Study of the block-sequential operator on Boolean networks. Application to discrete network analysis

Luis Cabrera Crot

Thesis for the PhD degree in Computer Science.[1]

Thesis Advisors: Julio Aracena, Adrien Richard, Lilian Salinas

January 22$^{\text{nd}}$, 2024

[1] Funded by CONICYT-PFCHA/Doctorado Nacional/2016-21160885

Boolean networks

Boolean Networks

- A global activation function
  $f = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$
  - Composed by local activation functions
    $f_v : \{0,1\}^n \to \{0,1\}$

$$
\begin{aligned}
f_1(x) &= x_2 \wedge x_4 \\
f_2(x) &= x_1 \\
f_3(x) &= x_2 \vee x_3 \\
f_4(x) &= x_3 \wedge x_4
\end{aligned}
$$

Boolean networks

Boolean Networks

- A global activation function
  $f = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$
  - Composed by local activation functions
    $f_v : \{0,1\}^n \to \{0,1\}$
- Schedule $s : V \to \{1, \dots, n\}$

$$
\begin{aligned}
f_1(x) &= x_2 \wedge x_4 \\
f_2(x) &= x_1 \\
f_3(x) &= x_2 \vee x_3 \\
f_4(x) &= x_3 \wedge x_4
\end{aligned}
$$

$$s = \{1, 2, 3, 4\}$$

Boolean networks

Boolean Networks

- A global activation function
  $f = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$
  - Composed by local activation functions
    $f_v : \{0,1\}^n \to \{0,1\}$
- Schedule $s : V \to \{1, \ldots, n\}$

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$

$s = \{1, 2, 3, 4\}$

Parallel schedule

Boolean networks

Boolean Networks

- A global activation function
  $f = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$
  - Composed by local activation functions
    $f_v : \{0,1\}^n \to \{0,1\}$
- Schedule $s : V \to \{1, \ldots, n\}$

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$

$s = \{1\}\{2\}\{3\}\{4\}$
Sequential schedule

Boolean networks

Boolean Networks

- A global activation function $f = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$
  - Composed by local activation functions $f_v : \{0,1\}^n \to \{0,1\}$
- Schedule $s : V \to \{1, \ldots, n\}$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{3,4\}\,\{1,2\}$$

Block-sequential schedule

Boolean networks

$$
\begin{aligned}
f_1(x) &= x_2 \wedge x_4 \\
f_2(x) &= x_1 \\
f_3(x) &= x_2 \vee x_3 \\
f_4(x) &= x_3 \wedge x_4
\end{aligned}
$$

Interaction Digraph

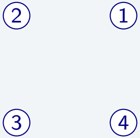
- A digraph $G(f) = (V, A)$:

Boolean networks

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$

Interaction Digraph

- A digraph $G(f) = (V, A)$:
  - $V = \{1, \ldots, n\}$

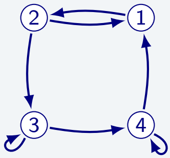


Boolean networks

Interaction Digraph

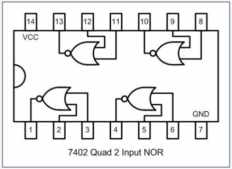
- A digraph $G(f) = (V, A)$:
  - $V = \{1, \ldots, n\}$
  - $(i, j) \in A \Leftrightarrow f_j$ depends on x_i

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$

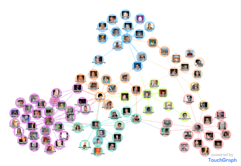


Applications of Boolean networks

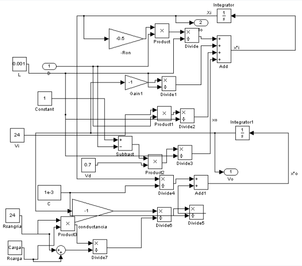
Circuit theory



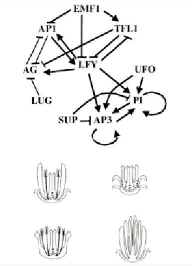
Social networks



Computer science



Biological systems



Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

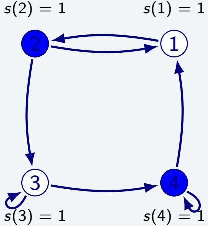
$$x_v(t+1) = f_v(x_u(t')\colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$
$$s = \{1, 2, 3, 4\}$$



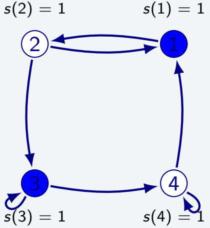
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{1, 2, 3, 4\}$$



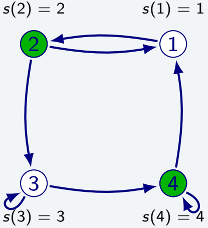
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t')\colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$s = \{1\}\,\{2\}\,\{3\}\,\{4\}$



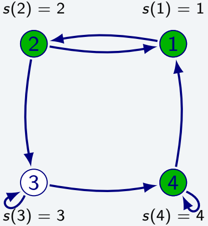
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{1\}\,\{2\}\,\{3\}\,\{4\}$$



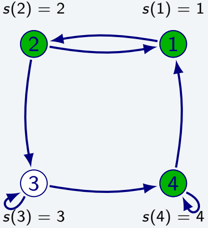
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{1\}\,\{2\}\,\{3\}\,\{4\}$$



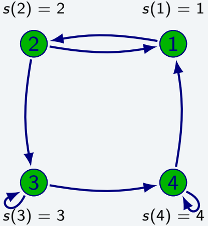
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{1\}\,\{2\}\,\{3\}\,\{4\}$$



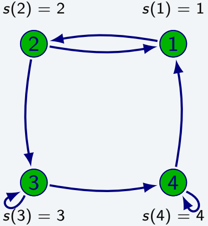
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$$s = \{1\}\,\{2\}\,\{3\}\,\{4\}$$



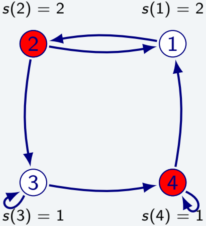
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t')\colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

$s = \{3,4\}\,\{1,2\}$



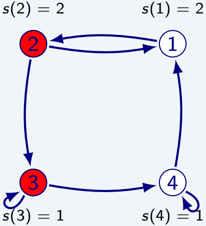
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$f_1(x) = x_2 \wedge x_4$$
$$f_2(x) = x_1$$
$$f_3(x) = x_2 \vee x_3$$
$$f_4(x) = x_3 \wedge x_4$$

$$s = \{3, 4\}\,\{1, 2\}$$



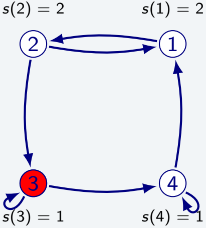
Dynamic of a Boolean network

The dynamic of a Boolean network (f, s) is given by:

$$x_v(t+1) = f_v(x_u(t') \colon u \in V), \quad t' = \begin{cases} t & \text{if } s(v) \leq s(u) \\ t+1 & \text{if } s(v) > s(u) \end{cases}$$

$$\begin{aligned} f_1(x) &= x_2 \wedge x_4 \\ f_2(x) &= x_1 \\ f_3(x) &= x_2 \vee x_3 \\ f_4(x) &= x_3 \wedge x_4 \end{aligned}$$

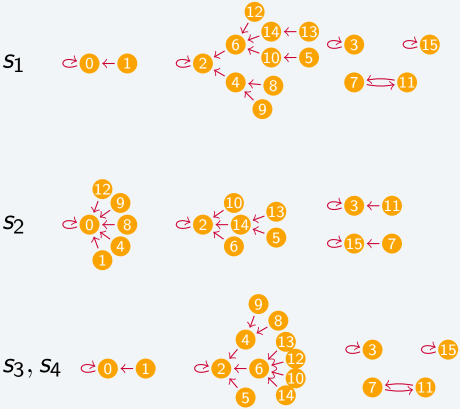
$$s = \{3, 4\}\,\{1, 2\}$$



Dynamical behavior

$s_1 = \{1,2,3,4\}$
$s_2 = \{1\}\{2\}\{3\}\{4\}$
$s_3 = \{3,4\}\{1,2\}$
$s_4 = \{4\}\{3\}\{1,2\}$

	State	f^{s_1}	f^{s_2}	$f^{s_3} = f^{s_4}$
0	0000	0000	0000	0000
1	0001	0000	0000	0000
2	0010	0010	0010	0010
3	0011	0011	0011	0011
4	0100	0010	0000	0010
5	0101	1010	1110	0010
6	0110	0010	0010	0010
7	0111	1011	1111	1011
8	1000	0100	0000	0100
9	1001	0100	0000	0100
10	1010	0110	0010	0110
11	1011	0111	0011	0111
12	1100	0110	0000	0110
13	1101	1110	1110	0110
14	1110	0110	0010	0110
15	1111	1111	1111	1111



Initial motivation

The initial motivation was to understand the variations of the interaction digraph of a Boolean network with respect to changes in the update schedule and its relation with some dynamic properties of the network.

