Advances in Applied Mathematics 48 (2012) 646–662

Contents lists available at SciVerse ScienceDirect

Advances in Applied Mathematics

www.elsevier.com/locate/yaama

Disjunctive networks and update schedules

Eric Goles [a,b], Mathilde Noual [c,d,*]

[a] Universidad Adolfo Ibañez, Peñalolen, Santiago, Chile
[b] ISCV, Complex Systems Institute of Valparaiso, Valparaiso, Chile
[c] Université de Lyon, ÉNS-Lyon, LIP, CNRS UMR5668, Lyon, France
[d] IXXI, Institut rhône-alpin des Systèmes Complexes, Lyon, France

ARTICLE INFO

Article history:
Received 15 December 2010
Accepted 25 November 2011
Available online 15 February 2012

MSC:
37B99

Keywords:
Regulation network
Linear Boolean network
Attractor
Limit cycle
Fixed point
Update schedule

ABSTRACT

In this paper, we present a study of the dynamics of disjunctive networks under all block-sequential update schedules. We also present an extension of this study to more general fair periodic update schedules, that is, periodic update schedules that do not update some elements much more often than some others. Our main aim is to classify disjunctive networks according to the robustness of their dynamics with respect to changes of their update schedules. To study this robustness, we focus on one property, that of being able to cycle dynamically.

0. Introduction

Set in the framework of discrete dynamical systems seen as models of biological regulation networks, this article studies the dynamics of disjunctive automata networks under different update schedules. Since discrete models of regulation networks were introduced [13,14,20], many studies of the dynamical properties of general and particular automata networks have been carried out [4,6,10,11,15,17,18]. Many of them have focused on dynamics that are induced either by a sequential updating of the elements of the networks, either by a parallel updating, or by both.

In the case of genetic regulation networks, for instance, although biological knowledge about the precise updating schedules of network components lack, one may argue reasonably that genes

* Corresponding author at: Laboratoire d'Informatique du Parallélisme de l'ÉNS Lyon, Site Jacques Monod, 15 parvis René Descartes, BP 7000 69342 Lyon Cedex 07, France.

E-mail address: mathilde.noual@ens-lyon.fr (M. Noual).

0196-8858/$ – see front matter
doi:10.1016/j.aam.2011.11.009

involved in a same cellular physiological function are highly unlikely to perform their regulations in perfect synchrony. Biologists, however, seem to agree that a certain amount of synchrony is not, on the whole, implausible [2,22]. Thus, it seams interesting to study updating modes that are somewhere between total synchrony and total asynchrony. A now classical generalisation of the parallel and sequential update schedules has been introduced by F. Robert [19]. These more general update schedules are known as *block-sequential* update schedules.[1] They define blocks of elements of a network whose states are to be updated in parallel while the block themselves are updated sequentially and they do so in a way that every element is updated exactly once during one update sequence. In [9], following a conjecture expressed by Thomas [21] concerning the role of underlying cycles in the dynamics of a network, we studied exhaustively and characterised the dynamics of Boolean automata networks whose underlying structures are directed cycles under all block-sequential update schedules. Here, we take a different approach.

Instead of restricting the underlying structure of the networks we study, we chose to focus on a particular instance of Boolean automata networks, namely disjunctive networks. One particularity of these networks make them an interesting starting point to answer some questions that reveal themselves yet too thorny for general Boolean automata networks: the fact that their underlying structures translate completely and straightforwardly every dependency between elements states. The purpose of the work we present here is thus to understand the dynamics of disjunctive networks under all block-sequential update schedules (Section 2) and more generally under all *fair periodic update schedules* (Section 3), that is, periodic update schedules that do not update some elements much more often than some others. To build this understanding, we set the aim of classifying disjunctive networks according to the robustness of their dynamics with respect to changes of their update schedules. To study this robustness, we focus on one property, that of being able to cycle dynamically. After a preliminary section of definitions (Section 1), in Section 2, we present our study of disjunctive networks under arbitrary block-sequential update schedules and generalise the particular case of disjunctive networks updated in parallel as studied in [8,12]. In the following section, Section 3, we extend this study to all fair update schedules and propose a classification of networks based on the classification suggested by Elena [7]. We also point out, the differences of our notion of robustness when we consider all fair periodic update schedules and when we only consider block-sequential update schedules.

1. Definitions

1.1. Disjunctive networks

A *disjunctive network* of size n is defined by a digraph $G = (V, A)$ of order $|V| = n$. The nodes of this digraph are assimilated to the elements of the network. Abusing language, we speak indifferently of the digraph or *network* G. In the sequel, we use the following notation

$$\forall m \in \mathbb{N}, \quad \mathbb{N}^{<m} = \{0, \dots, m-1\}$$

and the convention that, in the general case, the set of nodes of a disjunctive network $G = (V, A)$ equals $V = \mathbb{N}^{<n}$.

The set of nodes and the set of arcs of a subdigraph H of an arbitrary digraph $G = (V, A)$ are denoted respectively by V_H and A_H. In particular, strongly connected components are denoted by $C = (V_C, A_C)$ and are said to be *non-trivial* if they contain at least one arc. Thus, non-trivial strongly connected components either are *loop-nodes* (*i.e.* nodes $i \in V$ with a loop $(i, i) \in A$) or contain several nodes. Paths P_{ji} from a node j to a node i are denoted by the ordered list $\{i = i_0, i_1, \dots, i_\ell = j\}$ of nodes they involve. By default, they are supposed to be directed so that $\forall k < \ell$, $(i_k, i_{k+1}) \in A$ is an arc. Otherwise, when it is specified that P_{ji} is *undirected*, it is supposed that $\forall k < \ell$, either (i_k, i_{k+1}) is an arc, or (i_{k+1}, i_k) is. Closed pathes are called cycles. For a cycle $\mathcal{C} = \{i_0, i_1, \dots, i_\ell = i_0\}$, we denote

[1] F. Robert called them *série-parallèle*.

by $V_{\mathcal{C}}$ the set of nodes in $\mathcal{C}$. Cycles are not supposed here to be necessarily simple so it may be that $\exists k, k' \leqslant \ell$ such that $k \neq k'$ and $i_k = i_{k'}$. Thus, in the general case, $\{i_k \mid k \in \mathbb{N}/\ell\mathbb{N}\}$ is a *multiset* with underlying set $V_{\mathcal{C}}$. Cycles are not supposed either to run at most once through each arc: it may be that $\exists k, k' \leqslant \ell$ such that $k \neq k'$ and $(i_k, i_k + 1) = (i_{k'}, i_{k'+1})$. Generally, the set of arcs of a cycle $\mathcal{C}$, which is denoted by $A_{\mathcal{C}}$, is the underlying set of the multiset $\{(i_k, i_{k+1}) \mid k \in \mathbb{N}/\ell\mathbb{N}\}$.

As in any Boolean automata network, nodes in a disjunctive network can take two states, either 0 or 1. Thus, if n is the size of the network, then $\{0,1\}^n$ is the set of network *configurations*. For any network configuration $x = (x_0 \ldots, x_{n-1}) \in \{0,1\}^n$, the coefficient x_i is the state of node $i \in V$. More generally, if $H = (V_H, A_H)$ is a subdigraph of G, then x_H denotes the configuration of H, *i.e.* the restriction of x to the coefficients x_i, $i \in V_H$. When all nodes of a digraph or subdigraph of size n have the same state $a \in \{0,1\}$, its configuration is written a^n.

