ApoptosisNetwork | 41 | 7 | 8 | 12 | 8 | 7.27 | 581.07 | - |
| Treatment_of_Castration_Resistant | 42 | 14 | 16384 | 0 | 0 | 0.13 | 18.18 | 0.73 |
| GuardCellAbscisicAcidSignaling | 44 | 8 | 28 | 32 | 15 | 1.33 | 7.90 | 0.83 |
| InflammatoryBowelDisease | 47 | 22 | 1 | 960 | 1 | 2.47 | - | - |
| Stomatal_Opening_Model | 49 | 13 | 48 | 243 | 14 | 10.99 | 31.22 | 2.38 |
| Differentiation_of_T_lymphocytes | 50 | 18 | 2050 | 5581 | 0 | 627.76 | - | 89.75 |
| Senescence | 51 | 12 | 17 | 84 | 2 | 9.93 | 18.05 | 3.00 |
| Drosophila | 52 | 14 | 128 | 84 | 0 | 4.88 | - | 1984.40 |
| MAPK | 53 | 10 | 18 | 226 | 6 | 8.15 | - | - |
| B_bronchiseptica_T_retortaeformis | 53 | 15 | 30 | 298 | 0 | 15.61 | 3556.85 | 440.16 |
| TcellLGL | 60 | 23 | 142 | 11156 | 108 | 55.56 | 21198.63 | 916.23 |
| TLGLSurvival | 61 | 25 | 318 | 18276 | 260 | 174.66 | - | - |
| PC12CellDifferentiation | 62 | 3 | 3 | 0 | 0 | 0.20 | 5.01 | 0.59 |
| ButanolProduction | 66 | 18 | 8192 | 12416 | 6144 | 324.22 | - | - |
| HumanMyelomaCells | 67 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | - |
| HGF_Signaling_in_Keratinocytes | 68 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 8.75 |
| Colitis_associated_colon_cancer | 70 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | - |
| Bcell | 72 | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 29.84 |
| YeastApoptosis | 73 | 17 | 8448 | 4352 | 4352 | [illegible] | 45.85 | 1.16 |
| IL_6_Signalling | 86 | 21 | 32768 | 20480 | 4096 | 297.51 | - | - |
| T_Cell_Receptor_Signaling | 101 | 10 | 128 | 72 | 24 | 5.27 | 3596.65 | - |

FVS-ABN can handle large networks in reasonable time.

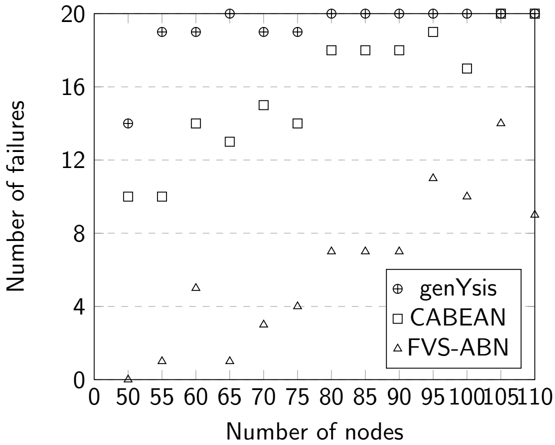
# Results on real biological networks (cont.)

- We here omit the analysis on the impact of the picked FVS to the performance of FVS-ABN. See the paper for more details.
- In general, the picked FVS may largely impact the performance of FVS-ABN and using a smaller FVS may get better performance.

# Results on randomly generated networks

- We randomly generated a set of $N$-$K$ BNs [Kauffman, 1992] with network size $n$ in the set $\{50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 105, 110\}$ and $K = 2$ (i.e., each node has exactly $K = 2$ input nodes). For each network size, 20 instances were generated. In total, we have 260 random BNs.
- We then applied FVS-ABN, genYsis, and CABEAN to the 260 random BNs and recorded the number of failures (i.e., failed to obtain the result within 30 minutes).

# Results on randomly generated networks (cont.)




- The number of failures of genYsis or CABEAN rapidly approaches 20.
- FVS-ABN can even handle 30 or 55 percent of networks for $n = 105$ or $n = 110$, respectively.
- In each network size, the number of failures of genYsis or CABEAN is always larger than that of FVS-ABN.

