UNIVERSIDAD DE CONCEPCIÓN

Centro de Investigación en Ingeniería Matemática (CI^{2}MA)

Centro de Investigación en Ingeniería Matemática

Complexity of limit cycle existence and feasibility problems in Boolean networks

Julio Aracena, Luis Gomez,
Lilian Salinas

PREPRINT 2014-30

SERIE DE PRE-PUBLICACIONES

Complexity of limit cycle existence and feasibility problems in Boolean networks

Julio Aracena[a,b], Luis Gómez[a,*], Lilian Salinas[c,b]

[a] *Departamento de Ingeniería Matemática, Universidad de Concepción, Av. Esteban Iturra s/n, Casilla 160-C, Concepción, Chile.*
[b] *CI²MA*
[c] *Departamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Concepción, Edmundo Larenas 215, Piso 3, Concepción, Chile*

Abstract

Boolean networks have been used as models of gene regulation and other biological networks. One key element in the dynamical behavior of the networks are the limit cycles, which are very sensitive to changes in the update schedule used. In this paper we study two problems related to the inferring of update schemes and limit cycles in Boolean networks: Limit Cycle Existence problem and Feasible Limit Cycle problem. We explore in families of Boolean networks with different types of local activation function and structural properties of the interaction digraph to define the sharp delineation of the algorithmic complexity for both problems. We show that they are NP-Hard for different deterministic update schedules, even in AND-OR Boolean networks or with symmetric interaction digraph. However, they are polynomial problems in the case of verifying both conditions. As particular example of this, we prove that in the case of AND-OR networks with symmetric interaction digraph, there exists a limit cycle in a network iterated with a block-sequential update if and only if there exists a limit cycle with parallel scheme. This last condition is equivalent to a topological property on the network which can be verified in polynomial time.

Keywords: Boolean network, limit cycle, update schedule, NP-Hardness.

☆Partially supported by Project FONDECYT 1131013

*Corresponding author.

Email addresses: jaracena@ing-mat.udec.cl (Julio Aracena), lgomez@ing-mat.udec.cl (Luis Gómez), lilisalinas@udec.cl (Lilian Salinas)

1. Introduction

A Boolean network is a system of n interacting Boolean variables, which evolve, in a discrete time, according to a predefined rule. They have applications in many areas, including circuit theory, computer science and social systems (Green et al., 2007; Tocci and Widmer, 2001). In particular, from the seminal works of Kauffman (1969, 1993) and Thomas (Thomas, 1973; Schaefer, 1978), they are extensively used as models of gene networks. Despite their simplicity, they provide a useful model in which different phenomena can be reproduced and studied, and indeed, many regulatory models published in the biological literature fit within their framework (Huang, 1999; Shmulevich et al., 2003; Fauré et al., 2006; Bornholdt, 2008).

Since Boolean networks have a finite number of states, the long-run dynamic trajectories always reach a periodic sequence of states, called attractor. When the period is one, the attractor is said to be a fixed point, and when the period is greater than one, it is called limit cycle. In the modeling of genetic regulatory networks, the attractors are associated to distinct types of cells defined by patterns of gene activity. In particular, the limit cycles are often associated with mitotic cycles in cells. (Huang, 1999; Fauré et al., 2006).

The update schedule in a Boolean network, that is the order in which each node is updated, is of great importance in the dynamics of the network. In general and probably due to the difficulty of really knowing the order (if any) in which events take place in the cell, regulatory networks are usually studied with synchronous schedule (parallel scheme). Other types of deterministic update schedules, introduced by Robert (1986, 1995), and used in the discrete modeling of genetic regulatory networks (see Ruz et al. (2014); Goles et al. (2013); Meng and Feng (2014)) and other dynamical systems are: the sequential update (nodes are updated one by one in a prescribed order) and block-sequential updates (which are sequential over the sets of a partition, but parallel inside of each set).

Many theoretical and analytic studies have been done about the limit cycles of a Boolean network when different update schedules are used (Aracena et al., 2013; Demongeot et al., 2008; Goles and Noual, 2012; Elena, 2009; Macauley and Mortveit, 2009). Most of them show that the limit cycles are very sensitive to changes in the update schedule of the network, in opposition to the fixed points which do not depend on the scheme used. In particular, in Goles and Salinas (2008) is proved that for networks without negative loops it is not possible that the parallel and the sequential update share limit cycles.

One of the major problems in the understanding of the function of many biological complex systems, such as genetic networks or molecular signaling pathways, is the inferring of the network with an given update schedule from observed data, as for example a limit cycle. In this sense, the reconstruction of a genetic regulatory network has been so far done considering mainly synchronous update (see for example Shmulevich et al. (2002); Akutsu et al. (1999)). However, as mentioned above, there are limit cycles, under sequential or block-sequential schedules, which cannot be yielded with parallel update.

In this paper we study the Feasible Limit Cycle problem, which consists in given a Boolean network and a closed sequence of global configurations of the network, determining the existence of an deterministic update schedule (parallel, sequential or block-sequential) such that the sequence of configurations is a limit cycle of the network iterated with this scheme. A problem directly related to FLC is the Limit Cycle Existence problem, i.e., which consists in determining the existence of an update schedule, for a given Boolean network, such that it yields a limit cycle. The specific problem of determining the existence of limit cycles of a Boolean network with parallel update is known to be NP-Hard (Just, 2006). Here, we study this problem in the case of other kinds of update schedules (sequential and block-sequential).

We explore in families of Boolean networks with different types of local activation function and structural properties of the interaction digraph, to define the sharp delineation of the algorithmic complexity for both problems. We show that they are NP-Hard for the different deterministic update schedules, even in networks having either local activation function of type conjunctive (AND) or disjunctive (OR), or with symmetric interaction digraph. However, they are polynomial problems in the case of verifying both conditions. As particular example of this, we prove that in the case of AND-OR networks with symmetric interaction digraph, there exists a limit cycle in a network iterated with a block-sequential update if and only if there exists a limit cycle with parallel scheme. This last condition is equivalent to a topological property on the network which can be verified in polynomial time.

This paper is organized as follows. In Section 2, we introduce Boolean networks with deterministic update schedule and the basic concepts related to them. In Section 3, we study the complexity of the Limit Cycle Existence problem, and we prove that without restrictions on the interaction digraph, this problem is NP-Hard even in the case of AND-OR networks for any kind of update schedule (synchronous, sequential and block-sequential). In Section 4 we study the complexity of the Feasible Limit Cycle problem. We prove that for any interaction digraph, this problem is NP-Complete even in the case of Disjunctive networks. In Section 5 we give certain families of the networks in which each problem is polynomial, this classes are in AND-OR networks and depend strongly on the topology of the interaction digraph of the network. polynomial, this classes are in AND-OR networks and depend strongly on the topology of the interaction digraph of the network.

2. Definitions and Notation

Let V be a set of n elements. We denote by $F = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$ a function, where each component, $f_v : \{0,1\}^n \to \{0,1\}$ is a Boolean function, and such that $\forall x \in \{0,1\}^n$, $\forall v \in V : F(x)_v = f_v(x)$.

Given $x = (x_v)_{v \in V} \in \{0,1\}^n$ and $u \in V$, we define $\bar{x}^u \in \{0,1\}^n$ as:

$$\forall v \in V : \bar{x}^u_v = \begin{cases} x_v & \text{if } v \neq u \\ \neg x_u & \text{if } v = u \end{cases}$$

Where $\forall a \in \{0,1\} : \neg a = 1 \iff a = 0$.

We also define $\bar{x} \in \{0,1\}^n$ as: $\forall v \in V,\ \bar{x}_v = \neg x_v$.

A *Boolean network* $N = (F, s)$ is defined by a finite set V of n elements; n state variables $x_v \in \{0,1\},\ v \in V$; a function $F = (f_v)_{v \in V}$ called *global activation function*, where its component functions f_v are called *local activation functions*, and an *update schedule* defined by a function $s : V \to \{1, \ldots, n\}$ such that $s(V) = \{1, \ldots, m\}$ for some $m \leq n$. A *block* of an update schedule s is a set $B_i = \{v \in V : s(v) = i\}$, $1 \leq i \leq m$. An update schedule s is also denoted by $s = \{v \in B_1\}\{v \in B_2\} \cdots \{v \in B_m\}$. A *synchronous* or *parallel* update is given by an update schedule s such that $\forall v \in V,\ s(v) = 1$. A *sequential* update corresponds to a bijective function. Other kinds of update schedules can be considered as *block-sequential* updates. Block-sequential update schedules were introduced in Robert (1986).

The update of value states of the Boolean network with an update function s is given by:

$$x_v^{k+1} \quad = \quad f_v(x_u^{l_u} : u \in V)$$

where $l_u = k$ if $s(v) \leq s(u)$ and $l_u = k+1$ if $s(v) > s(u)$.

This is equivalent to applying a function $F^s : \{0,1\}^n \to \{0,1\}^n$ in a parallel way, with $F^s(x) = (f_v^s(x))_{v \in V}$ defined by:

$$f_v^s(x) = f_v(g_{v,u}^s(x) : u \in V),$$

where the function $g_{v,u}^s$ is defined by $g_{v,u}^s(x) = x_u$ if $s(v) \leq s(u)$ and $g_{v,u}^s(x) = f_u^s(x)$ if $s(v) > s(u)$. Thus, the function F^s corresponds to the dynamical behavior of the network $N = (F, s)$. We note that F^s was called Serial-Parallel operator in Robert (1986), and in the particular case of sequential updates it was called Gauss-Seidel operator.

We say that two Boolean networks $N_1 = (F_1, s_1)$ and $N_2 = (F_2, s_2)$ have the same dynamical behavior if $F_1^{s_1} = F_2^{s_2}$.

Since $\{0,1\}^n$ is a finite set, we have two limit behaviors for the iteration of a network:

- *Fixed Point.* We define a fixed point as $x \in \{0,1\}^n$ such that $F^s(x) = x$.

