Japan Advanced Institute of Science and Technology
Graduate School of Advanced Science and Technology

On Attractor Detection and Optimal Control of Boolean Networks

PhD Student: Trinh Van Giang, 1820424
Supervisor: Kunihiko Hiraishi

November 05, 2021

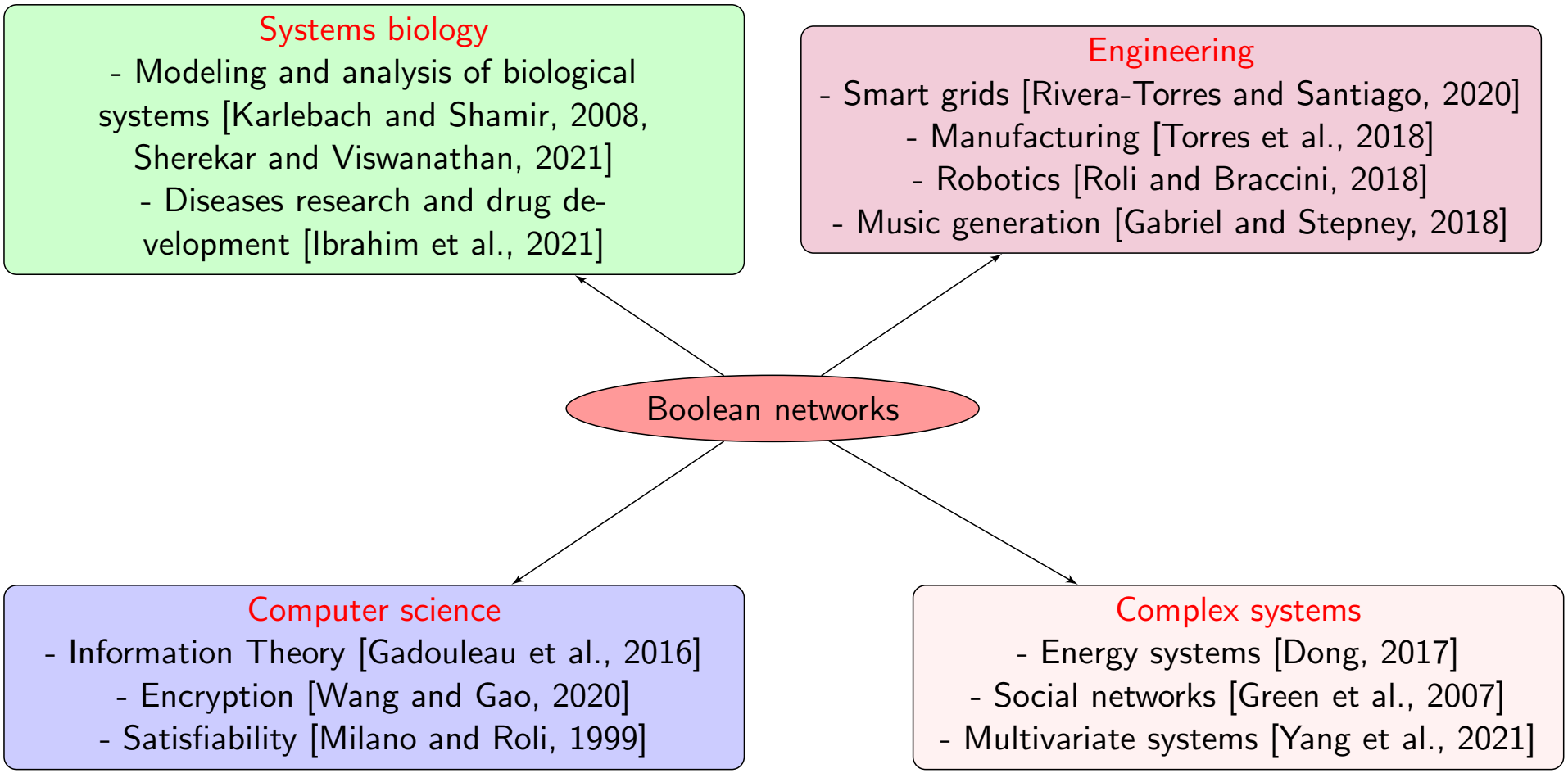
Contents

Contents

Boolean networks

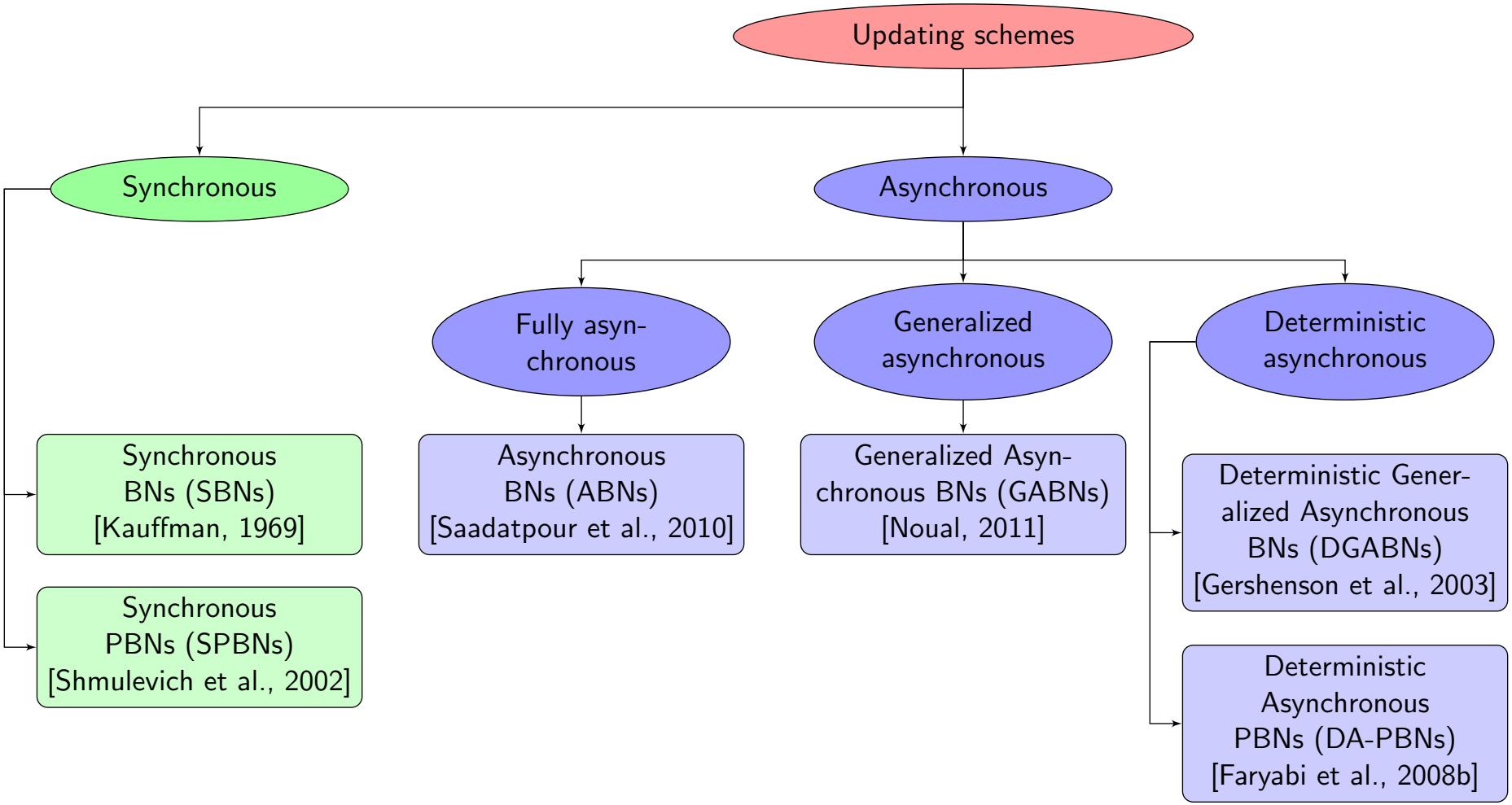
- Boolean Networks (BNs) are simple but efficient mathematical formalism for modeling and analyzing complex biological systems [Schwab et al., 2020].
  - A BN includes n nodes; each node can receive either 0 or 1, and can be associated with one Boolean function [Gershenson, 2004].
  - Probabilistic Boolean Networks (PBNs) are a stochastic extension of BNs where each node can be associated with one or more Boolean functions, and each Boolean function has a probability for selection [Shmulevich et al., 2002].
- BNs and PBNs are interesting mathematical objects that have recently attracted various work in theory [Schwab et al., 2020].
- Furthermore, they have widely been applied to various areas from science to engineering [Valverde et al., 2020].

Applications of BNs



... and many other applications.

Classification of BNs



These typical types of BNs have been widely studied as well as found various applications [Schwab et al., 2020].

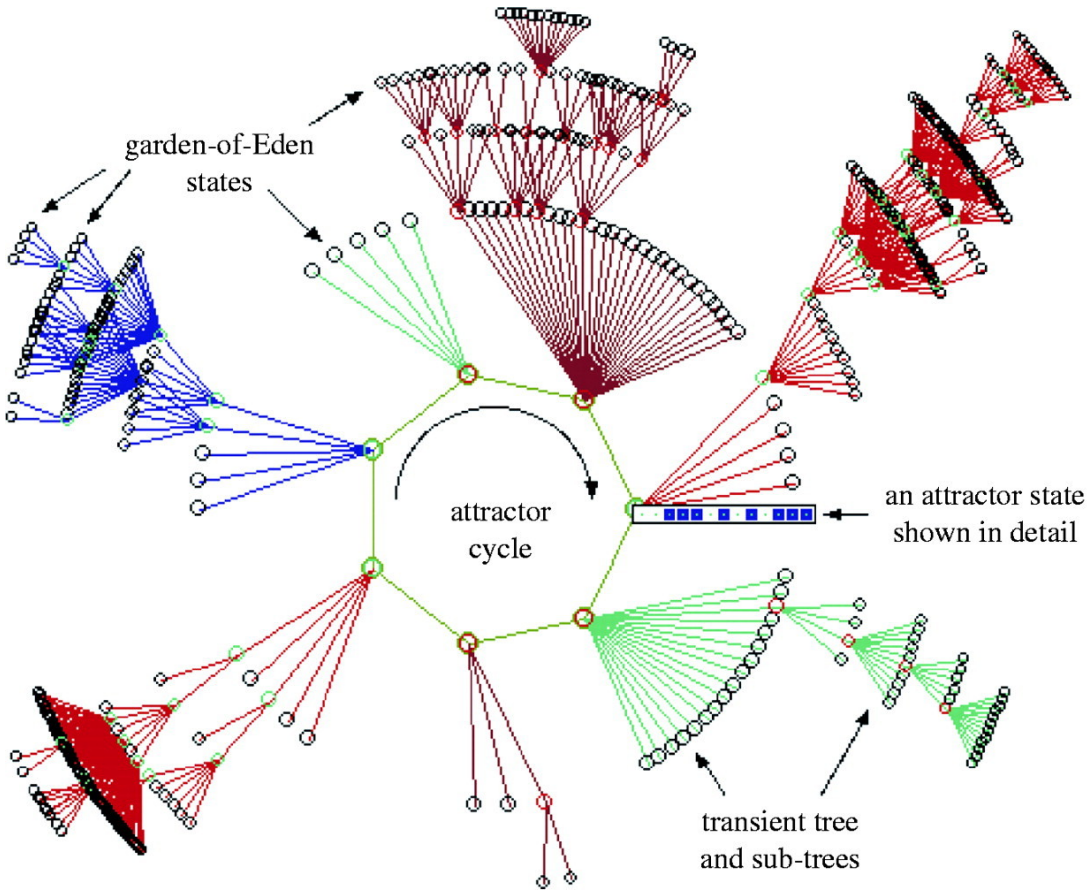
Attractor detection and optimal control of BNs

Attractor detection and optimal control of BNs are difficult and interesting in theory but also have a plenty of applications in many areas [Akutsu, 2018].

In this research, we focus on the two above problems on BNs.

Attractor detection in BNs

An *attractor* of a BN is a set of states such that the BN cannot escape from this set once entered it (the long-run dynamics).



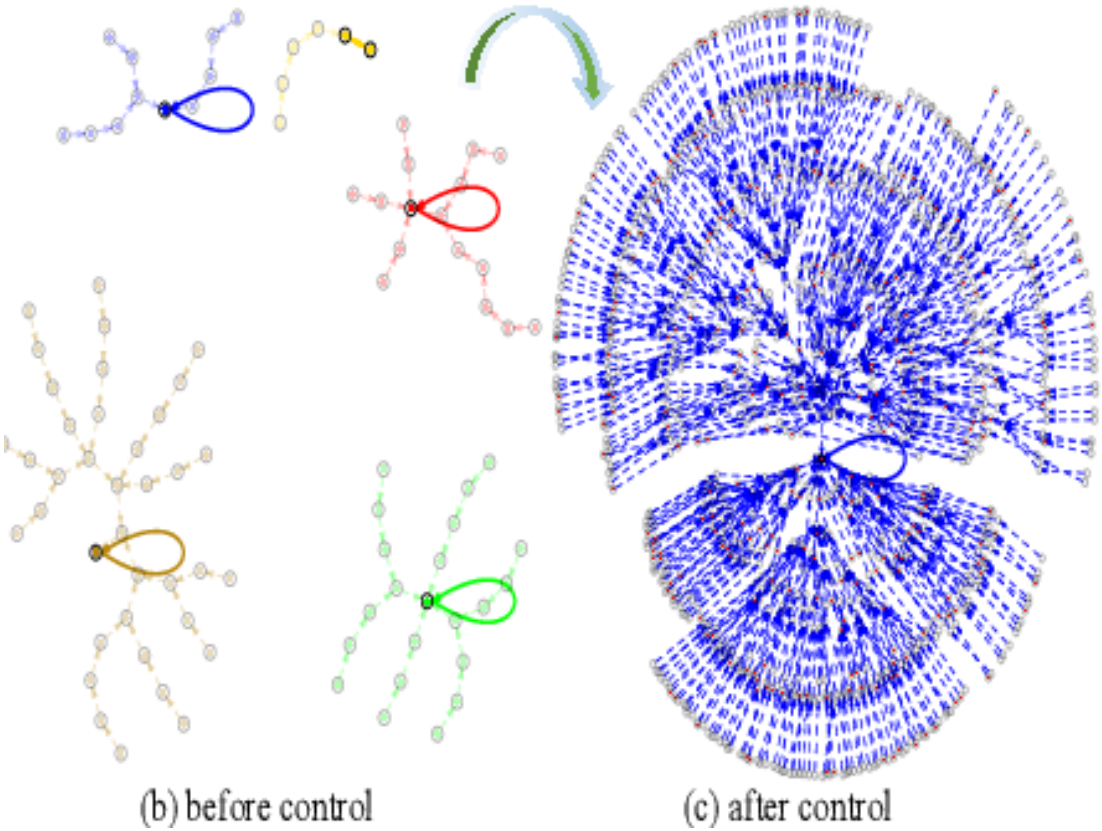
Source: [Bornholdt, 2008]

Attractor detection in BNs (cont.)

- Analysis of attractors could provide new insights into systems biology [Albert and Thakar, 2014] (e.g., the origins of cancers [Béal et al., 2021], SARS-CoV-2 [Ibrahim et al., 2021], HIV [Oyeyemi et al., 2014]).
- Attractors also play an important role in the development of new drugs [Putnins and Androulakis, 2019].
- Attractors of BNs have been also used to study various other systems, such as, multivariate systems [Yang et al., 2021], complex systems [Gates et al., 2021].

Optimal control of BNs

Optimal control of BNs is defined as the design of intervention strategies (control policies) to beneficially alter the dynamics of the considered system [Shmulevich and Dougherty, 2010].



Source: [Zhong et al., 2019]

Optimal control of BNs (cont.)

- Since BNs are logical dynamical and highly non-linear systems, control of BNs has become a hot topic in the control community [Cheng and Qi, 2009].
- It has been found in various applications in many areas, such as, systems biology [Biane and Delaplace, 2018], fault detection of logic circuits [Fornasini and Valcher, 2015], industry [Torres et al., 2018].
- Note that attractor detection also gives a starting point for many control approaches for biological systems [Biane and Delaplace, 2018].

Challenges in theories of BNs

- There are very few studies exploring the relations in dynamics among different types of BNs [Paulevé and Richard, 2012].
- The dynamics of some types of BNs (e.g., DGABNs and DA-PBNs) is not well-formulated.

Challenges in attractor detection of BNs

- Many methods and tools have been proposed, but they are mainly designed for SBNs, the simplest type of BNs.
- Few methods and tools have been proposed for ABNs, the more complex type of BNs but considered more suitable than SBNs in modeling biological systems [Saadatpour et al., 2010].
- The previous methods for attractor detection in ABNs, such as, the BDD-based methods [Garg et al., 2008], the decomposition-based methods [Mizera et al., 2018], are unable to handle large networks (e.g., networks with more than 100 nodes).
- In particular, there is the lack of practical methods for other more complex types of BNs such as GABNs and DGABNs.

Challenges in optimal control of BNs

- Many methods and tools have been proposed in recent years. However, they are mainly designed for SBNs, ABNs, or especially SPBNs.
- Very few methods [Li and Li, 2021, Faryabi et al., 2008a] have been proposed for optimal control of DGABNs or DA-PBNs; they are impractical for large networks due to they require to compute transition probability matrices of size exponential in n.
- Moreover, although some of the proposed methods for optimal control of SPBNs can avoid computing transition probability matrices of exponential size [Kobayashi and Hiraishi, 2012a, Kobayashi and Hiraishi, 2012b, Wei et al., 2017], they are still needed to improve due to their applicable ranges are still limited to medium problem instances [Akutsu, 2018].

Goal of this dissertation

In this dissertation, we aim to develop theories as well as efficient methods for attractor detection and optimal control of different types of BNs.

Why different types of BNs?

- Each type of BNs has its part in real life and can be suitable for modeling a specific type of systems; the choice among them in a specific circumstance depends on the available data and application [Faryabi et al., 2008a].
- Relations in dynamics between different types of BNs can be exploited to efficiently analyze BNs. For example, attractors of an SBN can be used to efficiently find attractors of its ABN counterpart [Garg et al., 2008].
- This consideration may provide new theoretical insights into the theory of BNs [Paulevé and Richard, 2012].

Contents

Boolean networks

Boolean Network (BN)

A Boolean Network (BN) is defined as a 2-tuple (V, F), where $V = \{x_1, ..., x_n\}$ $(n \geq 1)$ is the set of nodes and $F = \{f_1, ..., f_n\}$ is the set of Boolean functions. Each node x_i is identified as a Boolean variable, and is associated with a Boolean function $f_i : \mathbb{B}^{|IN(f_i)|} \to \mathbb{B}$, where $IN(f_i)$ is the set of input nodes of f_i. $x_i(t) \in \mathbb{B}$ and $x(t) = (x_1(t), ..., x_n(t))^\top$ denote the state of node x_i and the state of the BN at time t, respectively.

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

Dynamics of BNs

- Node x_i can update its state by

$$x_i(t+1) = f_i(x(t)).$$

- Following the updating scheme, the BN transits from a state to another state (possibly identical). This transition is called the *state transition*.
- Then, the dynamics of a BN can be represented by all possible states of the BN along with all possible state transitions from each state.

Synchronous Boolean Networks (SBNs)

- An SBN is the simplest BN model. At each time step, all its nodes will update their values simultaneously.
- Since the state transitions from a state of the SBN are time-invariant, the whole dynamics of an SBN can be captured by a State Transition Graph (STG).
- An STG is a directed graph in which each node corresponds to a state of the BN and each arc corresponds to a state transition between two states (possibly identical).
- The STG of an SBN of size n has 2^n nodes and 2^n arcs.

Asynchronous Boolean Networks (ABNs)

- An ABN can be seen as the most popular BN model.
- The updating scheme of an ABN is fully asynchronous. That is, at each time step, a single node is selected uniformly at random to be updated.
- By this updating scheme, a state of the ABN has n outgoing state transitions, making the dynamics of an ABN non-deterministic.
- Like SBNs, the whole dynamics of an ABN can be also captured by an STG. The STG of an ABN of size n has 2^n nodes and $n \times 2^n$ arcs.

Generalized Asynchronous Boolean Networks (GABNs)

- GABNs can be seen as a generalization of ABNs. At each time step, a GABN randomly and uniformly selects any number of nodes to update synchronously.
- The whole dynamics of a GABN can be also captured by an STG. The STG of a GABN of size n has 2^n nodes and $2^n \times 2^n$ arcs, making its analysis more difficult.

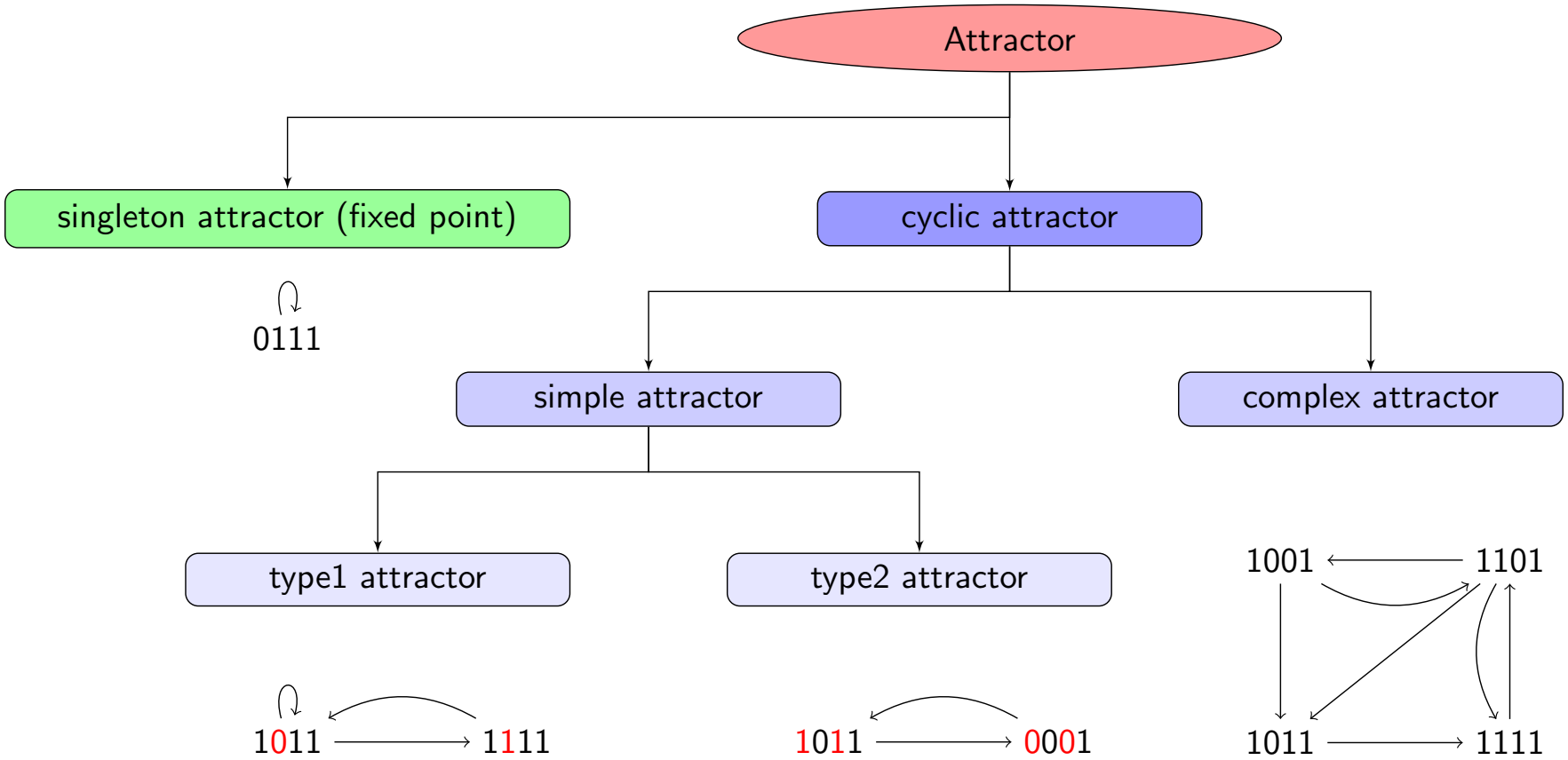
Attractors

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.

- In general, an attractor of a BN is equivalent to a bottom (terminal) Strongly Connected Component (SCC) of the STG of this BN [Garg et al., 2008].
- Since the STG of a BN has 2^n nodes and at least 2^n arcs, naive approaches for finding attractors (e.g, explicitly building the STG and then applying graph algorithms) are intractable when n is large.
- Based on [Garg et al., 2008], we can classify different types of attractors.