- Given a Boolean network, which structural properties of the interaction network are sensitive to the change in the update schedule?
- Given the invariance of fixed points with respect to different update schedules, how can we take advantage of that in an algorithm to find them?
- What are the properties that must be satisfied to find networks that have the same dynamic behavior?

Labeled digraph

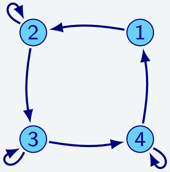
Given a interaction graph $G(f)$ and a block-sequential schedule s, a labeled digraph $(G(f), lab_s)$ is defined, where:

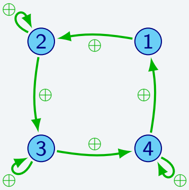
$$lab_s : A(G) \to \{\oplus, \ominus\}$$

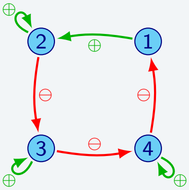
$$lab_s(u, v) = \oplus \iff s(u) \geq s(v)$$

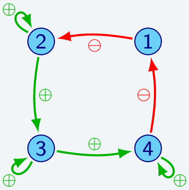
$G(f)$ $s_1 = \{1, 2, 3, 4\}$ $s_2 = \{2\}\{3\}\{4\}\{1\}$ $s_3 = \{3, 4\}\{1\}\{2\}$

$s_4 = \{4\}\{1\}\{3, 2\}$









Why are we interested in labeled digraphs?

Theorem (Aracena, Goles, Moreira, Salinas (2009))

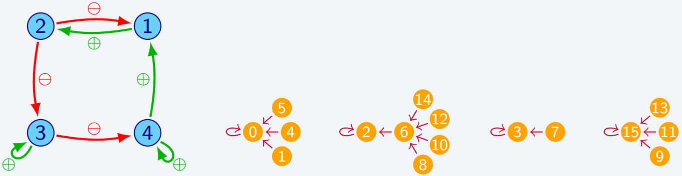
Given two Boolean networks (f, s) and (f, s') which differ only in the update schedule. If the labeled digraphs associated to them are equal, then both networks have the same dynamical behavior.

Why are we interested in labeled digraphs?

Theorem (Aracena, Goles, Moreira, Salinas (2009))

Given two Boolean networks (f, s) and (f, s') which differ only in the update schedule. If the labeled digraphs associated to them are equal, then both networks have the same dynamical behavior.

$$s = \{2\}\{1\}\{3\}\{4\} \text{ and } s' = \{2\}\{3\}\{1\}\{4\}$$

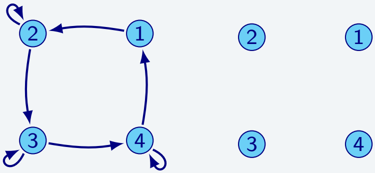


Parallel digraph

The *parallel digraph* is a digraph that represent the potential dependence of the components of the Boolean network, according with the block-sequential schedule s.

$s = \{2\}\{3\}\{4\}\{1\}$

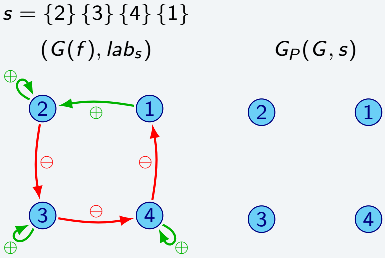
$G(f)$ $\quad$ $G_P(G, s)$



Parallel digraph

$\forall (u, v) \in V(G) \times V(G), (u, v) \in A(G_P(G, s))$ if and only if:

- (u, v) is labeled $\oplus$
- $\exists w \in V(G), (u, w)$ is labeled $\oplus$ and exists a path from w to v labeled $\ominus$



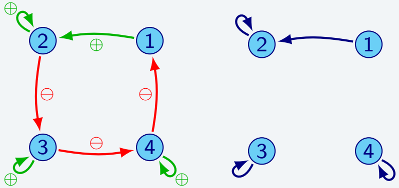
Parallel digraph

$\forall (u,v) \in V(G) \times V(G), (u,v) \in A(G_P(G,s))$ if and only if:

- (u,v) is labeled $\oplus$
- $\exists w \in V(G), (u,w)$ is labeled $\oplus$ and exists a path from w to v labeled $\ominus$

$s = \{2\}\,\{3\}\,\{4\}\,\{1\}$

$(G(f), lab_s)$ $\quad\quad$ $G_P(G,s)$



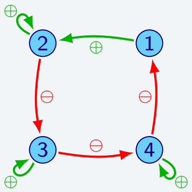
Parallel digraph

$\forall (u,v) \in V(G) \times V(G), (u,v) \in A(G_P(G,s))$ if and only if:

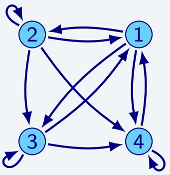
- (u,v) is labeled $\oplus$
- $\exists w \in V(G), (u,w)$ is labeled $\oplus$ and exists a path from w to v labeled $\ominus$

$s = \{2\}\{3\}\{4\}\{1\}$

$(G(f), lab_s)$



$G_P(G,s)$



Parallel digraph

Proposition

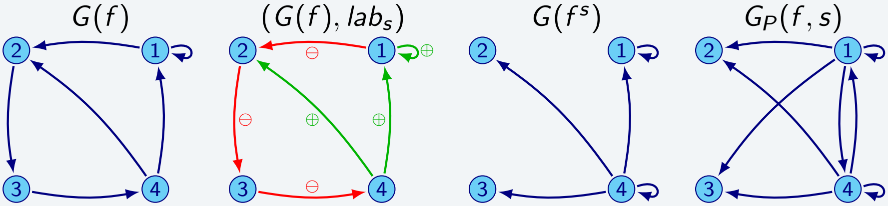
$G(f^s) \subseteq G_P(f,s)$.

Remark

If (f,s) is a conjunctive (or disyuntive) network, then $G(f^s) = G_P(f,s)$. (Goles and Noual, 2012)

Parallel digraph

The inclusion could be strict in monotone Boolean networks:



$$\begin{aligned} f_1(x) &= x_1 \wedge x_4 \\ f_2(x) &= x_1 \vee x_4 \\ f_3(x) &= x_2 \\ f_4(x) &= x_3 \end{aligned}$$

$$s = \{1\}\{2\}\{3\}\{4\}$$

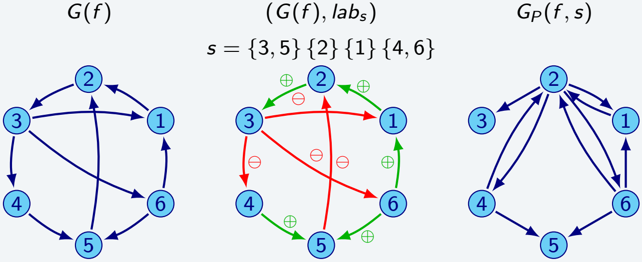
$$\begin{aligned} f_1^s(x) &= x_1 \wedge x_4 \\ f_2^s(x) &= x_4 \\ f_3^s(x) &= x_4 \\ f_4^s(x) &= x_4 \end{aligned}$$

Contents

Characterization of the parallel digraph

Proposition

Let (f, s) be a Boolean Network, and $G(f)$ the interaction digraph associated with f. If $G(f)$ is a strongly connected digraph, then $G_P(f, s)$ has one unique strongly connected component composed by the vertices with an arc with $\oplus$, and the other vertices are pending nodes of $G(f)$.



This is an extension of the result obtained by Goles and Noual (2012).

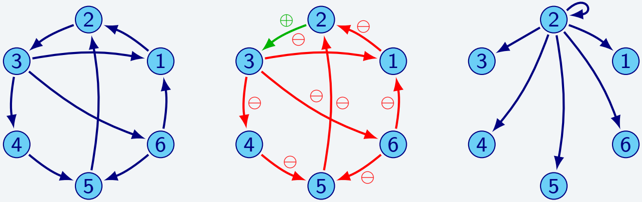
Characterization of the parallel digraph

Proposition

The size of the strongly connected component of $G_P(f,s)$ is greater than or equal to $\tau(G(f))$. Moreover, there exists a schedule s such that the size of the strongly connected component of $G_P(f,s)$ is equal to $\tau(G(f))$.