In a digraph $G = (V, A)$, the *in-* (resp. *out-*) *neighbourhood* of a node $i \in V$ is the set of nodes defined and denoted by $N_G^-(i) = \{j \in V \mid (j,i) \in A\}$ (resp. $N_G^+(i) = \{j \in V \mid (i,j) \in A\}$). The *in-* (resp. *out-*) *degree* of i is the cardinal of this set, denoted by $deg_G^-(i)$ (resp. by $deg_G^+(i)$). Arcs of the set A represent dependencies between automata states which are specified by the *local transition functions* f_i that are associated to each node i:

$$f_i : \begin{cases} \{0,1\}^n \to \{0,1\}, \\ x \mapsto \bigvee_{j \in N_G^-(i)} x_j. \end{cases} \tag{1}$$

The idea is that if x is the current network configuration and if the state of node i is updated, then it becomes $f_i(x)$. In other words, i takes state 1 if and only if $\exists j \in N_G^-(i)$, $x_j = 1$. Let us note that a general Boolean automata network is defined similarly. The only difference is that f_i can be any Boolean function satisfying $(i,j) \in A \Leftrightarrow \exists x \in \{0,1\}^n,\ f_i(x) \neq f_i(\bar{x}^j)$ where $\bar{x}^j$ is the configuration defined by $\forall k \neq j$, $\bar{x}_k^j = x_k$ and $\bar{x}_j^j = \neg x_j$.

In this paper, without loss of generality, all digraphs considered are supposed to be connected and to have non-null minimal in-degrees ($\forall i \in V$, $deg_G^-(i) > 0$). The reason why the second restriction can be done safely in the context of disjunctive networks follows from the definition of the local transition functions in (1). Indeed, in a disjunctive network, nodes with a constant state (which necessarily equals 0) have no effective impact on the states of other nodes. They can thus be removed from the digraph without any consequences on the states of the remaining nodes and this "pruning" of the original digraph can be carried out polynomial time.

Now, in order to see networks as dynamical systems and define a *global* transition function $F : \{0,1\}^n \to \{0,1\}^n$, an update schedule needs to be specified.

1.2. Update schedules

In this paper, we are interested in periodic update schedules that update all nodes in a way that there are no nodes that are updated much more often than others. We call these update schedules *fair* update schedules. They constitute the important class of update schedules that have a finite period which involves each node at least once.

Let us first define a general (*periodic*) *update schedule* as a function $s : V \to \mathcal{P}(\mathbb{N}^{<m})$ such that $\forall i \in V$, $s(i)$ refers to the set of dates at which node i is updated during one update sequence and $\forall t < m$, $s^{-1}(t)$ refers to the set of nodes updated at date t of the update sequence (*i.e.,* in t-th position). In other words, nodes are updated according to the sequence $s^{-1}(0), s^{-1}(1), \ldots, s^{-1}(m-1)$ (see Example 1 below). Let us note that some nodes may never be updated and some nodes may be updated more than once during each update sequence. Let us also note that when $|s^{-1}(t)| > 1$, there are $|s^{-1}(t)|$ nodes that are updated synchronously at step t of the sequence. Without loss of generality, we suppose that update schedules impose no "waiting period" within a complete sequence of updates: $\forall t < m$, $s^{-1}(t) \neq \emptyset$.

A network G updated according to an update schedule s is denoted by $G(s)$ and will simply be called the network $G(s)$ in the sequel.

Example 1. Let G be a network of size 5 and let $s : V \to \mathcal{P}(\mathbb{N}^{<3})$ be the update schedule such that $s(0) = \emptyset$, $s(1) = s(4) = \{0, 2\}$, $s(2) = \{0\}$, and $s(3) = \{1\}$. Then, the network $G(s)$ has the following dynamics starting from $x(0) \in \{0, 1\}^n$:

$$\begin{aligned} x(0) \to x(1) &= \big(x_0(0), f_1\big(x(0)\big), f_2\big(x(0)\big), x_3(0), f_4\big(x(0)\big)\big) \\ \to x(2) &= \big(x_0(0), f_1\big(x(0)\big), f_2\big(x(0)\big), f_3\big(x(1)\big), f_4\big(x(0)\big)\big) \\ \to x(3) &= \big(x_0(0), f_1\big(x(2)\big), f_2\big(x(0)\big), f_3\big(x(1)\big), f_4\big(x(2)\big)\big) \\ \to x(4) &= \big(x_0(0), f_1\big(x(3)\big), f_2\big(x(3)\big), f_3\big(x(1)\big), f_4\big(x(3)\big)\big) \ldots \end{aligned}$$

where $x(t)$ denotes the configuration of the network at step t of the update sequence.

We say that an update schedule s is *k-fair* when it doesn't update any node more than k times the number of times it updates another node:

$$\forall i, j \in V, \quad \big|s(i)\big| \leqslant k \times \big|s(j)\big|.$$

In this paper, we pay special attention to a well-known class of 1-fair update schedules, namely, *block-sequential update schedules* (see Example 2). These update schedules are such that $m \leqslant n = |V|$ and every node is updated exactly once during an update sequence: $\forall i \in V$, $|s(i)| = 1$. For such update schedules, we abuse notations and define s as a function of $V \to \mathbb{N}^{<m}$ so that $s(i)$ refers to the only date of update of node i within the update sequence (and no longer to a *set* of dates). When $m = 1$ and $s^{-1}(0) = V$, s is called the *parallel update schedule* and is denoted by π. It updates all nodes at once. On the contrary, a *sequential update schedule* is a block-sequential update schedule that updates nodes one at a time: $\forall t < m = n$, $|s^{-1}(t)| = 1$. The number of different block-sequential update schedules of a set of n elements is known to be exponential in n [5].

Example 2. Let $V = \mathbb{N}^{<6}$. The function $r : V \to \mathbb{N}^{<3}$ such that $r(0) = r(1) = r(5) = 2$, $r(3) = r(4) = 1$ and $r(2) = 0$ is a block sequential update schedule. The function $s : V \to \mathbb{N}^{<6}$ such that $s(0) = 3$, $s(1) = 2$, $s(2) = 4$, $s(3) = 1$, $s(4) = 5$ and $s(5) = 0$ is a sequential update schedule. A more practical way of denoting r, s and the parallel update schedule π is the following:

$$r \equiv \{2\}\{3, 4\}\{0, 1, 5\}, \qquad s \equiv \{5\}\{3\}\{1\}\{0\}\{2\}\{4\}, \qquad \pi \equiv \{0, 1, 2, 3, 4, 5\}.$$

1.3. Transition functions

Let us now define a *transition function* local *to a set* $W \subseteq V$ *of nodes* that are updated in parallel:

$$f_W(x)_i = \begin{cases} f_i(x) & \text{if } i \in W, \\ x_i & \text{otherwise.} \end{cases}$$

Then, we can define the *global transition function* $F_s : \{0, 1\}^n \to \{0, 1\}^n$ of a network updated with a periodic update schedule $s : V \to \mathcal{P}(\mathbb{N}^{<m})$:

$$\forall x \in \{0, 1\}^n, \quad F_s(x) = f_{s^{-1}(m-1)}(x) \circ \cdots \circ f_{s^{-1}(1)}(x) \circ f_{s^{-1}(0)}(x).$$

F_s is the function that gives the new configuration of the network after a whole sequence of updates by s. For instance, for the parallel update schedule, $F_\pi(x)$ equals $f_V(x)$. When there is no ambiguity as to what network $G = (V, A)$ and what update schedule s are being considered, for any initial configuration $x \in \{0, 1\}^n$, we write $x = x(0)$ and $x(t) = F_s^t(x)$ (where F_s^t is the t-th iterate of F_s).