Classification of attractors



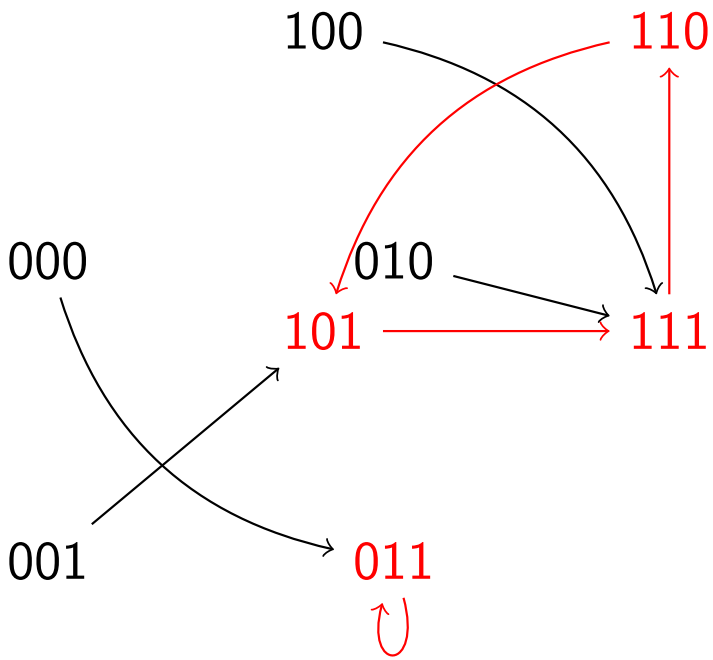
Example

Consider a BN $\mathcal{N}$ of three nodes associated to three variables (x_1, x_2, x_3). Its Boolean functions are given by

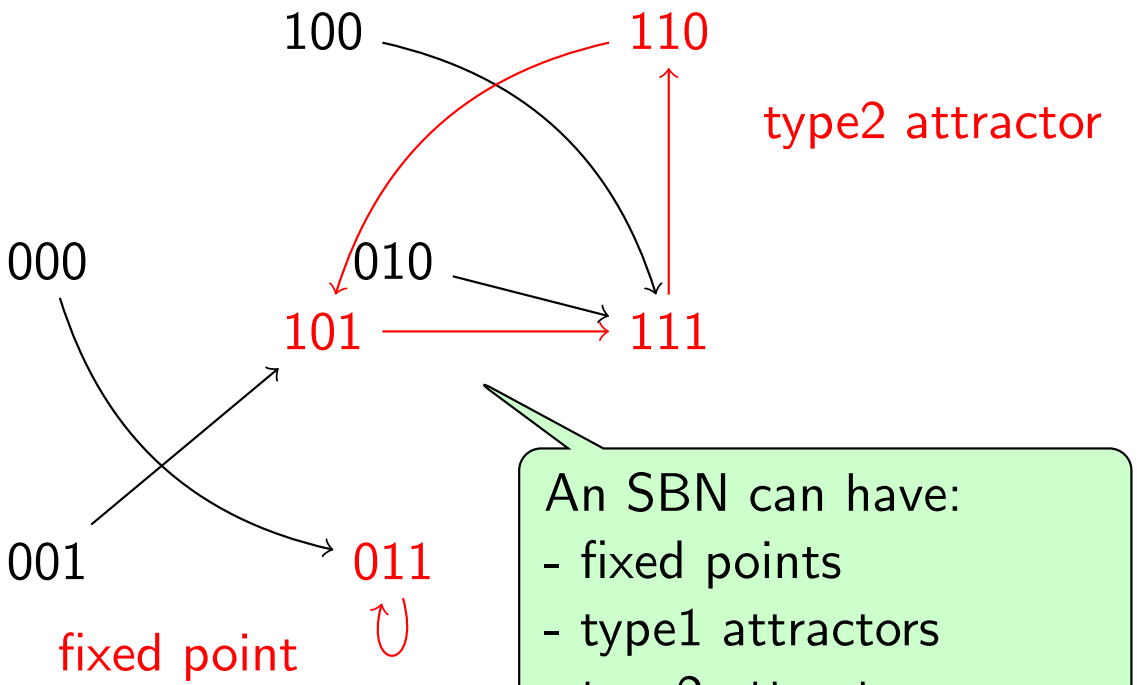
$$\begin{cases} f_1 = x_1 \vee (\neg x_1 \wedge ((\neg x_2 \wedge x_3) \vee (x_2 \wedge \neg x_3))), \\ f_2 = (\neg x_1 \wedge \neg x_3) \vee (x_2 \wedge x_3) \vee (x_1 \wedge \neg x_2), \\ f_3 = \neg x_1 \vee (x_1 \wedge (\neg x_2 \vee (x_2 \wedge \neg x_3))), \end{cases}$$

where "$\wedge$", "$\vee$", and "$\neg$" denote CONJUNCTION, DISJUNCTION, and NEGATION operators, respectively.

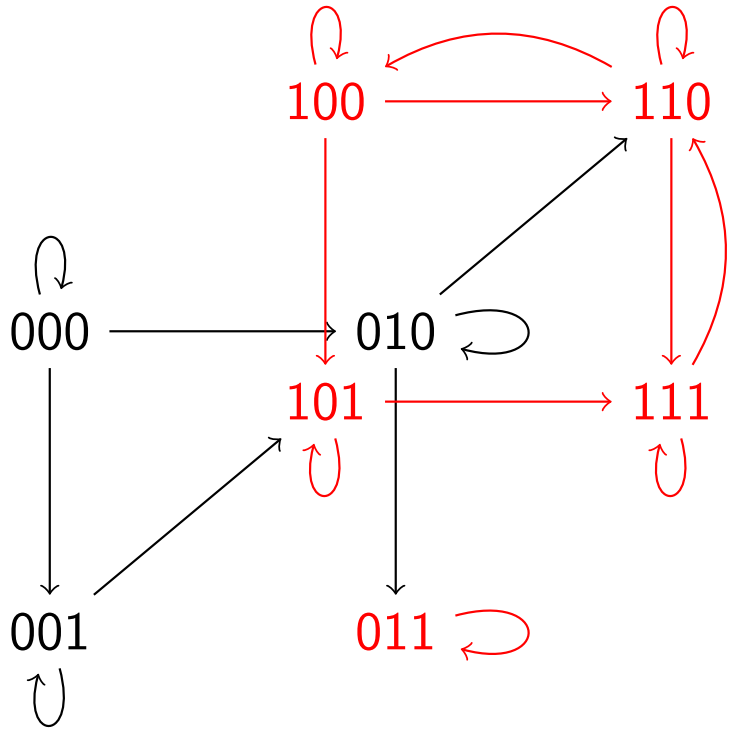
STG of the SBN counterpart



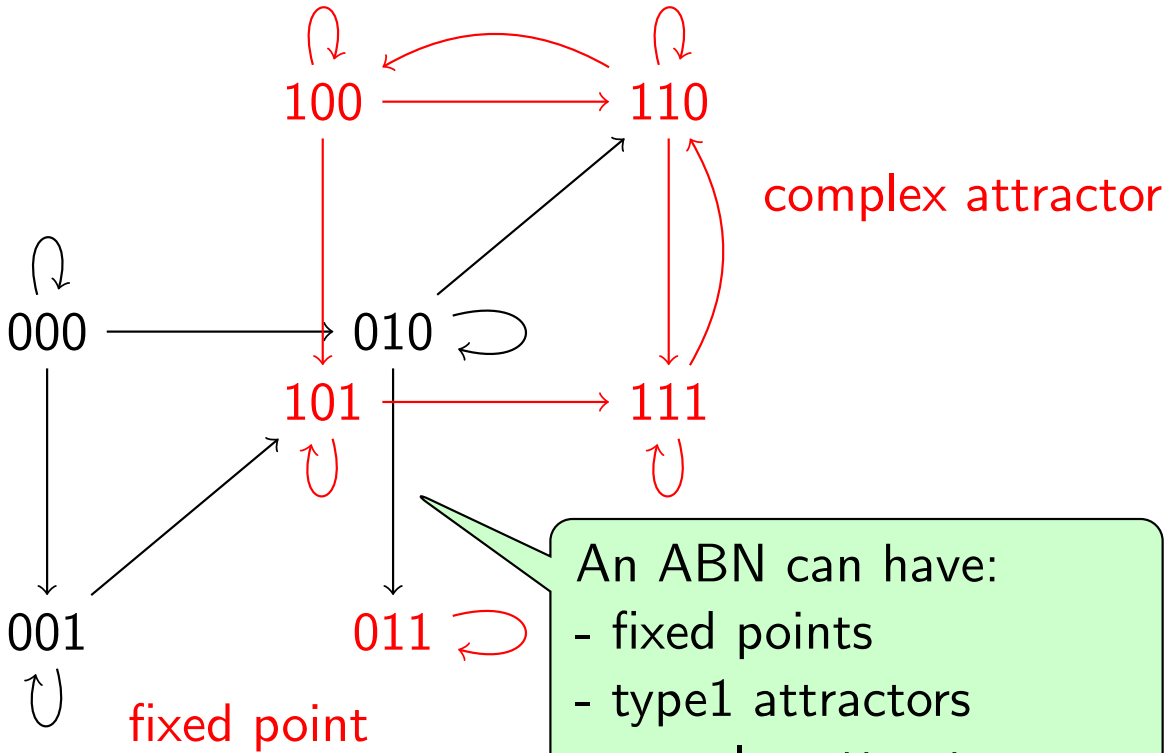
STG of the SBN counterpart



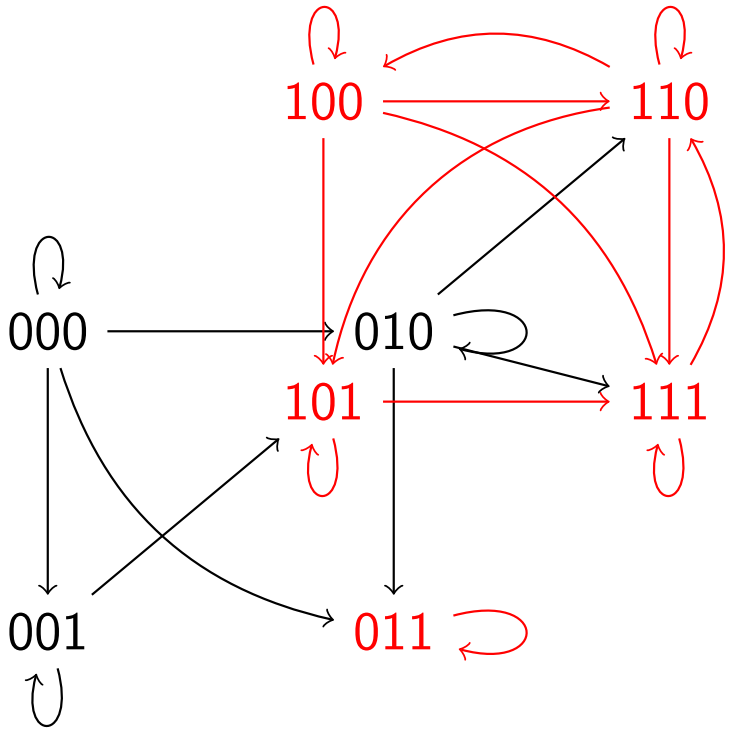
STG of the ABN counterpart



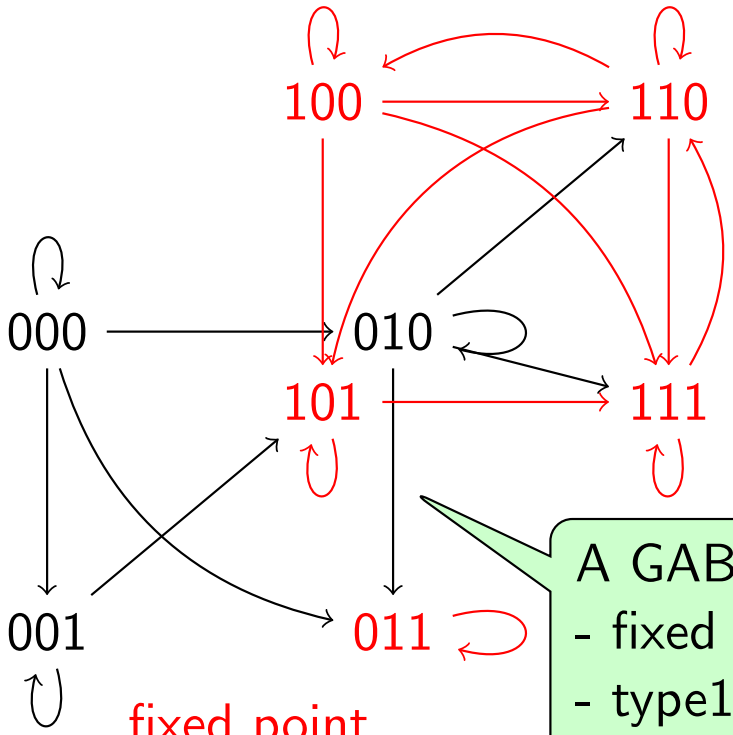
STG of the ABN counterpart



STG of the GABN counterpart



STG of the GABN counterpart



A GABN can have:
- fixed points
- type1 attractors
- complex attractors

Probabilistic Boolean networks

Probabilistic Boolean Network (PBN)

A Probabilistic Boolean Network (PBN) is defined as a triple (V, F, C), where $V = \{x_1, ..., x_n\}$ $(n \geq 1)$ is the set of nodes, $F = \{F_1, ..., F_n\}$, and $C = \{C_1, ..., C_n\}$. Each node x_i is identified as a Boolean variable, and is associated with a non-empty set of Boolean functions, $F_i = \{f_1^{(i)}, ..., f_{l_i}^{(i)}\}, l_i \geq 1$. Each Boolean function $f_j^{(i)}$ has a probability of selection associated with it, $c_j^{(i)}$. Thus, $C_i = \left\{c_1^{(i)}, ..., c_{l_i}^{(i)}\right\}$ such that $\sum_{j=1}^{l_i} c_j^{(i)} = 1$. The state of a node or a PBN at time t is defined as same as that of a BN.

Dynamics of PBNs

- At each time step, node x_i updates its state by

$$x_i(t+1) = f_j^{(i)}(x(t)),$$

  where $f_j^{(i)}$ is selected from F_i with the probability $c_j^{(i)}$.
- Following the updating scheme, the PBN transits from a state to another state (possibly identical) with a probability. This transition is called the *probability transition*.
- Then, the dynamics of a PBN can be represented by all possible states of the PBN along with all possible probability transitions from each state.

Synchronous PBNs (SPBNs)

- Like SBNs, all the nodes of an SPBN update their values synchronously at each time step.
- Then, the whole dynamics of an SPBN can be captured by an STG in which a transition probability is attached to an arc.

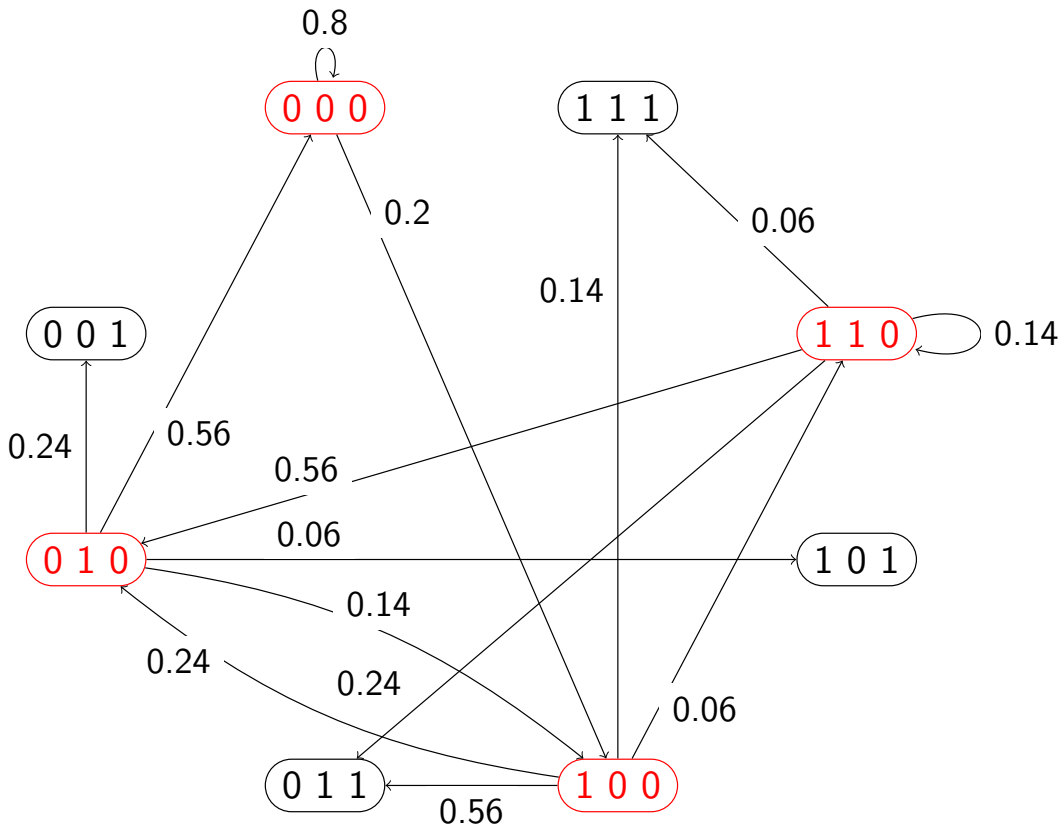
Example

Consider an SPBN $\mathcal{SP}$ [Akutsu, 2018]

$$f^{(1)} = \begin{cases} f_1^{(1)} = x_3, c_1^{(1)} = 0.8, \\ f_2^{(1)} = \neg x_3, c_2^{(1)} = 0.2, \end{cases}$$

$$f^{(2)} = f_1^{(2)} = x_1 \wedge \neg x_3, c_1^{(2)} = 1.0,$$

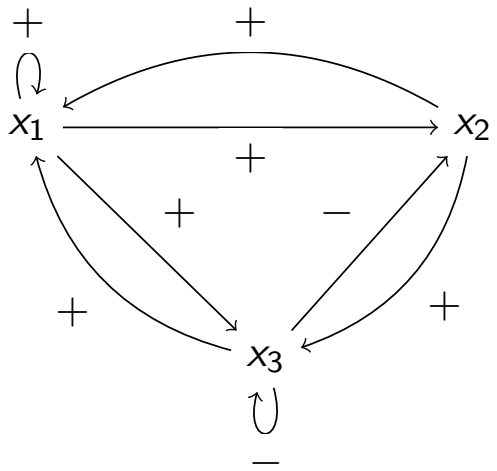
$$f^{(3)} = \begin{cases} f_1^{(3)} = x_1 \wedge \neg x_2, c_1^{(3)} = 0.7, \\ f_2^{(3)} = x_2, c_2^{(3)} = 0.3. \end{cases}$$



Interaction graph

The *interaction graph* of a BN depicts the qualitative interactions between nodes and is usually represented as a signed directed graph on the set of nodes. An interaction between two nodes can be positive (+) or negative (-). Consider a BN

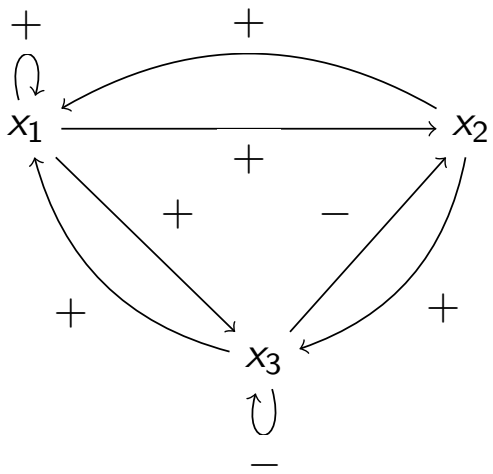
$$f_1 = x_1 \wedge x_2 \wedge x_3,$$
$$f_2 = x_1 \vee \neg x_3,$$
$$f_3 = (x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$



Interaction graph

The *interaction graph* of a BN depicts the qualitative interactions between nodes and is usually represented as a signed directed graph on the set of nodes. An interaction between two nodes can be positive (+) or negative (-). Consider a BN

$$f_1 = x_1 \wedge x_2 \wedge x_3,$$
$$f_2 = x_1 \vee \neg x_3,$$
$$f_3 = (x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

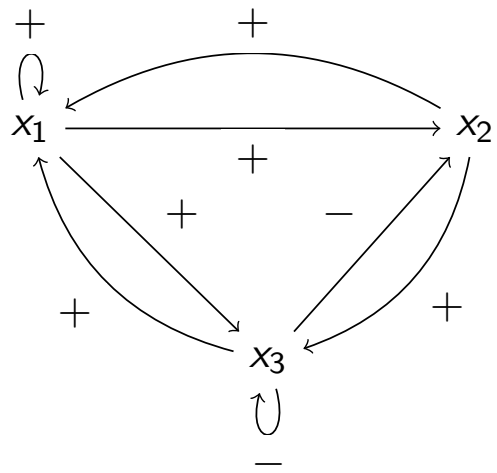


A *Feedback Vertex Set* (FVS) of G is a set of vertices U such that $G - U$ contains no cycle. This graph has two FVSs:
- $\{x_1, x_3\}$ (the minimum one),
- $\{x_1, x_2, x_3\}$.

Interaction graph

The *interaction graph* of a BN depicts the qualitative interactions between nodes and is usually represented as a signed directed graph on the set of nodes. An interaction between two nodes can be positive (+) or negative (-). Consider a BN

$$f_1 = x_1 \wedge x_2 \wedge x_3,$$
$$f_2 = x_1 \vee \neg x_3,$$
$$f_3 = (x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$

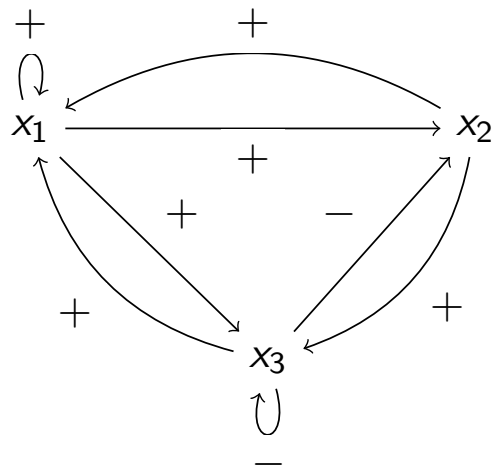


A *Negative Feedback Vertex Set* (NFVS) of G is a set of vertices U such that $G - U$ contains no negative cycle (cycle with an odd number of negative arcs). This graph has four NFVSs:
- $\{x_3\}$ (the minimum one),
- $\{x_1, x_3\}$,
- $\{x_2, x_3\}$,
- $\{x_1, x_2, x_3\}$.

Interaction graph

The *interaction graph* of a BN depicts the qualitative interactions between nodes and is usually represented as a signed directed graph on the set of nodes. An interaction between two nodes can be positive (+) or negative (-). Consider a BN

$$f_1 = x_1 \wedge x_2 \wedge x_3,$$
$$f_2 = x_1 \vee \neg x_3,$$
$$f_3 = (x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (x_1 \wedge x_2 \wedge x_3).$$



A *Positive Feedback Vertex Set* (PFVS) of G is a set of vertices U such that $G - U$ contains no positive cycle (cycle with an even number of negative arcs). This graph has four PFVSs:
- $\{x_1\}$ (the minimum one),
- $\{x_1, x_2\}$,
- $\{x_1, x_3\}$,
- $\{x_1, x_2, x_3\}$.

Optimal control of BNs

Standard formulation

Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

Optimal control of BNs

internal nodes stand for usual nodes (i.e., genes or proteins), external (control) nodes can stand for external interventions (e.g., drugs, radiation, or chemotherapy)

Standard formulation

Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

Optimal control of BNs

the initial state can stand for a disease or cancerous state, and the desired state can stand for a healthy or normal state

Standard formulation

Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

Optimal control of BNs

$u_i(k) = 0$ implies that u_i is not applied at time k, whereas $u_i(k) = 1$ implies that u_i is applied at time k with the application cost $g(u_i)$.

Standard formulation

Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

Optimal control of BNs

Standard formulation

Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

There are two control modes as follows.

- The time-sensitive mode: The condition $x(M) = x^{des}$ must be strictly satisfied, i.e., the BN must reach the desired state at exactly the target time M.
- The non-time-sensitive mode: The condition $x(M) = x^{des}$ can be relaxed, i.e., the BN can reach the desired state before or at time step M.