$G(f)$ | $G(f, lab_s)$ | $G_P(f,s)$

$\tau(G(f)) = 1$ | $s = \{3\}\{4,6\}\{5,1\}\{2\}$

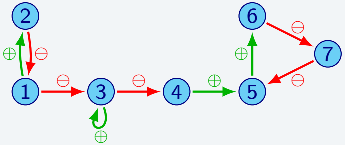


Characterization of the parallel digraph

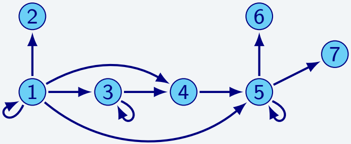
This result can be extended to not strongly connected digraphs.

$$s = \{2\}\,\{1,6\}\,\{3,7\}\,\{4,5\}$$

$(G(f), lab_s)$



$G_P(f,s)$



Feedback vertex set in parallel digraph

Proposition

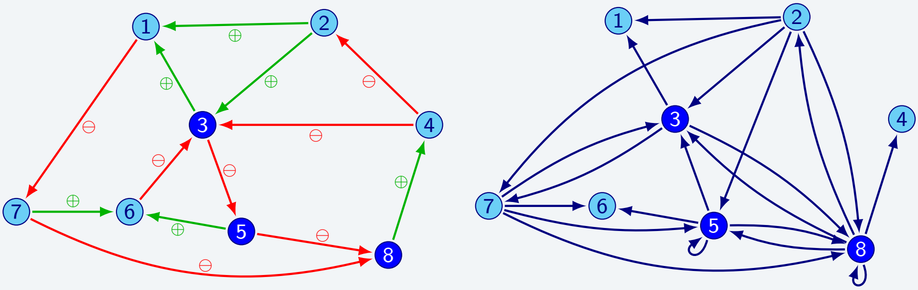
Let (f, s) be a Boolean network. Given $U \subseteq V(f)$,

$$U \text{ is a FVS of } G_P(f,s) \implies U \text{ is a FVS of } G(f)$$

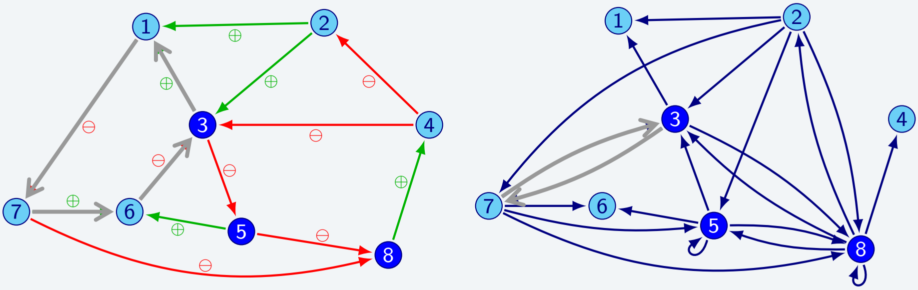
Thus,

$$\tau(G_P(f,s)) \geq \tau(G(f)).$$

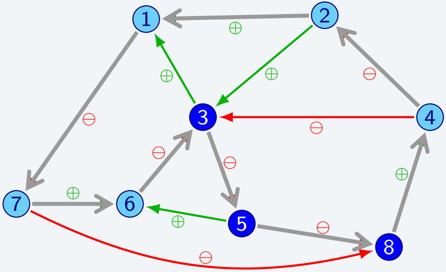
Feedback vertex set in parallel digraph

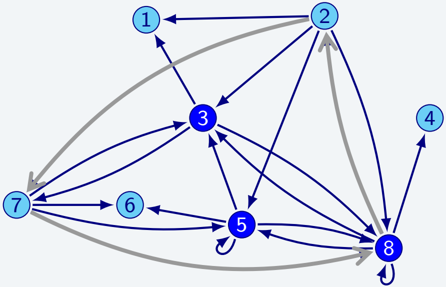


Feedback vertex set in parallel digraph

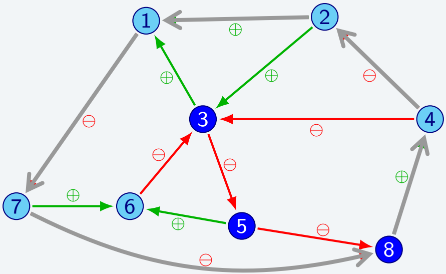


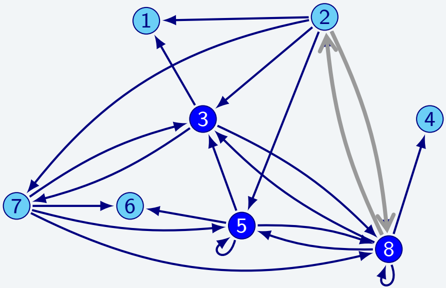
Feedback vertex set in parallel digraph



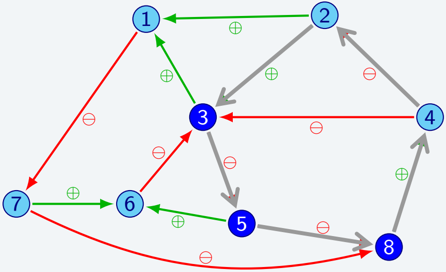


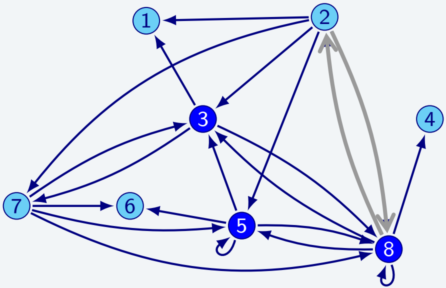
Feedback vertex set in parallel digraph





Feedback vertex set in parallel digraph





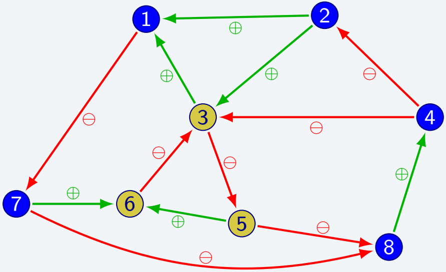
Packing number in parallel digraph

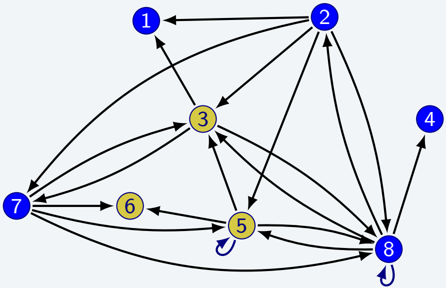
Proposition

Let (f, s) be a Boolean network. Then,

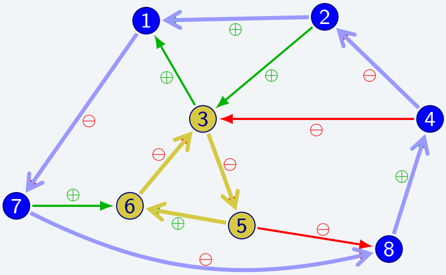
$$\nu(G_P(f, s)) \geq \nu(G(f)).$$

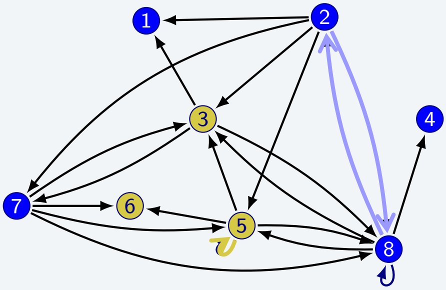
Packing number in parallel digraph



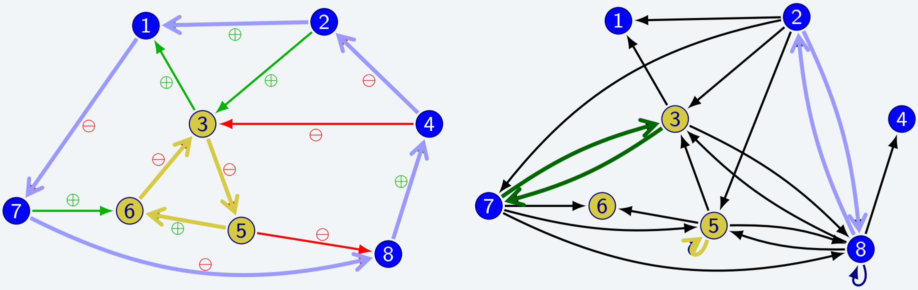


Packing number in parallel digraph





Packing number in parallel digraph



Motivation

Since $\tau(G_P(f,s)) \geq \tau(G(f))$ and $\nu(G_P(f,s)) \geq \nu(G(f))$ and $\tau(G(f)) \geq \nu(G(f))$.