- *Limit Cycle.* We define a cycle of length $p > 1$ as the sequence $[x^k]_{k=0}^p = [x^0, \ldots, x^{p-1}, x^0]$ such that $x^k \in \{0,1\}^n$, x^k are pairwise distinct and $F^s(x^k) = x^{k+1}$, for all $k = 0, \ldots, p-1$ and $x^p \equiv x^0$. We note that any cyclic permutation of a sequence represents the same limit cycle. The set of limit cycles of N is denoted by $LC(N)$.

Fixed points and limit cycles are called *attractors* of the network.

We say that a node is *frozen* for a limit cycle if its state is constant on it.

Given a digraph G, the node set of G is referred to as $V(G)$, and its arc set as $A(G)$. Given a node $v \in V(G)$, the set of incoming nodes to v is denoted by $N_G^-(v) = \{u \in V(G) : (u,v) \in A(G)\}$. Analogously, the set of outgoing nodes from v is denoted by $N_G^+(v) = \{u \in V(G) : (v,u) \in A(G)\}$. An arc $(v,v) \in A(G)$ is called *loop* of G. Given $U \subseteq V(G)$, $G[U]$ is the digraph obtained from G by removing all nodes in $V(G) \setminus U$ and all arcs incoming to or outgoing from these nodes. $G[U]$ is called the *subdigraph generated* by U.

The digraph associated to a function $F = (f_v)_{v \in V}$, called *interaction digraph*, is the directed graph $G^F = (V, A)$, where $(u,v) \in A$ if and only if f_v

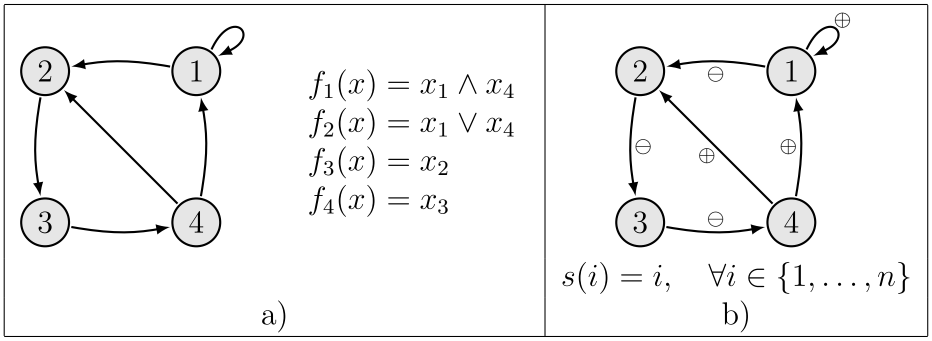


Figure 1: a) Digraph associated to a Boolean network. b) Update Digraph associated to a Boolean network and an update schedule.

depends on x_u, i.e., if there exists $x \in \{0,1\}^n$ such that $f_v(x) \neq f_v(\bar{x}^u)$. Note that if f_v is constant, then $N^-_{G^F}(v) = \emptyset$. See an example of an interaction digraph in Figure 1a.

Given $G = (V, A)$ a digraph with node set V of n elements and $s : V \to \{1, \dots, n\}$ an update schedule, we denote by $G_s = (G, \text{lab}_s)$ the labeled digraph, called *update digraph*, where the function $\text{lab}_s : A \to \{\ominus, \oplus\}$ is defined by:

$$\text{lab}_s(u, v) = \begin{cases} \oplus & \text{if } s(u) \geq s(v) \\ \ominus & \text{if } s(u) < s(v) \end{cases}$$

The update digraph associated to a Boolean network $N = (F, s)$ is defined by $G^F_s = (G^F, \text{lab}_s)$ (see an example of update digraph G_s in Figure 1b). Note that the label on a loop will always be $\oplus$. It was proven in Aracena et al. (2009) that if two different update schedules induce same update digraph, then they yield the same dynamical behavior.

Given a finite set U of k elements, we say that a Boolean function $f : \{0,1\}^k \to \{0,1\}$ is *monotonic* on input $v \in U$ if for every $x \in \{0,1\}^k$ such that $x_v = 0$, we have that $f(x) \leq f(\bar{x}^v)$. A loop $(v, v) \in A\left(G^F\right)$ is *monotonic* if f_v is monotonic on input v. In particular, a monotonic function f is said to be an *AND function*, denoted $f(x) = \bigwedge_{v \in U} x_v$, if and only if $f(x) = 1 \iff \forall v \in U : x_v = 1$. We say that a monotonic function f is an *OR function*, denoted $f(x) = \bigvee_{v \in U} x_v$, if and only if $f(x) = 1 \iff \exists v \in U : x_v = 1$.

In this way, we say that a function $F : \{0,1\}^n \to \{0,1\}^n$ is *monotonic* if each local activation function is monotonic. We say that F is an *AND-OR* *function* if each local activation function is either an AND or an OR function. In this case, we define $V_{\text{AND}}(F) \subseteq V\left(G^F\right)$ $\left(V_{\text{OR}}(F) \subseteq V\left(G^F\right)\right)$ as the nodes that have an AND (OR) local activation function. In particular, we say that F is an *OR function* if each local activation function is an OR function.

An AND-OR function F can be completely described by its interaction digraph, labeling AND and OR nodes differently (in the figures of this paper, white nodes represent OR nodes, and dark gray nodes represent AND nodes). That is, given $G = (V, A)$ a digraph and $\{V_{\text{AND}}, V_{\text{OR}}\}$ a partition of V, we define $F : \{0,1\}^{|V|} \to \{0,1\}^{|V|}$ as follows:

$$\forall v \in V : f_v(x) = \begin{cases} \bigwedge\limits_{u \in N_G^-(v)} x_u & \text{if } v \in V_{\text{AND}} \\ \bigvee\limits_{u \in N_G^-(v)} x_u & \text{if } v \in V_{\text{OR}} \end{cases}$$

Note that if $N_G^-(v) = \emptyset$, then

$$f_v(x) = \begin{cases} 1 & \text{if } v \in V_{\text{AND}} \\ 0 & \text{if } v \in V_{\text{OR}} \end{cases}$$

In this paper we are interested in the existence of update schedules which yields a given limit cycle. More precisely we define the following problem:

> Feasible Limit Cycle Problem (FLC): Given a set V of n elements and $F = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$ and a sequence $\mathcal{C} = \left[x^k\right]_{k=0}^p$ such that $x^k \in \{0,1\}^n$, x^k are pairwise distinct and $x^p \equiv x^0$. Does there exist an update schedule s such that $\mathcal{C} \in LC(F, s)$?

Previously, we study a well known problem in discrete dynamical systems synchronously updated, which consists of determining the existence of a limit cycle. This problem has been proven to be NP-Hard in different families of Boolean networks(cite). In this paper we extend these results in the case of deterministic update schedules different from parallel schedule.

> Limit Cycle Existence Problem (LCE): Given a set V of n elements and $F = (f_v)_{v \in V} : \{0,1\}^n \to \{0,1\}^n$. Does there exists an update schedule s such that $LC(F, s) \neq \emptyset$?

MON LCE, **AND-OR LCE** and **OR LCE** problems are the corresponding LCE problems when F is a monotonic, an AND-OR and an OR function, respectively. **MON FLC**, **AND-OR FLC** and **OR FLC** problems are defined analogously.

3. Limit cycle existence problem

In this section we study the complexity of deciding when there exists an update schedule that generates limit cycles when a given Boolean function is updated under it. A specific and directly related problem is to determine the existence of a limit cyle in a given Boolean network with synchronous schedule. This problem was proved to be NP-Hard even for AND-OR functions (Just, 2006). Here, we are interested in determing for a given Boolean network the existence of a deterministic update schedule that yields a limit cycle.

First, we prove that the general case of this problem is NP-Hard.

Theorem 1. *LCE is NP-Hard*

Proof. We show that SAT $\leq_p$ LCE.

Given a normal conjunctive formula (ncf) ϕ in variables $w_1, \ldots, w_n$, we consider $F = (f_v)_{v \in V} \colon \{0,1\}^{n+3} \to \{0,1\}^{n+3}$, where $V = \{v_1, \ldots, v_n, v_\phi, z_1, z_2\}$, as follows (see Figure 2):

$$\begin{array}{rcl} \forall i \in \{1, \ldots, n\}, \; f_{v_i}(x) & = & x_{v_i} \\ f_{v_\phi}(x) & = & \phi\left(x_{v_i} : i \in \{1, \ldots, n\}\right) \\ f_{z_1}(x) & = & x_{v_\phi} \wedge x_{z_2} \\ f_{z_2}(x) & = & x_{v_\phi} \wedge x_{z_1} \end{array}$$

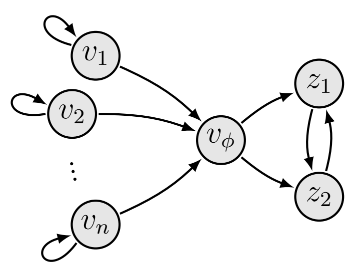


Figure 2: Connection digraph of the transformation defined in Theorem 1.

Then, we have:

($\Longrightarrow$) Let w be such that $\phi(w) = 1$. Then, if we consider the update schedule $s = \{v_1, \ldots, v_n, v_\phi\}\{z_1, z_2\}$, it is clear that $\mathcal{C} = [(w,1,0,1),(w,1,1,0),(w,1,0,1)] \in LC(F,s)$.
($\Longleftarrow$) Let us suppose that $\forall w\colon \phi(w) = 0$. Then, for every update schedule s, we have that:

- $\forall x \in \{0,1\}^{n+3}$, $\forall i \in \{1,\ldots,n\} : f^s_{v_i}(x) = x_{v_i}$.
- $\forall x \in \{0,1\}^{n+3}\colon f^s_{v_\phi}(x) = 0$. Therefore,
- $\forall x \in \{0,1\}^{n+3}\colon f^s_{z_i}(F^s(x)) = 0, i \in \{1,2\}$.