For a network $G = (V, A)$ updated with a block-sequential update schedule s we also introduce the *local transition functions* f_i^s *relative to* s so that:

$$F_s(x) = \left(f_0^s(x), \ldots, f_{n-1}^s(x)\right) \tag{2}$$

and, in particular, $F_\pi(x) = f_V(x) = (f_0(x), \ldots, f_{n-1}(x))$. These functions are defined as follows:

$$\forall x \in \{0,1\}^n, \quad f_i^s(x) = f_i(x_0^*, \ldots, x_{n-1}^*) \quad \text{where } x_j^* = \begin{cases} x_j & \text{if } s(i) \leqslant s(j), \\ f_j^s(x) & \text{if } s(i) > s(j). \end{cases} \tag{3}$$

This way, under a block-sequential update schedule s, if $x = x(t)$ is the network configuration at time t, then $x_i(t+1) = F_s(x)_i = f_i^s(x)$ is the state that node i has between the time of its sole update within the update sequence following time t and the time of its next update in the next update sequence.

Let us note that the definitions above of (local) transition functions are given for arbitrary Boolean automata networks. For the case of disjunctive networks in which all local transition functions equal the OR function, local transition functions relative to s defined in (3) can be written as follows:

$$\forall x \in \{0,1\}^n, \quad f_i^s(x) = \bigvee_{\substack{j \in N_G^-(i) \\ s(j) \geqslant s(i)}} x_j \vee \bigvee_{\substack{j \in N_G^-(i) \\ s(j) < s(i)}} f_j(x). \tag{4}$$

As a consequence, if M is the adjacency matrix of G (*i.e.*, the $n \times n$ matrix such that $\forall i, j \in V$, $M_{ij} = 1 \Leftrightarrow (i, j) \in A$), then, under the parallel update schedule, $\forall x(t) \in \{0,1\}^n$, $\forall k \in \mathbb{N}$, $x(t+k) = x(t) \cdot M^k$.

1.4. Dynamics of networks

Let $G = (V, A)$ be a network updated with a fair update schedule $s : V \to \mathcal{P}(\mathbb{N}^{<m})$. Since the set $\{0,1\}^n$ of configurations of $G(s)$ is finite, all trajectories $x(0), x(1), \ldots, x(t), \ldots$ of $G(s)$ necessarily end up looping: $\forall x(0) \in \{0,1\}^n$, $\exists t, p$ s.t. $x(t+p) = x(t)$. *Attractors* are orbits of such periodic configurations $x(t)$. Attractors of size 1 involving a unique periodic configuration x ($x = x(t)$, $\forall t \in \mathbb{N}$) that satisfies $f_i(x) = x_i$, $\forall i \in V$, are called fixed points. Since fixed point configurations can equivalently be defined as the configurations x such that $\forall W \subseteq V$, $f_W(x) = x$, in the case of a 1-fair update schedule s, they are exactly the fixed points of the global transition function F_s. Other attractors are called *limit cycles*.

Let us note that with 1-fair update schedules, the set of attractors of size 1 equals the set of fixed points. This is not necessarily true with other update schedules. Generally, it may exist attractors of size 1 that are limit cycles rather than fixed points. Indeed, update schedules that are not 1-fair may update some nodes several times during a sequence of updates. As a consequence, the states of some nodes may cycle within an update sequence while the network will globally appear to be in a stable configuration if it is observed only after the end of each update sequence. Since one of our main aims here is to distinguish between networks that cycle and networks that don't, while focusing on 1-fair update schedules such as block-sequential update schedules, it will suffice to characterise the dynamics of networks by observing the network configurations only once at the end of each update sequence.

In other terms, in the special case of 1-fair update schedules, and more specifically, that of block-sequential update schedules, we will consider network transitions of the form $(x, F_s(x))$ rather than their decomposition into elementary transitions of the form $(x, f_{s^{-1}(t)}(x))$. With a 1-fair update schedule s, we consider the *transition graph* $\mathcal{I}(G(s))$ whose nodes are the configurations of $G(s)$ and whose arcs are the transitions $(x(t), x(t+1))$ (*i.e.* $\mathcal{I}(G(s))$ is the graph of the function F_s). This digraph describes the dynamics of $G(s)$. In this context, an attractor that corresponds to an underlying cycle of length p in $\mathcal{I}(G(s))$, is said to have *period* p. Configurations in an attractor of period p are also said to have *period* p. For such configurations $x(t)$, p is the minimal integer such that $x(t+p) = x(t)$ is satisfied. In particular, fixed points have period 1.

Generally, if s is a fair update schedule and if H is a subdigraph of G, we say that H cycles (resp. is fixed) in an attractor of $G(s)$ if its state changes in that attractor (resp. if it remains the same). If s is block-sequential, the transition graph relative to H under s is denoted by $\mathcal{I}(H(s))$ and defined as the digraph whose nodes are the configurations $x_H \in \{0,1\}^{|V_H|}$ of H and whose arcs are the transitions $(x_H(t), x_H(t+1))$ where $x_H(t+1) = F_s(x(t))_H$. It is important to note that when $H = C$ is a non-trivial strongly connected component of G, it can be made to behave exactly as if it were an isolated digraph rather than the subdigraph of another. Indeed, it suffices to set to 0 the states of all nodes in $V \setminus V_C$ that are on a path that ends in C. This way, by definition of the local transition functions in a disjunctive network (see (4)), these nodes remain fixed in state 0 (all their in-neighbours are and remain in state 0) and they cannot influence other nodes of the network, *i.e.* they cannot cause another node to change states. In particular, they cannot influence nodes of C and nor can any other node in $V \setminus V_C$ (since other nodes are not on paths that end in C). Thus, a node of C only undergoes influences by nodes of C, *i.e.* C is free to behave as if it were isolated. If s block-sequential, this means that there is a subdigraph of $\mathcal{I}(C(s))$ that is isomorphic to the transition graph that C would have if it were an isolated strongly connected digraph.

Now, before we move on, we give a preliminary lemma that will be extensively used in the sequel. This lemma concerns the impact of the state of a node i being fixed to a certain value $a \in \{0,1\}$ (*i.e.* $\exists t \in \mathbb{N}$, $\forall t' \geqslant t$, $x_i(t') = a$) on the state of other nodes in the network.

Lemma 1. *Let $G = (V, A)$ be a disjunctive network updated with the update schedule s.*

(i) *If s is a fair update schedule and if a node $i \in V$ becomes fixed in state* 1*, then all nodes on a path that starts in i also become fixed in state* 1.
(ii) *If s is a block-sequential update schedule, and if, in an attractor $\mathcal{A}$ of G, a node $i \in V$ is fixed in state* 0*, then all nodes on a path that ends in i are also fixed in state* 0 *in $\mathcal{A}$.*

A consequence is that a strongly connected network updated with a fair update schedule cannot cycle if it contains a loop-node in state 1 *and a strongly connected network updated with a block-sequential update schedule cannot cycle if it contains a loop-node.*

Proof. Point (i) is proven by induction the length ℓ_{ij} of a path P_{ij} from the node i (whose state is fixed to 1) to a node j. The proof relies on the two following facts. By (4), in a disjunctive network, any node takes state 1 if it is updated while one of its in-neighbours is currently in state 1. A fair update schedule updates all nodes eventually. Thus, nodes on P_{ij} gradually become fixed in state 1.