Optimal control of BNs

Standard formulation

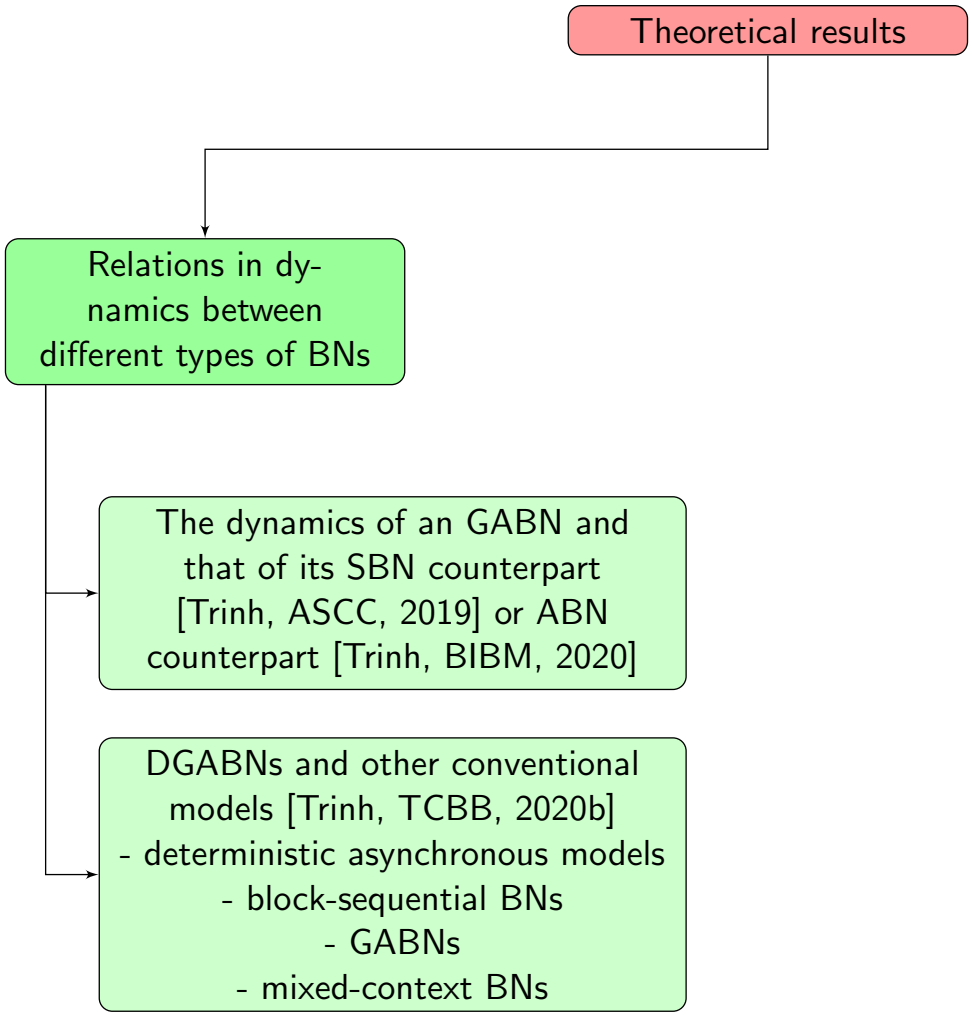
Given a BN including a set of internal nodes ($X = \{x_1, ..., x_n\}$) and a set of external (control) nodes ($U = \{u_1, ..., u_m\}$), an initial state $x^{ini} \in \{0,1\}^{1\times n}$, a desired state $x^{des} \in \{0,1\}^{1\times n}$, a target time M, and a cost vector $g \in \mathbb{N}^{1\times m}$. Let decide whether or not there exists a control sequence of 0-1 control vectors $\langle u(0), ..., u(M-1)\rangle$ such that $x(0) = x^{ini}$, $x(M) = x^{des}$, and the linear cost function $C = \sum_{j=0}^{M-1}(\sum_{i=1}^{m}(u_i(j) \times g(u_i)))$ is minimum. Then, output one if it exists.

Note that there are several variants of the standard formulation designed for different types of BNs as well as different aims of control, e.g.,

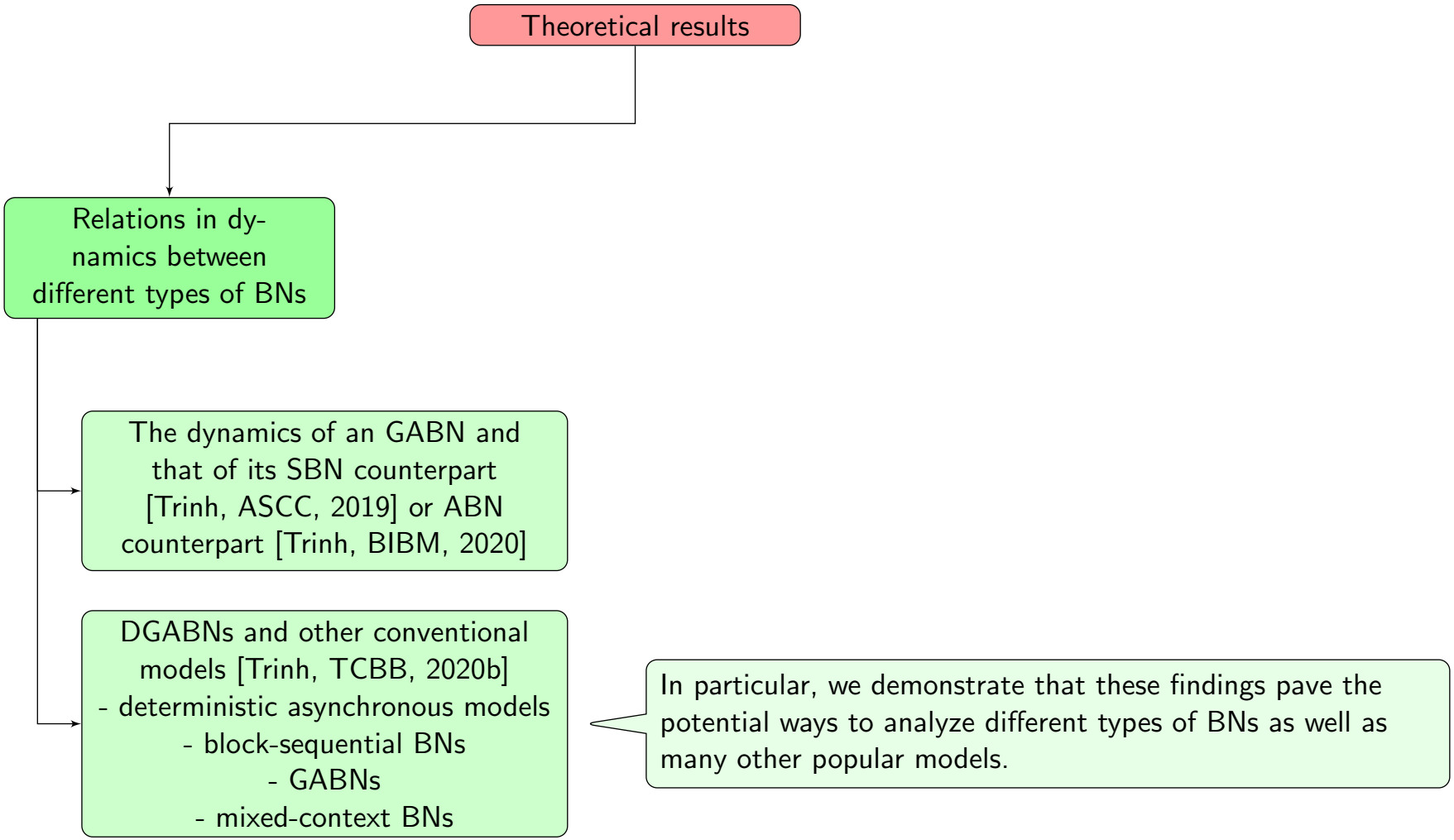
- removing the cost vector and the cost function but considering the maximum (or minimum) probability of reaching the desired state for the case of SPBNs [Kobayashi and Hiraishi, 2012b];
- removing the desired state but considering the average expected cost for the case of SPBNs [Datta et al., 2003, Kobayashi and Hiraishi, 2012a].