Thus, $LC(F,s) = \emptyset$, for every update schedule s. □

Now we prove that the LCE problem restricted to AND-OR functions is also NP-Hard.

Theorem 2. *AND-OR LCE is NP-Hard.*

Proof. We show that SAT $\leq_p$ AND-OR LCE.

Given a ncf ϕ in variables $w_1, \ldots, w_n$ with clauses $C_1, \ldots, C_m$ and let us define $F = (f_v)_{v\in V}\colon \{0,1\}^{4n+m+5} \to \{0,1\}^{4n+m+5}$ according to the following table:

$v \in V$	Type	$N^-_{G^F}(v)$
$v_i,\ i \in \{1,\ldots,n\}$	AND	$\{v_i\}$
$\bar{v}_i,\ i \in \{1,\ldots,n\}$	AND	$\{\bar{v}_i\}$
$o_i,\ i \in \{1,\ldots,n\}$	OR	$\{v_i, \bar{v}_i\}$
$a_i,\ i \in \{1,\ldots,n\}$	AND	$\{v_i, \bar{v}_i\}$
A	AND	$\{o_1, \cdots, o_n\}$
O	OR	$\{a_1, \cdots, a_n\}$
$v_{C_j},\ j \in \{1,\ldots,m\}$	OR	$\{v_i\colon w_i \in C_j\} \cup \{\bar{v}_i\colon \neg w_i \in C_j\}$
v_ϕ	AND	$\{v_{C_1}, \cdots, v_{C_m}\}$
z_1	AND	$\{z_2, v_\phi, A\}$
z_2	OR	$\{z_3, O\}$
z_3	OR	$\{z_1\}$

Table 1: Definition of F in the transformation defined in Theorem 2.

See G^F in Figure 3. Here, $\forall i \in \{1,\ldots,n\}$, nodes v_i represent literals w_i and nodes $\bar{v}_i$ represent literals $\neg w_i$.

Now, we note that:

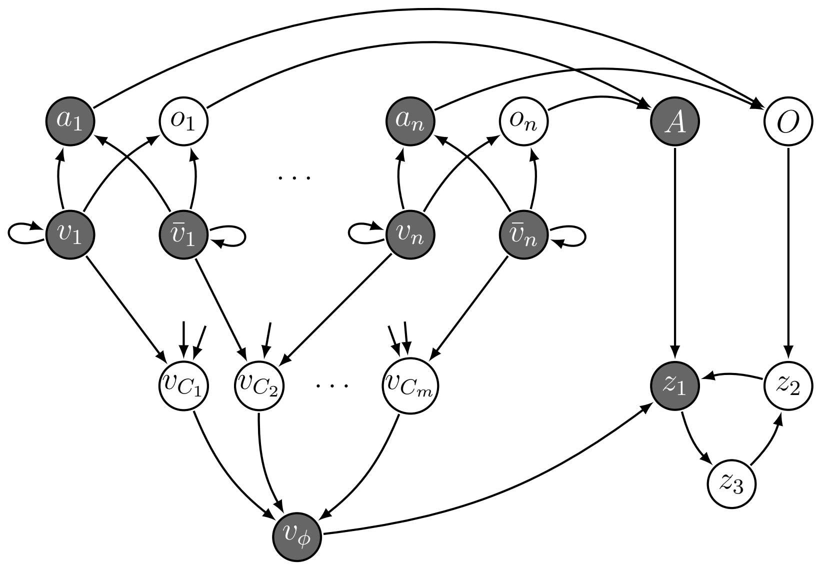


Figure 3: Connection digraph of the transformation defined in Theorem 2.

1. For any update schedule s, $\forall x^0 \in \{0,1\}^{4n+m+5}$:

   - $\forall k \geq 1,\ \forall i \in \{1,\dots,n\},\ \forall v \in \{v_i, \bar{v}_i\} : x_v^{k+1} = f_v^s(x^k) = x_v^0$
   - $\forall k \geq 2,\ \forall i \in \{1,\dots,n\},\ \forall v \in \{o_i, a_i\} : x_v^{k+1} = f_v^s(x^k) = x_v^0$
   - $\forall k \geq 2,\ \forall j \in \{1,\dots,m\} : x_{v_{C_j}}^{k+1} = f_{v_{C_j}}^s(x^k) = x_{v_{C_j}}^1$
   - $\forall k \geq 2,\ \forall v \in \{A, O, v_\phi, z_1, z_2, z_3\} : x_v^{k+1} = f_v^s(x^k) = x_v^2$

2. $f_A(x) = 1 \wedge f_O(x) = 0 \iff \forall i \in \{1,\dots,n\} : x_{\bar{v}_i} = \neg x_{v_i}$

($\Longrightarrow$) If $\exists \hat{w} \colon \phi(\hat{w}) = 1$, and we consider the update schedule s and the limit cycle $\mathcal{C} = [x^0, x^1, x^0]$ as described in the following table:

$v \in V$	v_i	$\bar{v}_i$	o_i	a_i	C_j	A	O	v_ϕ	z_1	z_2	z_3
$s(v)$	1	1	1	1	1	1	1	1	2	3	2
x_v^0	$\hat{w}_{v_i}$	$\neg\hat{w}_{v_i}$	1	0	1	1	0	1	1	0	0
x_v^1	$\hat{w}_{v_i}$	$\neg\hat{w}_{v_i}$	1	0	1	1	0	1	0	1	1

Clearly, $\mathcal{C} \in LC(F,s)$.

($\Longleftarrow$) Let s be an update schedule such that $\mathcal{C} = \left[x^k\right]_{k=0}^{p} \in LC(F, s)$.

For the first note above, only nodes z_1, z_2 or z_3 can cycle. For these nodes to cycle, it is necessary that:

$$f_{v_\phi}(x^0) = 1$$
$$f_A(x^0) = 1$$
$$f_O(x^0) = 0$$

From the first equation, we have that $\phi\left(x^0_{v_i}, x^0_{\bar{v}_i} : i \in \{1, \dots, n\}\right) = 1$. Second and third equations imply that $\forall i \in \{1, \dots, n\} : x_{\bar{v}_i} = \neg x_{v_i}$. Therefore, $\phi(x^0_i : i \in \{1, \dots, n\}) = 1$. □

Corollary 3. *AND-OR LCE is NP-Hard in the following cases:*

i.- Restricted to the parallel update schedule.

ii.- Restricted to sequential update schedules.

iii.- Restricted to limit cycles of length 2.

iv.- Restricted to maximum in-degree equal to 2.

Proof. i.- In this case we remove the vertex z_3 and we add an arc from z_1 to z_2.

ii.- It is easy to find a sequential udpate schedule equivalent to the update schedule in the proof of Theorem 2.

iii.- In the proof of Theorem 2 we restrict to limit cycles of length 2.

iv.- To see this, we just need to add intermediary nodes before every node that has in-degree greater than two as is exemplified in Figure 4. We note that the nodes that fulfill this condition are z_1, A, O, v_ϕ and the clause nodes. To simplify the transformation for clause nodes, we could consider 3-SAT instead of SAT. This transformation is enough because the only nodes that cycle are z_1, z_2 and z_3. □

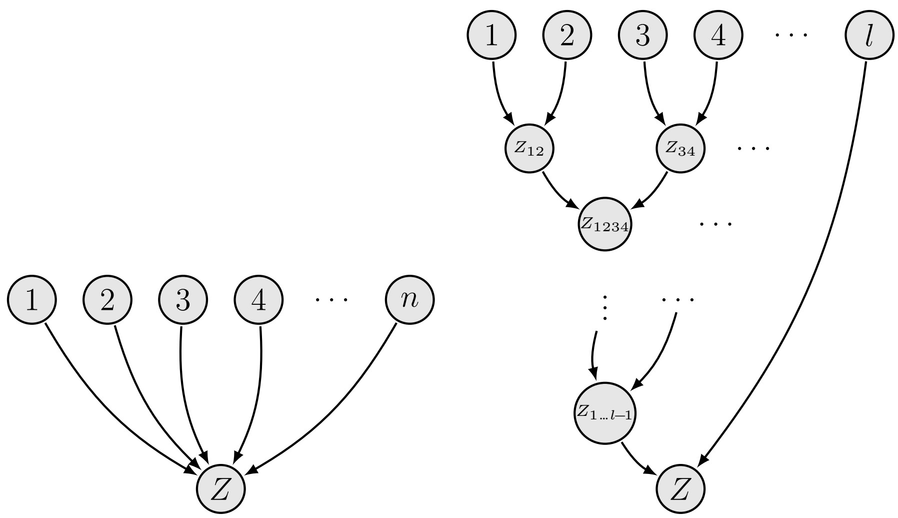


Figure 4: Example for odd l of the transformation mentioned in the proof of Corollary 3 to deal with nodes with in-degree in greater than two in Theorem 2.

4. Feasible limit cycle problem

In this section we study the complexity of determine the existence of an update schedule such that a given sequence of state vectors is a limit cycle for a given global activation function. We note that this problem gain importance when we consider several kind of update schedules because if it is restricted to the parallel update schedule is trivially polynomial.

Not-All-Equal Satisfiability (NAESAT) is a special case of the general satisfiability problem (SAT), which is defined as follows:

Not-All-Equal Satisfiability (NAESAT) Given ϕ a cnf in variables $w_1, \ldots, w_n$. Does there exist w such that $\phi(w) = 1$ and there is no clause in ϕ all literals of which are set to 1?

NAESAT is known to be NP-Complete Schaefer (1978).

Observe that NAESAT is equivalent to: given a cnf ϕ, does there exists w such that $\phi(w) = \phi(\bar{w}) = 1$?