Given an Boolean network f. Is there a schedule s such that:

$$\nu(G_P(f,s)) = \tau(G_P(f,s)) = \tau(G(f))?$$

Labeling function lab_U

lab_U

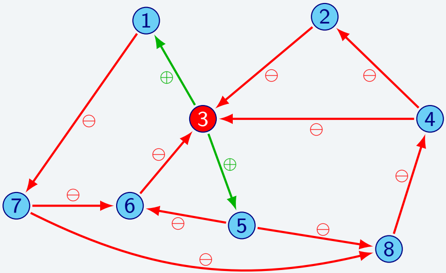
Let $U \subseteq V(f)$ a feedback vertex set of $G(f)$. We define a labeling function $lab_U : A(f) \to \{\oplus, \ominus\}$ as:

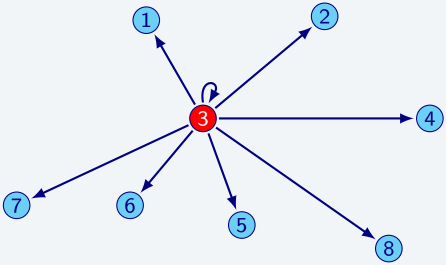
$$\forall (i,j) \in A(G) \quad lab_U(i,j) = \begin{cases} \oplus & \text{if } i \in U \\ \ominus & \text{otherwise.} \end{cases}$$

Proposition

If $|U| = \tau(G(f))$, then $\tau(G_P(f,s)) = \tau(G(f)) = \nu(G_P(f,s))$.

Labeling function lab_U





Contents

Algorithms to find fixed points

First, it is important to note that the process of finding fixed points in a Boolean network is a difficult task. In 1985, Noga Alon show that answering if a network has a fixed point is NP-complete, and in 1989, Patrik Floréen and Pekka Orponen show that counting how many fixed points a Boolean network has is #P-complete.

Algorithms to find fixed points

Strategies used

- Reduction methods - (Veliz-Cuba, 2011; Zañudo and R. Albert, 2013; Veliz-Cuba et al., 2015)
- Representation as polynomial functions - (Hinkelmann et al., 2011; Zou, 2013)
- SAT-based methods - (Devloo et al., 2003; Tamura and Akutsu, 2009; Melkman et al., 2010; Akutsu et al., 2011; Dubrova and Teslenko, 2011)
- Methods based on Integer Programming - (Akutsu et al., 2009)
- Strategic Sampling - (Zhang et al., 2007)
- Methods based on Minimal Feedback Vertex Sets - (Akutsu et al., 1998)

Motivation

$$|FP(f)| \leq 2^{\tau(G(f))}$$

Where:

Motivation

$$|FP(f)| \leq 2^{\tau(G(f))}$$

Where:

- $FP(f)$ is the set of fixed points for a network f.

Motivation

$$|FP(f)| \leq 2^{\tau(G(f))}$$

Where:

- $FP(f)$ is the set of fixed points for a network f.
- $\tau(G(f))$ is the minimum size of a feedback vertex set in $G(f)$.

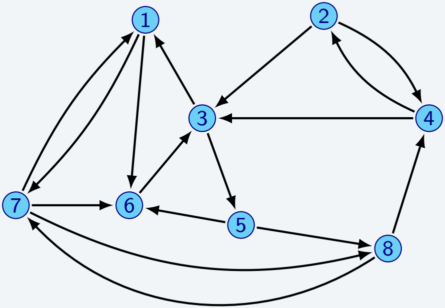
Motivation

$$|FP(f)| \leq 2^{\tau^+(G(f))} \leq 2^{\tau(G(f))}$$

Where:

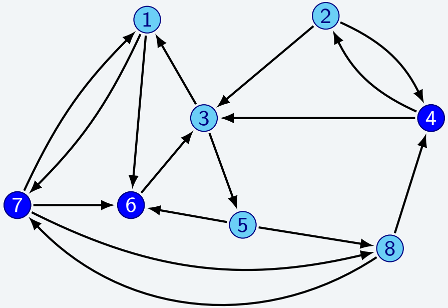
- $FP(f)$ is the set of fixed points for a network f.
- $\tau(G(f))$ is the minimum size of a feedback vertex set in $G(f)$.
- $\tau^+(G(f))$ is the minimum size of a positive feedback vertex set in $G(f)$.

List the fixed points of a Boolean network



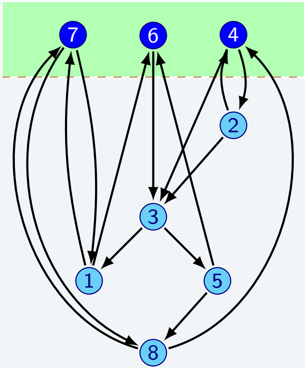
Let G be the interaction digraph of a Boolean network

List the fixed points of a Boolean network



And let F be a FVS (not necessarily minimal) of G

List the fixed points of a Boolean network



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

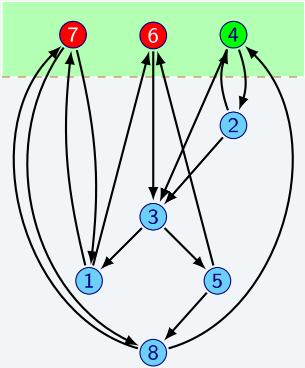
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

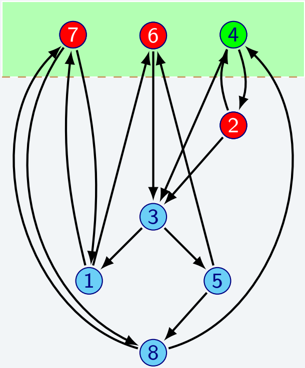
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

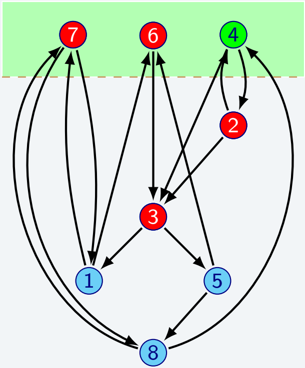
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

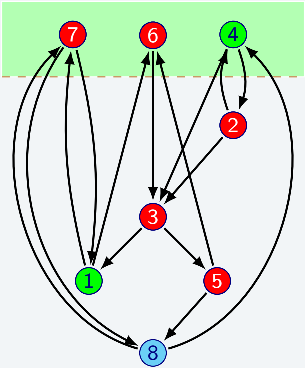
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

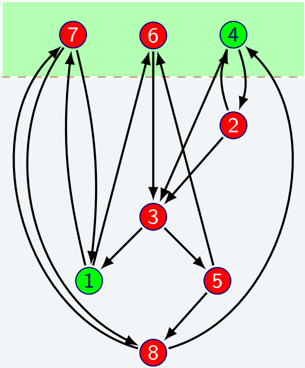
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

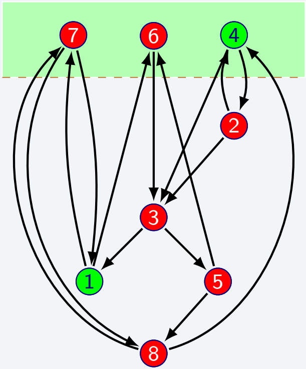
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



10010000 is a fixed point of f

List the fixed points of a Boolean network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

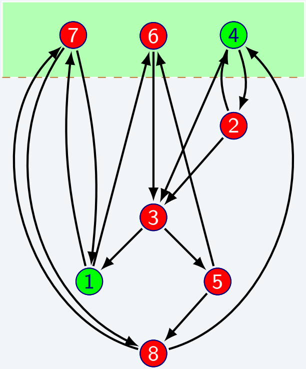
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

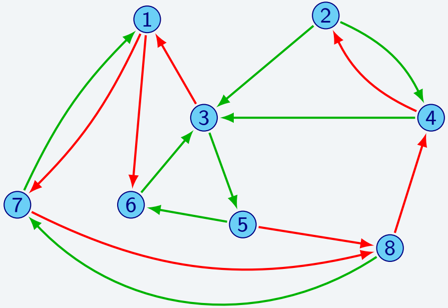
$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



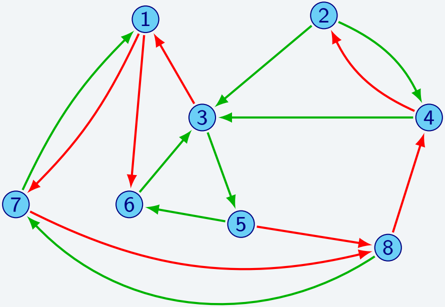
The complexity of this algorithm is $n \cdot 2^{|F|}$

Algorithm for regulatory networks



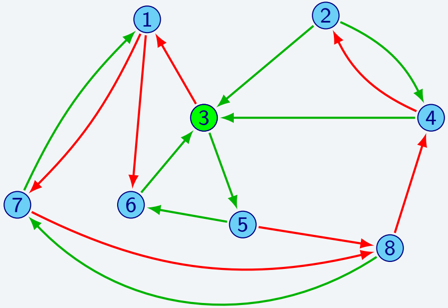
In the case of regulatory networks, the arcs of the interaction digraph can be weighted positive or negative.