Contents

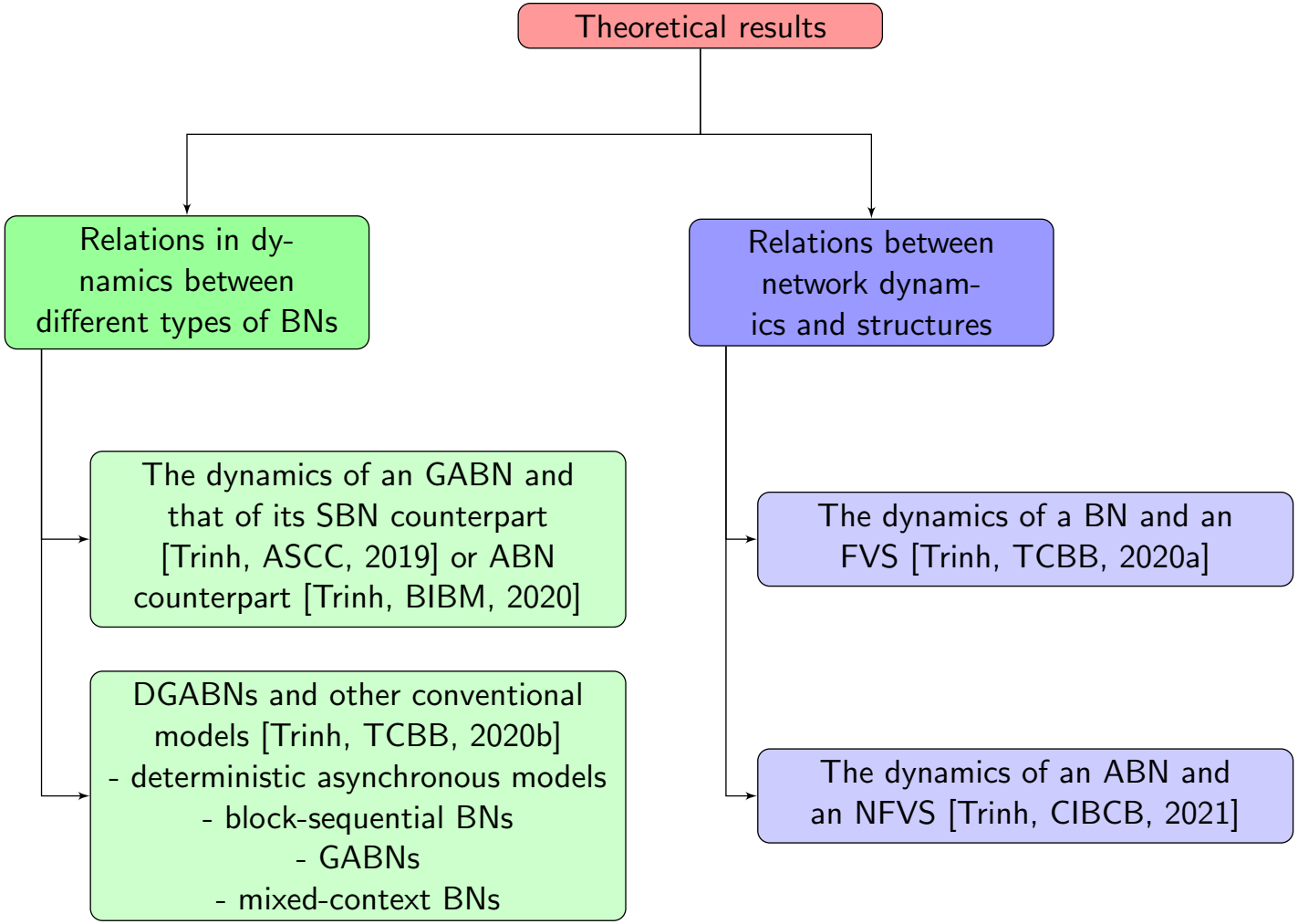
Theoretical contributions



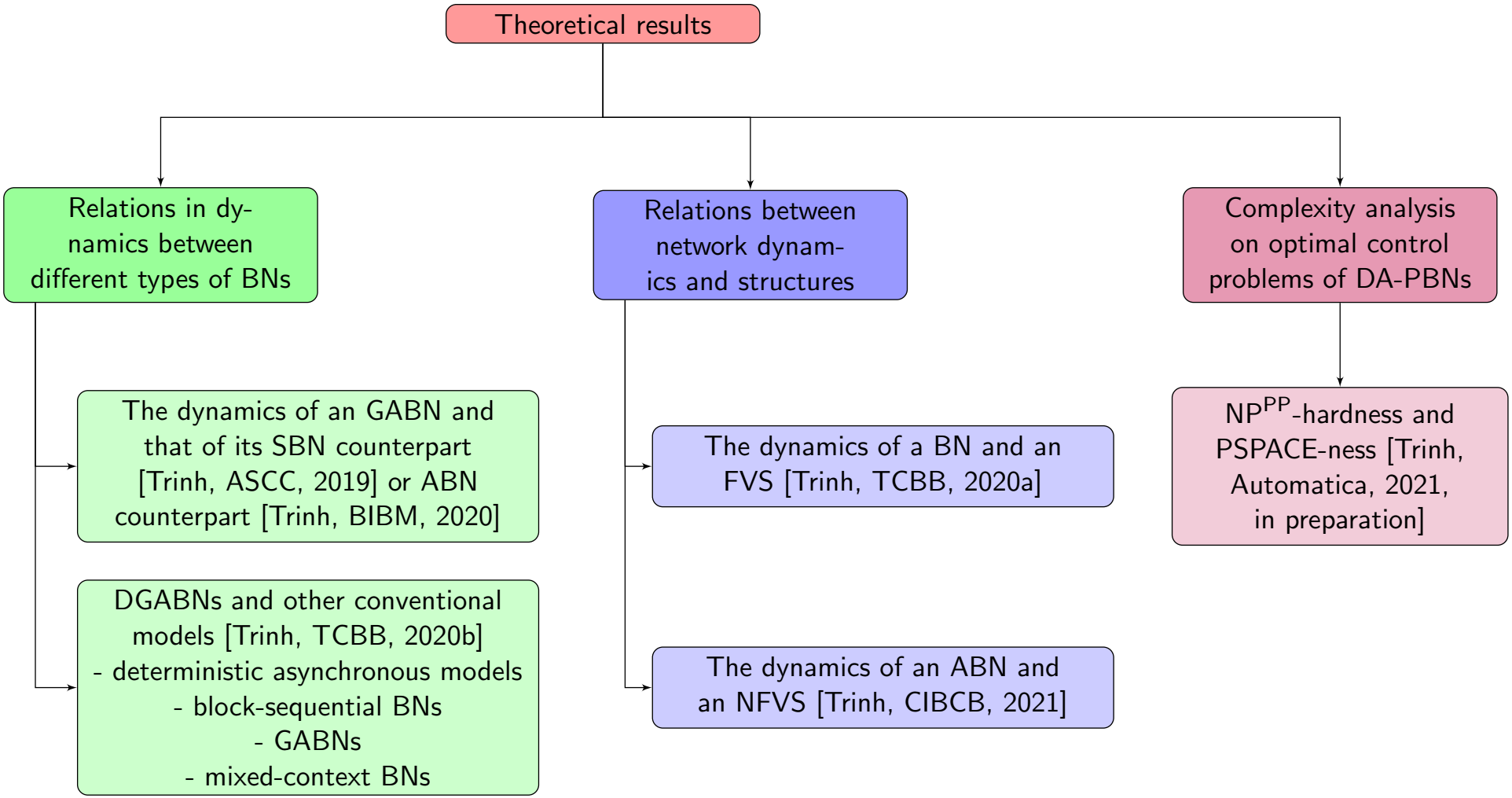
Theoretical contributions



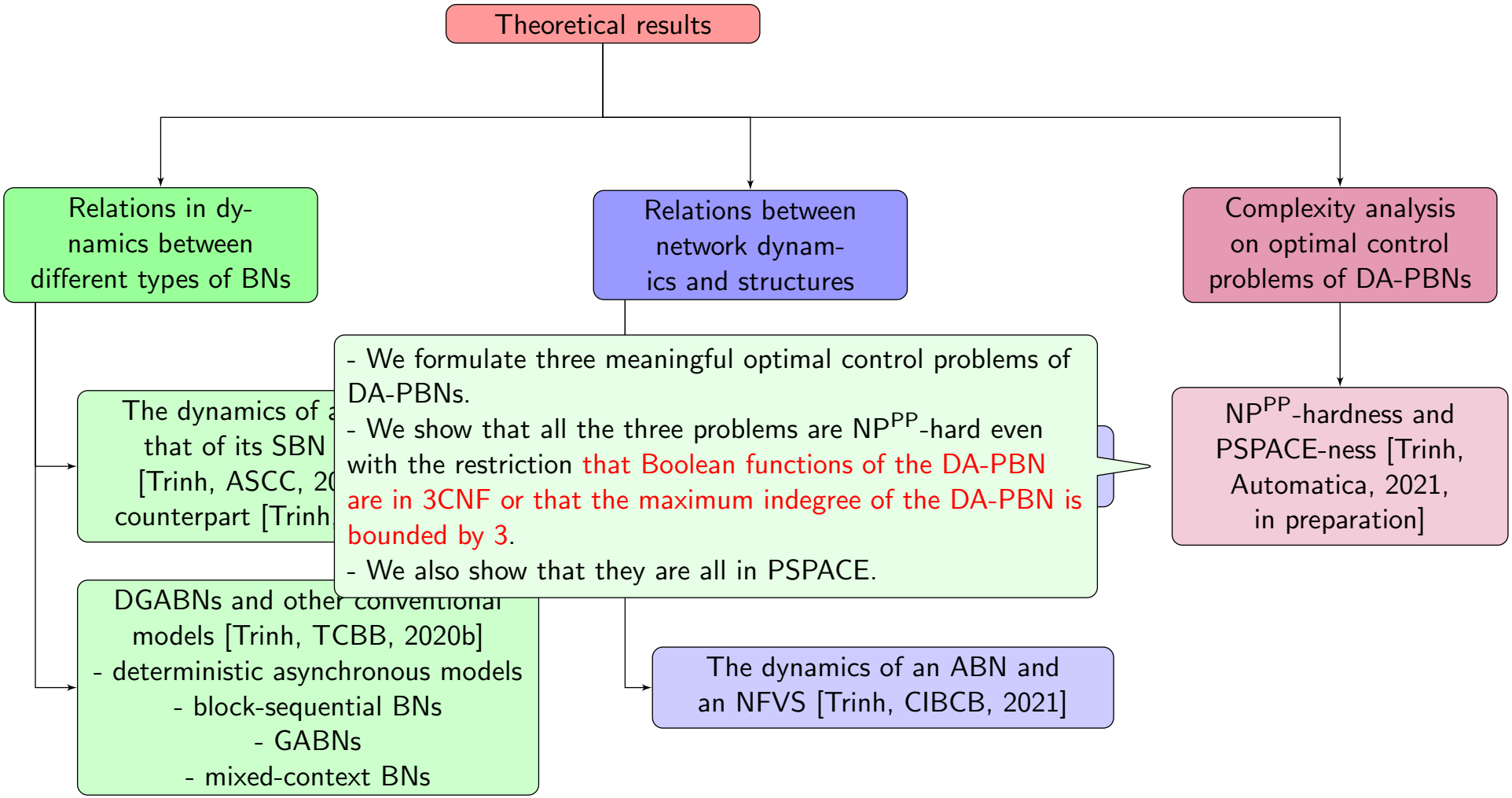
Theoretical contributions



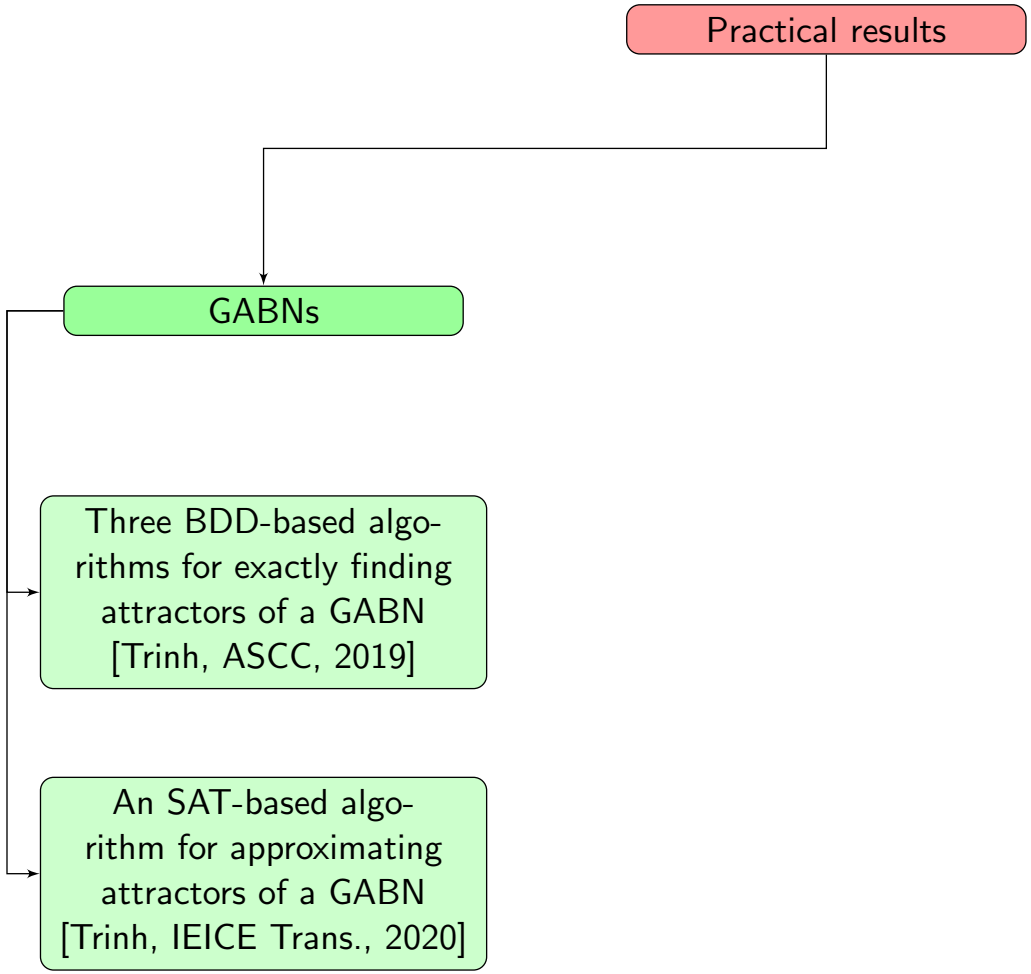
Theoretical contributions



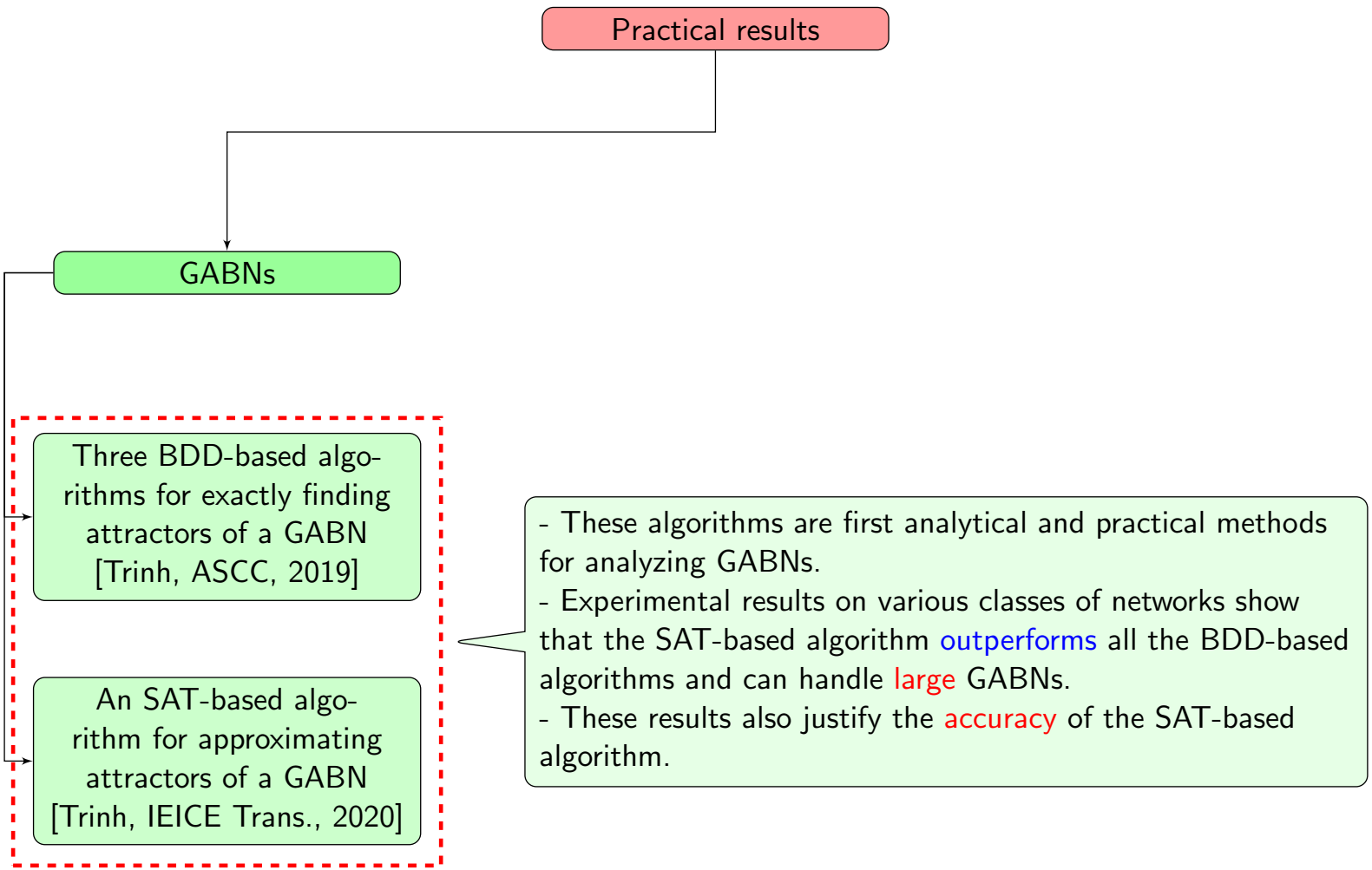
Theoretical contributions



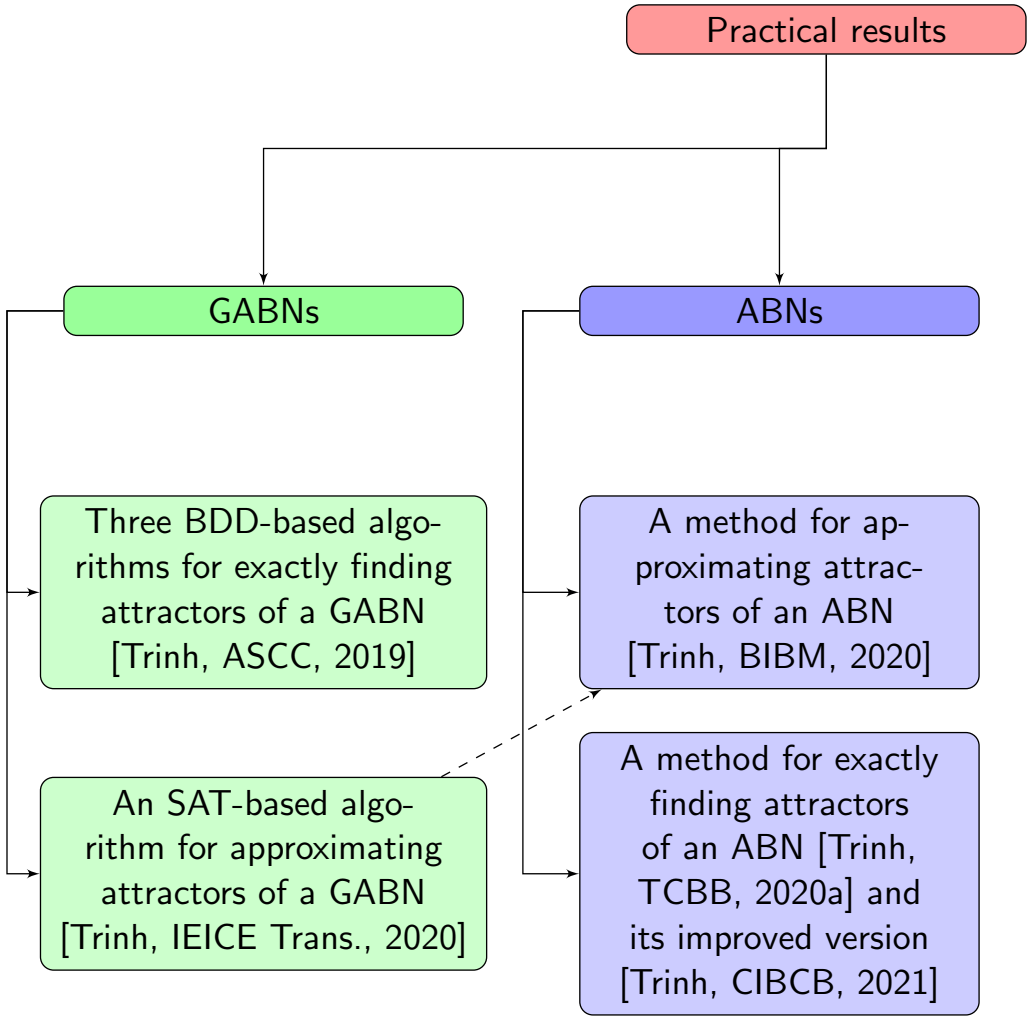
Practical contributions



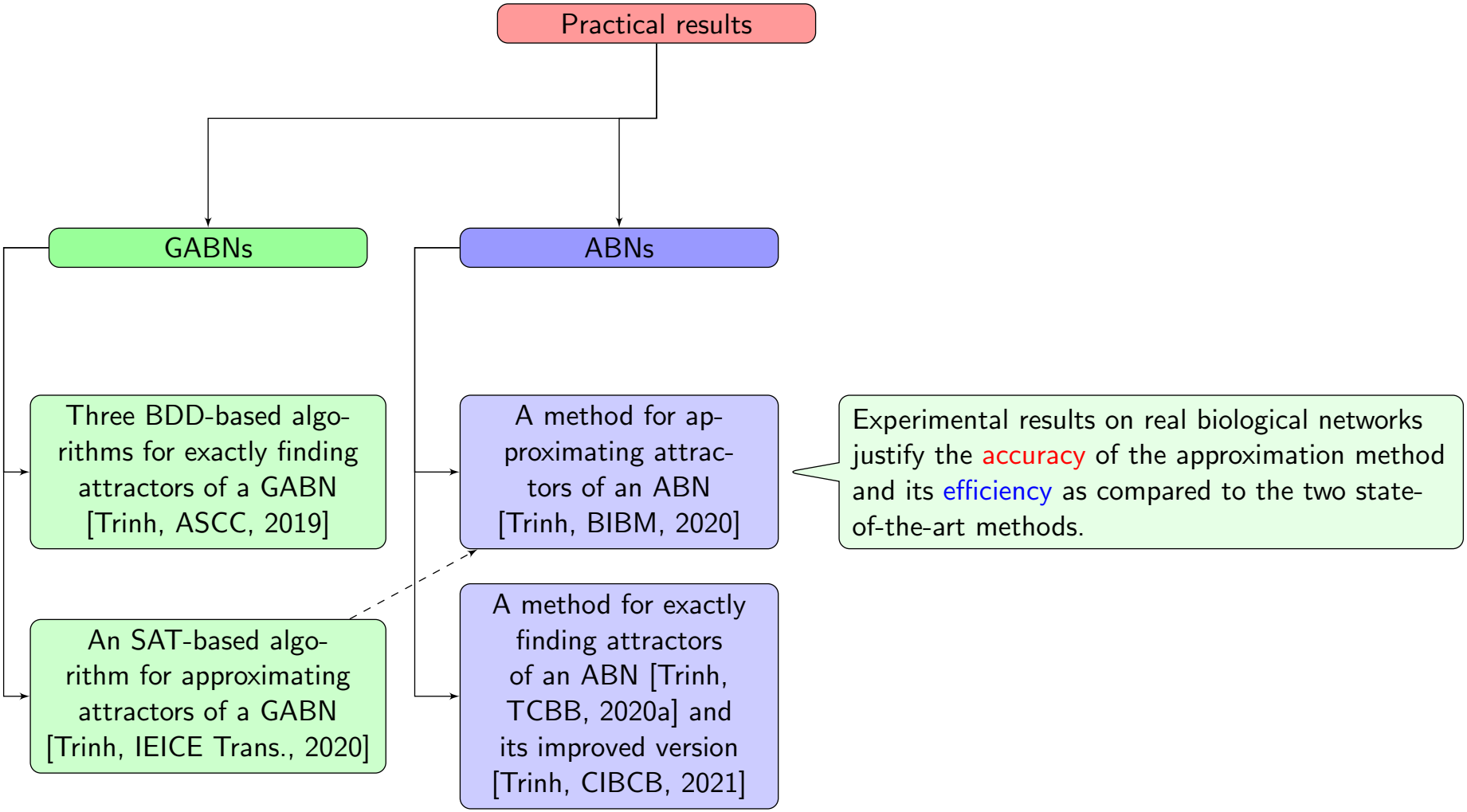
Practical contributions



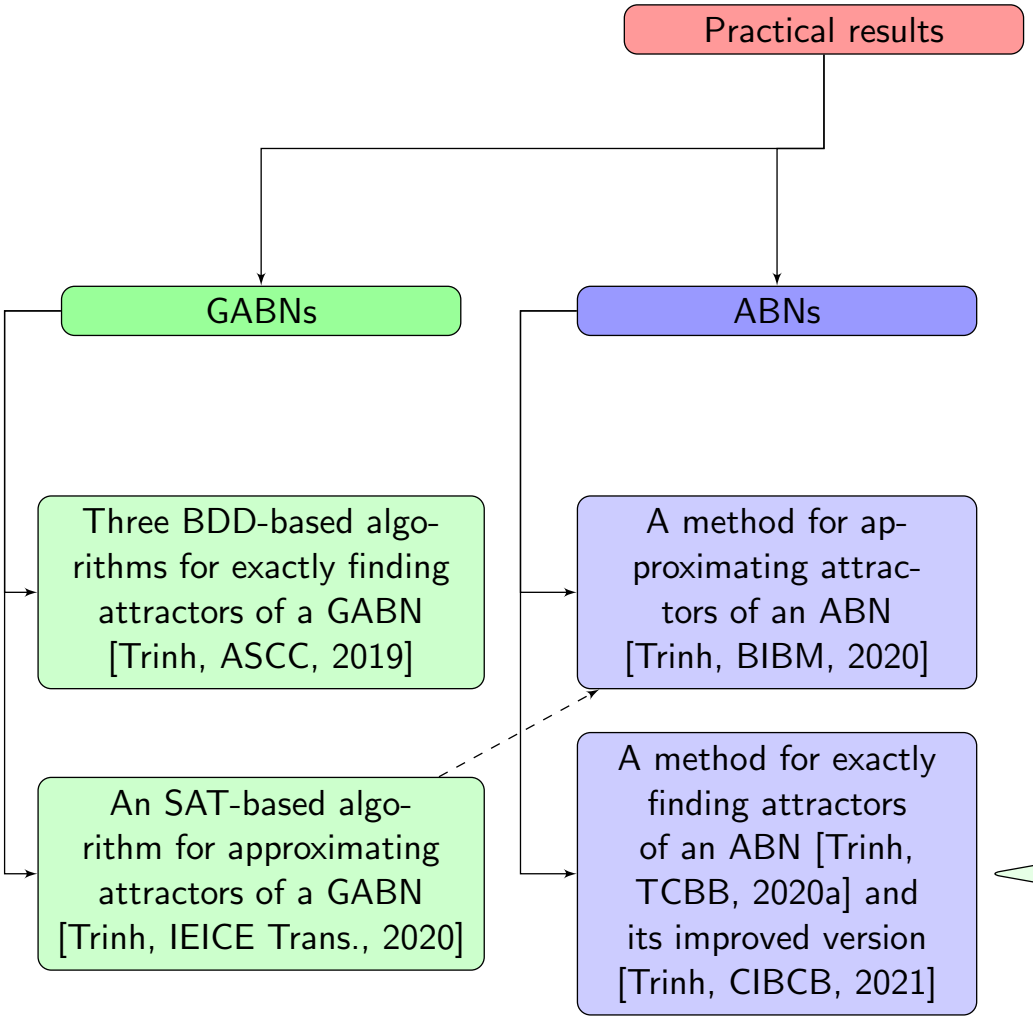
Practical contributions



Practical contributions

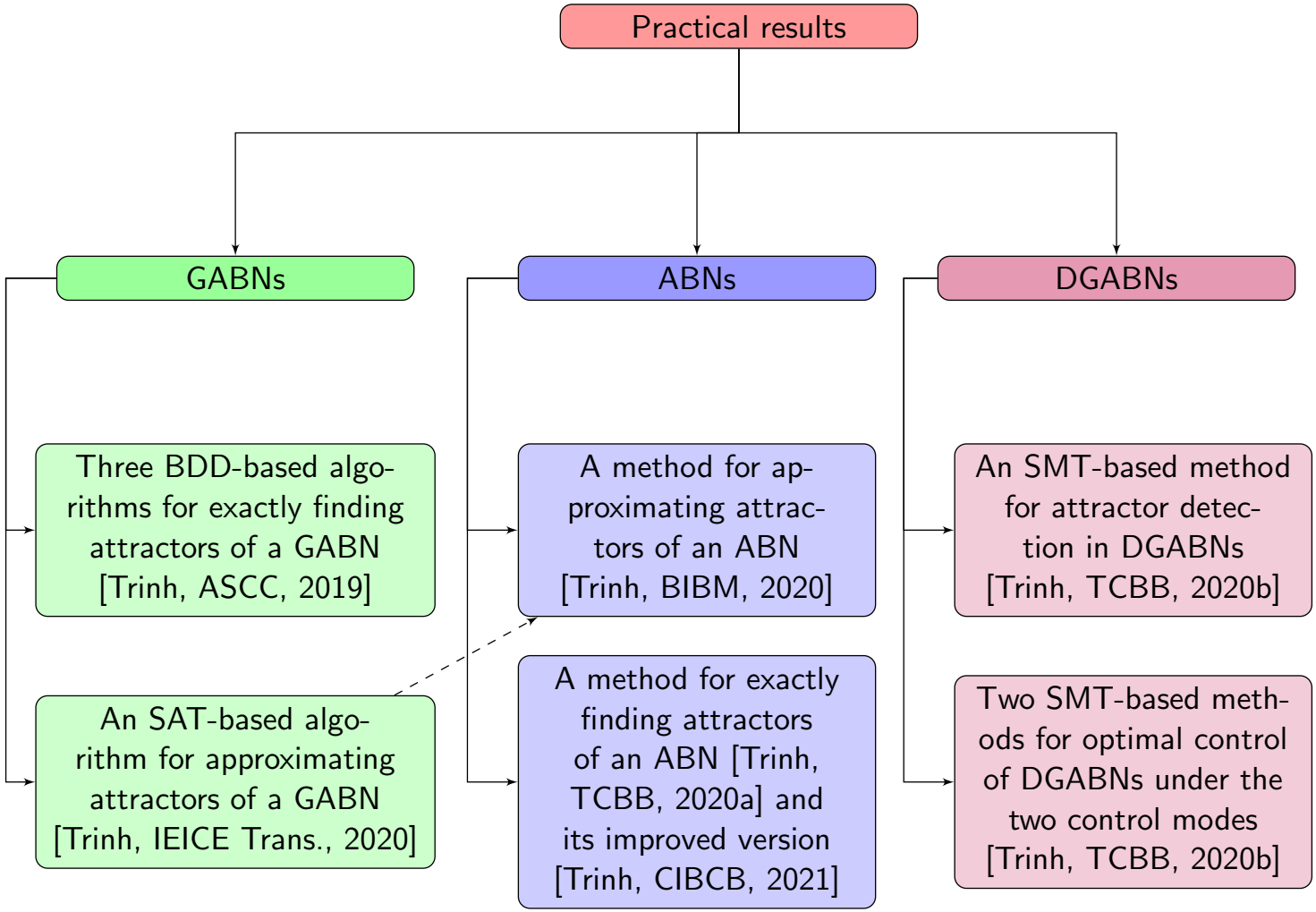


Practical contributions

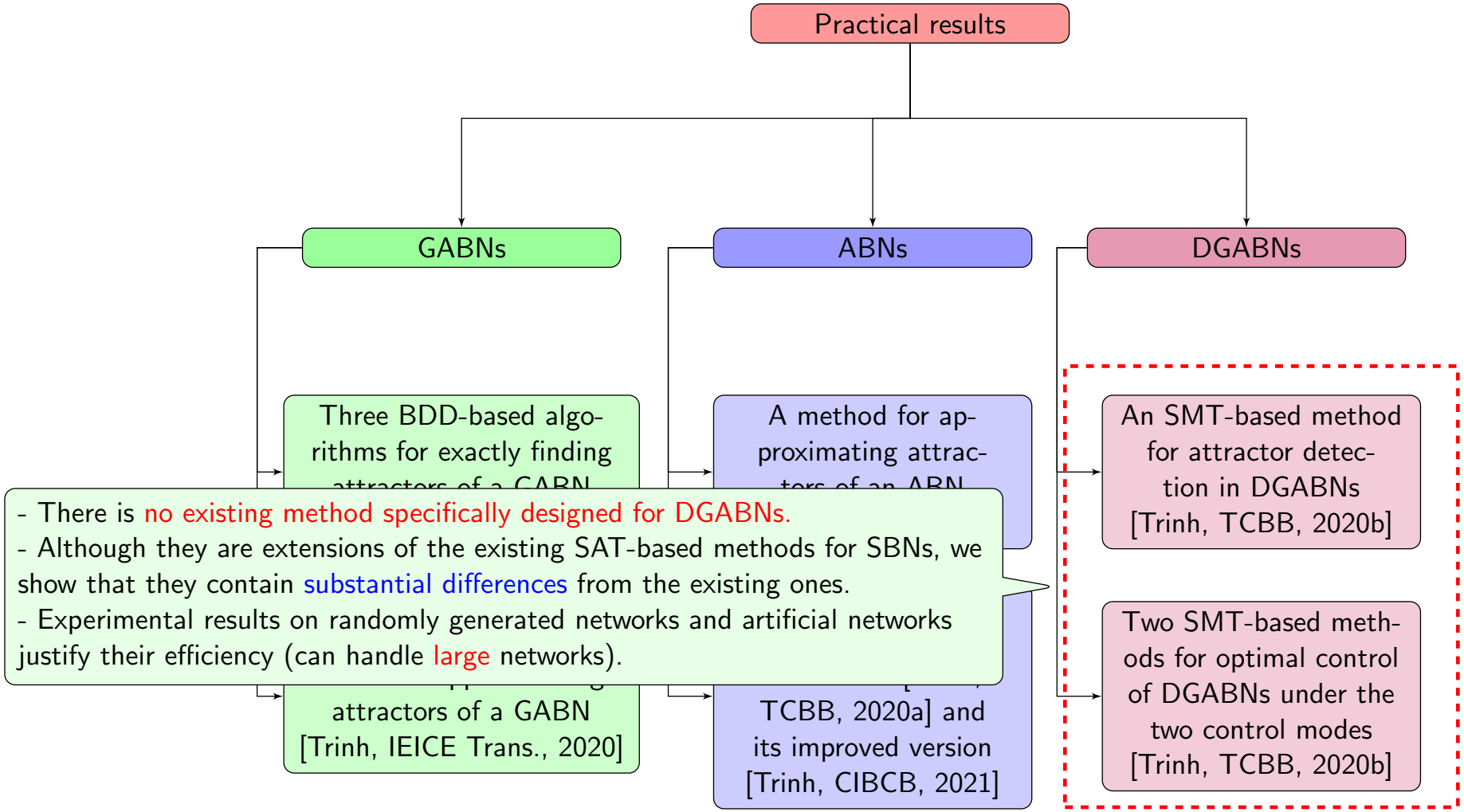


This method outperforms the state-of-the-art methods and can handle very large networks with up to 1000 nodes in terms of randomly generated networks and more than 300 nodes in terms of real biological networks.