To prove point (ii), suppose that $\forall x(t) \in \mathcal{A} = \{x(t) \mid t \in \mathbb{N}/p\mathbb{N}\}$, $x_i(t) = 0$. Suppose in addition that there exists a node $j \in N_G^-(i)$ satisfying $\exists t \in \mathbb{N}/p\mathbb{N}$, $x_j(t) = 1$. By (3) or (4), either $s(i) \leqslant s(j)$ and then $x_i(t+1) = f_i^s(x(t)) = 1$, or $s(i) > s(j)$ and then $x_i(t) = f_i^s(x(t-1)) = 1$. Both cases contradict $\forall t \in \mathbb{N}/p\mathbb{N}$, $x_i(t) = 0$ so it holds that $\forall t \in \mathbb{N}/p\mathbb{N}$, $\forall j \in N_G^-(i)$, $x_j(t) = 0$. Point (ii) follows by induction on the length ℓ_{ji} of a path from j to i.

Let us prove the last part of Lemma 1. Let $\mathcal{A}$ be an attractor of the strongly connected digraph $G = (V, A)$ and let $(i,i) \in A$. By (4), as soon as i takes state 1, it remains in state 1. Point (i) then implies that $\mathcal{A}$ corresponds to the fixed point $1^{|V|}$. And if s is block-sequential then, either the state of i is fixed to 0 in $\mathcal{A}$ in which case, by point (ii), $\mathcal{A}$ corresponds to the fixed point $0^{|V|}$, either, it takes state 1. □

2. Dynamics of block-sequential update schedules

In this section we describe the dynamics of a network induced by an arbitrary block-sequential update schedule. This description will help us in our general network classification of Section 3.

2.1. Digraph G^s associated to a network $G(s)$

Given a disjunctive network G and a block-sequential update schedule s, we define a new network (see Example 3) $G^s = (V, A^s)$ with the same set of nodes and with a set of arcs equal to:

$$A^s = \{(j,i) \mid f_i^s(x) \text{ depends on } x_j\}$$

so that $\mathcal{I}(G(s)) = \mathcal{I}(G^s(\pi))$.

Lemma 2. *The set of arcs of the network G^s is characterised by:*

$$(j,i) \in A^s \iff \begin{array}{l} \text{in } G\text{, there exists a path } \{i_0, i_1 \dots i_\ell\} \\ \text{of length } \ell \geqslant 1 \text{ from } j = i_0 \text{ to } i = i_\ell \text{ s.t.} \\ s(i_0) \geqslant s(i_1) \text{ and } \forall 1 \leqslant k < \ell,\ s(i_k) < s(i_{k+1}). \end{array}$$

Proof. First, let us suppose that there exists such a path from j to i in G. Let $x = x(t)$ be an arbitrary configuration. By definition, configuration $x(t+1) = F_s(x)$ satisfies $\forall i \in V$, $x_i(t+1) = f_i^s(x)$. By (3), it also satisfies $\forall 1 \leqslant k < \ell$, $x_{i_{k+1}}(t+1)$ depends on $x_{i_k}(t+1)$. With an induction on k this leads to $\forall 1 \leqslant k < \ell$, $x_{i_{k+1}}(t+1)$ depends on $x_{i_1}(t+1)$. By (3) again, $x_{i_1}(t+1)$, in turn, depends on $x_{i_0} = x_j$. As a consequence, $\forall 1 \leqslant k \leqslant \ell$, $x_{i_{k+1}}(t+1)$, and in particular, $x_i(t+1) = f_i^s(x)$ depends on x_j so $(j,i) \in A^s$.

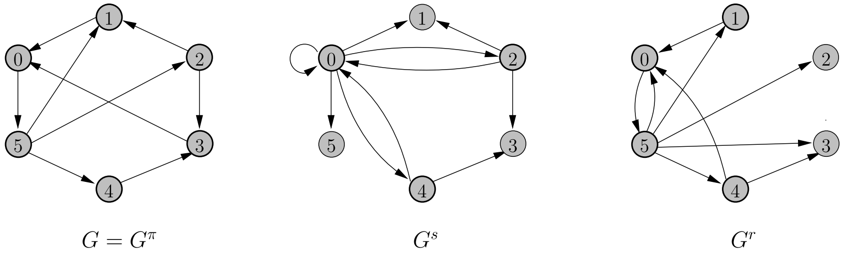
Now, let us suppose that $(j,i) \in A^s$ so that $x_i(t+1)$ depends on $x_j(t)$. The converse of Lemma 2 is proven by induction on $s(i)$. First suppose that $s(i) = 0$. It can only be that $s(i) \leqslant s(j)$ and, by (2), $(j,i) \in A$ (so that $f_i(x)$ does indeed depend on x_j). Next, suppose that $s(i) > 0$, then either, again, $s(i) \leqslant s(j)$ and $(j,i) \in A$, or there is a $k \in N_G^-(i)$ such that $s(k) < s(i)$ (so that $x_i(t+1)$ depends on $x_k(t+1)$) and $x_k(t+1)$ depends on $x_j(t)$. In the second case, by induction hypothesis there exists a path with the desired properties from j to k and, using the arc (k,i), this path can be extended to a path with the desired properties from j to i. □

Let us note that in particular, $G^\pi = G$. And since, by the definition of G^s, for any block-sequential update schedule s, the dynamics of G^s updated in parallel is identical to that of $G(s)$ ($\mathcal{I}(G(s)) = \mathcal{I}(G^s(\pi))$), in the sequel, we speak indifferently of the dynamics of $G(s)$ and $G^s(\pi)$. Thus, provided a characterisation of the digraphs G^s with respect to G, we may bring our study of networks updated with arbitrary block-sequential update schedules back to the study of networks updated in parallel.

Let us mention that for a general Boolean network N defined by a digraph $G = (V, A)$ and a set of local transition functions $\{f_i \mid i \in V\}$, it also holds that $\mathcal{I}(N(s)) = \mathcal{I}(N^s(\pi))$ where N^s is the network defined by G^s and the set of local transition functions $\{f_i^s \mid i \in V\}$.

Example 3. Let $V = \mathbb{N}^{<6}$ and let s and r be as in Example 2. Fig. 1 pictures the digraphs $G = G^\pi$, G^s and G^r associated to a disjunctive network G updated respectively with the update schedules π, s and r. As mentioned above, $G(s)$ and $G^s(\pi)$ have the same dynamics and similarly for r. This can be checked in the table below that gives the dependencies between states of nodes according to the update schedule.

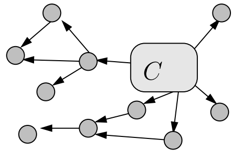
Let us now give some precisions on the structure of the digraphs G^s with respect to that of the digraphs G from which they issue.



$i \in V$	$\pi \equiv \{0,1,2,3,4,5\}$	$s \equiv \{5\}\{3\}\{1\}\{0\}\{2\}\{4\}$	$r \equiv \{2\}\{3,4\}\{0,1,5\}$
	$x_i(t+1)$		
0	$x_1(t) \vee x_3(t)$	$x_1(t+1) \vee x_3(t+1) = x_0(t) \vee x_2(t) \vee x_4(t)$	$x_1(t) \vee x_3(t+1) = x_1(t) \vee x_4(t) \vee x_5(t)$
1	$x_2(t) \vee x_5(t)$	$x_2(t) \vee x_5(t+1) = x_2(t) \vee x_0(t)$	$x_2(t+1) \vee x_5(t) = x_5(t)$
2	$x_5(t)$	$x_5(t+1) = x_0(t)$	$x_5(t)$
3	$x_2(t) \vee x_4(t)$	$x_2(t) \vee x_4(t)$	$x_2(t+1) \vee x_4(t) = x_5(t) \vee x_4(t)$
4	$x_5(t)$	$x_5(t+1) = x_0(t)$	$x_5(t)$
5	$x_0(t)$	$x_0(t)$	$x_0(t)$

Fig. 1. Digraphs and node state dependencies associated to three different block-sequential update schedules (see Example 3). For an arbitrary configuration $x(t) \in \{0,1\}^n$ of the network $G(\delta)$, $\delta \in \{\pi, s, r\}$, we write $x(t+1) = F_\delta(x(t))$ and $x_i(t+1) = f_i^\delta(x(t))$.