Algorithm for regulatory networks



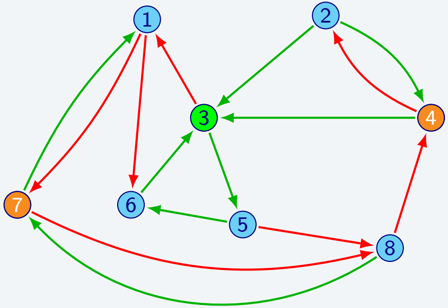
Let G be the interaction digraph of a Boolean network

Algorithm for regulatory networks



And let F^+ be a PFVS (not necessarily minimal) of G

Algorithm for regulatory networks



And let $F \setminus F^+$ the complement of F^+ to be a FVS

List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

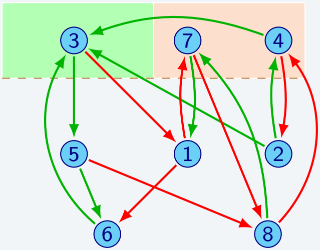
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

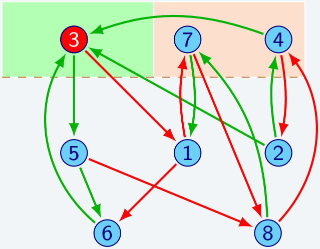
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

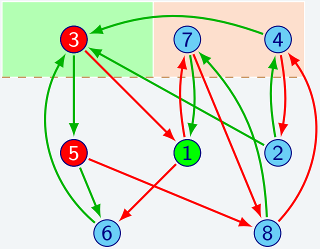
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

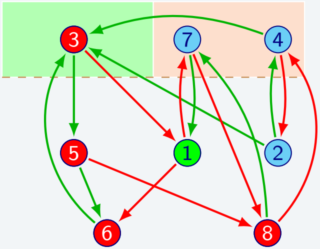
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

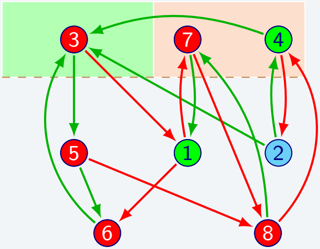
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

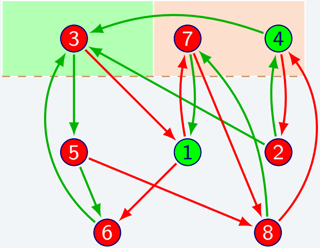
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

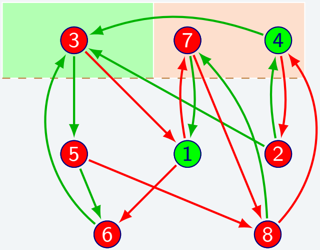
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



10010000 is a fixed point of f

List the fixed points of a regulatory network

Legend:

Vertex with True value

Vertex with False value

$f_1 = \overline{x_3} \vee x_7$

$f_2 = \overline{x_4}$

$f_3 = (x_2 \wedge x_4) \vee (x_2 \wedge x_6) \vee (x_4 \wedge x_6)$

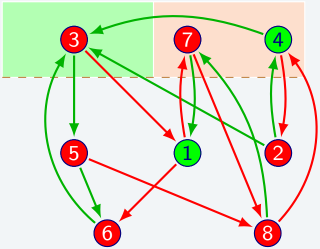
$f_4 = x_2 \vee \overline{x_8}$

$f_5 = x_3$

$f_6 = \overline{x_1} \vee x_5$

$f_7 = \overline{x_1} \vee x_8$

$f_8 = x_5 \wedge x_7$



The complexity of this algorithm is $((|F| - |F^+|)(n - |F^+|) + n) \cdot 2^{|F^+|}$

Complexity analysis

The complexity of listing fixed points based on a PFVS is

Complexity analysis

The complexity of listing fixed points based on a PFVS is

$$((|F| - |F^+|)(n - |F^+|) + n) \cdot 2^{|F^+|}$$

Complexity analysis

The complexity of listing fixed points based on a PFVS is

$$((|F| - |F^+|)(n - |F^+|) + n) \cdot 2^{|F^+|}$$

If $|F| = |F^+|$:

Complexity analysis

The complexity of listing fixed points based on a PFVS is

$$((|F| - |F^+|)(n - |F^+|) + n) \cdot 2^{|F^+|}$$

If $|F| = |F^+|$:

$$n \cdot 2^{|F|}$$

Complexity analysis

The complexity of listing fixed points based on a PFVS is

$$((|F| - |F^+|)(n - |F^+|) + n) \cdot 2^{|F^+|}$$

If $|F| = |F^+|$:

$$n \cdot 2^{|F|}$$

It is equivalent to the execution of FVS-algorithm

Contents

Motivation

State of Art

- (Goles and Salinas, 2008) and (Aracena et al, 2009) define equivalence classes between different update schedules.
- (Goles and Noual, 2012) classify the disjunctive networks according to the robustness of their dynamics concerning changes in the update schedule, all this using parallel digraphs.
- (Goles and Salinas, 2010) study how the network changes when it is updated with a given sequential schedule.

Motivation

However, to our knowledge, the following questions have been little explored:

- What other networks have the same dynamics as that of a given network?
- What dynamics are only yielded by a parallel schedule?

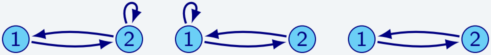
Dynamically equivalent network

Definition

Let f, h be two Boolean networks and s an update schedule. We say that (h, s) is *dynamically equivalent to* f if $h^s = f$. Moreover, if $h \neq f$, or $h = f$ and $s \not\sim_f s_p$, we say that (h, s) and f are *non-trivially dynamically equivalent*.

Counter-examples

By reflexivity, any network is dynamically equivalent to itself (considering the parallel schedule). But there are networks that do not have another network that is non-trivial dynamically equivalent, for example:



Dynamically equivalent networks problem

Dynamically equivalent networks problem (DEN problem)

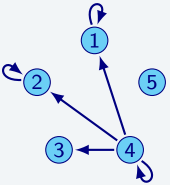
Input: *A Boolean network f (encoded as a Boolean formula for each f_i).*

Question: *does there exists a Boolean network h and an update schedule s, such that (h, s) is non-trivially dynamically equivalent to f?*

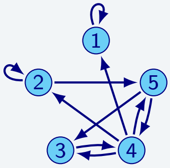
What do we need to solve this problem?

Lemma

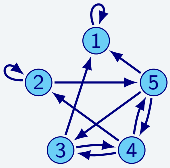
Let h, f be Boolean networks and $s = B_1, B_2, \ldots, B_m$ with $m > 1$. If $h^s = f$, then there exists $\bar{h}$ and $\bar{s}$ such that $\bar{h}^{\bar{s}} = f$ where $\bar{s} = B_1, B_2, \ldots, B_{m-1} \cup B_m$.



$f_1(x) = x_1 \vee x_4$
$f_2(x) = x_2 \wedge x_4$
$f_3(x) = x_4$
$f_4(x) = x_4$
$f_5(x) = 0$



$h_1(x) = x_1 \vee x_4$
$h_2(x) = x_2 \wedge x_4$
$h_3(x) = x_4 \wedge \neg x_5$
$h_4(x) = x_3 \wedge \neg x_5$
$h_5(x) = x_2 \wedge \neg x_4$
$s = \{2\}\{5\}\{3\}\{4\}\{1\}$



$\bar{h}_1(x) = x_1 \vee (x_3 \wedge \neg x_5)$
$\bar{h}_2(x) = x_2 \wedge x_4$
$\bar{h}_3(x) = x_4 \wedge \neg x_5$
$\bar{h}_4(x) = x_3 \wedge \neg x_5$
$\bar{h}_5(x) = x_2 \wedge \neg x_4$
$\bar{s} = \{2\}\{5\}\{3\}\{4, 1\}$

How hard is it to solve the problem?

Theorem

DEN is NP-Hard.