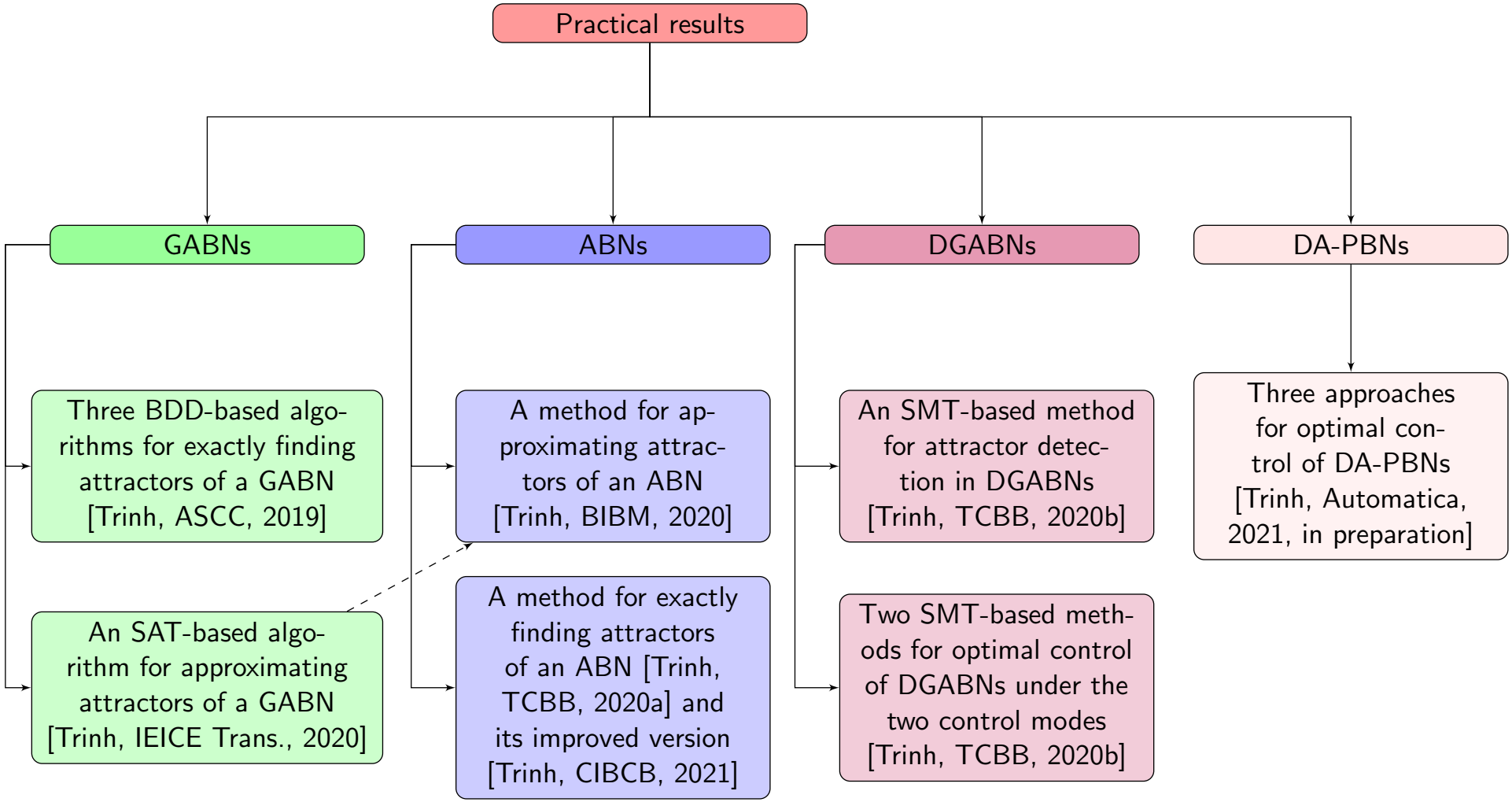
Practical contributions



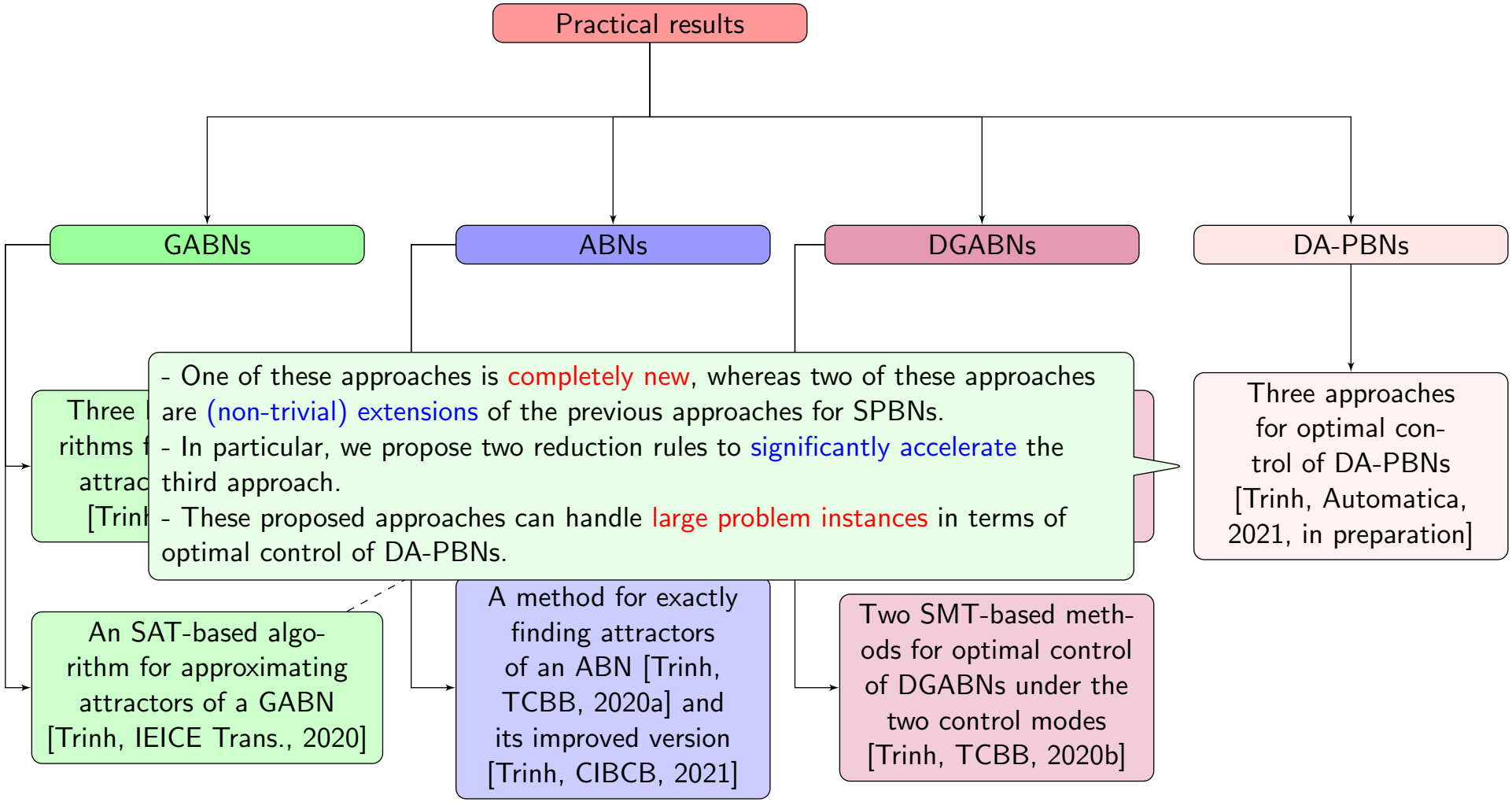
Practical contributions



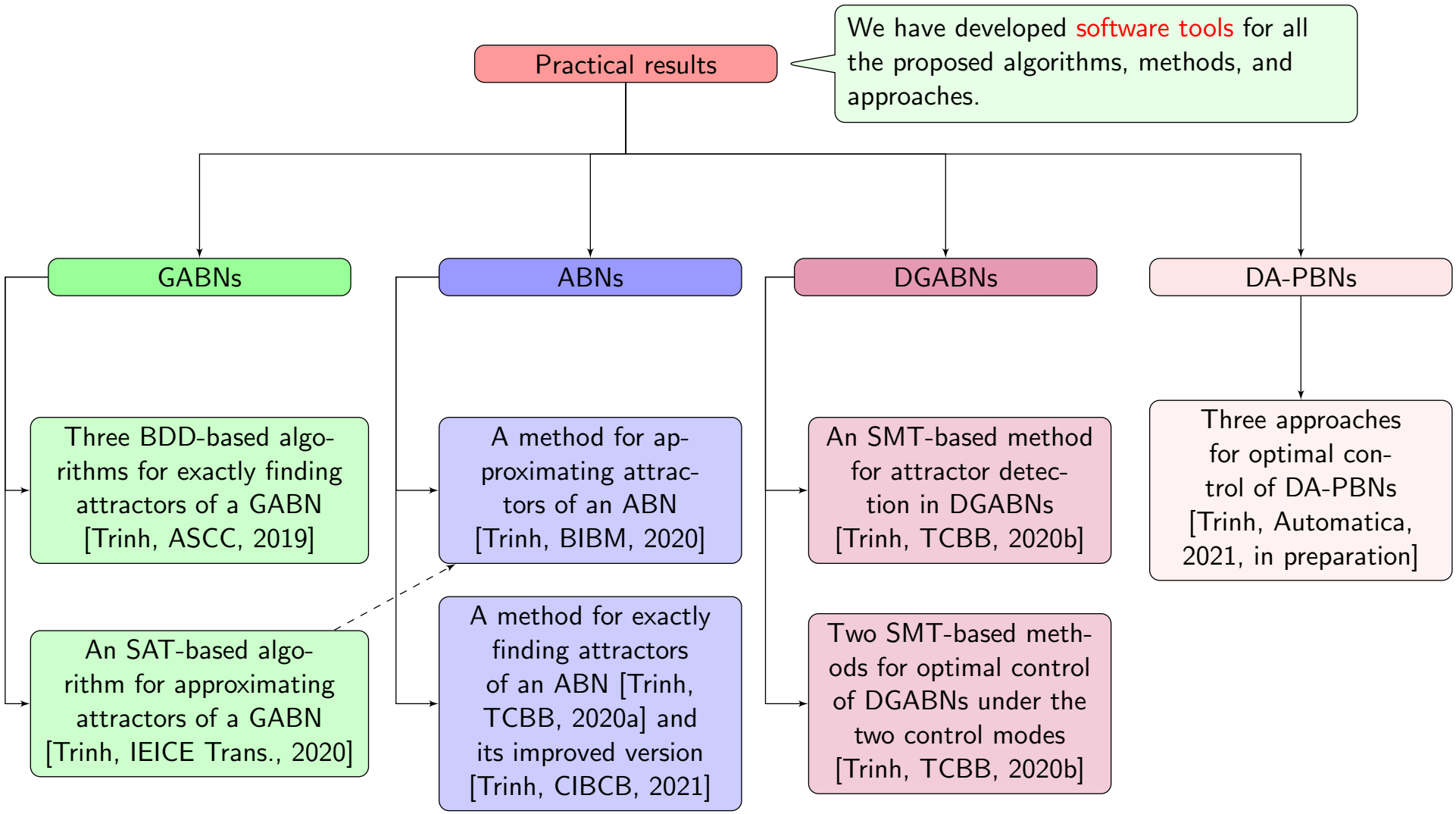
Practical contributions



Practical contributions



Practical contributions



Contents

Relations in dynamics between GABNs and SBNs

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.

Relations in dynamics between GABNs and SBNs

Lemma 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. If s is a state of $\mathcal{S}$, then $FR^{\mathcal{S}}(\{s\}) \subseteq FR^{\mathcal{G}}(\{s\})$.

Relations in dynamics between GABNs and SBNs

Lemma 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. If s is a state of $\mathcal{S}$, then $FR^{\mathcal{S}}(\{s\}) \subseteq FR^{\mathcal{G}}(\{s\})$.

Lemma 3.3.2 [Gershenson, 2002]

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. $\mathcal{G}$ and $\mathcal{S}$ have the same set of singleton attractors.

Lemma 3.3.3

Let $\mathcal{G}$ be a GABN and and $\mathcal{S}$ be its SBN counterpart. $\mathcal{G}$ and $\mathcal{S}$ have the same set of type1 attractors.

Relations in dynamics between GABNs and SBNs

Lemma 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. If s is a state of $\mathcal{S}$, then $FR^{\mathcal{S}}(\{s\}) \subseteq FR^{\mathcal{G}}(\{s\})$.

Lemma 3.3.2 [Gershenson, 2002]

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. $\mathcal{G}$ and $\mathcal{S}$ have the same set of singleton attractors.

Lemma 3.3.3

Let $\mathcal{G}$ be a GABN and and $\mathcal{S}$ be its SBN counterpart. $\mathcal{G}$ and $\mathcal{S}$ have the same set of type1 attractors.

Theorem 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. Any attractor of $\mathcal{G}$ always contains an attractor of $\mathcal{S}$.

Relations in dynamics between GABNs and SBNs

Lemma 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. If s is a state of $\mathcal{S}$, then

Based on these relations, we
- propose three BDD-based algorithms and an SAT-based algorithm for attractor detection in GABNs,
- state and prove several relations in dynamics between GABNs and ABNs.

set of singleton attractors.

Lemma 3.3.3

Let $\mathcal{G}$ be a GABN and and $\mathcal{S}$ be its SBN counterpart. $\mathcal{G}$ and $\mathcal{S}$ have the same set of type1 attractors.

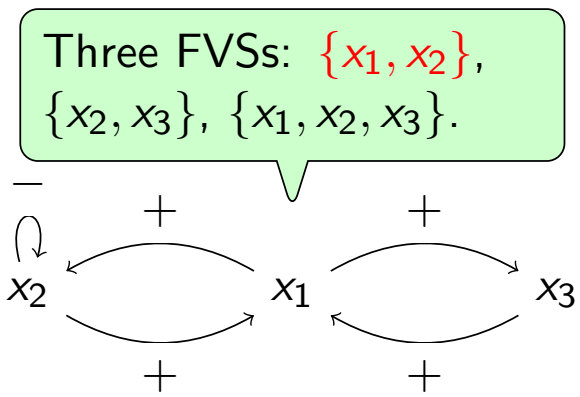
Theorem 3.3.1

Let $\mathcal{G}$ be a GABN and $\mathcal{S}$ be its SBN counterpart. Any attractor of $\mathcal{G}$ always contains an attractor of $\mathcal{S}$.

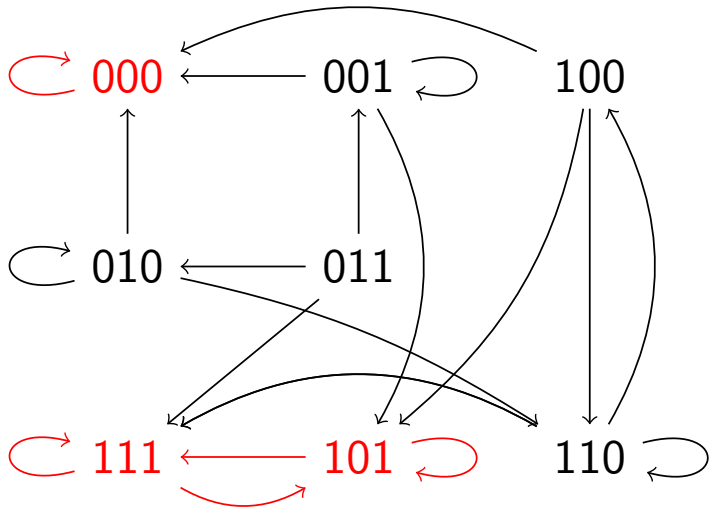
Illustrative example

Example BN

$$\begin{cases} f_1 = x_2 \vee x_3, \\ f_2 = x_1 \wedge \neg x_2, \\ f_3 = x_1. \end{cases}$$



(a) Interaction graph



(b) STG of the ABN counterpart

Relations between FVSs and BNs

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

Lemma 4.3.1

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

Lemma 4.3.2

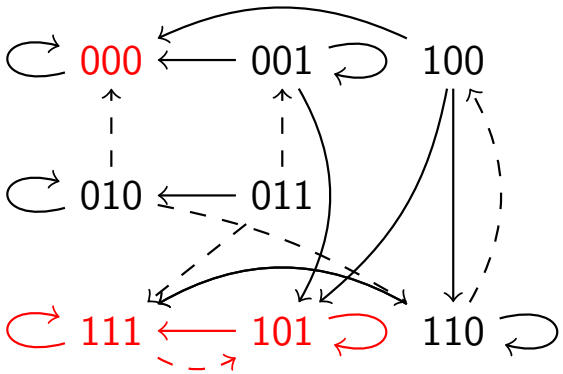
Let $\mathcal{N}$ be a BN and its STG be G. Let U be an FVS of $\mathcal{N}$. Then G has no cycles such that all the values of the nodes in U do not change through these cycles.

Trinh Van Giang, Tatsuya Akutsu and Kunihiko Hiraishi: "An FVS-based approach to attractor detection in asynchronous random Boolean networks," *IEEE/ACM TCBB*, 2020, in press.

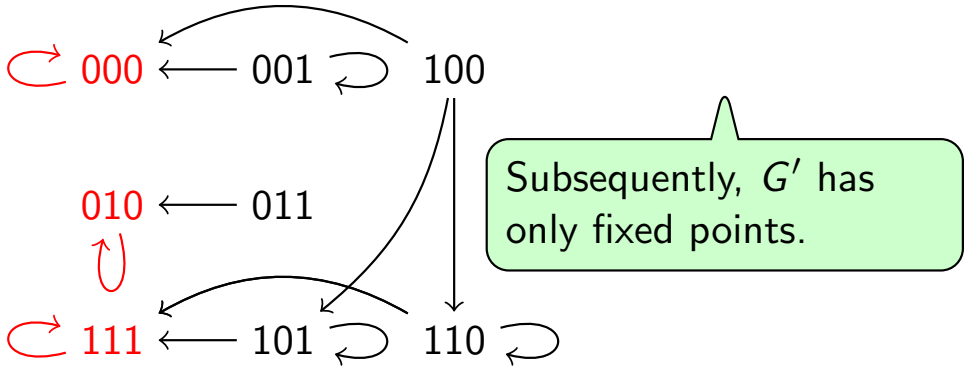
Relations between FVSs and BNs (cont.)

Theorem 4.3.1

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 a set of Boolean 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 $1 - b_{i_j}$. In other words, the value of a node $x_{i_j} \in U$ in a state in G' is retained if this value is equal to b_{i_j}. With this meaning, we call B as a set of "retained" values. Then G' has no cycles.



(a) G

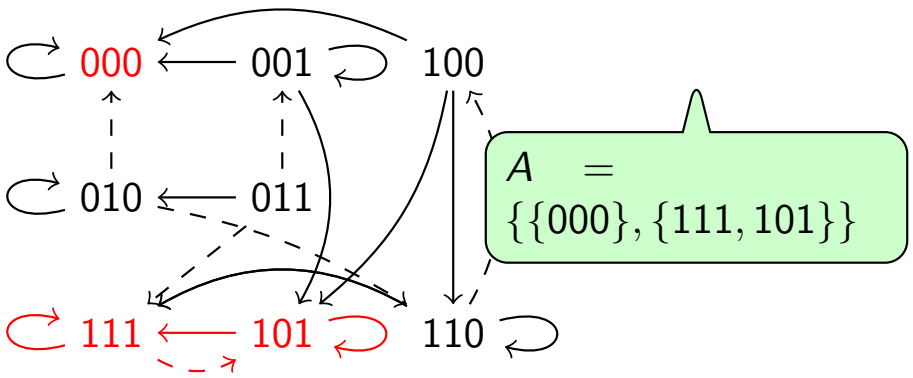


(b) G' with $b_1 = 0, b_2 = 1$

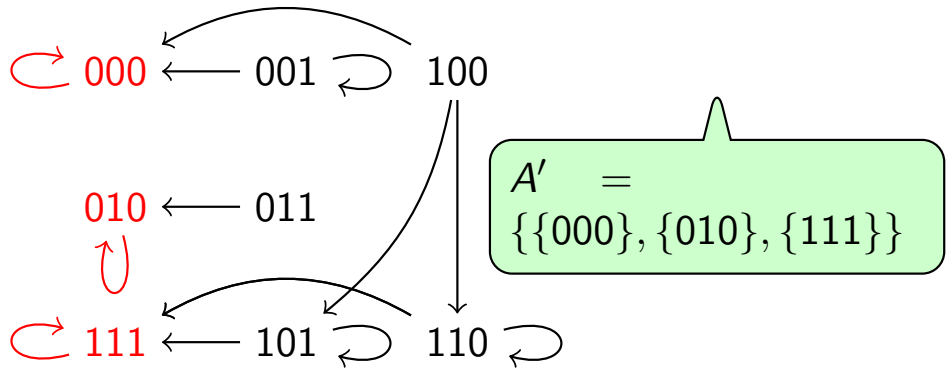
Relations between FVSs and BNs (cont.)

Theorem 4.3.2

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'.

General approach of **FVS-ABN**

- **FVS-ABN** uses an FVS to systematically remove arcs in the STG of the ABN to get a candidate set of states that covers all attractors of the ABN (by Theorems 4.3.1 and 4.3.2).
- 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 formally prove the correctness of **FVS-ABN**.

Constituent steps of **FVS-ABN**

- This method includes several constituent steps.
- 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.
- We here show the two most important steps.

Preprocessing

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

Output: The set A of states.

1: Find an FVS $U = \{x_{i_1}, ..., x_{i_k}\}$ of $IG(\mathcal{A})$
2: Choose a set $B = \{b_{i_1}, ..., b_{i_k}\}$ of retained values corresponding to the nodes of U
3: Let G be the STG of $\mathcal{A}$
4: 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
5: $F_{fix} \leftarrow$ the set of fixed points of G
6: $F \leftarrow$ the set of fixed points of G'
7: $F \leftarrow F \backslash F_{fix}$
8: Perform **Preprocessing-SSF** to shrink the set F
9: $F \leftarrow F \backslash F_{fix}$
16: **end while**
17: **return** A

In each iteration 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**.

Reachability analysis

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

Output: The set A of states.

1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11: **while** $F \neq \emptyset$ **do**
12: Remove a state s from F
13: **if** **UnfReach**$(\mathcal{A}, s, A \cup F) = false$ **then**
14: $A \leftarrow A \cup \{s\}$
15: **end if**
16: **end while**
17: **return** A

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 procedure 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 target set) 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.)

Results on real biological networks

				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	-	-
				8	0	15.61	3556.85	440.16
				6	108	55.56	21198.63	916.23
				6	260	174.66	-	-
				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 **genY-sis** [Garg et al., 2008] 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	-	-
				8	0	15.61	3556.85	440.16
				6	108	55.56	21198.63	916.23
				6	260	174.66	-	-
				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** [Mizera et al., 2018] 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							-
HGF_Signaling_in_Keratinocytes	68							8.75
Colitis_associated_colon_cancer	70							-
Bcell	72							29.84
YeastApoptosis	73	17	8448	4352	4352	[illegible]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 can handle large networks in reasonable time.

Relations between NFVs and ABNs

Trinh Van Giang and Kunihiko Hiraishi: "An improved method for finding attractors of large-scale asynchronous Boolean networks", *IEEE CIBCB*, 2021.

Theorem 4.6.2

Let $\mathcal{A}$ be an ABN. Let U^- be an NFVS of $IG(\mathcal{A})$ and B^- be a set of retained values corresponding to the nodes of U^-. Let att be an attractor of $\mathcal{A}$. Then there exists a state s such that $s \in att$ and s is a fixed point of the reduced STG with respect to U^- and B^-.

We also show that Theorem 4.6.2 does not hold for the case of PFVSs.

Relations between NFVs and ABNs

Trinh Van Giang and Kunihiko Hiraishi: "An improved method for finding attractors of large-scale asynchronous Boolean networks", *IEEE CIBCB*,

Based on this theorem, we improve **FVS-ABN** by using an NFVS instead of an FVS to get the candidate set. The use of NFVSs opens a chance to get a smaller candidate set.

Let $\mathcal{A}$ be an ABN. Let U^- be an NFVS of $IG(\mathcal{A})$ and B^- be a set of retained values corresponding to the nodes of U^-. Let att be an attractor of $\mathcal{A}$. Then there exists a state s such that $s \in att$ and s is a fixed point of the reduced STG with respect to U^- and B^-.

We also show that Theorem 4.6.2 does not hold for the case of PFVSs.

Results on real biological networks

			FVS-ABN		iFVS-ABN	
name	n	$\|A\|$	$\|U\|$	time (secs)	$\|U^-\|$	time (secs)
Differentiation_of_T_lymphocytes	50	2050	18	952.73	6	578.09
HumanMyelomaCells	67	83	13	173.25	6	105.42
HGF_Signaling_in_Keratinocytes	68	72	10	2.34	0	0.54
Influenza_A_Virus_Replication_Cycle	131	524	29	-	10	42.33
Signaling_in_Macrophage_Activation	321	4096	16	21216.07	1	6712.42
Wg_Pathway_of_Drosophila	26	16384	15	3.67	1	3.67
TumourCell	32	9	10	0.81	5	0.54
TCellSignaling	40	8	5	0.46	2	0.45
Treatment_of_Castration_Resistant	42	16384	14	0.40	0	0.36
Senescence	51	17	10	22.91	5	18.94
PC12CellDifferentiation	62	3	2	0.45	0	0.46
YeastApoptosis	73	8448	16	76.20	3	76.74
IL_6_Signalling	86	32768	21	2127.45	10	2056.36

Results on real biological networks

the improved version of **FVS-ABN**

			FVS-ABN		**iFVS-ABN**	
name	n	$\|A\|$	$\|U\|$	time (secs)	$\|U^-\|$	time (secs)
Differentiation_of_T_lymphocytes	50	2050	18	952.73	6	578.09
HumanMyelomaCells	67	83	13	173.25	6	105.42
HGF_Signaling_in_Keratinocytes	68	72	10	2.34	0	0.54
Influenza_A_	not obtain the result within 10 hours			→ -	10	42.33
Signaling_in_Macrophage_Activation	321	4096	16	21216.07	1	6712.42
Wg_Pathway_of_Drosophila	26	16384	15	3.67	1	3.67
TumourCell	32	9	10	0.81	5	0.54
TCellSignaling	40	8	5	0.46	2	0.45
Treatment_of_Castration_Resistant	42	16384	14	0.40	0	0.36
Senescence	51	17	10	22.91	5	18.94
PC12CellDifferentiation	62	3	2	0.45	0	0.46
YeastApoptosis	73	8448	16	76.20	3	76.74
IL_6_Signalling	86	32768	21	2127.45	10	2056.36

Results on real biological networks

In 5/13 BNs, **iFVS-ABN** is much faster than **FVS-ABN**

			FVS-ABN		iFVS-ABN	
name	n	$\lvert A\rvert$	$\lvert U\rvert$	time (secs)	$\lvert U^-\rvert$	time (secs)
Differentiation_of_T_lymphocytes	50	2050	18	952.73	6	578.09
HumanMyelomaCells	67	83	13	173.25	6	105.42
HGF_Signaling_in_Keratinocytes	68	72	10	2.34	0	0.54
Influenza_A_Virus_Replication_Cycle	131	524	29	-	10	42.33
Signaling_in_Macrophage_Activation	321	4096	16	21216.07	1	6712.42
Wg_Pathway_of_Drosophila	26	16384	15	3.67	1	3.67
TumourCell	32	9	10	0.81	5	0.54
TCellSignaling	40	8	5	0.46	2	0.45
Treatment_of_Castration_Resistant	42	16384	14	0.40	0	0.36
Senescence	51	17	10	22.91	5	18.94
PC12CellDifferentiation	62	3	2	0.45	0	0.46
YeastApoptosis	73	8448	16	76.20	3	76.74
IL_6_Signalling	86	32768	21	2127.45	10	2056.36

Results on real biological networks

In the 8/13 remaining BNs, the running time of **iFVS-ABN** is comparable to that of **FVS-ABN**.

			FVS-ABN		**iFVS-ABN**	
name	n	$\|A\|$	$\|U\|$	time (secs)	$\|U^-\|$	time (secs)
Differentiation_of_T_lymphocytes	50	2050	18	952.73	6	578.09
HumanMyelomaCells	67	83	13	173.25	6	105.42
HGF_Signaling_in_Keratinocytes	68	72	10	2.34	0	0.54
Influenza_A_Virus_Replication_Cycle	131	524	29	-	10	42.33
Signaling_in_Macrophage_Activation	321	4096	16	21216.07	1	6712.42
Wg_Pathway_of_Drosophila	26	16384	15	3.67	1	3.67
TumourCell	32	9	10	0.81	5	0.54
TCellSignaling	40	8	5	0.46	2	0.45
Treatment_of_Castration_Resistant	42	16384	14	0.40	0	0.36
Senescence	51	17	10	22.91	5	18.94
PC12CellDifferentiation	62	3	2	0.45	0	0.46
YeastApoptosis	73	8448	16	76.20	3	76.74
IL_6_Signalling	86	32768	21	2127.45	10	2056.36

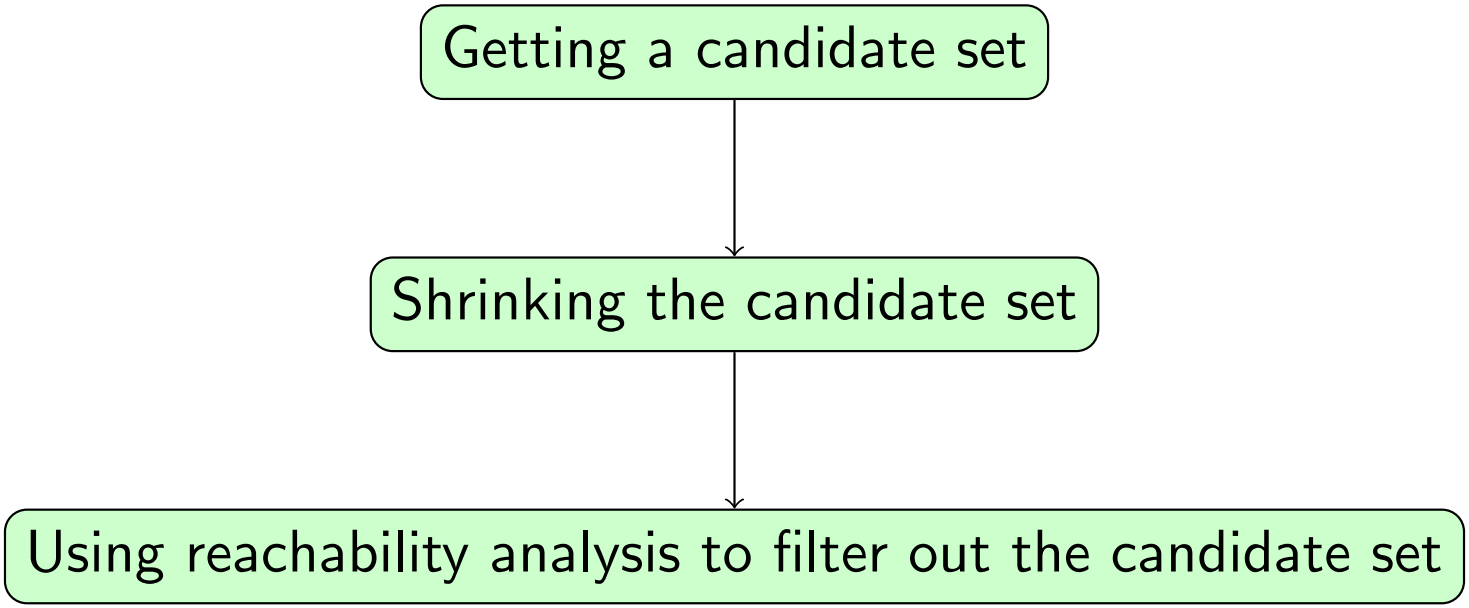
Results on real biological networks

			FVS-ABN		iFVS-ABN	
name	n	$\|A\|$	$\|U\|$	time (secs)	$\|U^-\|$	time (secs)
Differentiation_of_T_lymphocytes	50	2050	18	952.73	6	578.09
HumanMyelomaCells	67	83	13	173.25	6	105.42
HGF_Signaling_in_Keratinocytes	68	72	10	2.34	0	0.54
Influenza_A_Virus_Replication_Cycle	131	524	29	-	10	42.33
Signaling_in_Macrophage_Activation	321	4096	16	21216.07	1	6712.42
Wg_Pathway_of_Drosophila	26	16384	15	3.67	1	3.67
TumourCell	32	9	10	0.81	5	0.54
TCellSignaling	40	8	5	0.46	2	0.45
Treatment_of_Castration_Resistant	42	16384	14	0.40	0	0.36
Senescence	51	17	10	22.91	5	18.94
PC12CellDifferentiation	62	3	2	0.45	0	0.46
YeastApoptosis	73	8448	16	76.20	3	76.74
IL_6_Signalling	86	32768	21	2127.45	10	2056.36

These observations show the efficiency of the use of NFVSs.