Lemma 3. *Let G be a strongly connected digraph and s a block-sequential update schedule of its nodes. Then, G^s is comprised of one unique non-trivial strongly connected component and possibly some outgoing acyclic subdigraphs (see the following figure).*



Proof. Because, $deg^-_G(i) > 0$, $\forall i \in V$, it also holds that $deg^-_{G^s}(i) > 0, \forall i \in V$, *i.e.* with both update schedules π and s, the state of any node depends on the state of at least one other node. Thus, there exists a non-trivial strongly connected component in G^s.

Let us suppose that there exists two distinct non-trivial strongly connected components C_1 and C_2 in G^s and let $j \in V_{C_1}$ and $i \in V_{C_2}$ be two nodes in each of them. Also, let $j' \in V_{C_1}$ be another (or the same) node of C_1 such that $(j', j) \in A^s$ (this node exists because C_1 is non-trivial). Because G is strongly connected, it contains a path $\mathcal{P}_{ji} = \{j = i_0, i_1, \ldots, i_\ell = i\}$ from j to i. Let $r < \ell$ be the smallest integer such that $s(i_r) \geqslant s(i_{r+1})$, if it exists, and let $r = \ell$, otherwise. It can be shown that the sub-path of $\mathcal{P}_{ji}$ that starts in i_r and ends in i is necessarily a series of zero (if $i_r = i$), one or several pathes of the form described in Lemma 2. By Lemma 2, each of these pathes are turned into single arcs in G^s so that, in this digraph, there exists a path from i_r to i. Using (3) (the definition of f_i^s) and that $\forall 0 < k \leqslant r$, $s(i_{k-1}) < s(i_k)$, it can also be shown that $\forall x \in \{0,1\}^n$, $\forall 0 < k \leqslant r$, $f_{i_k}(x)$ and in particular $f^s_{i_r}(x)$ depends on $f^s_j(x)$ which in turn depends on $x_{j'}$. Thus, in G^s, there exists a path from j' to i_r, and consequently, a path from $j' \in V_{C_1}$ to $i \in V_{C_2}$. For similar reasons, there also exists a path from a node of C_2 to one of C_1 so that both components can not be distinct. From this contradiction we derive that there only is one non-trivial strongly connected component C in G^s. All nodes $i \notin V_C$ can be reached from C but belong to no non-trivial strongly connected component. Thus, they necessarily constitute acyclic subdigraphs outgoing C. $\square$

Lemma 4. *Let $G = (V, A)$ be a digraph and s a block-sequential update schedule of its nodes. Then, G^s is strongly connected if and only if G is strongly connected and $\forall i \in V$, $\exists j \in N_G^+(i)$, $s(i) \geqslant s(j)$.*

Proof. First, suppose that G is strongly connected and $\forall i \in V$, $\exists j \in N_G^+(i)$, $s(i) \geqslant s(j)$. By Lemma 3, G^s contains a unique non-trivial strongly connected component C. By Lemma 2, $\forall i \in V$, $\exists j \in N_G^+(i)$, $(i, j) \in A^s$ so $\forall i \in V$, $deg_{G^s}^+(i) > 0$. As a result, C contains all nodes of V. Conversely, if G is not strongly connected then, there exists $i, j \in V$ with no path in G from i to j. By Lemma 2, pathes in G^s are induced by pathes in G: there cannot be a path from i to j in G^s unless there is one in G. Thus, G^s is not strongly connected. If G is strongly connected but there exists $i \in V$ such that $\forall j \in N_G^+(i)$, $s(j) > s(i)$ then, by Lemma 2 again, $deg_{G^s}^+(i) = 0$ and G^s is not strongly connected. □

2.2. Structure and dynamics

As it has been mentioned above, to define a general Boolean network, one needs a set of local functions as well as an underlying digraph. As a result, the dynamics of such networks cannot be derived directly from the knowledge of their structure. Disjunctive networks, on the contrary, are completely defined by their underlying digraphs. And as we have seen in the previous section with the fact that $\mathcal{I}(G(s)) = \mathcal{I}(G^s(\pi))$, this remains true when the network is updated with an arbitrary block-sequential update schedule s and if the underlying digraph considered is G^s. The following lemma can be derived from results in [8] or from the formula in [12] giving the number of fixed points of a conjunctive or disjunctive network. We however give here a simple proof. Let us recall that by definition, fixed points of a network do not depend on its update schedule.

Lemma 5. *A disjunctive network $G = (V, A)$ (such that $\forall i \in V$, $deg^+(i) > 0$) has a unique non-trivial strongly connected component if and only if for any block-sequential update schedule s, $0^{|V|}$ and $1^{|V|}$ are the only fixed points of $G(s)$.*

Proof. 0^n and 1^n are obviously fixed points of any disjunctive network (such that $\forall i \in V$, $deg^+(i) > 0$). Suppose first that G has one unique non-trivial strongly connected component C. If x is a fixed point of G such that $\exists i \in V \setminus V_C$, $x_i = 1$, then (by (4)) $\exists j \in N_G^-(i)$, $x_j = 1$. In this case, there exists a path from C to i that contains only nodes whose states equal 1 in x (this can be shown by induction on the length of the path) and in particular, there exists a node of C whose state equals 1 in x. Now, if $\exists i \in V_C$, $x_i = 1$, then, by point (i) of Lemma 1, x equals $x = 1^{|V|}$. Any fixed point of G which is not equal to $0^{|V|}$ is equal to $1^{|V|}$.

Conversely, suppose that G contains more than one non-trivial strongly connected component. Then, G contains a non-trivial strongly connected component C with no in-coming arcs and there exists a fixed point x in which the state of C is fixed to $x_C = 0^{|C|}$ and the states of all other non-trivial strongly connected components C' are fixed to $x_{C'} = 1^{|C'|}$ (they can remain in that state because they are non-trivial). □

Now, let us define $\eta(G)$ as the greatest common divisor of the lengths of all cycles in a strongly connected digraph or component $G = (V, A)$. $\eta(G)$ is called the *index of imprimitivity* of G in [3] and the *loop number* of G in [12]. It is known [3] that the property $\eta(G) = 1$ is equivalent to the adjacency matrix M of G being *primitive*, *i.e.*, M is an irreducible square matrix for which there exists a positive integer N such that $\forall k \geqslant N$, M^k is a strictly positive matrix. This means that if $x(t) \neq 0^n$ then there exists an integer k such that $x(t + k) = x(t) \cdot M^k = 1^n$. More generally, we have the following lemma:

Lemma 6. *Let G be a disjunctive network updated with the block-sequential update schedule s and let C be a non-trivial strongly connected component of G^s. Then, the period of C in any attractor $\mathcal{A}$ of $G(s)$ divides $\eta(C)$ and there exists an attractor $\mathcal{A}$ in which C cycles with period $\eta(C)$. In particular, $\eta(C) = 1$ if and only if the state of C is fixed in all attractors of $G(s)$.*

Proof. We prove Lemma 6 for a strongly connected digraph $C = G$ updated in parallel. The general case follows from $\mathcal{I}(G(s)) = \mathcal{I}(G^s(\pi))$ and from the remark made on page 651 concerning the dynamics of non-trivial strongly connected components.