# Contents

# Summary

- We have formally stated and proved the relations between FVSs and dynamics of BNs.

# Summary

- We have formally stated and proved the relations between FVSs and dynamics of BNs.
- From these relations, we have proposed an FVS-based method (called FVS-ABN) for detecting attractors in ABNs. Our approach relies on the principle of removing arcs in the STG to get a candidate set and the reachability analysis to filter out the candidate set. The correctness of our method has been formally proved.

# Summary

- We have formally stated and proved the relations between FVSs and dynamics of BNs.
- From these relations, we have proposed an FVS-based method (called FVS-ABN) for detecting attractors in ABNs. Our approach relies on the principle of removing arcs in the STG to get a candidate set and the reachability analysis to filter out the candidate set. The correctness of our method has been formally proved.
- We have also proposed Preprocessing SSF to reduce the computational burden while preserving the correctness of FVS-ABN. Then, we have proposed an unfolding-based and on-the-fly method for checking the reachability in ABNs.

# Summary

- We have formally stated and proved the relations between FVSs and dynamics of BNs.
- From these relations, we have proposed an FVS-based method (called FVS-ABN) for detecting attractors in ABNs. Our approach relies on the principle of removing arcs in the STG to get a candidate set and the reachability analysis to filter out the candidate set. The correctness of our method has been formally proved.
- We have also proposed Preprocessing SSF to reduce the computational burden while preserving the correctness of FVS-ABN. Then, we have proposed an unfolding-based and on-the-fly method for checking the reachability in ABNs.
- The experimental results are very promising. FVS-ABN outperforms the two state-of-the-art methods, genYsis and CABEAN. Furthermore, FVS-ABN can handle large real biological networks in reasonable time.

# Discussion

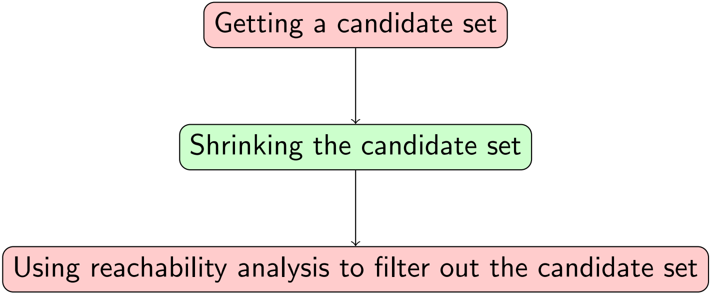
- In principle, FVS-ABN works well on large networks having relatively small FVSs and not too large attractors. Fortunately, these characteristics are often found in real biological networks [Gershenson, 2002, Skodawessely and Klemm, 2011].

# Discussion

- In principle, FVS-ABN works well on large networks having relatively small FVSs and not too large attractors. Fortunately, these characteristics are often found in real biological networks [Gershenson, 2002, Skodawessely and Klemm, 2011].
- In systems biology, the analysis of large real biological networks (e.g, IL_6_Signalling, T_Cell_Receptor_Signaling) was usually performed by using the reduced versions of these networks due to the performance limitations of the existing tools. Hence, the advanced computation capability of FVS-ABN can enable biologists to conduct more accurate analysis on large networks, then to discover more biological insights.

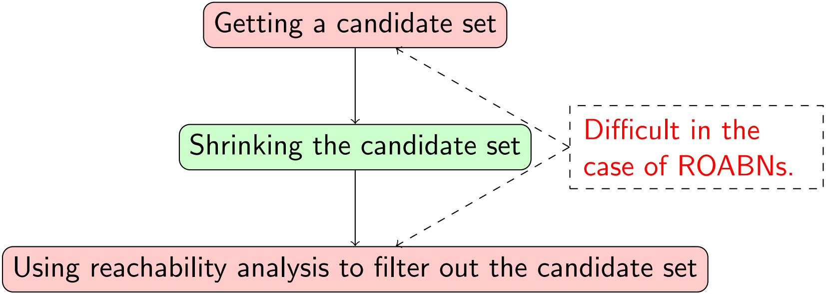
# Discussion (cont.)