First, we prove that feasible limit cycle problem is NP-Complete.

Theorem 4. *FLC is NP-Complete.*

Proof. It is clear that FLC is NP. To prove NP-Hardness we show that NAESAT $\leq_p$ FLC.

Given a 3-ncf ϕ in variables $w_1, \dots, w_n$, we consider $V = \{v_1, \dots, v_n, v_\phi\}$, $x^0 = (\vec{0}, 1)$, $x^1 = (\vec{1}, 1)$ and $\mathcal{C} = [x^0, x^1, x^0]$, where $\vec{0} = (0, \dots, 0)$, $\vec{1} = (1, \dots, 1) \in \{0,1\}^n$ and $F = (f_v)_{v \in V} \colon \{0,1\}^{n+1} \to \{0,1\}^{n+1}$ as follows:

$$\forall i \in \{1, \dots, n\} \colon f_{v_i}(x) = \neg x_{v_i}$$
$$f_{v_\phi}(x) = \phi(x_{v_i} : i \in \{1, \dots, n\})$$

See G^F in Figure 5.

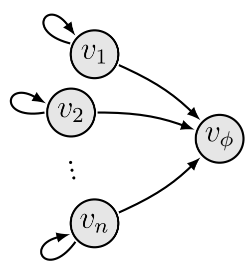


Figure 5: Interaction digraph of the transformation defined in Theorem 4.

($\Longrightarrow$) If there exists w such that $\phi(w) = \phi(\bar{w}) = 1$, then by defining $s = \{v_i \colon w_i = 1\}\{v_\phi\}\{v_i \colon w_i = 0\}$, it is clear that $\mathcal{C} \in LC(F, s)$.
($\Longleftarrow$) Let us suppose that there exists an update schedule s such that $\mathcal{C} \in LC(F, s)$. Then we define $w \in \{0,1\}^n$ such that $w_i = 1 \iff s(v_i) < s(v_\phi)$. It is easy to check that $\phi(w) = \phi(\bar{w}) = 1$. $\square$

Next, we show that the ideas of the previous proof can be extended to the Monotonic case.

Theorem 5. *MON FLC is NP-Complete.*

Proof. As in the general case, this problem is NP. To prove NP-Hardness, we show that SAT $\leq_p$ MON FLC. Given a ncf ϕ in variables $w_1, \dots, w_n$, we build $F = (f_v)_{v \in V} \colon \{0,1\}^{2n+3} \to \{0,1\}^{2n+3}$, $\mathcal{C} = [x^0, x^1, x^2 \equiv x^0]$ where $x^0, x^1 \in \{0,1\}^{2n+3}$ and $V = \{v_1 \dots, v_n, \bar{v}_1, \dots, \bar{v}_n, z_1, z_2, z_3\}$, as follows:

$$\forall i \in \{1, \dots, n\} \qquad f_{v_i}(x) = x_{\bar{v}_i} \wedge (x_{z_1} \vee x_{z_3})$$
$$\forall i \in \{1, \dots, n\} \qquad f_{\bar{v}_i}(x) = x_{v_i}$$
$$f_{z_1}(x) = \bigwedge_{i=1}^{n} x_{v_i}$$
$$f_{z_2}(x) = x_{z_1} \wedge \hat{\phi}\left(x_{v_i}, x_{\bar{v}_i} : i \in \{1 \dots, n\}\right)$$
$$f_{z_3}(x) = x_{z_2}$$

where nodes v_i represent literals w_i; nodes $\bar{v}_i$ represent literals $\neg w_i$ and $\hat{\phi}$ is the monotonic version of ϕ, in variables $x_{v_1}, \ldots, x_{v_n}, x_{\bar{v}_1}, \ldots, x_{\bar{v}_n}$, that comes from ϕ replacing literals w_i by x_{v_i} and $\neg w_i$ by $x_{\bar{v}_i}$ (see Figure 6). Finally, we define $x^1 = \overline{x^0}$ and:

$$x_u^0 = \begin{cases} 1 & \text{if } u \in \{v_1, \ldots, v_n\} \\ 0 & \text{if } u \in \{\bar{v}_1, \ldots, \bar{v}_n, z_1, z_2, z_3\} \end{cases}$$

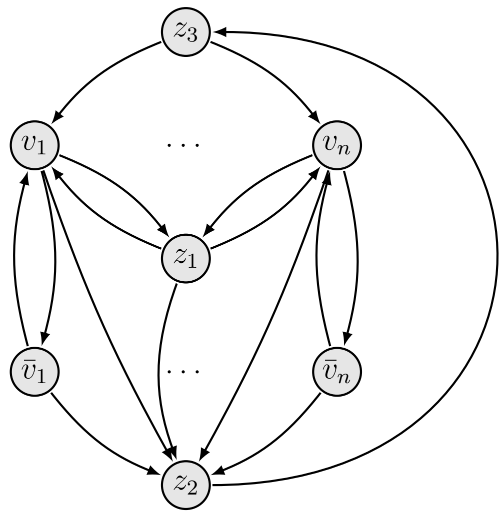


Figure 6: Interaction digraph of the transformation defined in Theorem 5.

The definition of F is similar than in Theorem 4, but monotonically. In order to achieve the monotony property, we add the $\bar{v}_i$ nodes that allow us to use $\hat{\phi}$ instead of ϕ, and the role of v_ϕ back there, that is to allow cycling, is monotonically done here by nodes z_1, z_2 and z_3.

($\Longrightarrow$) Let w be such that $\phi(w) = 1$, then if we consider the update schedule:

$$s = \{z_1\}\{v_i, \bar{v}_i \colon w_i = 0\}\{z_2\}\{v_i, \bar{v}_i \colon w_i = 1\}\{z_3\}$$

From Table 2, it is clear that $\forall k \in \{0,1\} : F^s(x^k) = x^{k+1}$ and $x^2 \equiv x^0$. Therefore, $\mathcal{C} \in LC(F, s)$.

($\Longleftarrow$) Let s be an update schedule such that $\forall k \in \{0,1\} : F^s(x^k) = x^{k+1}$ and let x^s be the global state just before node z_1 get updated. Since $1 = x^1_{z_2} = f^s_{z_1}(x^0) = f_{z_1}(x^s)$, we have that $\hat{\phi}(x^s_{v_i}, x^s_{\bar{v}_i} \colon i \in \{1, \ldots, n\}) = 1$. On another hand, we note that $\forall i \in \{1, \ldots, n\}$:

$v \in V$	$v_i : w_i = 1$	$v_i : w_i = 0$	$\bar{v}_i : w_i = 1$	$\bar{v}_i : w_i = 0$	z_1	z_2	z_3
$s(v)$	4	2	4	2	1	3	5
x_v^0	1	1	0	0	0	0	0
$x_v^1 = F^s(x^0)_v$	0	0	1	1	1	1	1
$x_v^0 = F^s(x^1)_v$	1	1	0	0	0	0	0

Table 2: Transition table of the states defined in Theorem 5.

1) $x_{v_i}^0 = 1, x_{v_i}^1 = 0, x_{\bar{v}_i}^0 = 0, x_{\bar{v}_i}^1 = 1 \implies s(\bar{v}_i) \leq s(v_i)$.

2) $x_{v_i}^1 = 0, x_{v_i}^0 = 1, x_{\bar{v}_i}^1 = 1, x_{\bar{v}_i}^0 = 0 \implies s(v_i) \leq s(\bar{v}_i)$.

3) Since v_i and $\bar{v}_i$ are connected by a cycle of length 2, necessarily

$$(s(v_i) \geq s(z_1) \wedge s(\bar{v}_i) \geq s(z_1)) \veebar (s(v_i) < s(z_1) \wedge s(\bar{v}_i) < s(z_1))$$

Thus, 1) and 2) imply that $\forall i \in \{1, \dots, n\} : s(v_i) = s(\bar{v}_i)$ and then $\forall i \in \{1, \dots, n\}, \forall k \in \{0, 1\} : x_{\bar{v}_i}^k = \neg x_{v_i}^k$. From this and 3), we have that $\forall i \in \{1, \dots, n\}, \forall k \in \{0, 1\} : x_{\bar{v}_i}^s = \neg x_{v_i}^s$.

Therefore, $\phi(\hat{x}) = 1$, with $\hat{x} = \left(x_{v_i}^s\right)_{i=1}^n$. □

To prove the OR case, we need a completely different approach, since above ideas sufficient when we are restricted to OR functions. First we need some previous results. In Lemma 6 we prove the SAT variation we are going to use and Lemma 7 is a technical result.

We define:

SAT_{01}: Given ϕ a ncf such that $\phi(\vec{0}) = \phi(\vec{1}) = 1$. Does there exists $x \notin \left\{\vec{0}, \vec{1}\right\}$ such that $\phi(x) = 1$?

Lemma 6. *SAT_{01} is NP-Complete.*

Proof. It is clear that SAT_{01} is NP. To prove NP-Hardness, we show that $\text{SAT} \leq_p \text{SAT}_{01}$.

Let ϕ be a ncf in variables $x_1 \dots, x_n$ and clauses $C_1, \dots, C_n$. We define $\hat{\phi}$ a ncf as follows:

$$\hat{\phi}(x) = \begin{cases} \bigwedge\limits_{j=1}^{m} \bigwedge\limits_{\substack{i,k=1 \\ i \neq k}}^{n} (C_j \vee \neg x_i \vee x_k) & \text{if } \phi(\vec{0}) = \phi(\vec{1}) = 0 \\ x_1 \vee \neg x_2 & \text{if } \phi(\vec{0}) = 1 \vee \phi(\vec{1}) = 1 \end{cases}$$

Clearly, $\hat{\phi}(\vec{0}) = \hat{\phi}(\vec{1}) = 1$.
($\Longrightarrow$) Let $x \in \{0,1\}^n$ be such that $\phi(x) = 1$. Hence,

- If $x \in \left\{\vec{0}, \vec{1}\right\}$, then $\hat{\phi}(x) = x_1 \vee \neg x_2$ and considering $\hat{x} = (1,0)$ we have that $\hat{\phi}(\hat{x}) = 1$.
- If $x \notin \left\{\vec{0}, \vec{1}\right\}$, then considering $\hat{x} = x$ we have that $\hat{\phi}(\hat{x}) = 1$.