First, let us consider two arbitrary nodes i and j that are connected by a path of length ℓ_{ij} from i to j. Given that $x_i(t) = 1$ is true for some time $t \in \mathbb{N}$, we can show by induction on ℓ_{ij} that $x_j(t + \ell_{ij}) = 1$ is true at some ulterior time $t + \ell_{ij}$. In particular, $x_i(t + \ell_{ii}) = x_i(t) = 1$ holds and so does $\forall k \in \mathbb{N}$, $x_i(t + k \cdot \ell_{ii}) = x_i(t) = 1$.

Now, suppose that a node i satisfies $x_i(t) = 0$ and for some integer $k \geqslant 1$, $x_i(t + k \cdot \ell_{ii}) = 1$. Then by the previous remark it holds that $\forall k' \geqslant k$, $x_i(t + k' \cdot \ell_{ii}) = 1$. In particular, if $x(t)$ belongs to an attractor $\mathcal{A}$ of period p, with $k' = k \cdot p \geqslant k$, it holds that $0 = x_i(t) = x_i(t + k \cdot p \cdot \ell_{ii}) = 1$. This contradiction proves that if $x_i(t) = 0$ in an attractor configuration $x(t)$, then, there is no $k \geqslant 1$ such that $x_i(t + k \cdot \ell_{ii}) = 1$.

Thus, generally, for a configuration $x(t)$ belonging to an attractor $\mathcal{A}$ of period p, it holds that $\forall i \in V,\ \forall k \in \mathbb{N},\ x_i(t + k \cdot \ell_{ii}) = x_i(t)$. Consequently, p divides the lengths ℓ_{ii} of all cycles in G. It also divides their greatest common divisor $\eta(G)$ and if $\eta(G) = 1$, all attractors have period $p = 1$.

Now, finally, let us suppose that $\eta(G) > 1$ so that (see [3]) the set of nodes of G may be partitioned as follows: $V = V_1 \uplus V_1 \uplus \cdots \uplus V_{\eta(G)}$ where $\forall 1 \leqslant k \leqslant \eta(G)$, $V_k \neq \emptyset$, and, supposing that $V_{\eta(G)+1} = V_1$, all arcs of A are of the form $(j, i) \in V_k \times V_{k+1}$ for some $1 \leqslant k \leqslant \eta(G)$. Then, the configuration $x \in \{0, 1\}^n$ defined by $\forall i \in V_1$, $x_i = 0$ and $\forall i \notin V_1$, $x_i = 1$, for example, belongs to a limit cycle of period $\eta(G)$ of G. □

Lemma 6 extends to all block-sequential update schedules a result given in [8] and again in [12] concerning the particular case of the parallel update schedule. A direct consequence of this lemma is that a network G that may cycle under block-sequential update schedules is a network for which there exists at least one block-sequential update schedule s and one non-trivial strongly connected component C of G^s satisfying $\eta(C) > 1$.

Following Lemma 6, we know that any network G such that $\eta(G) > 1$ cycles with the parallel update schedule. A natural question is "If G is such that $\eta(G) = 1$, can we determine whether G may cycle for some block-sequential update schedule (*i.e.*, whether $\eta(G^s) > 1$ for some s)?". As mentioned earlier, the number of block-sequential update schedules of n nodes is exponential. Thus, to answer this question, we must keep relying on the structural properties of G. The two following propositions answer this question for some particular classes of disjunctive networks.

Proposition 1. *A symmetric digraph*[2] *G cycles under no other block-sequential update schedule than the parallel*: $\forall s \neq \pi$, $\eta(G^s) = 1$. *It cycles under the parallel update schedule if and only if it contains no cycle of odd length. In this case $\eta(G) = 2$ and by Lemma* 6*, all limit cycles have period* 2.

Proof. For any nodes i and j such that $(i, j), (j, i) \in A$, if $s(i) > s(j)$ then i is a loop-node in G^s so that $\eta(G^s) = 1$ and $G(s)$ does not cycle by Lemma 6. Thus, the only update schedule of G that can possibly induce limit cycles is π. If there exists an odd length cycle in G, then, since there also are cycles of length 2, $\eta(G) = 1$. Otherwise, $\eta(G) = 2$. □

To give the next example of a class of disjunctive networks illustrating Lemma 6, let us define *cycle-edges* of a strongly connected digraph G as maximal paths $\{i_0, \ldots, i_k\}$ such that (i) all nodes $i_r, r \leqslant k$ are distinct, except possibly i_0 and i_k, and (ii) all nodes i_r, $r \leqslant k$, except possibly i_0 and i_k, have in- and out-degree 1. With this definition, loops are special instances of cycle-edges of length 1 and so are the arcs (j, i) of a cycle that are such that i and j both have in- or out-degree greater than 1.

Proposition 2. *A digraph containing a non-trivial strongly connected component with no cycle-edges of length* 1 *can cycle for a certain block-sequential update schedule.*

[2] $\forall i, j \in V$, $(i, j) \in A \Rightarrow (j, i) \in A$.

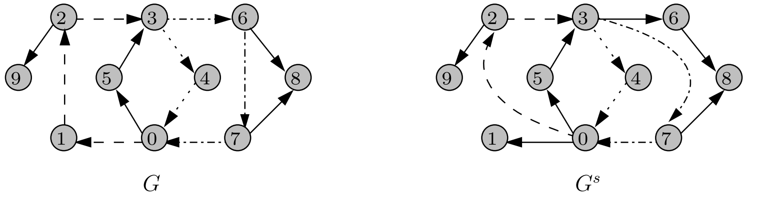


Fig. 2. Left: a digraph G as in Proposition 2 and such that $\eta(G) = 1$. Right: the digraph G^s satisfying $\eta(G(s)) = 4$, where $s \equiv \{0, 1, 3, 4, 5, 6, 8, 9\}\{2, 7\}$. Arcs with different dotted or dashed lines belong to different cycle-edges. Arcs in full lines do not belong to any cycle-edge.

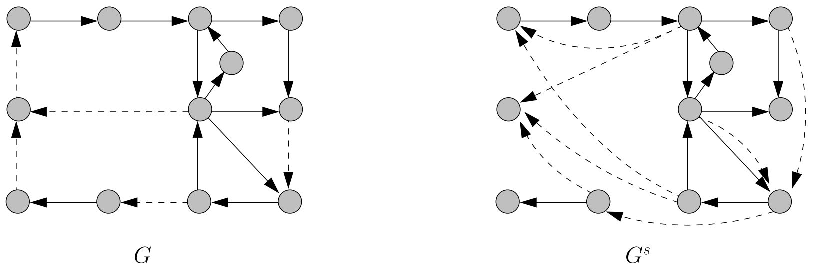


Fig. 3. Left: a network G. The arcs (j, i) in dashed lines are those that satisfy $s(i) > s(j)$. All other arcs are those in a 3-cycle-cover of G. Right: the digraph G^s where s is associated to the 3-cycle-cover of G as in proof of Proposition 3.