- Since the relations presented above do not depend on the updating scheme, the approach of FVS-ABN can be a blueprint for attractor detection in other types of Boolean networks, such as ROABNs, GABNs. However, we would have to consider many issues due to the characteristic of the updating scheme of a specific type.



# Discussion (cont.)

- Since the relations presented above do not depend on the updating scheme, the approach of FVS-ABN can be a blueprint for attractor detection in other types of Boolean networks, such as, ROABNs, GABNs. However, we would have to consider many issues due to the characteristic of the updating scheme of a specific type.



# Potential improvements

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$

- Use an existing exact method or propose a more efficient heuristic method for finding FVSs.
- Propose a more efficient heuristic for setting the retained set $B$.

These aim at reducing the size of the candidate set $F$.

$G$ where

Perform Preprocessing SSF to shrink the set $F$
$F \leftarrow F \setminus F_{fix}$
$A \leftarrow F_{fix}$
**while** $F \neq \emptyset$ **do**
    Remove a state $s$ from $F$
    **if** $UnfReach(\mathcal{A}, s, A \cup F) = false$ **then**
        $A \leftarrow A \cup \{s\}$
    **end if**
**end while**
**return** $A$

# Potential improvements (cont.)

**Input:** An ABN $\mathcal{A}$.

**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$of $IG(\mathcal{A})$

Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$

Let $G$ be the STG of $\mathcal{A}$

Let $G'$ be the STG obtained by removing all arcs $(x, x')$ from $G$ where $\bigvee_{j=1}^{k}(x_{i_j} \leftrightarrow b_{i_j} \wedge x'_{i_j} \leftrightarrow 1 - b_{i_j})$ holds

$F_{fix} \leftarrow$ the set of fixed points of $G$

$F \leftarrow$ the set of fixed points of $G'$

$F \leftarrow F \backslash F_{fix}$

In some failed networks of the experiments, the reason for the failure is the computation of fixed points. It is reasonable because the used techniques (i.e., BDD and SAT) are inefficient for large networks especially the networks comprising complex Boolean functions (e.g., N-K networks). Hence, we can use a more efficient method for computing $F_{fix}$ and $F$, such as, ILP-based methods [Akutsu et al., 2009], ASP-based methods [Klarner et al., 2017].

**return** $A$

# Potential improvements (cont.)

**Input:** An ABN $\mathcal{A}$.
**Output:** The set $A$ of states.

Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of $U$
Let $G$ be the STG of $\mathcal{A}$

In some failed networks of the experiments, the reason for the failure is the reachability analysis. Hence, we can use a more efficient method for checking the reachability in ABNs.
- Other techniques in terms of Petri net unfoldings, such as, contextual Petri nets [Baldan et al., 2012], merged processes [Rodríguez et al., 2013].
- Some static analyzers [Paulevé, 2017, Chai et al., 2018], which can handle very large networks, in a preprocessing step.

    Remove a state $s$ from $F$
    **if** $UnfReach(\mathcal{A}, s, A \cup F) = false$ **then**
        $A \leftarrow A \cup \{s\}$
    **end if**
**end while**
**return** $A$

# Future work

- Improve FVS-ABN by improving its constituent steps (see the potential improvements presented above). Our target is making FVS-ABN enable to handle very large networks with thousands of nodes.

# Future work

- Improve FVS-ABN by improving its constituent steps (see the potential improvements presented above). Our target is making FVS-ABN enable to handle very large networks with thousands of nodes.
- Extend FVS-ABN to that for other popular types of asynchronous Boolean models. We now focus on ROABNs [Saadatpour et al., 2010]. Theoretically, we also consider the relations in dynamics between ROABNs and ABNs.

# Future work

- Improve FVS-ABN by improving its constituent steps (see the potential improvements presented above). Our target is making FVS-ABN enable to handle very large networks with thousands of nodes.
- Extend FVS-ABN to that for other popular types of asynchronous Boolean models. We now focus on ROABNs [Saadatpour et al., 2010]. Theoretically, we also consider the relations in dynamics between ROABNs and ABNs.
- Extend FVS-ABN to that for multi-valued networks where each gene can have more than two expression levels (more expressive than Boolean networks). Of course, we can encode a multi-valued network as a BN [Didier et al., 2011], and then apply FVS-ABN to the encoded BN. However, the size of the encoded BN may be large, making its analysis unmanageable. Thus, a directed approach on multi-valued networks is needed.