($\Longleftarrow$) Let $\hat{x}$ be such that $\hat{\phi}\left(\hat{x}\right) = 1$ and let us suppose that $\forall x\colon \phi(x) = 0$, then there exist $j \in \{1, \ldots, m\}$ such that $C_j\left(\hat{x}\right) = 0$.

Thus, $\forall i \neq k \in \{1, \ldots, n\} : C_j\left(\hat{x}\right) \vee \neg \hat{x}_i \vee \hat{x}_k = \neg \hat{x}_i \vee \hat{x}_k = 1$.

Now, if there exists $k \in \{1, \ldots, n\}$ such that $\hat{x}_k = 0$, then for each $i \neq k \in \{1, \ldots, n\}$ we have that $\hat{x}_i = 0$ and therefore, $\hat{x} = \vec{0}$. Otherwise, $\hat{x} = \vec{1}$.

Analogously, if there exists $i \in \{1, \ldots, n\}$ such that $\hat{x}_i = 1$, then for each $k \neq i \in \{1, \ldots, n\}$ we have that $\hat{x}_k = 1$ and therefore, $\hat{x} = \vec{1}$. Otherwise, $\hat{x} = \vec{0}$.

Thus, $\hat{\phi}$ is only satisfiable by $\vec{0}$ and $\vec{1}$. $\square$

Definition 1. *Given $F : \{0,1\}^n \to \{0,1\}^n$, $x, y \in \{0,1\}^n$, we define for each $q, r \in \{0,1\}$:*

$$V_{qr}\left(x,y\right) := \left\{v \in V\left(G^F\right) : x_v = q \ \wedge \ y_v = r\right\}$$

And the set of constant nodes:

$$V_c\left(x,y\right) = V_{00}\left(x,y\right) \cup V_{11}\left(x,y\right)$$

When there is no confusion, we will ignore the argument (x, y) in the previous definitions.

Lemma 7. *Let $F : \{0,1\}^n \to \{0,1\}^n$ be an AND-OR function, $x, y \in \{0,1\}^n$ and s an update schedule. If $F^s(x) = y$ then, for each $v \in V\left(G^F\right)$ we have that:*

1.- If $v \in V_{OR}\left(F\right)$ and

i.- $v \in V_{10} \cup V_{00}$, then $\forall u \in N^-_{G^F}(v)$:

$$(u \in V_{01} \wedge s(u) \geq s(v)) \vee (u \in V_{10} \wedge s(u) < s(v)) \vee u \in V_{00}$$

ii.- $v \in V_{01} \cup V_{11}$*, then* $\exists u \in N^-_{G^F}(v)$:

$$(u \in V_{01} \wedge s(u) < s(v)) \vee (u \in V_{10} \wedge s(u) \geq s(v)) \vee u \in V_{11}$$

2.- If $v \in V_{AND}(F)$ *and*

i.- $v \in V_{01} \cup V_{11}$*, then* $\forall u \in N^-_{G^F}(v)$:

$$(u \in V_{10} \wedge s(u) \geq s(v)) \vee (u \in V_{01} \wedge s(u) < s(v)) \vee u \in V_{11}$$

ii.- $v \in V_{10} \cup V_{00}$*, then* $\exists u \in N^-_{G^F}(v)$:

$$(u \in V_{10} \wedge s(u) < s(v)) \vee (u \in V_{01} \wedge s(u) \geq s(v)) \vee u \in V_{00}$$

Proof. Let $v \in V\left(G^F\right)$.

1.- If $v \in V_{\mathrm{OR}}(F)$,

i.- Let us suppose $v \in V_{10} \cup V_{00}$ and let $u \in N^-_{G^F}(v)$. Since $f^s_v(x) = 0$, necessarily $x_u = 0 \vee y_u = 0$. Now:

- If $u \in V_{01}$, then $x_u = 0 \wedge y_u = 1$. Since $f^s_v(x) = 0$, it is necessary that $s(u) \geq s(v)$.
- If $u \in V_{10}$, then $x_u = 1 \wedge y_u = 0$. Since $f^s_v(x) = 0$, it is necessary that $s(u) < s(v)$.
- Otherwise, j must necessarily be in V_{00}.

ii.- Straightforward from the definition of OR functions and analogous argument as before.

2.- If $v \in V_{\mathrm{AND}}(F)$, the proof is straightforward from the definition of AND functions and analogous argument as before. □

Remark 1. *We know from Aracena et al. (2009) that Boolean networks updated under different updates schedules that generate the same update digraph have the same dynamical behavior. Therefore, we focus on finding an update digraph wich satisfies certain restrictions. In this way, according to the definition of an update digraph and to the established in the previous lemma, we have that for OR nodes (AND nodes), all incoming arcs of the nodes in* $V_{00} \cup V_{10}$ *(*$V_{11} \cup V_{01}$*) have their labels uniquely defined. To satisfy the necessary conditions such that* $F^s(x) = y$*, at least one incoming arc to the nodes in* $V_{11} \cup V_{01}$ *(*$V_{00} \cup V_{10}$*) must be chosen and labeled accordingly. It is in this choice where the complexity of the problem arises.*

Now we prove that OR FLC is NP-Complete.

Theorem 8. *OR FLC is NP-Complete.*

PROOF. We prove that $\mathrm{SAT}_{01} \leq_p$ OR FLC.

Let ϕ be ncf in variables $w_1, \ldots, w_n$ with clauses $C_0, \ldots, C_{m-1}$ such that $\phi(\vec{0}) = \phi(\vec{1}) = 1$.

We define an OR function F and a limit cycle $\mathcal{C}$ such that each variable w_i is represented by a node $v_i \in V(G^F)$ and whose value is defined according to the relative order of schedule between node v_i and a given node v_ϕ. Besides, each clause of ϕ is associated to a transitions in the limit cycle $\mathcal{C}$.

More precisely, we define $F = (f_v)_{v\in V} \colon \{0,1\}^{3m+n+4} \to \{0,1\}^{3m+n+4}$ an OR function by its interaction digraph defined in Table 3 (see an example in Figure 7).

$\boldsymbol{v \in V}$	$\boldsymbol{N^-_{G^F}(v)}$
$v_i,\ i \in \{1, \ldots, n\}$	$\{z_0\} \cup \left\{C_j^1 \colon w_i \in C_j\right\} \cup \left\{C_j^2 \colon \neg w_i \in C_j\right\}$
$C_j^k,\ j \in \{0, \ldots, m-1\},\ k \in \{1, 2\}$	$\left\{C_j^{k-1}\right\}$
$C_j^0,\ j \in \{0, \ldots, m-1\}$	$\left\{C^2_{j-1 \bmod m}\right\}$
$z_k,\ k \in \{0, 1, 2\}$	$\left\{z_{k-1 \bmod 3}\right\}$
v_ϕ	$\{v_1, \ldots, v_n\}$

Table 3: Definition of G^F defined in Theorem 8.

And, we define $\mathcal{C} = [x^{0,0}, x^{1,0}, x^{2,0}, x^{0,1}, \ldots] = \left[x^{k,j}\right]_{k\in\mathbb{Z}_3, j\in\mathbb{Z}_m}$ of length $3m$ as:

$$
\begin{aligned}
x^{k,j}_{v_i} &= \begin{cases} 1 & \text{if } k = 0 \vee (k = 1 \wedge w_i \in C_j) \vee (k = 2 \wedge \neg w_i \in C_j) \\ 0 & \text{otherwise} \end{cases} \\
x^{k,j}_{C^{k'}_{j'}} &= \begin{cases} 1 & \text{if } j = j' \wedge k = k' \\ 0 & \text{otherwise} \end{cases} \\
x^{k,j}_{z_{k'}} &= \begin{cases} 1 & \text{if } k = k' \\ 0 & \text{otherwise} \end{cases} \\
x^{k,j}_{v_\phi} &= 1.
\end{aligned}
$$

See an example in Table 4.

In this way, each clause C_j in ϕ is represented by the vectors $x^{1,j}$ and $x^{2,j}$ such that: $x^{1,j}_{v_i} = 1$ and $x^{2,j}_{v_i} = 0$ if the literal w_i is in C_j; $x^{1,j}_{v_i} = 0$ and $x^{2,j}_{v_i} = 1$ if $\neg w_i$ is in C_j, and $x^{1,j}_{v_i} = 0$ and $x^{2,j}_{v_i} = 0$ otherwise. Hence, for all

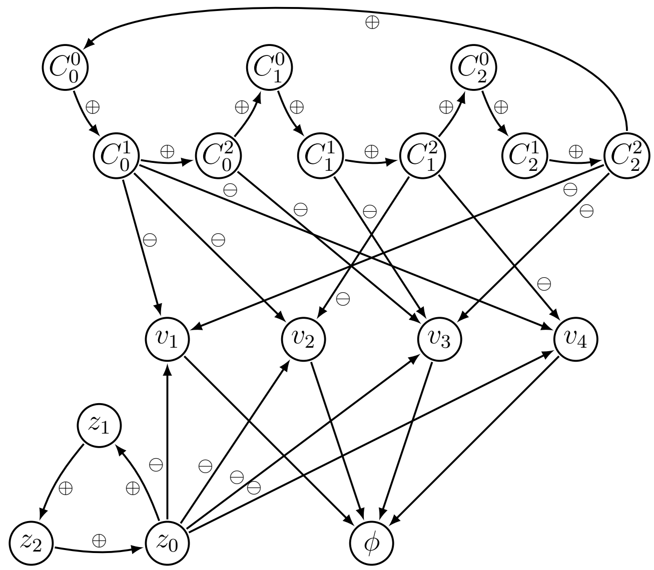


Figure 7: Example of G^F for $\phi(w) = (w_1 \vee w_2 \vee \neg w_3 \vee w_4) \wedge (\neg w_2 \vee w_3 \vee \neg w_4) \wedge (\neg w_1 \vee \neg w_3)$ according to the transformation defined in Theorem 8.