Proof. We prove Proposition 2 in the case where the digraph itself is strongly connected. The general case follows the remark made on page 651 concerning the dynamics of non-trivial strongly connected components. Let s be a block-sequential update schedule such that for any cycle-edge $\{i_0, \dots, i_k\}$ of odd length $k \geqslant 3$, $s(i_1) < s(i_2)$ and $\forall r \leqslant k$, $r \neq 1$, $s(i_r) \geqslant s(i_{r+1})$. Then, for any such cycle-edge of G, by Lemma 2, the following holds. The path $\{i_2, \dots, i_k\}$ remains unchanged in G^s. The path $\{i_0, i_2\}$ is replaced by the arc $(i_0, i_2) \in A^s$. The arc $(i_1, i_2) \in A \setminus A^s$ disappears. Thus, globally, the cycle-edge $\{i_0, i_1, \dots, i_k\}$ of odd length k in G becomes a cycle-edge $\{i_0, i_2, \dots, i_k\}$ of even length $k - 1$ in G^s. All cycle-edges and consequently all cycles of G^s have an even length. As a result, $\eta(G^s) > 1$ is even (see Fig. 2). □

We are now going to prove Proposition 3 given below. This proposition characterises, in terms of their structure, the strongly connected disjunctive networks that cycle for a certain block-sequential update schedule. First, let us define *k-cycle-covers* as follows. Let $W \subseteq A$ be a subset of arcs of a strongly connected digraph $G = (V, A)$ and let $\mathcal{C}$ be a cycle of this digraph. We say that $\mathcal{C}$ is covered by a multiple of $k \in \mathbb{N}$ arcs of W or is *k-covered by W* if $\exists q \geqslant 1$ s.t. $|A_{\mathcal{C}} \cap W| = k \cdot q$ (see Fig. 3). The set W is said to be a *k-cycle-cover* of G if and only if:

(i) all cycles of G are k-covered by W, and
(ii) there is no undirected cycle $\{i_0, i_1, \dots, i_m = i_0\}$ containing at least one arc of $A \setminus W$ and satisfying the following: $\forall r < m$, either $(i_r, i_{r+1}) \in A \cap W$ or $(i_{r+1}, i_r) \in A \setminus W$.

The purpose of k-cycle-covers is that we can easily associate to them block-sequential update schedules s such that k divides $\eta(G^s)$.

Proposition 3. *For any strongly connected digraph G and any integer $k > 1$, there exists a block-sequential update schedule s such that k divides $\eta(G^s)$ if and only if there exists a k-cycle-cover of G. As a consequence, an arbitrary network G cycles for some block-sequential update schedule s if and only if it contains a non-trivial strongly connected component C that has a k-cycle-cover for some $k > 1$.*

Proof. Suppose W is a k-cycle-cover of the strongly connected digraph G. Let s be a block sequential update schedule such that:

$$\forall (i,j) \in A,\ (i,j) \in W \quad \Leftrightarrow \quad s(j) \leqslant s(i).$$

The only reason for which s might not be well defined is if its definition induces that $\exists i \in V$, $s(i) > s(i)$. As proven in [1], this can only happen if there exists an undirected cycle $\{i = i_0, i_1, \ldots, i_m = i\}$ containing at least one arc (i_{r+1}, i_r) (such that $s(i_{r+1}) < s(i_r)$) and satisfying the following $\forall r < m$:

- if $(i_r, i_{r+1}) \in A$ then $s(i_{r+1}) \leqslant s(i_r)$, *i.e.*, $(i_r, i_{r+1}) \in W$, and
- if $(i_{r+1}, i_r) \in A$ then $s(i_{r+1}) < s(i_r)$, *i.e.*, $(i_{r+1}, i_r) \notin W$.

Point (ii) in the definition of a k-cycle-cover, however, excludes this. Therefore, s is well defined. Now, by Lemma 2, a cycle of length ℓ in G^s is necessarily induced by a cycle of G that contains ℓ arcs (i, j) such that $s(j) \leqslant s(i)$, *i.e.* such that $(i, j) \in W$. And since that cycle must be k-covered by W, k must divide ℓ. Thus, k divides the lengths of all cycles in G^s and as a result it divides $\eta(G^s)$.

Conversely, suppose that s is such that $k > 1$ divides $\eta(G^s)$. For any cycle $\mathcal{C}$ of G, let $W(\mathcal{C}) = \{(i,j) \in A_{\mathcal{C}} \mid s(j) \leqslant s(i)\}$. We claim that each cycle $\mathcal{C} = \{i_0, i_1, \ldots, i_\ell = i_0\}$ of G induces a cycle of G^s of length $|W(\mathcal{C})|$. To show this, consider the maximal subpaths of $\mathcal{C}$ that have the form described in Lemma 2. There necessarily exist some because s would not be well defined otherwise. The extremities of these maximal subpaths are all the nodes i_r, $r \in \mathbb{N}/\ell\mathbb{N}$ of $V_{\mathcal{C}}$ such that $s(i_{r+1}) \leqslant s(i_r)$, *i.e.* such that $(i_r, i_{r+1}) \in W(\mathcal{C})$. Thus, there are as many of these subpaths as there are arcs in $W(\mathcal{C})$. Further, by Lemma 2, these maximal subpaths of $\mathcal{C}$ are turned into arcs in G^s that connect their extremities in a way that all these arcs form a cycle $\mathcal{C}'$ of G^s. The length of this cycle equals the number of arcs it contains (counting each arc as many times as it is used), that is, the number of maximal subpaths of $\mathcal{C}$, that is, $|W(\mathcal{C})|$. Now, since, by definition, $\eta(G^s)$ divides all cycle sizes in G^s, it divides $|V_{\mathcal{C}'}| = |W(\mathcal{C})|$. As a result, $W = \bigcup W(\mathcal{C})$ is a k-cycle-cover of G (condition (ii) in the definition of a k-cycle-cover is satisfied because s is well defined).

The second part of Proposition 3 follows from the remark on page 651 on the dynamics of non-trivial strongly connected components. □

To complete this section, we give one last result concerning the dynamics of disjunctive networks under block-sequential update schedules. In the next section, we will focus on general fair update schedules.

Lemma 7. *For any disjunctive network G there exists a block-sequential update schedule s such that $G(s)$ only has fixed points.*

Proof. For every non-trivial strongly connected component C of G, let $i_C \in C$ be an arbitrary node in C and let $P_C = \{i_C = i_0, \ldots, i_k = i_C\}$ be a closed path starting and ending in i_C. We can define an update schedule s as follows. For every non-trivial strongly connected component C associated to the node i_C and to the path P_C, s satisfies $s(i_{r+1}) > s(i_r)$, $\forall 0 < r < k$ and, for any other arc $(j, i) \in A$ ($\forall C,\ \forall 0 < r < k$, $(j,i) \neq (i_r, i_{r+1})$), $s(i) \leqslant s(j)$. By Lemma 2, $\forall C$, i_C is a loop-node in G^s. By Lemmas 1 and 6, $G(s)$ does not cycle. □

3. Network classification

This section is devoted to the classification of disjunctive networks according to their dynamics with respect to all fair update schedules. The basis of our classification is the four classes (Fi, Cy, Mi and Ev) defined by Elena in his PhD Thesis [7]. Since Elena, focused on a different set of Boolean

Table 1
In column 1 of this table figures the names of the classes that are defined on the corresponding lines according to the properties given in line 1. In the second (resp. third) column figures a "Y" if networks of the class have fixed points different from 0^n and 1^n (resp. limit cycles) for all fair update schedules, and an "N" if they have for none. The two bottom right cells of the table mean that for some fair update schedules networks in the classes Ev and Ev' have limit cycles and for some they have not.

	$\exists FP \neq 0^n,\ 1^n$	$\exists LC$
Fi	N	N
Fi'	Y	N
Cy	N	Y
Mi	Y	Y
Ev	N	for some s Y, for others N
Ev'	Y	for some s Y, for others N

networks,[3] we have adapted the definition of these classes to fit our study of disjunctive networks. More precisely, we have changed Elena's definitions for the following reason. Any disjunctive network of size n with minimal in-degree greater than 0 obviously has at least two fixed points: 0^n and 1^n. As a result, there is no point in distinguishing networks on the criteria that they have fixed points or not. Thus, we chose to replace this criteria by that of having fixed points different from 0^n and 1^n. This way, we end up with the six classes defined in Table 1. We recall again that the set of fixed points of a network is independent of the update schedule. Thus, every disjunctive network belongs to one of the six classes of Table 1.