# Our recent publications on Boolean networks


- Trinh Van Giang and Kunihiko Hiraishi: "On attractor detection and optimal control of deterministic generalized asynchronous random Boolean networks," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2020, in press. https://doi.org/10.1109/TCBB.2020.3043785.
- Trinh Van Giang, Tatsuya Akutsu and Kunihiko Hiraishi: "An FVS-based approach to attractor detection in asynchronous random Boolean networks," IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2020, in press. https://doi.org/10.1109/TCBB.2020.3028862.
- Trinh Van Giang and Kunihiko Hiraishi: "A study on attractors of generalized asynchronous random Boolean networks," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 103(8), 987-994, 2020. https://doi.org/10.1587/transfun.2019EAP1163.

# Our recent publications on Boolean networks (cont.)

- Trinh Van Giang and Kunihiko Hiraishi: "An efficient method for approximating attractors in large-scale asynchronous Boolean models," 13th International Workshop on Biological Network Analysis and Integrative Graph-Based Approaches (IWBNA 2020), in Proc. 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2020), 1820-1826, 2020. https://doi.org/10.1109/BIBM49941.2020.9313230.
- Trinh Van Giang and Kunihiko Hiraishi: "Algorithms for finding attractors of generalized asynchronous random Boolean networks", in Proc. 2019 12th Asian Control Conference (ASCC 2019), 67-72, 2019. http://ieeexplore.ieee.org/document/8765169.

# Our older publications on Boolean networks


- Kobayashi Koichi and Kunihiko Hiraishi. "Optimization-based approaches to control of probabilistic Boolean networks." Algorithms 10.1 (2017): 31. https://doi.org/10.3390/a10010031.
- Kobayashi Koichi and Kunihiko Hiraishi. "Design of probabilistic Boolean networks based on network structure and steady-state probabilities." IEEE Transactions on Neural Networks and Learning Systems 28.8 (2016): 1966-1971. https://doi.org/10.1109/TNNLS.2016.2572063.
- Kobayashi Koichi and Kunihiko Hiraishi. "ILP/SMT-based method for design of Boolean networks based on singleton attractors." IEEE/ACM Transactions on Computational Biology and Bioinformatics 11.6 (2014): 1253-1259. https://doi.org/10.1109/TCBB.2014.2325011.

# References I


Akutsu, T., Hayashida, M., and Tamura, T. (2009).
Integer programming-based methods for attractor detection and control of Boolean networks.
In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 5610–5617. IEEE.

Baldan, P., Bruni, A., Corradini, A., König, B., Rodríguez, C., and Schwoon, S. (2012).
Efficient unfolding of contextual petri nets.
*Theoretical Computer Science*, 449:2–22.

Biane, C. and Delaplace, F. (2018).
Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery.
*IEEE/ACM Trans. Comput. Biol. Bioinf.*, 16(5):1574–1585.

# References II


Chai, X., Ribeiro, T., Magnin, M., Roux, O., and Inoue, K. (2018).
Static analysis and stochastic search for reachability problem.

Chatain, T., Haar, S., Jezequel, L., Paulevé, L., and Schwoon, S. (2014).
Characterization of reachable attractors using petri net unfoldings.
In *International Conference on Computational Methods in Systems Biology*, pages 129–142. Springer.

Didier, G., Remy, E., and Chaouiya, C. (2011).
Mapping multivalued onto Boolean dynamics.
*Journal of theoretical biology*, 270(1):177–184.

# References III

Dubrova, E. and Teslenko, M. (2011).
A sat-based algorithm for finding attractors in synchronous boolean networks.
*IEEE/ACM transactions on computational biology and bioinformatics*, 8(5):1393–1399.

Garg, A., Di Cara, A., Xenarios, I., Mendoza, L., and De Micheli, G. (2008).
Synchronous versus asynchronous modeling of gene regulatory networks.
*Bioinformatics*, 24(17):1917–1925.

Gershenson, C. (2002).
Classification of random boolean networks.
*arXiv preprint cs/0208001*.