$j \in \{0, \ldots, m-1\}$, $x_{v_\phi}^{2,j} = 1$ if and only if there exists $i \in \{1, \ldots, n\}$ such that either $x_{v_i}^{1,j} = 1$, $x_{v_i}^{2,j} = 0$ and $s(v_i) \geq s(v_\phi)$ or $x_{v_i}^{1,j} = 0$, $x_{v_i}^{2,j} = 1$ and $s(v_i) < s(v_\phi)$. Therefore, we obtain an equivalence between the relative order of nodes v_i and v_ϕ, and the value of the variable w_i as follows:

$$s(v_i) \geq s(v_\phi) \iff w_i = 1 \tag{1}$$

In this way, variable v_ϕ remains frozen with value equal to one if and only if all clauses are satisfiable.

Besides, we note that, by Lemma 7 applied to each transition in C, we have that for every update schedule s such that $\mathcal{C} \in LC(F, s)$:

$$\forall k \in \{1,2\},\ \forall j \in \{0,\ldots,m-1\}: \quad s\left(C_j^{k-1}\right) \geq s\left(C_j^k\right) \tag{2}$$

$$\forall j \in \{0,\ldots,m-1\}: \quad s\left(C_{j-1 \bmod m}^2\right) \geq s\left(C_j^0\right) \tag{3}$$

$$\forall k \in \{0,1,2\}: \quad s\left(z_{k-1 \bmod 3}\right) \geq s\left(z_k\right) \tag{4}$$

$$\forall i \in \{1,\ldots,n\}: \quad s\left(z_0\right) < s\left(v_i\right) \tag{5}$$

$$\forall i \in \{1,\ldots,n\},\ \forall k \in \{1,2\},\ \forall j \in \left\{t \colon C_t^k \in N_{G^F}^-(i)\right\}: \quad s\left(C_j^k\right) < s\left(v_i\right) \tag{6}$$

We note that conditions (2)–(6) define uniquely the labels of the arcs involved as shown in Figure 7.

	C_0^0	C_0^1	C_0^2	C_1^0	C_1^1	C_1^2	C_2^0	C_2^1	C_2^2	z_0	z_1	z_2	v_1	v_2	v_3	v_4	v_ϕ
$x^{0,0}$	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1
$x^{1,0}$	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1
$x^{2,0}$	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1
$x^{0,1}$	0	0	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1
$x^{1,1}$	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1
$x^{2,1}$	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	1
$x^{0,2}$	0	0	0	0	0	0	1	0	0	1	0	0	1	1	1	1	1
$x^{1,2}$	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1
$x^{2,2}$	0	0	0	0	0	0	0	0	1	0	0	1	1	0	1	0	1
$x^{0,3}$	1	0	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1

Table 4: Example of $\mathcal{C}$ for $\phi(w) = (w_1 \vee w_2 \vee \neg w_3 \vee w_4) \wedge (\neg w_2 \vee w_3 \vee \neg w_4) \wedge (\neg w_1 \vee \neg w_3)$ according to the transformation defined in Theorem 8.

We show now the details of the proof:

($\Longrightarrow$) Let $w \notin \left\{\vec{0}, \vec{1}\right\}$ be such that $\phi(w) = 1$, then we define the update schedule s as

$$s = B_1\, B_2\, B_3\, B_4\, B_5,$$

where

$$\begin{aligned}
B_1 &= \left\{C_j^k : j \in \{0, \ldots, m-1\},\ k \in \{0,1,2\}\right\}, \\
B_2 &= \left\{z_k : k \in \{0,1,2\}\right\}, \\
B_3 &= \left\{v_i : w_i = 0\right\}, \\
B_4 &= \left\{v_\phi\right\}, \\
B_5 &= \left\{v_i : w_i = 1\right\}.
\end{aligned}$$

It is easy to see that s satisfies conditions (1)–(6) and for all $v \in V \setminus \{v_\phi\}$,

$$x_v^{k,j} = \begin{cases} f_v^s(x^{2,j-1 \bmod m}) & \text{if } k = 0 \\ f_v^s(x^{k-1,j}) & \text{otherwise} \end{cases}$$

Also by condition (1) and by definition of F and $\mathcal{C}$ we have that:

$$f_{v_\phi}^{s}\left(x^{k,j}\right)=\begin{cases}\bigvee\limits_{\left\{i:s(v_i)<\left(v_\phi\right)\right\}} x_{v_i}^{0,j} \vee \bigvee\limits_{\left\{i:s(v_i)\geq\left(v_\phi\right)\right\}} x_{v_i}^{2,j-1 \bmod m} & \text{if } k=0\\ \bigvee\limits_{\left\{i:s(v_i)<\left(v_\phi\right)\right\}} x_{v_i}^{1,j} \vee \bigvee\limits_{\left\{i:s(v_i)\geq\left(v_\phi\right)\right\}} x_{v_i}^{0,j} & \text{if } k=1\\ \bigvee\limits_{\left\{i:s(v_i)<\left(v_\phi\right)\right\}} x_{v_i}^{2,j} \vee \bigvee\limits_{\left\{i:s(v_i)\geq\left(v_\phi\right)\right\}} x_{v_i}^{1,j} & \text{if } k=2\end{cases}$$

Moreover,

$$\begin{aligned} w \neq \vec{1} &\Longrightarrow \bigvee_{\left\{i:s(v_i)<\left(v_\phi\right)\right\}} x_{v_i}^{0,j}=1, \text{ and}\\ w \neq \vec{0} &\Longrightarrow \bigvee_{\left\{i:s(v_i)\geq\left(v_\phi\right)\right\}} x_{v_i}^{0,j}=1, \text{ and}\\ \phi(w)=1 &\Longrightarrow \bigvee_{\left\{i:s(v_i)<\left(v_\phi\right)\right\}} x_{v_i}^{2,j} \vee \bigvee_{\left\{i:s(v_i)\geq\left(v_\phi\right)\right\}} x_{v_i}^{1,j}=1. \end{aligned}$$

Hence $\forall j \in \{0,\ldots,m-1\}$, $\forall k \in \{0,1,2\} : f_{v_\phi}^{s}\left(x^{k,j}\right)=1$. Therefore, $\mathcal{C} \in LC\left(F,s\right)$.

($\Longleftarrow$) Let s be an update schedule such that $\mathcal{C} \in LC(F,s)$. Then, by definition of $\mathcal{C}$, conditions (2)–(6) are satisfied and we define w by condition (1). Clearly, $\phi(w)=1$.

Since $\phi(\vec{0})=\phi(\vec{1})=1$, then following configurations do not appear in $\mathcal{C}$:

$x^{k,j}$	v_1	$\cdots$	v_n	v_ϕ
$x^{0,j}$	1	$\vec{1}$	1	1
$x^{1,j}$	1	$\vec{1}$	1	1
$x^{2,j}$	0	$\vec{0}$	0	1
$x^{0,j+1}$	1	$\vec{1}$	1	1

$x^{k,j'}$	v_1	$\cdots$	v_n	v_ϕ
$x^{0,j'}$	1	$\vec{1}$	1	1
$x^{1,j'}$	0	$\vec{0}$	0	1
$x^{2,j'}$	1	$\vec{1}$	1	1
$x^{0,j'+1}$	1	$\vec{1}$	1	1

Thus, $\exists i,j \in \{1,\ldots,n\} : s\left(v_i\right) \geq s\left(v_\phi\right) \wedge s\left(v_j\right) < s\left(v_\phi\right)$ and therefore $w \notin \left\{\vec{0},\vec{1}\right\}$. $\square$

Remark 2. *Note that the proof of the previous theorem can be modified to prove that OR FLC is NP-Complete restricted to sequential update schedules. We just need to add two extra nodes in the digraph: a node a to the cycle with nodes C_j^k and a node b to the cycle with nodes z_k. In this way, the limit cycle $\mathcal{C}$ will be $x_b^{k,j} = x_{z_0}^{k,j}$ and $x_a^{k,j} = x_{C_0^0}^{k,j}$ in the new nodes. From Lemma 7 we deduce conditions about the update schedule compatible with a sequential update.*

5. Polynomial cases

In this Section we show some classes of Boolean networks in which LCE and FLC are polynomial. These classes share the property of having symmetric interaction digraph. However, this is condition is not sufficient as shown in the result below:

Proposition 9. *Symmetric-LCE and Symmetric-FLC are NP-Hard.*

Proof. For Symmetric LCE the proof is similar to that of Theorem 1, with the same limit cycle $\mathcal{C}$, but changing the local activation functions by the following:

$$\begin{aligned}
\forall i \in \{1, \ldots, n\}, \ f_{v_i}(x) &= x_{v_i} \wedge x_{v_\phi} \\
f_{v_\phi}(x) &= \phi\left(x_{v_i} : i \in \{1, \ldots, n\}\right) \wedge (x_{z_1} \vee x_{z_2}) \\
f_{z_1}(x) &= x_{v_\phi} \wedge x_{z_2} \\
f_{z_2}(x) &= x_{v_\phi} \wedge x_{z_1}
\end{aligned}$$

In this case G^F is symmetric as shown in Figure 8

For Symmetric FLC we keep the limit cycle of Theorem 4 but we change the local activation functions by the following:

$$\forall i \in \{1, \ldots, n\} \colon f_{v_i}(x) = \neg x_{v_i} \wedge x_{v_\phi}$$
$$f_{v_\phi}(x) = \phi(x_{v_i} : i \in \{1, \ldots, n\})$$

In this way G^F is symmetric, and the proof is similar to Theorem 4. $\square$

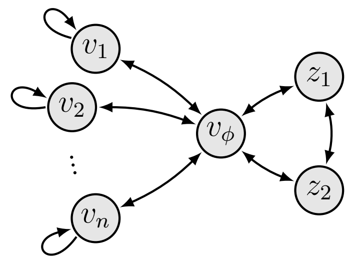


Figure 8: Connection digraph of the transformation defined in Proposition 9.