Notable implication

Especially, the principle that we developed for attractor detection in ABNs can be generalized as a blueprint for attractor detection in various types of BNs beyond ABNs.



Reasons for studying DGABNs

- DGABNs offer an interesting compromise between SBNs and ABNs, thus could provide a suitable modeling formalism of various types of systems [Greil et al., 2007]. Many applications of DGABNs in various fields can be found (see, e.g., [Waidyarathne and Samarasinghe, 2018, Sherekar and Viswanathan, 2021]).
- DGABNs are general and interesting mathematical objects.
  - SBNs are a special case of DGABNs [Gershenson, 2002, Greil et al., 2007].
  - Studying DGABNs can be a good starting point for further studies on more complex models such as deterministic asynchronous BNs [Gershenson, 2002] and mixed-context BNs [Gershenson et al., 2003] that are constructed based on DGABNs.
- To our best knowledge, all the previous studies on DGABNs are theoretical or simulation-based. There lack practical methods for DGABNs.

Deterministic Generalized Asynchronous Boolean Networks (DGABNs)

- A DGABN is a BN where each node x_i is also associated with two parameters: $p_i \in \mathbb{N}^+$ and $q_i \in \mathbb{N}$ $(q_i < p_i)$.
  - Let γ denote the least common multiple of all p's.
  - The set of all $p's$ and $q's$ is called the *context* of the DGABN.

Deterministic Generalized Asynchronous Boolean Networks (DGABNs)

- A DGABN is a BN where each node x_i is also associated with two parameters: $p_i \in \mathbb{N}^+$ and $q_i \in \mathbb{N}$ $(q_i < p_i)$.
  - Let γ denote the least common multiple of all p's.
  - The set of all $p's$ and $q's$ is called the *context* of the DGABN.
- At time t, node x_i will be updated by $x_i(t+1) = f_i(x(t))$ when the modulus of time t over p_i is equal to q_i (i.e., $t\%p_i = q_i$). If two or more nodes can be updated, they will be updated simultaneously.

Deterministic Generalized Asynchronous Boolean Networks (DGABNs)

- A DGABN is a BN where each node x_i is also associated with two parameters: $p_i \in \mathbb{N}^+$ and $q_i \in \mathbb{N}$ $(q_i < p_i)$.
  - Let γ denote the least common multiple of all p's.
  - The set of all $p's$ and $q's$ is called the *context* of the DGABN.
- At time t, node x_i will be updated by $x_i(t+1) = f_i(x(t))$ when the modulus of time t over p_i is equal to q_i (i.e., $t\%p_i = q_i$). If two or more nodes can be updated, they will be updated simultaneously.
- The evolution of the DGABN is specified by its context, and is time-dependent. Hence, the dynamics of the DGABN is not directly captured by an STG like SBNs, ABNs, or GABNs.

Extended State Transition Graph (ESTG)

Trinh Van Giang and Kunihiko Hiraishi: "On attractor detection and optimal control of deterministic generalized asynchronous random Boolean networks," *IEEE/ACM TCBB*, 2020, in press.

- We define an *extended state* of a DGABN, which includes a state of this DGABN and an embedded value that represents the scaled time t_{scaled} of time t (i.e., $t\%\gamma$) when reaching this state. Formally, $es \in \{0,1\}^n \times \{0,...,\gamma-1\}$ is an extended state, where es_i $(i = 1,...,n)$ denotes the value of internal node x_i and es_{n+1} denotes the value of the embedded time.

Extended State Transition Graph (ESTG)

- We define an *extended state* of a DGABN, which includes a state of this DGABN and an embedded value that represents the scaled time t_{scaled} of time t (i.e., $t\%\gamma$) when reaching this state. Formally, $es \in \{0,1\}^n \times \{0,...,\gamma-1\}$ is an extended state, where es_i $(i = 1,...,n)$ denotes the value of internal node x_i and es_{n+1} denotes the value of the embedded time.
- Then, the transition formula between two extended states is given as

$$T(es^j, es^{j+1}) := \left\{ es^{j+1}_{n+1} = (es^j_{n+1} + 1)\%\gamma \right\} \wedge$$

$$\bigwedge_{i=1}^{n} \left\{ \left[es^j_{n+1}\%p_i = q_i \wedge (es^{j+1}_i = f_i(es^j) \right] \vee \left[es^j_{n+1}\%p_i \neq q_i \wedge (es^{j+1}_i = es^j_i) \right] \right\}.$$

Extended State Transition Graph (ESTG)

- We define an *extended state* of a DGABN, which includes a state of this DGABN and an embedded value that represents the scaled time t_{scaled} of time t (i.e., $t\%\gamma$) when reaching this state. Formally, $es \in \{0,1\}^n \times \{0,...,\gamma-1\}$ is an extended state, where es_i $(i = 1,...,n)$ denotes the value of internal node x_i and es_{n+1} denotes the value of the embedded time.
- Then, the transition formula between two extended states is given as

$$T(es^j, es^{j+1}) := \left\{ es_{n+1}^{j+1} = (es_{n+1}^j + 1)\%\gamma \right\} \wedge$$

$$\bigwedge_{i=1}^{n} \left\{ \left[es_{n+1}^j \% p_i = q_i \wedge (es_i^{j+1} = f_i(es^j) \right] \vee \left[es_{n+1}^j \% p_i \neq q_i \wedge (es_i^{j+1} = es_i^j) \right] \right\}.$$

- From these definitions, the dynamics of a DGABN can be captured by an ESTG, a directed graph such that a node represents an extended state and an arc represents a transition between two extended states, respectively.

Extended State Transition Graph (ESTG)

- We define an *extended state* of a DGABN, which includes a state of this DGABN and an embedded value that represents the scaled time t_{scaled} of time t (i.e., $t\%\gamma$) when reaching this state. Formally,

The concept of an ESTG paves several results:
- Relations in dynamics between DGABNs and other conventional models.
- An SMT-based method and two SMT-based methods for attractor detection and optimal control of DGABNs, respectively.

$$T(es^j, es^{j+1}) := \left\{ es^{j+1}_{n+1} = (es^j_{n+1} + 1)\%\gamma \right\} \wedge$$

$$\bigwedge_{i=1}^{n} \left\{ \left[es^j_{n+1}\%p_i = q_i \wedge (es^{j+1}_i = f_i(es^j) \right] \vee \left[es^j_{n+1}\%p_i \neq q_i \wedge (es^{j+1}_i = es^j_i) \right] \right\}.$$

- From these definitions, the dynamics of a DGABN can be captured by an ESTG, a directed graph such that a node represents an extended state and an arc represents a transition between two extended states, respectively.

Extended State Transition Graph (ESTG)

- We define an *extended state* of a DGABN, which includes a state of this DGABN and an embedded value that represents the scaled time t_{scaled} of time t (i.e., $t\%\gamma$) when reaching this state. Formally,

Furthermore, the dynamics of a DA-PBN (the stochastic extension of DGABNs) can be captured by an ESTG where a probability is attached to an arc of the ESTG. With this underlying formulation, we propose three approaches for solving the optimal control problems of DA-PBNs.

$$T(es^j, es^{j+1}) := \left\{ es^{j+1}_{n+1} = (es^j_{n+1} + 1)\%\gamma \right\} \wedge$$

$$\bigwedge_{i=1}^{n} \left\{ \left[es^j_{n+1}\%p_i = q_i \wedge (es^{j+1}_i = f_i(es^j) \right] \vee \left[es^j_{n+1}\%p_i \neq q_i \wedge (es^{j+1}_i = es^j_i) \right] \right\}.$$

- From these definitions, the dynamics of a DGABN can be captured by an ESTG, a directed graph such that a node represents an extended state and an arc represents a transition between two extended states, respectively.

Contents

Future work

- This dissertation opens a number of research directions on BNs that could be pursued in the future.

Future work

- This dissertation opens a number of research directions on BNs that could be pursued in the future.
- First, the theoretical results obtained in this dissertation could be further explored to contribute more insights into the dynamics of BNs as well as pave potential ways for developing more efficient methods for attractor detection and optimal control of BNs.

Future work

- This dissertation opens a number of research directions on BNs that could be pursued in the future.
- First, the theoretical results obtained in this dissertation could be further explored to contribute more insights into the dynamics of BNs as well as pave potential ways for developing more efficient methods for attractor detection and optimal control of BNs.
- Second, the methods proposed in this dissertation could be further improved to handle larger networks (targeting genome-scale networks that can possess thousands of components [Mizera et al., 2018, Rozum et al., 2021]).
  Ongoing work (since the preliminary defense):
  - Trinh Van Giang and Kunihiko Hiraishi: "Computing attractors of large-scale asynchronous Boolean networks using minimal trap spaces," 2021, in preparation. Currently, this method can well handle networks of 3000+ nodes.

Future work (cont.)

- Third, both the theoretical and practical results of this dissertation could be extended to those for other less popular but more complex types of BNs such as random order asynchronous Boolean networks [Chaves et al., 2006] or generalized to those for more general models such as multi-valued networks [Luo and Wang, 2013] and hybrid models [Belta et al., 2001].
  Ongoing work:
  - Trinh Van Giang, Tatsuya Akutsu and Kunihiko Hiraishi: "On dynamics of random order asynchronous Boolean networks and an efficient FVS-based method for approximating their attractors," 2021, in preparation.
  - Extend **iFVS-ABN** for finding attractors of ABNs to that for finding attractors of asynchronous multi-valued networks [Gan and Albert, 2018]. The extension is not trivial. We have obtained some preliminary results since the preliminary defense.

Future work (cont.)

- Last, but not least, research on the dynamics of several special classes of BNs such as canalyzing and nested canalyzing BNs [Kauffman et al., 2004, Akutsu et al., 2011], AND-OR BNs [Melkman et al., 2010, Akutsu et al., 2012], conjunctive BNs [Weiss et al., 2018], as well as their attractor detection and optimal control problems, may be of interest. Because of their special structures, deeper theoretical results and more efficient methods may be obtained.

Journal papers

- Trinh Van Giang and Kunihiko Hiraishi: "On optimal control of deterministic asynchronous probabilistic Boolean networks," 2021, in preparation.
- 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.
- 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.
- 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.

Conference papers

- Trinh Van Giang and Kunihiko Hiraishi: "An improved method for finding attractors of large-scale asynchronous Boolean networks", in Proc. 18th *IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 1-9, 2021.
- 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)*, in Proc. *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1820-1826, 2020.
- Trinh Van Giang and Kunihiko Hiraishi: "Algorithms for finding attractors of generalized asynchronous random Boolean networks", in Proc. 12th *Asian Control Conference (ASCC)*, 67-72, 2019.

Other papers (not included in this dissertation)

- Trinh Van Giang, Tatsuya Akutsu and Kunihiko Hiraishi: "On dynamics of random order asynchronous Boolean networks and an efficient FVS-based method for approximating their attractors," 2021, in preparation.
- Trinh Van Giang and Kunihiko Hiraishi: "Computing attractors of large-scale asynchronous Boolean networks using minimal trap spaces," 2021, in preparation.

We believe that this dissertation will provide very useful tools for researchers in many fields as well as will be an important starting point for various potentially future work on Boolean networks research and beyond.

Thank you for your attention!

Appendix

Motivation

- Asynchronous Boolean networks are considered more suitable than synchronous ones in modeling biological systems [Saadatpour et al., 2010].
- ABNs and GABNs are conventional asynchronous models that are usually used in systems biology by the fact that precise information on time scales of components is usually missing [Schwab et al., 2020].
- Whereas many practical methods have been proposed for attractor detection in ABNs [Garg et al., 2008, Skodawessely and Klemm, 2011, Mizera et al., 2018], there is no practical method specifically designed for attractor detection in GABNs.
- In addition, it also lacks theoretical studies linking dynamics of GABNs and other models [Paulevé and Richard, 2012].

Algorithm **FR-BR-BDD-1**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
Compute all attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
$A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
while $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
    Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
    $FS \leftarrow FR^{\mathcal{G}}(att)$
    $BS \leftarrow BR^{\mathcal{G}}(att)$
    if $FS \subseteq BS$ **then**
        $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
        $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
    end if
end while
return $A^{\mathcal{G}}$

Algorithm **FR-BR-BDD-1**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
9: $FS \leftarrow FR^{\mathcal{G}}(att)$
10: $BS \leftarrow BR^{\mathcal{G}}(att)$
11: **if** $FS \subseteq BS$ **then**
12: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
13: $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
14: **end if**
15: **end while**
16: **return** $A^{\mathcal{G}}$

The GABN and its SBN counterpart have the same sets of singleton and type1 attractors (by Lemmas 3.3.1 and 3.3.2).

Algorithm **FR-BR-BDD-1**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
9: $FS \leftarrow FR^{\mathcal{G}}(att)$
10: $BS \leftarrow BR^{\mathcal{G}}(att)$
11: **if** $FS \subseteq BS$ **then**
12: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
13: $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
14: **end if**
15: **end while**
16: **return** $A^{\mathcal{G}}$

Check whether a type2 attractor of the SBN belongs to a complex attractor of the GABN (by Theorem 3.2.1). If yes, add the new attractor (FS) to the result set ($A^{\mathcal{G}}$).

Algorithm **FR-BR-BDD-1**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
9: $FS \leftarrow FR^{\mathcal{G}}(att)$
10: $BS \leftarrow BR^{\mathcal{G}}(att)$
11: **if** $FS \subseteq BS$ **then**
12: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
13: $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
14: **end if**
15: **end while**
16: **return** $A^{\mathcal{G}}$

Ignore the type2 SBN attractors that belong to the new GABN attractor.

Algorithm **FR-BR-BDD-2**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
Compute all attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
$A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
while $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
    Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
    $FS \leftarrow FR^{\mathcal{G}}(att)$
    $BS_{res} \leftarrow BR^{\mathcal{G}}_{res}(att, FS)$
    if $BS_{res} = FS$ **then**
        $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
        $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
    end if
end while
return $A^{\mathcal{G}}$

Algorithm **FR-BR-BDD-2**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor att from $A^{\mathcal{S}}_{type2}$
9: $FS \leftarrow FR^{\mathcal{G}}(att)$
10: $BS_{res} \leftarrow BR^{\mathcal{G}}_{res}(att, FS)$
11: **if** $BS_{res} = FS$ **then**
12: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FS$ {a new attractor}
13: $A^{\mathcal{S}}_{type2} \leftarrow A^{\mathcal{S}}_{type2} - FS$
14: **end if**
15: **end while**
16: **return** $A^{\mathcal{G}}$

This algorithm improves **FR-BR-BDD-1** by only calculating restricted backward reachable sets.

Algorithm **filtBDD**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
Compute all attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
$A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
$A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
while $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
    Randomly remove a type2 attractor $att^{\mathcal{S}}$ from $A^{\mathcal{S}}_{type2}$
    if $att^{\mathcal{S}}$ does not reach in $G(\mathcal{G})$ any attractor in $A^{\mathcal{S}}_{type2} \cup A^{\mathcal{G}}$ **then**
        $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FR^{\mathcal{G}}(att^{\mathcal{S}})$ {a new attractor}
    end if
end while
return $A^{\mathcal{G}}$

Algorithm **filtBDD**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor $att^{\mathcal{S}}$ from $A^{\mathcal{S}}_{type2}$
9: **if** $att^{\mathcal{S}}$ does not reach in $G(\mathcal{G})$ any attractor in $A^{\mathcal{S}}_{type2} \cup A^{\mathcal{G}}$ **then**
10: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FR^{\mathcal{G}}(att^{\mathcal{S}})$ {a new attractor}
11: **end if**
12: **end while**
13: **return** $A^{\mathcal{G}}$

This algorithm calculates neither total nor restricted backward reachable sets (which are usually very large) but filters out the set of type2 attractors of the SBN counterpart by checking the reachability in the GABN.

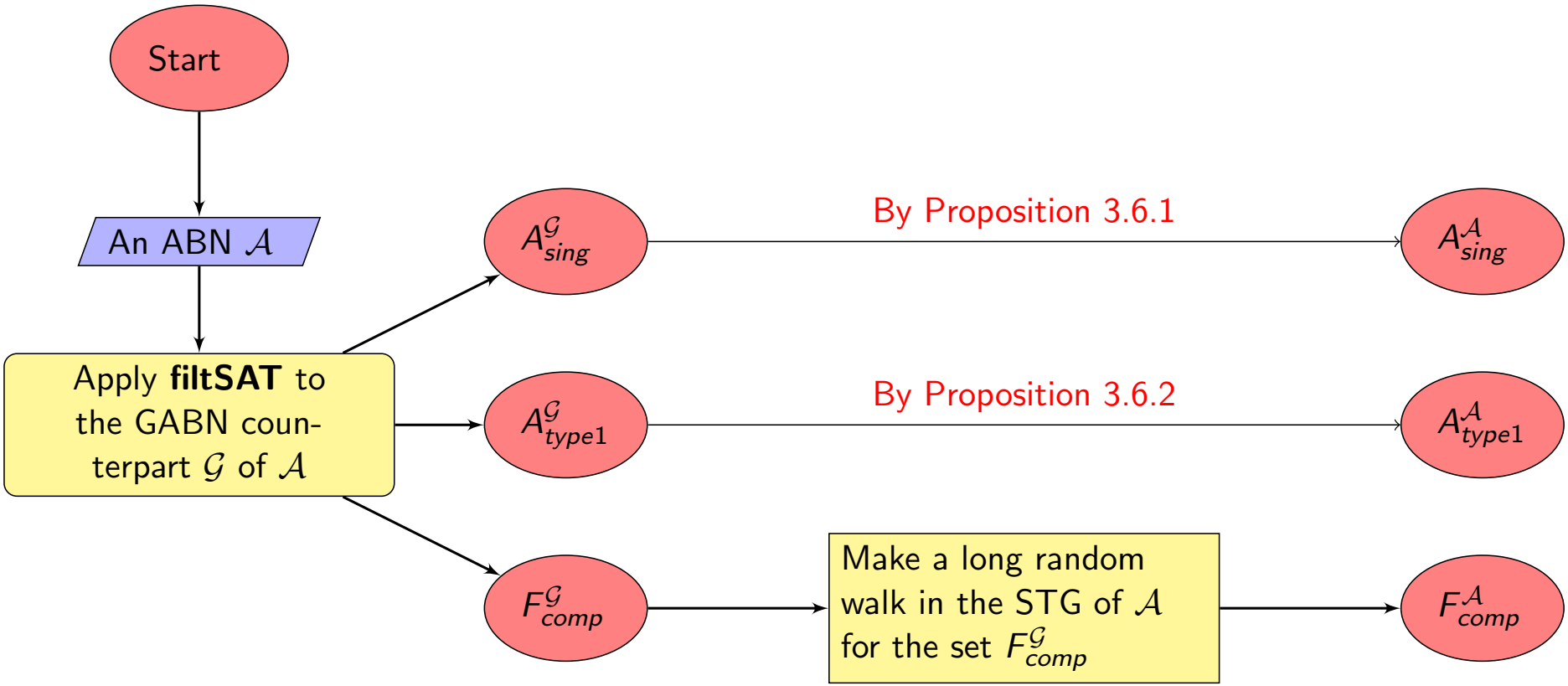
Algorithm **filtBDD**

Input: A GABN $\mathcal{G}$.
Output: The set of attractors of $\mathcal{G}$.

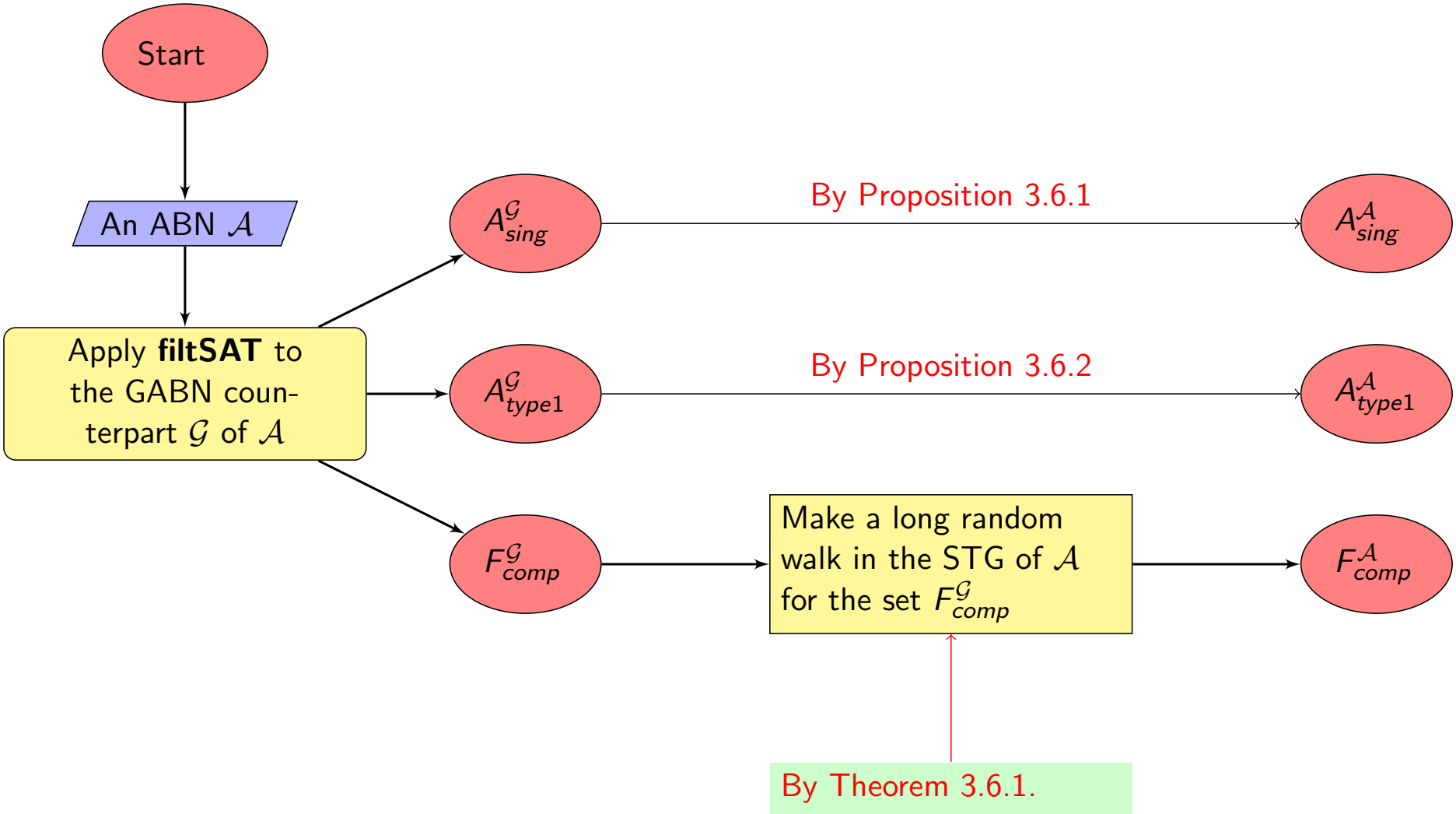
1: Let $\mathcal{S}$ be the SBN counterpart of $\mathcal{G}$
2: Compute all attractors of $\mathcal{S}$
3: $A^{\mathcal{S}}_{sing} \leftarrow$ the set of singleton attractors of $\mathcal{S}$
4: $A^{\mathcal{S}}_{type1} \leftarrow$ the set of type1 attractors of $\mathcal{S}$
5: $A^{\mathcal{S}}_{type2} \leftarrow$ the set of type2 attractors of $\mathcal{S}$
6: $A^{\mathcal{G}} \leftarrow A^{\mathcal{S}}_{sing} \cup A^{\mathcal{S}}_{type1}$
7: **while** $A^{\mathcal{S}}_{type2} \neq \emptyset$ **do**
8: Randomly remove a type2 attractor $att^{\mathcal{S}}$ from $A^{\mathcal{S}}_{type2}$
9: **if** $att^{\mathcal{S}}$ does not reach in $G(\mathcal{G})$ any attractor in $A^{\mathcal{S}}_{type2} \cup A^{\mathcal{G}}$ **then**
10: $A^{\mathcal{G}} \leftarrow A^{\mathcal{G}} + FR^{\mathcal{G}}(att^{\mathcal{S}})$ {a new attractor}
11: **end if**
12: **end while**
13: **return** $A^{\mathcal{G}}$

The reachability is checked by on-the-fly calculating the forward reachable set.