# References IV


Gershenson, C. (2004).
Introduction to random boolean networks.
*arXiv preprint nlin/0408006*.

Gershenson, C., Broekaert, J., and Aerts, D. (2003).
Contextual random Boolean networks.
In *European Conference on Artificial Life*, pages 615–624. Springer.

Harvey, I. and Bossomaier, T. (1997).
Time out of joint: Attractors in asynchronous random boolean networks.
In *Proceedings of the Fourth European Conference on Artificial Life*, pages 67–75. MIT Press, Cambridge.

# References V

He, Q., Xia, Z., and Lin, B. (2018).
P_UNSAT approach of attractor calculation for Boolean gene regulatory networks.
*J. Theor. Biol.*, 447:171–177.

Helikar, T., Kowal, B., McClenathan, S., Bruckner, M., Rowley, T., Madrahimov, A., Wicks, B., Shrestha, M., Limbu, K., and Rogers, J. A. (2012).
The Cell Collective: toward an open and collaborative approach to systems biology.
*BMC Syst. Biol.*, 6(1):96:1–96:14.

Hopfensitz, M., Müssel, C., Maucher, M., and Kestler, H. A. (2013).
Attractors in boolean networks: a tutorial.
*Computational Statistics*, 28(1):19–36.

# References VI

Johnson, D. S. and Garey, M. R. (1979).
*Computers and intractability: A guide to the theory of NP-completeness.*
WH Freeman.

Kauffman, S. A. (1992).
The origins of order: Self-organization and selection in evolution.
In *Spin glasses and biology*, pages 61–100. World Scientific.

Kauffman, S. A. (1993).
*The origins of order: Self-organization and selection in evolution.*
OUP USA.

Klarner, H., Streck, A., and Siebert, H. (2017).
Pyboolnet: a python package for the generation, analysis and visualization of Boolean networks.
*Bioinformatics*, 33(5):770–772.

# References VII

Mizera, A., Pang, J., Qu, H., and Yuan, Q. (2018).
Taming asynchrony for attractor detection in large boolean networks.
*IEEE/ACM transactions on computational biology and bioinformatics*, 16(1):31–42.

Paulevé, L. (2017).
Pint: A static analyzer for transient dynamics of qualitative networks with ipython interface.
In *International Conference on Computational Methods in Systems Biology*, pages 309–316. Springer.

Rodríguez, C., Schwoon, S., and Khomenko, V. (2013).
Contextual merged processes.
In *International Conference on Applications and Theory of Petri Nets and Concurrency*, pages 29–48. Springer.

# References VIII

Saadatpour, A., Albert, I., and Albert, R. (2010).
Attractor analysis of asynchronous Boolean models of signal transduction networks.
*Journal of Theoretical Biology*, 266(4):641–656.

Schwoon, S. and Romer, S. (2016).
Mole—a petri net unfolder.
http://www.lsv.fr/~schwoon/tools/mole/.

Skodawessely, T. and Klemm, K. (2011).
Finding attractors in asynchronous boolean dynamics.
*Advances in Complex Systems*, 14(03):439–449.

Thomas, R. (1991).
Regulatory networks seen as asynchronous automata: a logical description.
*Journal of theoretical biology*, 153:1–23.

# References IX


Tun, K., Menghini, M., D'Andrea, L., Dhar, P., Tanaka, H., and Giuliani, A. (2011).
Why so few drug targets: a mathematical explanation?
*Current computer-aided drug design*, 7(3):206–213.

Valverde, J. C., Mortveit, H. S., Gershenson, C., and Shi, Y. (2020).
Boolean networks and their applications in science and engineering.
*Complexity*, 2020:6183798:1–6183798:3.

Yamanaka, S. (2009).
Elite and stochastic models for induced pluripotent stem cell generation.
*Nature*, 460(7251):49.

# References X

Yuan, Q., Mizera, A., Pang, J., and Qu, H. (2019).
A new decomposition-based method for detecting attractors in synchronous Boolean networks.
*Sci. Comput. Program.*, 180:18–35.

Yuan, Q., Qu, H., Pang, J., and Mizera, A. (2016).
Improving bdd-based attractor detection for synchronous boolean networks.
*Science China Information Sciences*, 59(8):080101.

Thank you for your attention!