5.1. Limit cycle existence problem

We deal here with LCE problem. We will show that LCE is polynomial in the case of symmetric AND-OR function. The LCE problem here will be referred as SYMMETRIC AND-OR LCE.

First, OR networks were studied in Goles and Noual (2012), where, in our wording, was shown the following result:

Proposition 10. *Let F be an OR function with symmetric G^F. Then, for all $s \neq s^p$ $LC(F,s) = \emptyset$. Furthermore, $LC(F,s^p) \neq \emptyset$ if and only if G^F is bipartite. In this case, all limit cycles are of length 2.*

It is clear that the condition in the above proposition can be tested in polynomial time and therefore SYMMETRIC OR LCE is polynomial. We are now extending this result to AND-OR functions, but first we need some previous definitions.

Definition 2. *Given F an AND-OR function with symmetric G^F.*

- *We denote each non trivial connected component of $G\left[V_{OR}(F)\right]$ by $G_1^{OR}, \ldots, G_{k_{OR}}^{OR}$. We call them OR components of G^F.*
- *We denote each non trivial connected component of $G\left[V_{AND}(F)\right]$ by $G_1^{AND}, \ldots, G_{k_{AND}}^{AND}$. We call them AND components of G^F.*
- *We define the alternated nodes as*

$$V_{AO} = V \setminus \left(\bigcup_{i=1}^{k_{OR}} V\left(G_i^{OR}\right) \cup \bigcup_{i=1}^{k_{AND}} V\left(G_i^{AND}\right) \right)$$

and we denote by $G_1^{AO}, \ldots, G_{k_{AO}}^{AO}$, to the connected component of $G\left[V_{AO}\right]$. We call them alternated components of G^F.

- *We call to the set* $\left\{G_1^{OR}, \dots, G_{k_{OR}}^{OR}, G_1^{AND}, \dots, G_{k_{AND}}^{AND}, G_1^{AO}, \dots, G_{k_{AO}}^{AO}\right\}$, *an AOA (AND-OR ALTERNATED) decomposition of* G^F.

Remark 3.

1. *The set* $\left\{V\left(G_1^{OR}\right), \dots, V\left(G_{k_{OR}}^{OR}\right), V\left(G_1^{AND}\right), \dots, V\left(G_{k_{AND}}^{AND}\right), V\left(G_1^{AO}\right), \dots, V\left(G_{k_{AO}}^{AO}\right)\right\}$ *is a partition of* $V(G^F)$.

2. *Given* $i \in \{1, \dots, k_{AO}\}$, *we note that* $\forall u \in V\left(G_i^{AO}\right)$:

$$\begin{aligned} u \in V_{OR} &\implies N^-_{G^F}(u) \subseteq V_{AND} \\ u \in V_{AND} &\implies N^-_{G^F}(u) \subseteq V_{OR} \end{aligned}$$

Therefore, the non trivial alternate components of G^F *are bipartite.*

Next Lemma shows that every non bipartite AND or OR component of G^F is frozen in any limit cycle.

Lemma 11. *Given* F *an AND-OR function with symmetric* G^F, $\mathcal{C} \in LC(F, s^p)$ *and* D *an OR or an AND component of* G^F. *If* D *is non bipartite, then every node in* $V(D)$ *is frozen in* $\mathcal{C}$.

Proof. Let $\mathcal{C} = \left[x^k\right]_{k=0}^p \in LC(F, s^p)$ and D be a non bipartite OR component of G^F (the AND case is analogous). Then, there exists cycle of vertices $C = v_1 \dots v_{2N+1} v_1$ in D (cycle of odd length).

Observe that if there exists a path of length l from a vertex u to a vertex v and $x_u^t = 1$ then $x_v^{t+l} = 1$. Hence, for all vertex $v_i \in V(C)$, $v_i^k = 1 \Rightarrow v_i^{k+2N+1} = 1$. Besides, since G^F is symmetric, for all $v \in V(D)$, and for all $k \in \{0, \dots, p-1\}$, $x_v^k = 1 \Rightarrow x_v^{k+2} = 1$. Therefore, if there exists $v \in V(C)$ and $k \in \{0, \dots, p-1\}$ such that $x_v^k = 1$, then for all $v \in V(C)$ and for all $k \in \{0, \dots, p-1\}$, $x_v^k = 1$. Thus, every vertex in the cycle C is frozen. Finally, since G^F is strongly connected, the result holds. $\square$

Observe that, the neighbors vertices of $V(D)$ are not involved in the property of every node in $V(D)$ is frozen in $\mathcal{C}$, but in the value of the vertices of $V(D)$ in $\mathcal{C}$.

Next Theorem give a polynomial testable characterization of when a symmetric AND-OR function can generate limit cycles.

Proposition 12. *Given F an AND-OR function with symmetric G^F. Then, $LC(F, s^p) \neq \emptyset$ if and only if there exists a bipartite element in the AOA decomposition of G^F.*

Proof. ($\Longrightarrow$) Let $\mathcal{C} \in LC(F, s^p)$. We have two cases:

1.- There exists an OR or an AND bipartite component of G^F, and therefore the result holds.

2.- All OR and AND components of G^F are non bipartite. By Lemma 11, all of them are frozen in $\mathcal{C}$. Thus, there exists an alternated component of G^F.

If every alternated component of G^F is trivial, then each one of them has only frozen incoming neighbors, and therefore are also frozen. Then, there exists a non trivial alternated component of G^F which, by Remark 3, is bipartite.

($\Longleftarrow$) Let D be a bipartite element in the AOA decomposition of G^F.

1.- Let us suppose that D is an alternated connected component of G^F. We note that

$$\forall v \in V_{\text{OR}} \cap V(D) : N^-_{G^F}(v) \subseteq \bigcup_{i=1}^{k_{\text{AND}}} V\left(G_i^{\text{AND}}\right)$$
$$\forall v \in V_{\text{AND}} \cap V(D) : N^-_{G^F}(v) \subseteq \bigcup_{i=1}^{k_{\text{OR}}} V\left(G_i^{\text{OR}}\right)$$

If we consider $\mathcal{C} = [x^0, x^1, x^0]$, defined according to the following table:

$\boldsymbol{v \in V\left(G^F\right)}$	$\boldsymbol{V_{\text{OR}} \cap V(D)}$	$\boldsymbol{V_{\text{AND}} \cap V(D)}$	$\boldsymbol{V\left(G^F\right) \setminus V(D)}$
$\boldsymbol{x_v^0}$	1	0	0
$\boldsymbol{x_v^1}$	0	1	0

Table 5: Limit cycle from Proposition 12 if an alternated component of G^F is considered.

it is clear that $\mathcal{C} \in LC(F, s^p)$.

2.- Let us suppose that D is an OR component of G^F.

Let us denote by D^1 and D^2 the two sets of the bi-partition of D, and by V_T the set that contains the nodes of all the trivial alternated components of G^F. We define the following sets:

$$V_T^1 = \left\{ v \in V_T : N^-_{G^F}(v) \subseteq V(D^1) \right\} \subseteq V_{\text{AND}}$$
$$V_T^2 = \left\{ v \in V_T : N^-_{G^F}(v) \subseteq V(D^2) \right\}$$

Now, we define $\mathcal{C} = [x^0, x^1, x^0]$ according to the following table:

$\boldsymbol{v \in V\left(G^F\right)}$	$\boldsymbol{V\left(D^1\right) \cup V_T^1}$	$\boldsymbol{V\left(D^2\right) \cup V_T^2}$	$\boldsymbol{V\left(G^F\right) \setminus \left(V(D) \cup V_T^1 \cup V_T^2\right)}$
$\boldsymbol{x_v^0}$	1	0	0
$\boldsymbol{x_v^1}$	0	1	0

Table 6: Limit cycle from Proposition 12 if a bipartite OR component of G^F is considered.

It is clear that $\mathcal{C} \in LC(F, s^p)$.

3.- If D is an AND component of G^F, then the proof is analogous that in the OR case.

Proposition 13. *Given F an AND-OR function with symmetric G^F. If $LC(F, s^p) = \emptyset$, then for every update schedule $s \neq s^p$, $LC(F, s) = \emptyset$.*

Proof. We prove that if $LC(F, s^p) = \emptyset$, then for each update schedule $s \neq s^p$, $LC(F, s) = \emptyset$.

Let $s \neq s^p$ be an update schedule. We note that, since G^F is symmetric, then $s \neq s^p$ if an only if there exists $(u, v) \in A\left(G^F\right)$ such that $s(u) < s(v)$.

Let $\mathcal{C} = \left[x^k\right]_{k=0}^{p} \in LC(F, s)$. Since $LC(F, s^p) = \emptyset$, then all elements in the AOA decomposition of G^F are not bipartite, by Proposition 12. Therefore, by Lemma 11, every OR and AND component updated in parallel is frozen in

C. Besides, by the equivalence proved above, there are only trivial alternated components of G^F.

Now, let $u, v \in V_{\mathrm{OR}}$ (the AND case is analogous) such that $(u, v) \in A\left(G^F\right)$ and $s(u) < s(v)$. If there exists $k \in \{0, \ldots, p-1\}$ such that $x_u^k = 1$, we have that:

$$x_u^k = 1 \implies x_v^k = 1 \implies x_u^{k+1} = 1 \implies x_v^{k+1} = 1 \cdots \implies x_u^{k-1} = 1 \implies x_v^{k-1} = 1$$

Thus, u and v are frozen in $\mathcal{C}$ at value 1 as well as every node in the same connected component. Otherwise, u is frozen in $\mathcal{C}$ at value 0 as well as every node in the same connected component. In either case, all the OR component is frozen in $\mathcal{C}$.