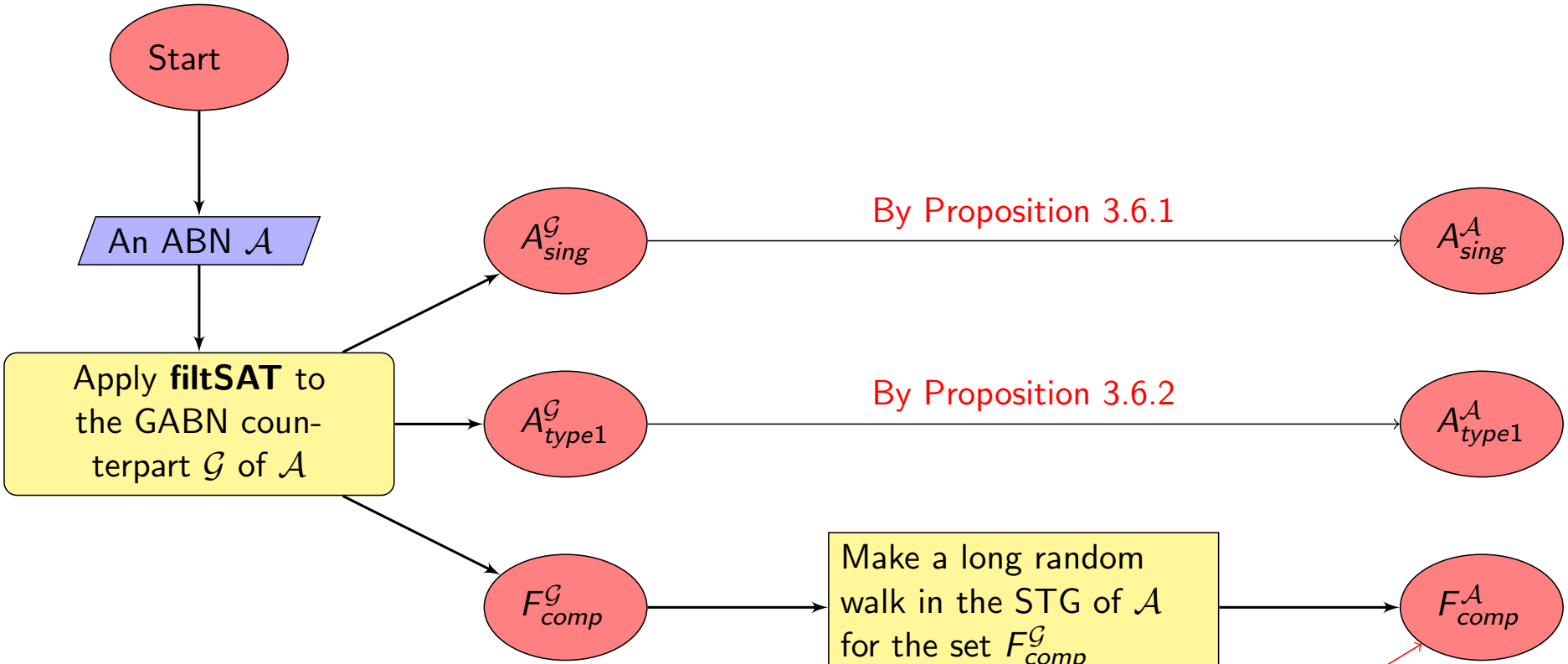
The **ApproABN** method



The **ApproABN** method



The **ApproABN** method



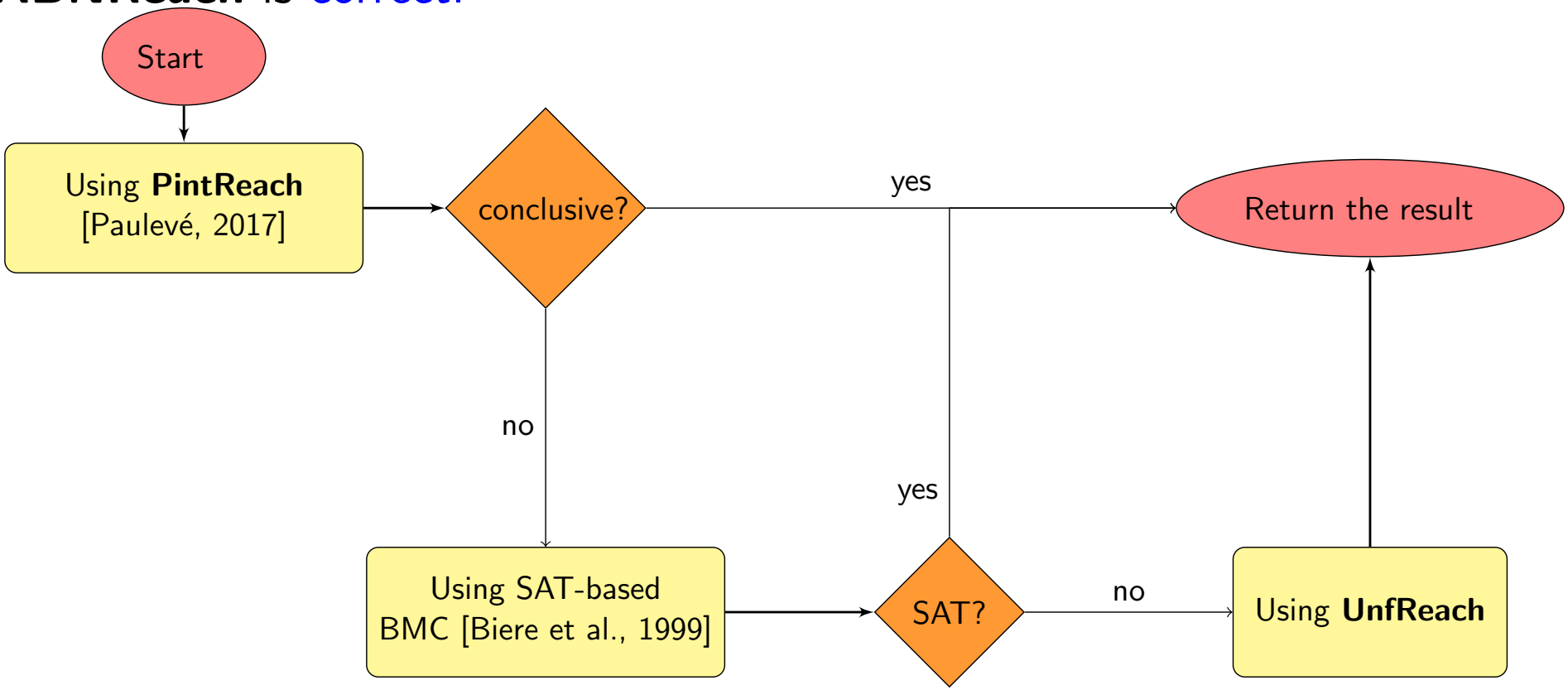
In the dissertation, we have analyzed in detail some cases in which the result of **ApproABN** is incorrect. We have also provided some reasons convincing the accuracy of **ApproABN** that will be confirmed by the conducted experiments on real biological networks.

Motivation

- ABNs are considered more suitable [Saadatpour et al., 2010] for representing various time scales as well as dealing with the lacking of knowledge on time scales.
- However, whereas many efficient algorithms and tools have been developed for attractor detection in SBNs, few methods have been proposed for attractor detection in ABNs due to the high complexity of the 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.
- A more detailed literature review on computational methods for attractor detection in BNs is provided in Section 4.2 of the dissertation.
- Therefore, it is important and interesting to develop efficient methods that can handle larger ABNs.

First improvement

In general, **ABNReach** is a reasonable combination of multiple previous techniques for checking the reachability in ABNs. The result of **ABNReach** is correct.



SMT-based method for attractor detection in DGABNs

- Dubrova and Teslenko proposed an efficient SAT-based method for finding attractors of an SBN [Dubrova and Teslenko, 2011].
- The ESTG of a DGABN is deterministic like the STG of a SBN, opening a chance to extend the SAT-based method for SBNs to that for DGABNs.
- We here propose an SMT-based method (called **DA-SMT-Att**) for exactly finding attractors of a DGABN. This method is a non-trivial extension of [Dubrova and Teslenko, 2011].
- We also provide a formal proof for the correctness of **DA-SMT-Att**. Note that such a proof in the case of SBNs is lacking in [Dubrova and Teslenko, 2011].

Applications

- We applied **DA-SMT-Att** to two real biological networks and compare the obtained results to the previous insights into these networks found in the literature.
- We also used **DA-SMT-Att** to verify several previous insights into the dynamics of random Boolean networks (i.e., N-K models) presented in [Gershenson, 2002, Gershenson et al., 2003].

Boolean networks with control nodes

BNs with control nodes

A BN with control nodes is defined as a triple (X, F, U), where $X = \{x_1, ..., x_n\}$ $(n \geq 1)$ is the set of internal nodes, $F = \{f_1, ..., f_n\}$ is the set of Boolean functions, and $U = \{u_1, ..., u_m\}$ $(m \geq 0)$ is the set of external (control) nodes. Each node x_i is identified as a Boolean variable, and is associated with a Boolean function $f_i : \mathbb{B}^n \times \mathbb{B}^m \to \mathbb{B}$. Also, each node u_i identified as a Boolean variable. $x_i(t) \in \mathbb{B}$ and $u_i(t) \in \mathbb{B}$ denote the state of internal node x_i and the state of external node u_i at time t, respectively. $x(t) = (x_1(t), ..., x_n(t))^\top$ and $u(t) = (u_1(t), ..., u_m(t))^\top$ denote the state and the control input of the BN at time t, respectively.

Boolean networks with control nodes

BNs with control nodes

A BN with control nodes is defined as a triple (X, F, U), where $X = \{x_1, ..., x_n\}$

Internal node x_i updates its state by $x_i(t+1) = f_i(x(t), u(t))$, whereas a control node can receive an arbitrary Boolean value at each time step.

Each node x_i is identified as a Boolean variable, and is associated with a Boolean function $f_i : \mathbb{B}^n \times \mathbb{B}^m \to \mathbb{B}$. Also, each node u_i identified as a Boolean variable. $x_i(t) \in \mathbb{B}$ and $u_i(t) \in \mathbb{B}$ denote the state of internal node x_i and the state of external node u_i at time t, respectively. $x(t) = (x_1(t), ..., x_n(t))^\top$ and $u(t) = (u_1(t), ..., u_m(t))^\top$ denote the state and the control input of the BN at time t, respectively.

Boolean networks with control nodes

BNs with control nodes

A BN with control nodes is defined as a triple (X, F, U), where $X = \{x_1, ..., x_n\}$ $(n \geq 1)$ is the set of internal nodes, $F = \{f_1, ..., f_n\}$ is the set of Boolean functions, and $U = \{u_1, ..., u_m\}$ $(m \geq 0)$ is the set of external (control) nodes. Each node x_i is identified as a Boolean variable, and is associated with a Boolean function $f_i : \mathbb{B}^n \times \mathbb{B}^m \to \mathbb{B}$. Also, each node u_i identified as a Boolean variable. $x_i(t) \in \mathbb{B}$ and $u_i(t) \in \mathbb{B}$ denote the state of internal node x_i and the state of external node u_i at time t, respectively. $x(t) = (x_1(t), ..., x_n(t))^\top$ and $u(t) = (u_1(t), ..., u_m(t))^\top$ denote the state and the control input of the BN at time t, respectively.

DGABNs with control nodes

A DGABN with control nodes is a BN with control nodes such that each internal node x_i is associated with two parameters, $p_i \in \mathbb{N}^+$ and $q_i \in \mathbb{N}, q_i < p_i$. Internal node x_i can be updated at time t if $t\%p_i = q_i$. If multiple internal nodes can be updated, then all of them are updated simultaneously.

The **DA-SMT-Con-TS** method

- Let es^t be the corresponding extended state of state $x(t)$, where $es_i^t = x_i(t), i \in \{1, ..., n\}$ and $es_{n+1}^t = t\%\gamma$.
- First, we encode an M-length path from es^0 (i.e., starts with x^{ini} at time $t = 0$) to es^M (i.e., ends with x^{des} at time $t = M$) in the ESTG of the DGABN $\mathcal{D}$ as an SMT formula P, which is based on the transition formula between two extended states of the DGABN.
- We then solve P under minimizing the cost function C in Z3 (see [Bjørner et al., 2015] for optimization in Z3).
- If **SAT**(P), then a control sequence and an optimum cost, which can be easily obtained from the satisfying assignments of the corresponding SMT variables, are released. Otherwise, "there are no control policies" is released.

The **DA-SMT-Con-NTS** method

- Based on the method for the time-sensitive mode, we modify the SMT formula P to represent an M-length path from es^0 to es^M in the ESTG of the DGABN such that along with this path, once we reach an extended state satisfying the following condition, all next extended states of the path will be equal to this extended state (i.e., there are no updates).
- The condition means that the values of internal nodes of the extended state are same as those of the desired state x^{des}.
- Note that we add a new variable r^j to indicate either the updating case ($r^j = 1$) or the non-updating case ($r^j = 0$). This helps us to easily obtain the real control sequence.

Reasons for studying optimal control of DA-PBNs

- DA-PBNs are general models: a DGABN is a special DA-PBN and an SPBN is a special DA-PBN [Shmulevich and Dougherty, 2010]. Clearly, optimal control of DA-PBNs is harder than that of DGABNs or SPBNs. In addition, developed methods for optimal control of DA-PBNs can be directly applied for those of DGABNs or SPBNs.
- In the context of systems biology, DA-PBNs seem to be more suitable to model GRNs, since a DA-PBN comprises all the synchronous, asynchronous, and probabilistic
  natures [Faryabi et al., 2008a, Shmulevich and Dougherty, 2010].
- It lacks efficient methods for optimal control of DA-PBNs.
  - To our best knowledge, [Faryabi et al., 2008a] is the sole method for optimal control of DA-PBNs. However, this method is inefficient because it requires to build transition probability matrices of size $(\gamma 2^n) \times (\gamma 2^n)$, where γ is a given constant.

Optimal control problems of DA-PBNs

Problem OptC-1

Given a DA-PBN $\mathcal{DP}$. Suppose that an initial state of $\mathcal{DP}$ is x^{ini} and a desired state of $\mathcal{DP}$ is x^{des}. Find a control sequence $\langle u(0), ..., u(M-1)\rangle$ that maximizes the reachability probability from x^{ini} to x^{des} at time M.

Problem OptC-2

Given a DA-PBN $\mathcal{DP}$. Suppose that an initial state of $\mathcal{DP}$ is x^{ini}, the unsafe state of $\mathcal{DP}$ is x^{des}, and $\varepsilon \in [0, 1]$ is given. Find a control sequence $\langle u(0), ..., u(M-1)\rangle$ that minimizes the reachability probability from x^{ini} to x^{des} at time M. If the minimum probability is at most ε, then $\mathcal{DP}$ is said to be safe.

Problems OptC-1 and OptC-2 are generalized from the reachability problem and the safety problem of SPBNs [Kobayashi and Hiraishi, 2012b], respectively.

Optimal control problems of DA-PBNs (cont.)

Problem OptC-3

Given a DA-PBN $\mathcal{DP}$. Suppose that the initial state of $\mathcal{DP}$ is x^{ini} and the control time M is given. Find a control sequence $\langle u(0), ..., u(M-1)\rangle$ that minimizes the expected cost

$$J = E\left[\sum_{k=0}^{M-1}\{\mathcal{Q}x(k) + \mathcal{R}x(k)\} + \mathcal{Q}_f x(M) \,\middle|\, x(0) = x^{ini}\right],$$

where $\mathcal{Q}, \mathcal{Q}_f \in \mathbb{R}^{1\times n}, \mathcal{R} \in \mathbb{R}^{1\times m}$ are weighting vectors.

Problem OptC-3 is generalized from the expected cost problem of SPBNs [Kobayashi and Hiraishi, 2012a].

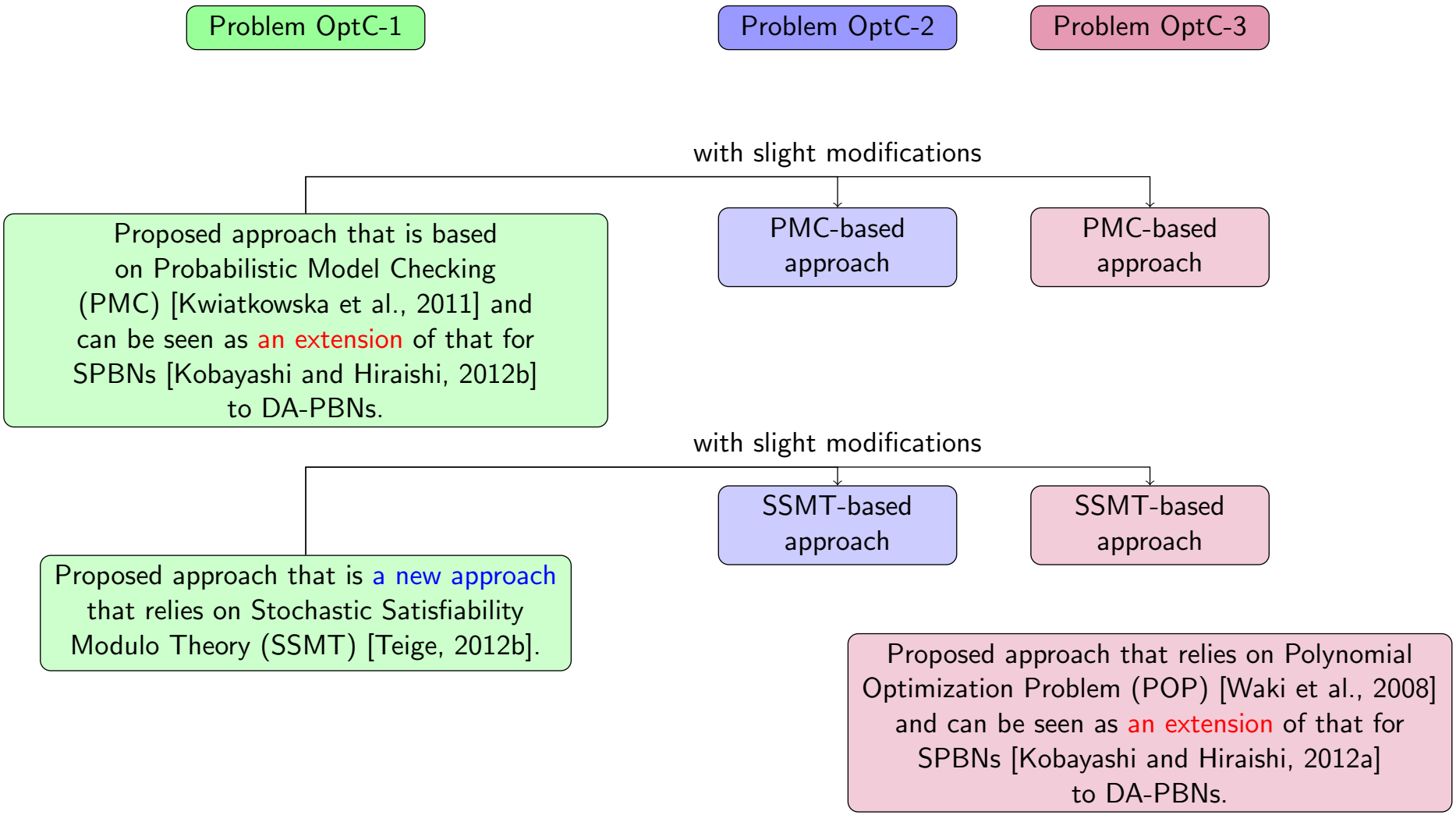
Each of these problems is suitable for a specific aim of control [Kobayashi and Hiraishi, 2012b, Kobayashi and Hiraishi, 2012a].

Important remarks

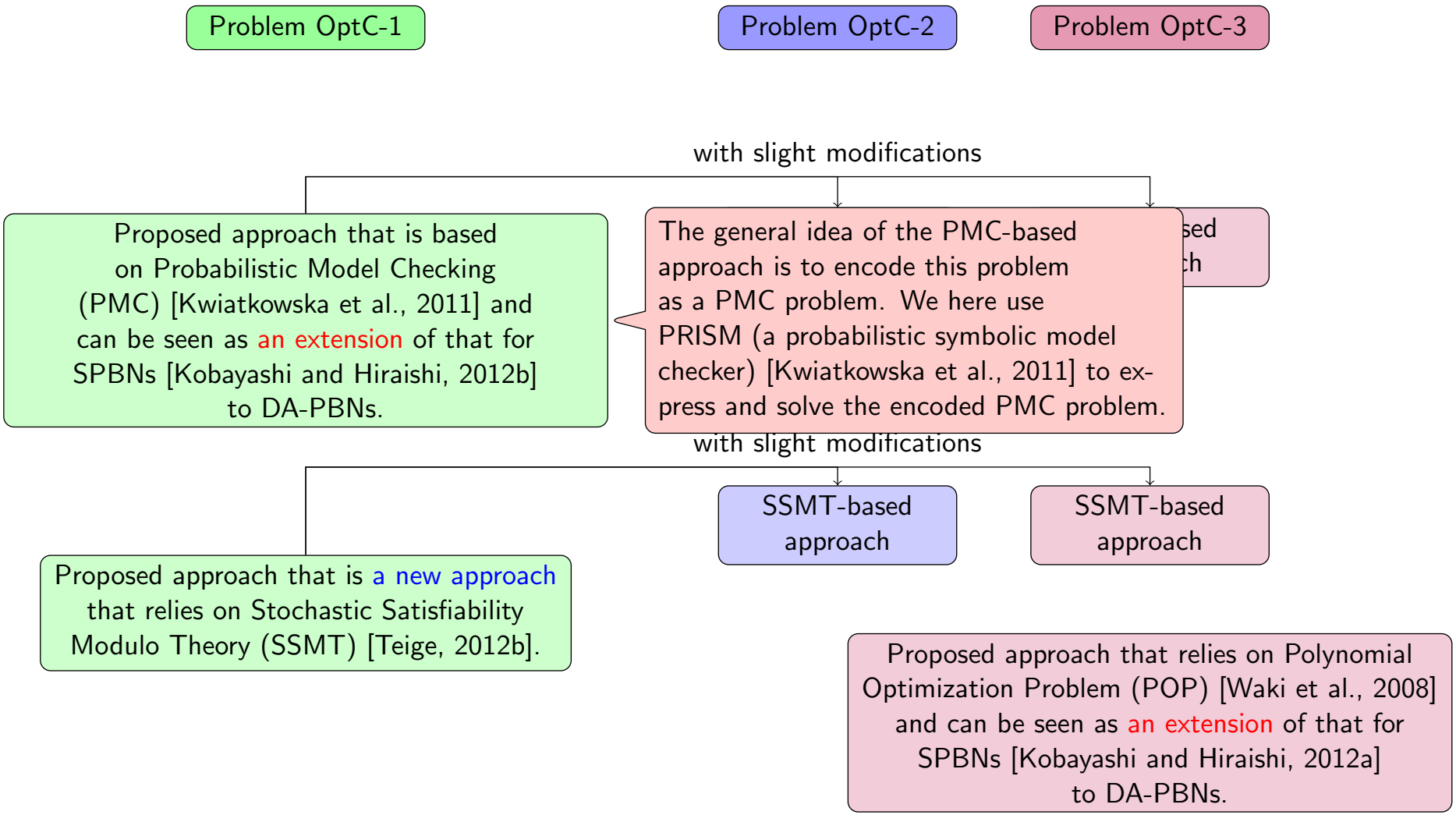
$NP \subseteq \Sigma_2^p \subseteq PH \subseteq NP^{PP} \subseteq PSPACE$ [Littman et al., 1998]

- The Σ_2^p-hardness of the counterpart of Problem OptC-3 for SPBNs is proved in [Chen et al., 2013]. By using similar reductions, we can easily obtain the Σ_2^p-hardness of the counterparts of Problems OptC-1 and OptC-2 for SPBNs. Thus, the optimal control problems of DA-PBNs seem harder than the corresponding optimal control problems of SPBNs. It is reasonable because an SPBN is a special DA-PBN.
- It is not plausible that efficient SAT-based or ILP-based algorithms exist even in the case of SPBNs because SAT and ILP are NP-complete.
- All the three problems are NP^{PP}-hard. Since all the three problems in PSPACE, their complexity is between NP^{PP}-hard and PSPACE-complete. Hence, these problems are hard to solve in general. Moreover, in a crude sense, NP^{PP} is very close to PSPACE. Therefore, to solve these problems, we may have to encode them as PSPACE-complete problems.