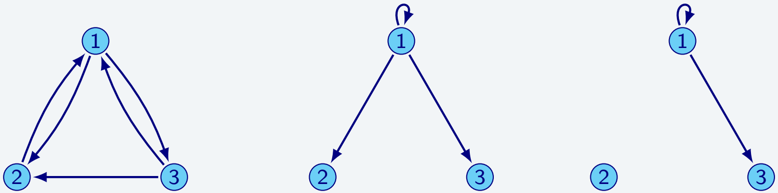
Proof

We prove that 3-SAT $\leq_p$ DEN.

Dynamically equivalent disjunctive networks problem

DYNAMICALLY EQUIVALENT DISJUNCTIVE NETWORKS PROBLEM (D-DEN PROBLEM)
Input: *A disjunctive Boolean network f (encoded as a Boolean formula for each f_i).*
Question: *does there exists a disjunctive Boolean network h and an update schedule s, such that (h, s) is non-trivially dynamically equivalent to f?*

Why only restrict ourselves to disjunctive Boolean networks h?



$$\begin{aligned} h_1(x) &= x_2 \veebar x_3 \\ h_2(x) &= x_1 \veebar x_3 \\ h_3(x) &= x_1 \end{aligned}$$

$$\begin{aligned} h_1^s(x) &= (x_1 \veebar x_1) \veebar x_1 \\ h_2^s(x) &= x_1 \veebar x_1 \\ h_3^s(x) &= x_1 \\ s &= \{3\}\,\{2\}\,\{1\} \end{aligned}$$

$$\begin{aligned} h_1^s(x) &= x_1 \\ h_2^s(x) &= 0 \\ h_3^s(x) &= x_1 \end{aligned}$$

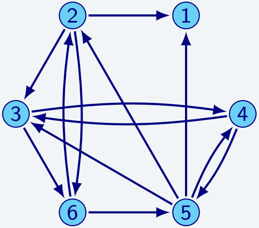
What do we need to solve this problem?

Proposition

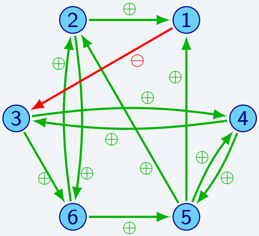
Let f be a disjunctive Boolean network. There exists a disjunctive Boolean network h and an update schedule s, such that (h, s) is non-trivially dynamically equivalent to f and with only one negative arc $(u, v) \in A(h)$ if and only if the following conditions are satisfied:

1. $N_f^-(u) \subseteq N_f^-(v)$,
2. $u \notin N_f^-(v) \setminus N_f^-(u)$,
3. For every vertex $w \in N_f^-(v) \setminus N_f^-(u)$, it does not exist a path from u to w in $G(f) - v$.

$G(f)$



$G(h)$



What do we need to solve this problem?

Corollary

Let f be a disjunctive Boolean network. If there exist $u, v \in [n]$ such that $N_f^-(u) = N_f^-(v)$, then there exists a disjunctive Boolean network h and an update schedule s such that (h, s) is non-trivially dynamically equivalent to f.

If $N_f^-(u^*) = N_f^-(v^*)$ then the three conditions are satisfied:

1. $N_f^-(u) \subseteq N_f^-(v)$,
2. $u \notin N_f^-(v) \setminus N_f^-(u)$,
3. For every vertex $w \in N_f^-(v) \setminus N_f^-(u)$, it does not exist a path from u to w in $G(f) - v$.

$A^{\ominus}$ set

Definition

Let f be a disjunctive Boolean network. We define the following set of arcs:

$$A^{\ominus}(f) = \{(u, v) \in [n] \times [n] : u \neq v \wedge N_f^-(u) \subseteq N_f^-(v)\}.$$

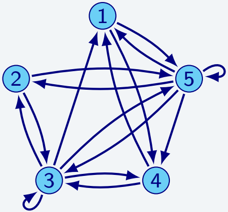
This set represents all arcs in $[n] \times [n]$ that can be labeled $\ominus$. Note that the inclusion relationship of these sets is transitive, because if $N_f^-(u) \subseteq N_f^-(w)$ and $N_f^-(w) \subseteq N_f^-(v)$, then $N_f^-(u) \subseteq N_f^-(v)$.

$A^{\ominus}$ set

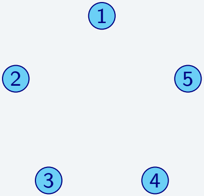
If $A^{\ominus}(f)$ has a cycle, then D-DEN problem has a solution (since at least two vertices would have the same in-neighborhood).

What occurs if $A^{\ominus}$ is acyclic?

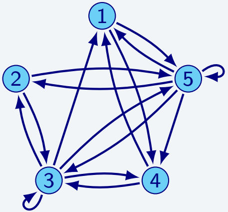
$\mathcal{G}_{lab}(f, A^-)$ operator



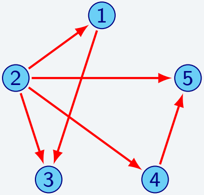
$$N^-(1) = \{3,4,5\}$$
$$N^-(2) = \{3,5\}$$
$$N^-(3) = \{2,3,4,5\}$$
$$N^-(4) = \{1,3,5\}$$
$$N^-(5) = \{1,2,3,5\}$$



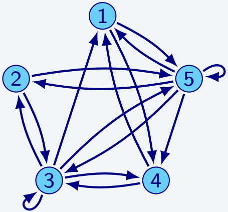
$\mathcal{G}_{lab}(f, A^-)$ operator



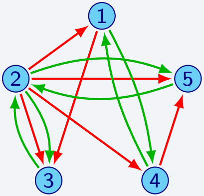
$$N^-(1) = \{3,4,5\}$$
$$N^-(2) = \{3,5\}$$
$$N^-(3) = \{2,3,4,5\}$$
$$N^-(4) = \{1,3,5\}$$
$$N^-(5) = \{1,2,3,5\}$$



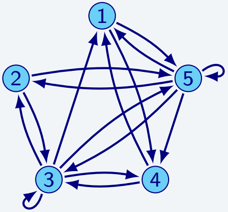
$\mathcal{G}_{lab}(f, A^-)$ operator



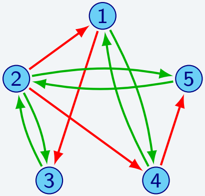
$$N^-(1) = \{3, 4, 5\}$$
$$N^-(2) = \{3, 5\}$$
$$N^-(3) = \{2, 3, 4, 5\}$$
$$N^-(4) = \{1, 3, 5\}$$
$$N^-(5) = \{1, 2, 3, 5\}$$



$\mathcal{G}_{lab}(f, A^-)$ operator



$N^-(1) = \{3, 4, 5\}$
$N^-(2) = \{3, 5\}$
$N^-(3) = \{2, 3, 4, 5\}$
$N^-(4) = \{1, 3, 5\}$
$N^-(5) = \{1, 2, 3, 5\}$



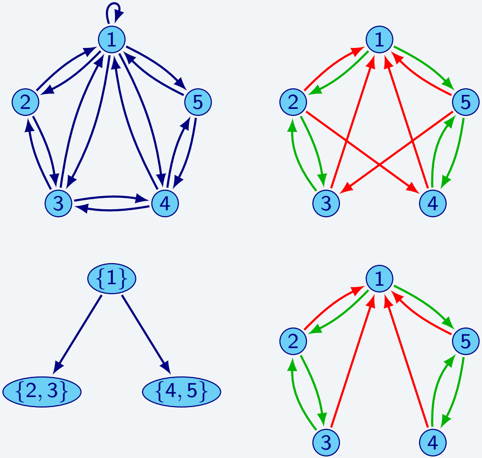
Admissible partition

Definition

Let (G, lab) be a labeled digraph. A partition $\{V_1, V_2\}$ of $[n]$ is said to be *admissible* if satisfies the following conditions:

1. $\exists (u, v) \in A(G), u \in V_1 \wedge v \in V_2$,
2. $\forall (u, v) \in A(G), u \in V_1 \wedge v \in V_2 \Rightarrow lab(u, v) = \ominus$,
3. $\forall (u, v) \in A(G), u, v \in V_2 \Rightarrow lab(u, v) = \oplus$.

Admissible partition



Remarks

- Different approaches to the problem of dynamically equivalent networks are presented.
- Solving the general problem is NP-Hard, since it is as difficult as 3-SAT
- An approach to finding a possible solution is presented: if there exists a solution with an update schedule with more than two blocks, then there exists a solution with an update schedule of only two blocks.
- The problem restricted to disjunctive networks can be solved in polynomial time.