Finally, all alternated nodes have only frozen neighbors, so they are also frozen. Therefore, every node is frozen in $\mathcal{C}$, which is a contradiction. □

Theorem 14. *Given F an AND-OR function with symmetric G^F. Then, $LC(F, s) \neq \emptyset$ if and only if there exists a bipartite element in the AOA decomposition of G^F.*

Proof. Straightforward from Proposition 12 and Proposition 13. □

The difference between Proposition 10 and Proposition 12 is that, in the OR case, the only schedule than can cycle is the parallel one and only when the interaction digraph is bipartite, and in the AND-OR case, there are others non parallel update schedules that can cycle when there exist the bipartite element in the AOA decomposition of the interaction digraph (for instance, keeping the nodes in the bipartite element in the same block and all the other nodes sequentially), but if such bipartite element does not exist, then no update schedule can cycle.

Corollary 15. *SYMMETRIC AND-OR LCE is polynomial.*

Proof. In Theorem 14 we characterized the existence of solution of this problem and such characterization is testable in polynomial time. □

5.2. Feasible limit cycle problem

Given the result of the previous section, we prove that SYMMETRIC OR FLC is also polynomial.

Proposition 16. *SYMMETRIC OR FLC is polynomial.*

Proof. Let F be an OR function with symmetric G^F and a sequence of different state vectors $\mathcal{C} = \left[x^k\right]_{k=0}^{p}, x^k \in \{0,1\}^n, x^p = x^0$. According to Proposition 10, to determine the solution to the problem is necessary and sufficient to check $p = 2$, $F(x^0) = x^1$ and $F(x^1) = x^0$, which can be done in polynomial time. $\square$

6. Conclusions

We have studied the algorithmic complexity of two problems about the existence of a deterministic update schedule (parallel, sequential and block-sequential) for a given Boolean network which yields any limit cycle (Existence Limit Cycle problem) or a particular limit cycle (Feasible Limit Cycle problem). We proved that both problems are NP-Hard even in AND-OR networks or in networks having a symmetric interaction digraph. However, they are both polynomial problems in OR networks with a symmetric interaction digraph. This because in such networks the existence of a limit cycle depends on a structural property of the interaction digraph which can be verified in polynomial time and the length of the limit cycle is at most two. The case of AND-OR networks with symmetric digraph remains an open problem.

References

Akutsu, T., Miyano, S., Kuhara, S., et al., 1999. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. In: Pacific Symposium on Biocomputing. Vol. 4. pp. 17–28.

Aracena, J., Goles, E., Moreira, A., Salinas, L., 2009. On the robustness of update schedules in Boolean networks. Biosystems 97, 1–8.

Aracena, J., Gómez, L., Salinas, L., 2013. Limit cycles and update digraphs in Boolean networks. Discrete Applied Mathematics 161, 1–2.

Bornholdt, S., 2008. Boolean network models of cellular regulation: prospects and limitations. Journal of the Royal Society Interface 5 (Suppl 1), S85–S94.

Demongeot, J., Elena, A., Sené, S., 2008. Robustness in regulatory networks: a multi-disciplinary approach. Acta Biotheoretica 56 (1-2), 27–49.

Elena, A., 2009. Robustesse des réseaux d'automates booléens à seuil aux modes d'itération. Application à la modélisation des réseaux de régulation génétique. Ph.D. thesis, Grenoble, France.

Fauré, A., Naldi, A., Chaouiya, C., Thieffry, D., 2006. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. Bioinformatics 22 (14), e124–e131.
URL http://bioinformatics.oxfordjournals.org/content/22/14/e124.abstract

Goles, E., Montalva, M., Ruz, G. A., 2013. Deconstruction and dynamical robustness of regulatory networks: Application to the yeast cell cycle networks. Bulletin of Mathematical Biology 75 (6), 939–966.

Goles, E., Noual, M., 2012. Disjunctive networks and update schedules. Advances in Applied Mathematics 48, 646–662.

Goles, E., Salinas, L., 2008. Comparison between parallel and serial dynamics of Boolean networks. Theoretical Computer Science 396 (1–3), 247 – 253.
URL http://www.sciencedirect.com/science/article/pii/S0304397507006846

Green, D. G., Leishman, T. G., Sadedin, S., 2007. The emergence of social consensus in Boolean networks. In: Artificial Life, 2007. ALIFE'07. IEEE Symposium on. IEEE, pp. 402–408.

Huang, S., 1999. Gene expression profiling, genetic networks and cellular states: an integrating concept for tumorigenesis and drug discovery. Journal of Molecular Medicine 77, 469–480.

Just, W., 2006. The steady state system problem is NP-hard even for monotone quadratic Boolean dynamical systems.

Kauffman, S., 1969. Metabolic stability and epigenesis in randomly connected nets. Journal of Theoretical Biology 22, 437–467.

Kauffman, S., 1993. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York.

Macauley, M., Mortveit, H. S., 2009. Cycle equivalence of graph dynamical systems. Nonlinearity 22 (2), 421.

Meng, M., Feng, J., 2014. Function perturbations in Boolean networks with its application in a d. melanogaster gene network. European Journal of Control 20 (2), 87 – 94.
URL http://www.sciencedirect.com/science/article/pii/S0947358014000028

Robert, F., 1986. Discrete Iterations: A Metric Study. Springer-Verlag, Berlin.

Robert, F., 1995. Les Systèmes Dynamiques Discrets. Mathématiques et Applications. Springer.
URL http://books.google.cl/books?id=Oztl9R30BW8C

Ruz, G. A., Goles, E., Montalva, M., Fogel, G. B., 2014. Dynamical and topological robustness of the mammalian cell cycle network: A reverse engineering approach. Biosystems 115, 23–32.

Schaefer, T. J., 1978. The complexity of satisfiability problems. In: Proceedings of the Tenth Annual ACM Symposium on Theory of Computing. STOC '78. ACM, New York, NY, USA, pp. 216–226.
URL http://doi.acm.org/10.1145/800133.804350

Shmulevich, I., Lähdesmäki, H., Dougherty, E., Astola, J., Zhang, W., 2003. The role of certain post classes in Boolean network models of genetic networks. Proceedings of the National Academy od Sciences USA 100, 10734–10739.

Shmulevich, I., Saarinen, A., Yli-Harja, O., Astola, J., 2002. Inference of genetic regulatory networks via best-fit extensions. In: Computational and Statistical Approaches to Genomics. Springer, pp. 197–210.

Thomas, R., 1973. Boolean formalization of genetic control circuits. Journal of Theoretical Biology 42, 563–585.

Tocci, R., Widmer, N., 2001. Digital Systems: Principles and Applications, seventh Edition. Prentice-Hall.

Centro de Investigación en Ingeniería Matemática (CI^{2}MA)

PRE-PUBLICACIONES 2014

2014-19 Veronica Anaya, David Mora, Carlos Reales, Ricardo Ruiz-Baier: *Stabilized mixed finite element approximation of axisymmetric Brinkman flows*

2014-20 Veronica Anaya, David Mora, Ricardo Oyarzúa, Ricardo Ruiz-Baier: *A priori and a posteriori error analysis for a vorticity-based mixed formulation of the generalized Stokes equations*

2014-21 Salim Meddahi, David Mora: *Nonconforming mixed finite element approximation of a fluid-structure interaction spectral problem*

2014-22 Eduardo Lara, Rodolfo Rodríguez, Pablo Venegas: *Spectral approximation of the curl operator in multiply connected domains*

2014-23 Gabriel N. Gatica, Filander A. Sequeira: *Analysis of an augmented HDG method for a class of quasi-Newtonian Stokes flows*

2014-24 Mario Álvarez, Gabriel N. Gatica, Ricardo Ruiz-Baier: *An augmented mixed-primal finite element method for a coupled flow-transport problem*

2014-25 Raimund Bürger, Sarvesh Kumar, Ricardo Ruiz-Baier: *Discontinuous finite volume element discretization for coupled flow-transport problems arising in models of sedimentation*

2014-26 Greg Barber, Muhammad Faryad, Akhlesh Lakhtakia, Thomas Mallouk, Peter Monk, Manuel Solano: *Buffer layer between a planar optical concentrator and a solar cell*

2014-27 David Mora, Gonzalo Rivera, Rodolfo Rodríguez: *A virtual element method for the Steklov eigenvalue problem*

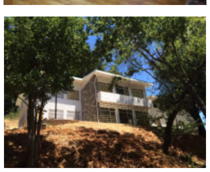
2014-28 Fabián Flores-Bazán, Giandomenico Mastroeni: *Characterizing FJ and KKT conditions in nonconvex mathematical programming with applications*

2014-29 Franco Fagnola, Carlos M. Mora: *On the relationship between a quantum Markov semigroup and its representation via linear stochastic Schrodinger equations*

2014-30 Julio Aracena, Luis Gomez, Lilian Salinas: *Complexity of limit cycle existence and feasibility problems in Boolean networks*

Para obtener copias de las Pre-Publicaciones, escribir o llamar a: Director, Centro de Investigación en Ingeniería Matemática, Universidad de Concepción, Casilla 160-C, Concepción, Chile, Tel.: 41-2661324, o bien, visitar la página web del centro: `http://www.ci2ma.udec.cl`



CENTRO DE INVESTIGACIÓN EN
INGENIERÍA MATEMÁTICA (CI^{2}MA)
Universidad de Concepción

Casilla 160-C, Concepción, Chile
Tel.: 56-41-2661324/2661554/2661316
http://www.ci2ma.udec.cl