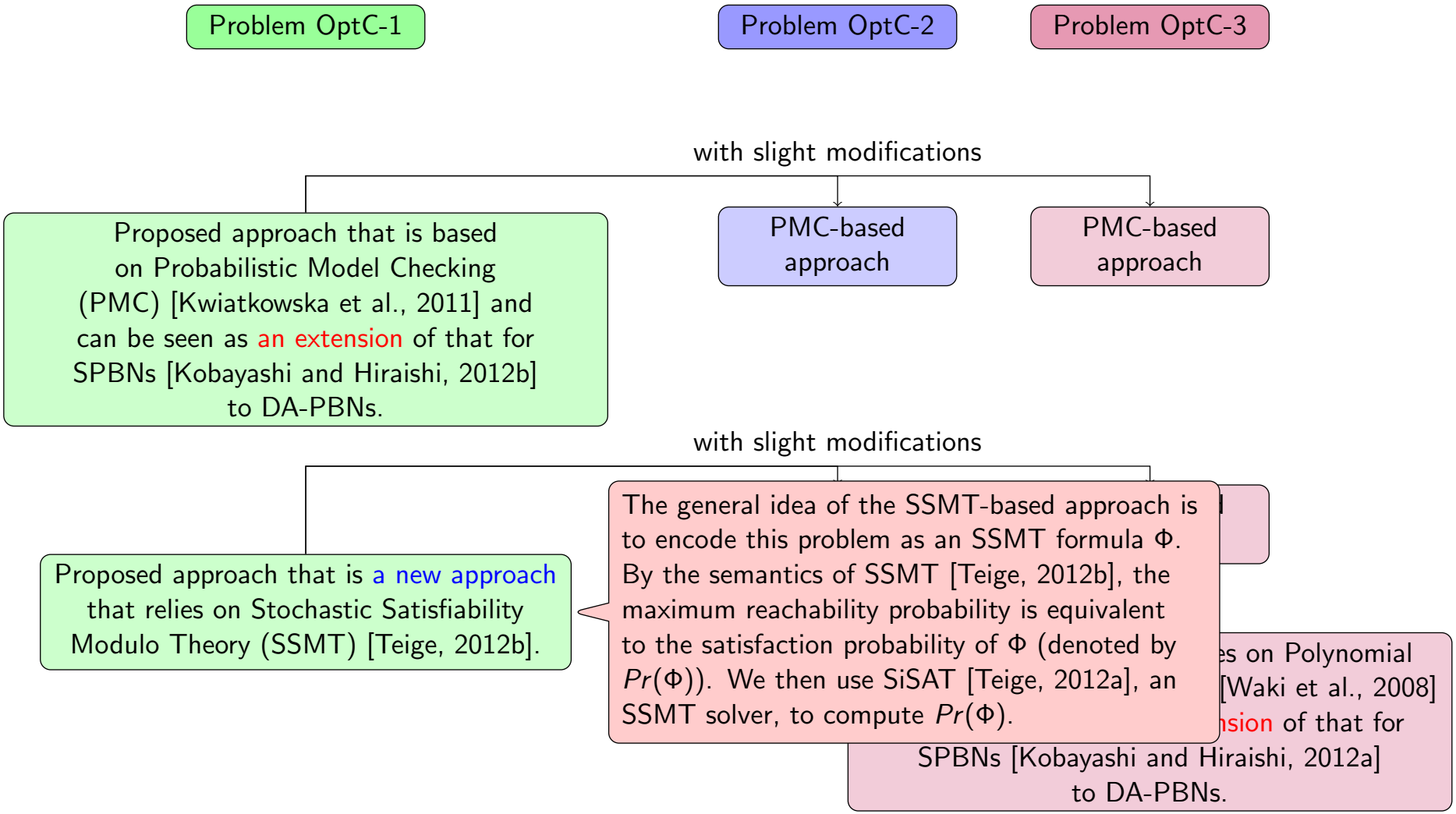
Proposed solution approaches



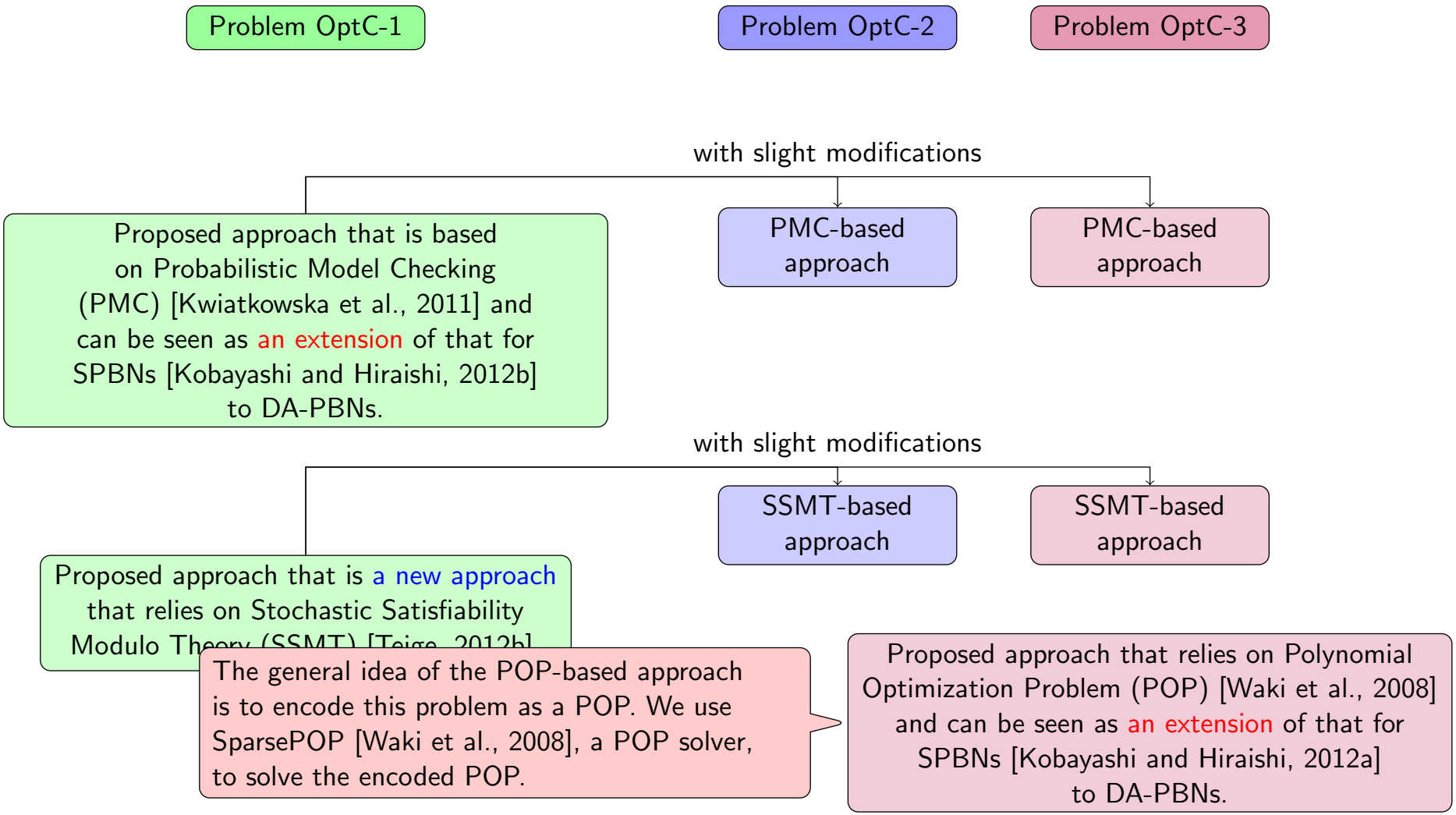
Proposed solution approaches



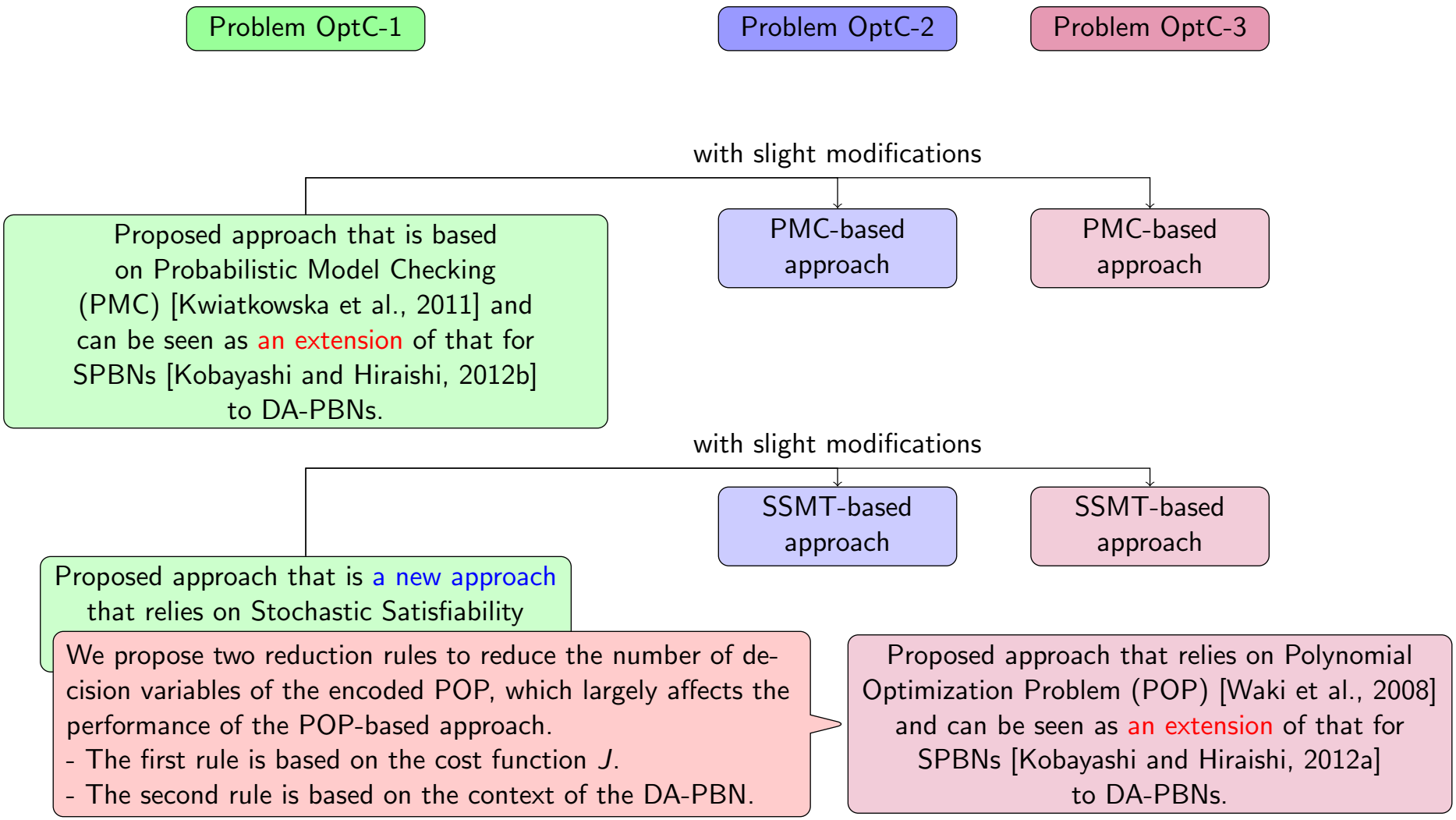
Proposed solution approaches



Proposed solution approaches



Proposed solution approaches



Important remarks

- We can consider other forms of the cost function in Problem OptC-3. For example, the weighting vectors can be replaced by functions of x and u [Wei et al., 2017, Akutsu, 2018]. By the expressive power of PRISM, SiSAT, or SparsePOP, we can easily modify the proposed approaches to handle a new form of the cost function.
- We can consider adding hard constraints (i.e., adding an upper bound H for the number of controls that can be applied to the network) into the three problems. The introduction of hard constraints is important for medical applications because the number of treatments such as radiation and chemo-therapy is usually limited [Chen et al., 2013].
  - Some little modifications to the SSMT-based and POP-based approaches.
  - But it is difficult to modify the PRISM-based approach to handle hard constraints.

Computational experiments

- In addition to a case study on a realistic network, computational experiments were performed to evaluate the performance of the three proposed approaches.
- Based on the experimental results, we present experimental analysis along with theoretical analysis on the effects of some factors (e.g., the number of nodes, the target time step) on the performance of the proposed approaches.
  $\Rightarrow$ A significant contribution because it lacks analysis in both theoretical and experimental aspects on how the running time of the proposed approaches depends on some factors in control settings.
- We also present a comprehensive comparison among these approaches.
  $\Rightarrow$ A significant contribution because in all the previous work, the proposed approaches for optimal control of SPBNs or DA-PBNs were only compared to the dynamic programming-based approach; it lacks a comparison among different proposed approaches.

Summary of the experimental results

- The experimental results confirm the advantages and disadvantages of each proposed approach as well as suggest that the proposed approaches can complement each other.
- For the PMC-based approach, the running time is linear or polynomial in M. However, it may meet OOM or takes extremely long time when the number of reachable states of the PRISM model is too large. Moreover, the number of reachable states is exponential in n and m.
- For the SSMT-based approach, the running time is exponential in M. However, it can handle the case of large n when the number of quantified variables is small.
- The POP-based approach with reduction gives the best performance overall, but the running time is still exponential in M.

Conclusions

- In theory, we have introduced a number of new theoretical results that contribute to the understanding of the dynamics of BNs.
- In practice, we have developed several efficient algorithms and methods for attractor detection and optimal control of different typical types of BNs.
  - The theoretical foundations for these algorithms and methods are the new theoretical results obtained in this research.
  - These algorithms and methods outperform the previous ones and can handle large-scale networks. Notably, **iFVS-ABN** can handle large ABNs with up to 1000 nodes.

Conclusions

- Finally, although systems biology has served as the main motivation for our research, applications of this dissertation are by no means limited to biological systems. Since we consider general BNs (i.e., there is no restriction in Boolean functions) as well as different types of BNs (GABNs, ABNs, DGABNs, DA-PBNs), the results (theoretical results and computational methods) introduced in this dissertation can be applied to a wide range of other systems.
  - Water quality networks [Gao et al., 2017]
  - Multivariate systems [Yang et al., 2021]
  - Multi-agent systems [Kochemazov and Semenov, 2014]
  - Social networks [Green et al., 2007]
  - Smart grids [Rivera-Torres and Santiago, 2020]
  - ...

References I

Akutsu, T. (2018).
Algorithms for analysis, inference, and control of Boolean networks.
World Scientific.

Akutsu, T., Kosub, S., Melkman, A. A., and Tamura, T. (2012).
Finding a periodic attractor of a Boolean network.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 9(5):1410–1421.

Akutsu, T., Melkman, A. A., Tamura, T., and Yamamoto, M. (2011).
Determining a singleton attractor of a Boolean network with nested canalyzing functions.
Journal of Computational Biology, 18(10):1275–1290.

Albert, R. and Thakar, J. (2014).
Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions.
Wiley Interdisciplinary Reviews: Systems Biology and Medicine, 6(5):353–369.

Béal, J., Pantolini, L., Noël, V., Barillot, E., and Calzone, L. (2021).
Personalized logical models to investigate cancer response to BRAF treatments in melanomas and colorectal cancers.
PLoS Computational Biology, 17(1):e1007900.

References II

Belta, C., Schug, J., Dang, T., Kumar, V., Pappas, G. J., Rubin, H., and Dunlap, P. (2001).
Stability and reachability analysis of a hybrid model of luminescence in the marine bacterium Vibrio fischeri.
In *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No. 01CH37228)*, volume 1, pages 869–874. IEEE.

Biane, C. and Delaplace, F. (2018).
Causal reasoning on Boolean control networks based on abduction: theory and application to cancer drug discovery.
IEEE/ACM Ttransactions on Computational Biology and Bioinformatics, 16(5):1574–1585.

Biere, A., Cimatti, A., Clarke, E. M., Fujita, M., and Zhu, Y. (1999).
Symbolic model checking using SAT procedures instead of BDDs.
In *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*, pages 317–320. IEEE.

Bjørner, N., Phan, A.-D., and Fleckenstein, L. (2015).
νZ - an optimizing SMT solver.
In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 194–199. Springer.

References III

Bornholdt, S. (2008).
Boolean network models of cellular regulation: prospects and limitations.
Journal of the Royal Society Interface, 5(suppl_1):S85–S94.

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.

Chaves, M., Sontag, E. D., and Albert, R. (2006).
Methods of robustness analysis for Boolean models of gene control networks.
IEE Proceedings-Systems Biology, 153(4):154–167.

Chen, X., Akutsu, T., Tamura, T., and Ching, W.-K. (2013).
Finding optimal control policy in probabilistic Boolean networks with hard constraints by using integer programming and dynamic programming.
International Journal of Data Mining and Bioinformatics, 7(3):1–22.

Cheng, D. and Qi, H. (2009).
Controllability and observability of Boolean control networks.
Automatica, 45(7):1659–1667.

References IV

Datta, A., Choudhary, A., Bittner, M. L., and Dougherty, E. R. (2003).
External control in Markovian genetic regulatory networks.
Machine Learning, 52(1):169–191.

Dong, Z. (2017).
Boolean network-based sensor selection with application to the fault diagnosis of a nuclear plant.
Energies, 10(12):2125.

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.

Faryabi, B., Chamberland, J.-F., Vahedi, G., Datta, A., and Dougherty, E. R. (2008a).
Optimal intervention in asynchronous genetic regulatory networks.
IEEE Journal of Selected Topics in Signal Processing, 2(3):412–423.

Faryabi, B., Chamberland, J.-F., Vahedi, G., Datta, A., and Dougherty, E. R. (2008b).
Optimal intervention in semi-Markov-based asynchronous genetic regulatory networks.
In *American Control Conference*, pages 1388–1393. IEEE.

References V

Fornasini, E. and Valcher, M. E. (2015).
Fault detection analysis of Boolean control networks.
IEEE Transactions on Automatic Control, 60(10):2734–2739.

Gabriel, G. and Stepney, S. (2018).
Dynamical music with musical Boolean networks.
In *International Conference on Computational Intelligence in Music, Sound, Art and Design*, pages 18–33. Springer.

Gadouleau, M., Richard, A., and Fanchon, E. (2016).
Reduction and fixed points of Boolean networks and linear network coding solvability.
IEEE Transactions on Information Theory, 62(5):2504–2519.

Gan, X. and Albert, R. (2018).
General method to find the attractors of discrete dynamic models of biological systems.
Physical Review E, 97(4).

Gao, Z., Chen, X., and Başar, T. (2017).
Controllability of conjunctive Boolean networks with application to gene regulation.
IEEE Transactions on Control of Network Systems, 5(2):770–781.

References VI

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.

Gates, A. J., Correia, R. B., Wang, X., and Rocha, L. M. (2021).
The effective graph reveals redundancy, canalization, and control pathways in biochemical regulation and signaling.
Proceedings of the National Academy of Sciences, 118(12):e2022598118.

Gershenson, C. (2002).
Classification of random Boolean networks.
In *Proceedings of the Eighth International Conference on Artificial Life*, pages 1–8. MIT Press.

Gershenson, C. (2004).
Introduction to random Boolean networks.
In *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, page 160–173. MIT Press.

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

References VII


Green, D. G., Leishman, T. G., and Sadedin, S. (2007).
The emergence of social consensus in Boolean networks.
In *2007 IEEE Symposium on Artificial Life*, pages 402–408.


Greil, F., Drossel, B., and Sattler, J. (2007).
Critical kauffman networks under deterministic asynchronous update.
New Journal of Physics, 9(10):373.


Ibrahim, M. A., Abdelrahman, A. H., Mohamed, T. A., Atia, M. A., Al-Hammady, M. A., Abdeljawaad, K. A., Elkady, E. M., Moustafa, M. F., Alrumaihi, F., Allemailem, K. S., et al. (2021).
In silico mining of terpenes from red-sea invertebrates for SARS-CoV-2 main protease (Mpro) inhibitors.
Molecules, 26(7):2082.


Karlebach, G. and Shamir, R. (2008).
Modelling and analysis of gene regulatory networks.
Nature Reviews Molecular Cell Biology, 9(10):770–780.


Kauffman, S., Peterson, C., Samuelsson, B., and Troein, C. (2004).
Genetic networks with canalyzing Boolean rules are always stable.
Proceedings of the National Academy of Sciences, 101(49):17102–17107.

References VIII

Kauffman, S. A. (1969).
Metabolic stability and epigenesis in randomly constructed genetic nets.
Journal of Theoretical Biology, 22(3):437–467.

Kobayashi, K. and Hiraishi, K. (2012a).
Optimal control of probabilistic Boolean networks using polynomial optimization.
IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 95(9):1512–1517.

Kobayashi, K. and Hiraishi, K. (2012b).
Symbolic approach to verification and control of deterministic/probabilistic Boolean networks.
IET Systems Biology, 6(6):215–222.

Kochemazov, S. and Semenov, A. (2014).
Using synchronous Boolean networks to model several phenomena of collective behavior.
PLoS ONE, 9(12):e115156.

Kwiatkowska, M., Norman, G., and Parker, D. (2011).
PRISM 4.0: Verification of probabilistic real-time systems.
In *International Conference on Computer Aided Verification*, pages 585–591. Springer.

References IX

Li, Y. and Li, H. (2021).
Controllability and stabilization of periodic switched Boolean control networks with application to asynchronous updating.
Nonlinear Analysis: Hybrid Systems, 41:101054.

Littman, M. L., Goldsmith, J., and Mundhenk, M. (1998).
The computational complexity of probabilistic planning.
Journal of Artificial Intelligence Research, 9:1–36.

Luo, C. and Wang, X. (2013).
Algebraic representation of asynchronous multiple-valued networks and its dynamics.
IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10(4):927–938.

Melkman, A. A., Tamura, T., and Akutsu, T. (2010).
Determining a singleton attractor of an AND/OR Boolean network in $O(1.587^n)$ time.
Information Processing Letters, 110(14-15):565–569.

Milano, M. and Roli, A. (1999).
Solving the satisfiability problem through Boolean networks.
In *Congress of the Italian Association for Artificial Intelligence*, pages 72–83. Springer.

References X

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.

Noual, M. (2011).
General iteration graphs and Boolean automata circuits.
arXiv preprint arXiv:1104.4044.

Oyeyemi, O. J., Davies, O., Robertson, D. L., and Schwartz, J.-M. (2014).
A logical model of HIV-1 interactions with the T-cell activation signalling pathway.
Bioinformatics, 31(7):1075–1083.

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.

Paulevé, L. and Richard, A. (2012).
Static analysis of Boolean networks based on interaction graphs: A survey.
Electronic Notes in Theoretical Computer Science, 284:93–104.

References XI

Putnins, M. and Androulakis, I. P. (2019).
Boolean modeling in quantitative systems pharmacology: Challenges and opportunities.
Critical Reviews™ in Biomedical Engineering, 47(6).

Rivera-Torres, P. J. and Santiago, O. L. (2020).
Fault detection and isolation in smart grid devices using probabilistic Boolean networks.
In *Computational Intelligence in Emerging Technologies for Engineering Applications*, pages 165–185. Springer.

Roli, A. and Braccini, M. (2018).
Attractor landscape: A bridge between robotics and synthetic biology.
Cell Differentiation, 4:5.

Rozum, J. C., Zañudo, J. G. T., Gan, X., Deritei, D., and Albert, R. (2021).
Parity and time reversal elucidate both decision-making in empirical models and attractor scaling in critical Boolean networks.
Science Advances, 7(29):eabf8124.

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.

References XII

Schwab, J. D., Kühlwein, S. D., Ikonomi, N., Kühl, M., and Kestler, H. A. (2020).
Concepts in Boolean network modeling: What do they all mean?
Computational and Structural Biotechnology Journal, 18:571–582.

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

Sherekar, S. and Viswanathan, G. A. (2021).
Boolean dynamic modeling of cancer signaling networks: Prognosis, progression, and therapeutics.
Computational and Systems Oncology, 1(2):e1017.

Shmulevich, I. and Dougherty, E. R. (2010).
Probabilistic Boolean networks: the modeling and control of gene regulatory networks.
SIAM.

Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002).
Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks.
Bioinformatics, 18(2):261–274.

References XIII


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


Teige, T. (2012a).
Quick start guide and tutorial.


Teige, T. (2012b).
Stochastic satisfiability modulo theories: a symbolic technique for the analysis of probabilistic hybrid systems.
PhD thesis, Universität Oldenburg.


Torres, P. J. R., Mercado, E. S., and Rifón, L. A. (2018).
Probabilistic Boolean network modeling of an industrial machine.
Journal of Intelligent Manufacturing, 29(4):875–890.


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.

References XIV

Waidyarathne, P. and Samarasinghe, S. (2018).
Boolean calcium signalling model predicts calcium role in acceleration and stability of abscisic acid-mediated stomatal closure.
Scientific Reports, 8(1):1–16.

Waki, H., Kim, S., Kojima, M., Muramatsu, M., and Sugimoto, H. (2008).
Algorithm 883: SparsePOP—a sparse semidefinite programming relaxation of polynomial optimization problems.
ACM Transactions on Mathematical Software (TOMS), 35(2):1–13.

Wang, X. and Gao, S. (2020).
Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network.
Information Sciences, 539:195–214.

Wei, O., Guo, Z., Niu, Y., and Liao, W. (2017).
Model checking optimal finite-horizon control for probabilistic gene regulatory networks.
BMC Systems Biology, 11(6):75–88.

Weiss, E., Margaliot, M., and Even, G. (2018).
Minimal controllability of conjunctive Boolean networks is NP-complete.
Automatica, 92:56–62.

References XV


Yang, X., Ram, N., Molenaar, P. C., and Cole, P. M. (2021).
Describing and controlling multivariate nonlinear dynamics: A Boolean network approach.
Multivariate Behavioral Research, pages 1–30.


Zhong, J., Ho, D. W., and Lu, J. (2019).
A new framework for pinning control of Boolean networks.
arXiv e-prints, pages arXiv–1912.