Contents

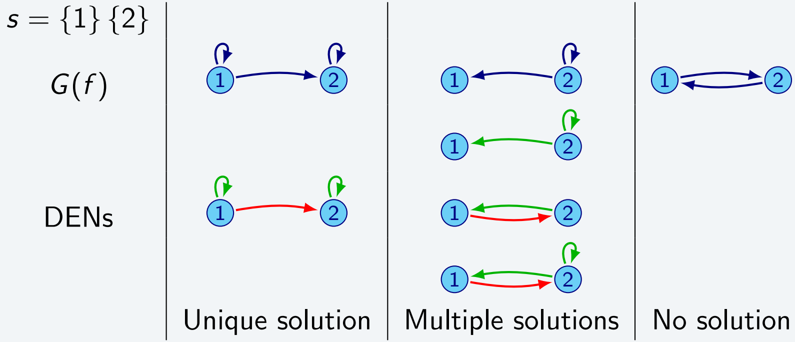
D-DEN problem with fixed update schedule

DYNAMICALLY EQUIVALENT DISJUNCTIVE BOOLEAN NETWORK WITH FIXED SCHEDULE PROBLEM

Input: *f is a disjunctive Boolean network and s is an update schedule.*

Question: *Does there exist h a disjunctive Boolean network such that $h^s = f$?*

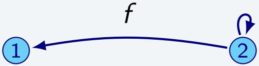
Scenarios of D-DEN problem with fixed update schedule



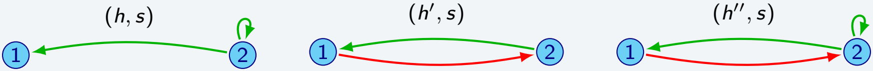
Results

Theorem

Let s be a block-sequential schedule and h and h' two Boolean networks such that $G_P(h, s) = G_P(h', s)$. Then $G_P(h \cup h', s) = G_P(h)$.



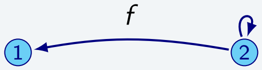
There are three preimages:



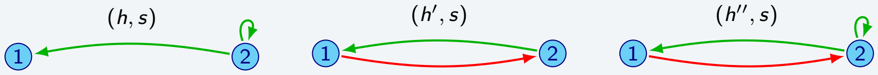
Results

Theorem

For this reason, if for a Boolean network f and a block-sequential schedule s there exists at least one DEN h, then there exists a maximum DEN.



There are three preimages:

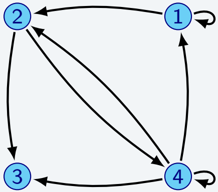


Algorithm MaximumDEDBN

Input

Given the following Boolean network f and a block-sequential schedule $s = \{3\}\{1\}\{2,4\}$.

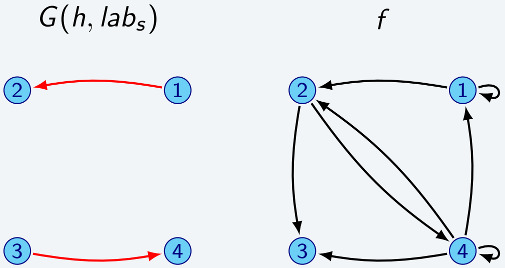
f



Algorithm MaximumDEDBN: Step 1 - Defining negative arcs

Rule

If there exists $(u, v) \in A^{\ominus}$, $s(u) < s(v)$, then (u, v) is a negative arc in $G(h, lab_s)$.

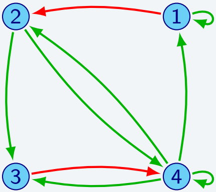


Algorithm MaximumDEDBN: Step 2 - Defining positive arcs

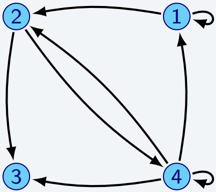
Rule

If there exists $(u, v) \in A(f)$, $s(u) \geq s(v)$, then (u, v) is a positive arc in $G(h, lab_s)$.

$G(h, lab_s)$

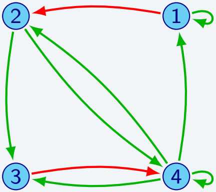


f

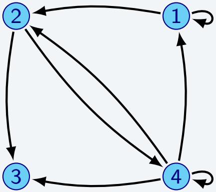


Algorithm MaximumDEDBN: Step 3 - Validation

$G(h, lab_s)$

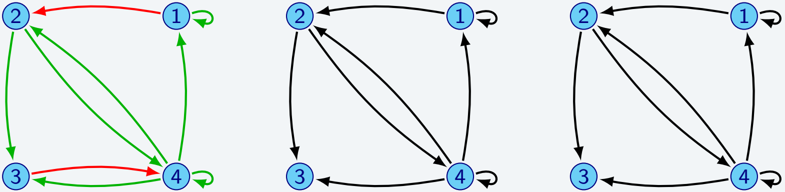


f



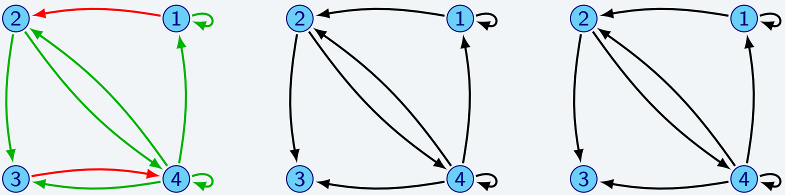
Algorithm MaximumDEDBN: Step 3 - Validation

$G(h, lab_s)$ $\quad$ $G_P(h, s)$ $\quad$ f



Algorithm MaximumDEDBN: Step 3 - Validation

$G(h, lab_s)$ $\quad$ $G_P(h, s)$ $\quad$ f



Output

Since $G_P(h, s) = f$, the algorithm return $G(h, lab_s)$ as maximun DEN of f with the update schedule s.

Enumeration of solutions with fixed schedule

ENUMERATION OF DYNAMICALLY EQUIVALENT DISJUNCTIVE BOOLEAN NETWORKS WITH FIXED UPDATE SCHEDULE

Input: *f is a disjunctive Boolean network and s is an update schedule.*

Question: *Is it possible to find the complete set of disjunctive Boolean networks h such that $h^s = f$?*

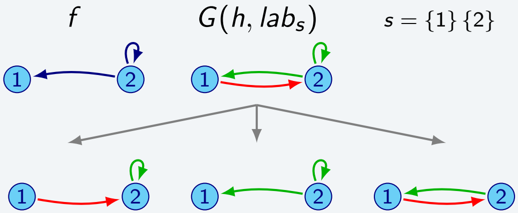
Enumeration of solutions with fixed schedule

f $\qquad G(h, lab_s) \qquad s = \{1\}\{2\}$

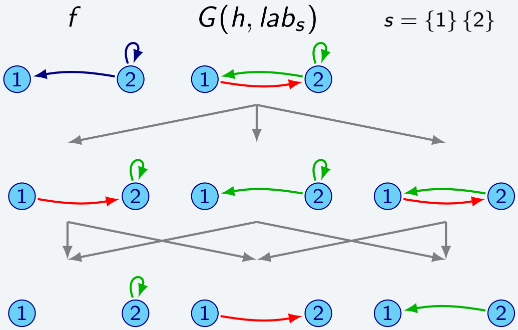
Enumeration of solutions with fixed schedule

f $\qquad$ $G(h, lab_s)$ $\qquad$ $s = \{1\}\{2\}$

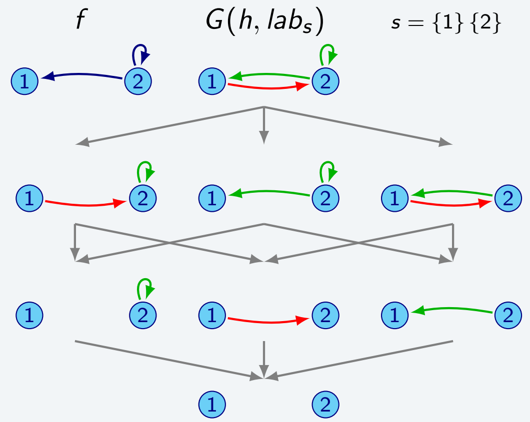
Enumeration of solutions with fixed schedule



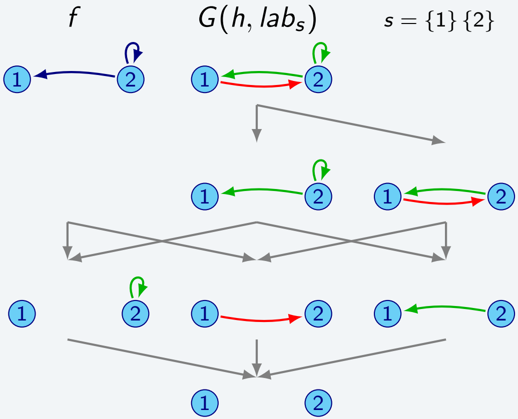
Enumeration of solutions with fixed schedule



Enumeration of solutions with fixed schedule

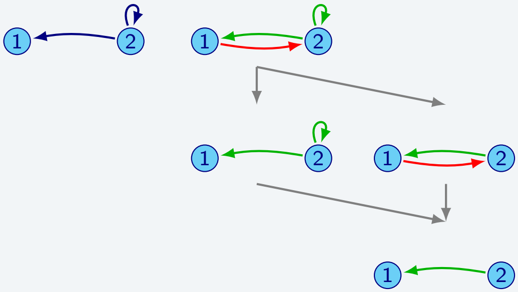


Enumeration of solutions with fixed schedule

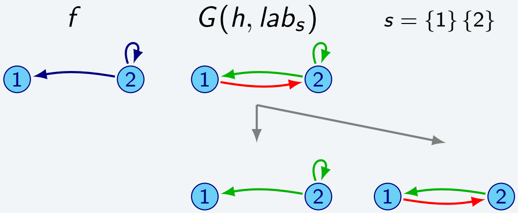


Enumeration of solutions with fixed schedule

f $\quad G(h, lab_s)$ $\quad s = \{1\}\{2\}$



Enumeration of solutions with fixed schedule



Enumeration of solutions with fixed schedule

Complexity

The complexity of this algorithm have a **polynomial delay**.

Enumeration of solutions with fixed schedule

Complexity

The complexity of this algorithm have a **polynomial delay**.
Since there are cases with an exponential number of DENs, listing all the DENs has an **exponential cost**

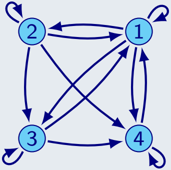
Enumeration of solutions with fixed schedule

Complexity

The complexity of this algorithm have a **polynomial delay**.

Since there are cases with an exponential number of DENs, listing all the DENs has an **exponential cost**

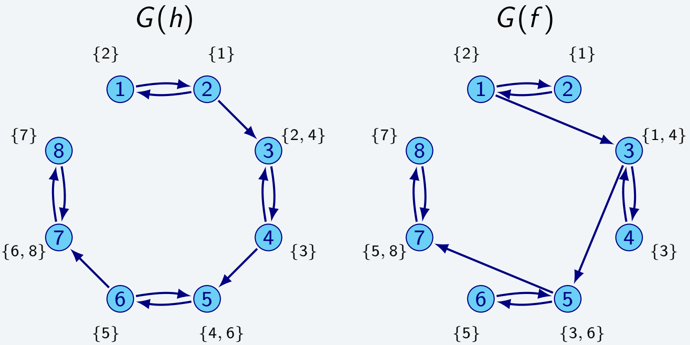
For example, this digraph with the block-sequential schedule $\{2\}\,\{3\}\,\{4\}\,\{1\}$ has 8 different DENs, corresponding to $2^{\frac{(n-2)(n-1)}{2}}$.



Dynamically equivalent disjunctive Boolean networks with fixed Boolean network problem

DYNAMICALLY EQUIVALENT DISJUNCTIVE BOOLEAN NETWORKS WITH FIXED BOOLEAN NETWORK PROBLEM
Input: *h and f be two disjunctive Boolean networks.*
Question: *Does there exist an update schedule $s \not\sim s_p$ such that $h^s = f$?*

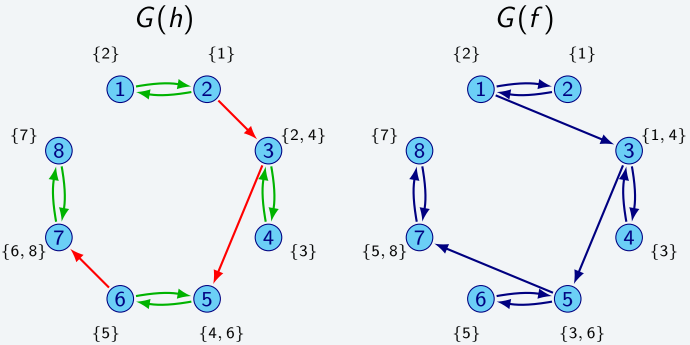
Preprocessing



Key concept

K-valid update schedule.

Preprocessing



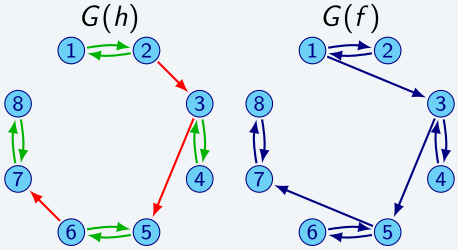
Key concept

K-valid update schedule.

s-divider procedure

Given a k-valid update schedule:

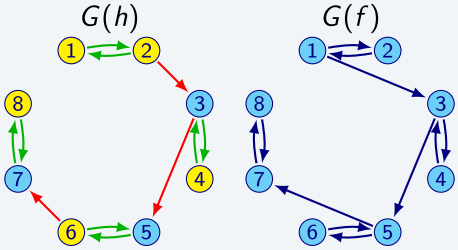
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

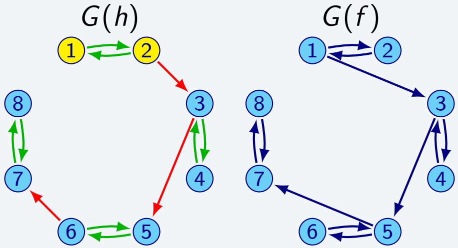
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

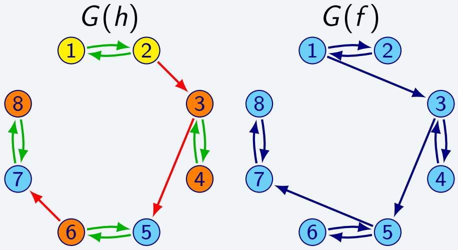
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

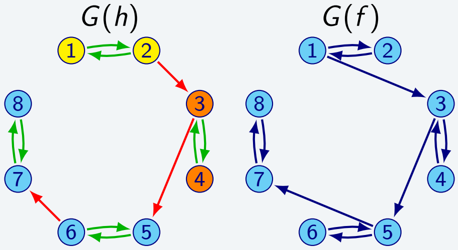
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

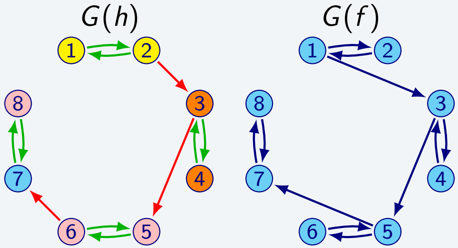
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

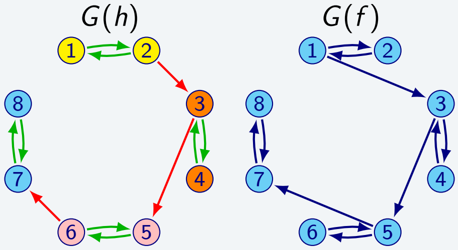
1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



s-divider procedure

Given a k-valid update schedule:

1. Select candidate vertices to belong to a block $k+1$.
2. It discards elements of this new block according to three criteria:
   - Negative criteria — Positive criteria — Potential criteria.
3. Finally, returns a $k+1$-valid update schedule (if it exists).



What occurs when $h = f$?

- A solution based on the s-divider procedure returns the parallel update schedule.
- A new strategy, based on two-block solutions, is generated, which gives us a non-trivial solution (if it exists).

Remarks

- Fixing one of the parameters of the original D-DEN problem does not generate a higher cost in its resolution.
- Moreover, in the case where the update schedule is fixed, a certain monotony is described that allows us to list all the solutions to the problem.
- The next step will be to find monotonicity in the case when we fix the Boolean network h so that we can find an algorithm that allows us to enumerate all the solutions.

Contents

Final conclusions

- In this thesis we have addressed different issues that allow us to satisfy the goal of studying how interaction graphs change when evaluated with different update schedules.
- All these results are interesting new tools that, combined in the right way, would allow to explore a number of topics in the field of Boolean networks that have yet to be explored.

Future Work: Open problems

- For FixedPoint algorithm, exploring methods to reduce the size of a network while conserving τ^+.
- Analyzing the comprehensive complexity of the DEN problem.
- Evaluating whether restricting conditions on the output of the D-DEN problem increases or decreases its time cost.
- Studying whether by applying D-DEN on a non-disjunctive network, it is possible to reconstruct its DEN.

Thank You!