Theorem 1 below details the content of the six classes of Table 1. It mentions *weakly-loop-free* components which are defined as non-trivial subdigraphs H of a digraph G that satisfy the following three conditions:

(i) H is strongly connected,
(ii) V_H contains no loop-node,
(iii) there is no node $i \in V$ such that $V_H \subseteq N_G^-(i)$ and there is a path from i to V_H.

To contain no weakly-loop-free component, an arbitrary graph must either be cycle-free or all of its non-trivial strongly connected components must contain loop-nodes (see Fig. 4 (right)). Indeed, any non-trivial strongly connected component with no loop-nodes is a weakly-loop-free component itself. Non-trivial strongly connected components containing loop-nodes can also, however, contain weakly-loop-free components (see Fig. 4 (left)). Let us note that it can be determined in polynomial time whether or not a strongly connected digraph, and as a consequence, any digraph contains a weakly-loop-free component: construct the subdigraph G' of the strongly connected digraph G where all loop-nodes and their incident arcs have been removed, find the non-trivial strongly connected components of G' and look for one of them, C, which is such that there is no node i of G satisfying $V_C \subseteq N_G^-(i)$.

Theorem 1. *Classes of disjunctive networks defined in Table* 1 *satisfy the following*:

(i) $Mi = Cy = \emptyset$.
(ii) $Fi \cup Ev$ *is the set of disjunctive networks that contain a unique non-trivial strongly connected component.* $Fi' \cup Ev'$ *is the set of disjunctive networks that contain several.*

[3] He studied threshold Boolean automata networks whose local transition functions are of the following form: $f_i(x) = \sum_j H(w_{ij} \cdot x_j - \theta_i)$ where H is the Heaviside function ($H(a) = 0$ if $a < 0$ and $H(a) = 1$ otherwise), w_{ij} is the weight attributed to the arc (j, i) and θ_i is the activation threshold of i.

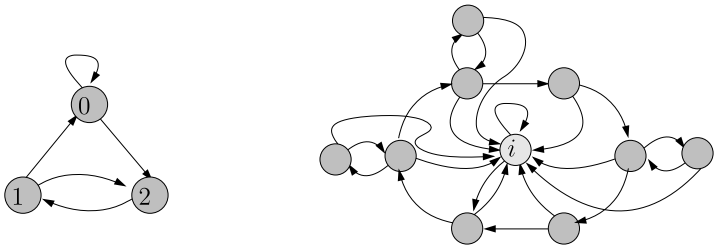


Fig. 4. Left. A strongly connected digraph in which nodes 1 and 2 induce a weakly-loop-free component. This graph has $\{(0,1,0),(0,0,1)\}$ as limit cycle under the fair update schedule $\{1,2\}\{0,1,2\}$. Right. A strongly connected digraph $G=(V,A)$ with no weakly-loop-free component. In any configuration $x \neq 0^{|V|}$, $\exists j \in N^-(i)=V$, $x_j=1$ so as soon as i is updated it takes state 1 and by Lemma 1, the network cannot cycle.

(iii) *$Ev \cup Ev'$ is the set of disjunctive networks that contain at least one* weakly-loop-free *component. $Fi \cup Fi'$ is the set of disjunctive networks that contain none.*

The first point of Theorem 1 derives directly from Lemma 7. Point (ii) comes from Lemma 5 and the fact that fixed points are not changed by fair update schedules. Finally, point (iii) which characterises the set of disjunctive networks that can cycle, follows from Proposition 4 below.

Proposition 4. *A disjunctive network G can cycle under some fair update schedule if and only if it contains a non-trivial strongly connected component in which there is a* weakly-loop-free *component. In addition, if G contains a* weakly-loop-free *simple cycle, then it cycles under a 2-fair update schedule.*

Proof. In this proof, given an update schedule $s: V \to \mathcal{P}(\mathbb{N}^{<m})$ and a configuration $x(0) \in \{0,1\}^n$, we change our notations and write, $\forall t < m$:

$$x(t) = f_{s^{-1}(t-1)}\big(x(t-1)\big) = f_{s^{-1}(t-1)} \circ \cdots \circ f_{s^{-1}(1)} \circ f_{s^{-1}(0)}\big(x(0)\big)$$

(rather than $x(t) = F_s^t(x(0))$ as before). Also, we write $\bar{0}^i$ to denote the configuration $x \in \{0,1\}^n$ such that $\forall j \neq i$, $x_j = 0$ and $x_i = 1$. Following the remark made on page 651, we prove Proposition 4 in the case where G is a non-trivial strongly connected digraph.

Let H be a maximal weakly-loop-free component of G and let $\mathcal{C} = \{i_0, \ldots, i_\ell = i_0\}$ be a cycle of G that runs through each arc of A_H at least once and through no other arc ($V_H = V_{\mathcal{C}}$ and $A_H = A_{\mathcal{C}}$). Then, let $s: V \to \mathcal{P}(\mathbb{N}^{<p})$ be the update schedule defined as follows: $\forall k \in \mathbb{N}/\ell\mathbb{N}$, $s(i_k) = \{k-1, k\}$ and $\forall j \notin V_H$, $s(j) = \{k\}$ where $k \in \mathbb{N}/\ell\mathbb{N}$ is such that $i_k \in V_H \setminus N_G^-(j)$ (k exists by the maximality of H and condition (iii) in the definition of a weakly-loop-free component). It can be checked that s is a well-defined fair update schedule. Let $x(0) = \bar{0}^{i_0}$. We claim that $\forall k \in \mathbb{N}/\ell\mathbb{N}$, $x(k) = \bar{0}^{i_k}$ and as a result, $\{x(k) = \bar{0}^{i_k} \mid k \in \mathbb{N}/\ell\mathbb{N}\}$ is a limit cycle of $G(s)$. Indeed, at each step $k \in \mathbb{N}/\ell\mathbb{N}$ of the update sequence, the set of nodes that are updated equals $s^{-1}(k) = \{i_k, i_{k+1}\} \cup \{j \notin V_H \mid s(j) = k\}$. If $x(k) = \bar{0}^{i_k}$, then the following holds: i_k (the unique node of G in state 1) has no in-neighbours in state 1, i_{k+1} has one in-neighbour (node i_k) in state 1, all other nodes in $s^{-1}(k)$ do not belong to V_H and have no in-neighbours in state 1. As a result, after the update of $s^{-1}(k)$, node i_k has taken state 0, node i_{k+1} has taken state 1, nodes $i \in s^{-1}(k) \setminus V_H$ remain in state 0 as do all other nodes that are not updated. Thus, $x(k+1) = \bar{0}^{i_{k+1}}$. With an induction on k, this allows to conclude.

If $\mathcal{C}$ is a simple cycle, then, because $V_{\mathcal{C}} = \{i_k \mid k < \ell\}$, each node $i_k \in V_{\mathcal{C}}$ is updated only twice: once at step k of the update sequence (when it takes state 1) and once at step $k+1$ (when it goes back to state 0). All other nodes are updated once. Thus, in this case, s is 2-fair.

Finally, let us prove the converse. Suppose that G contains no weakly-loop-free component but $\mathcal{A} = \{x(t) \mid t < p\}$ is a limit cycle of $G(s)$, where s is fair. Let H be the subdigraph of G induced by the nodes whose states are not fixed in $\mathcal{A}$. Necessarily, H is not cycle-free (its minimal in-degree must be at least one for all of its nodes to be able to cycle). Let C be a non-trivial strongly-connected

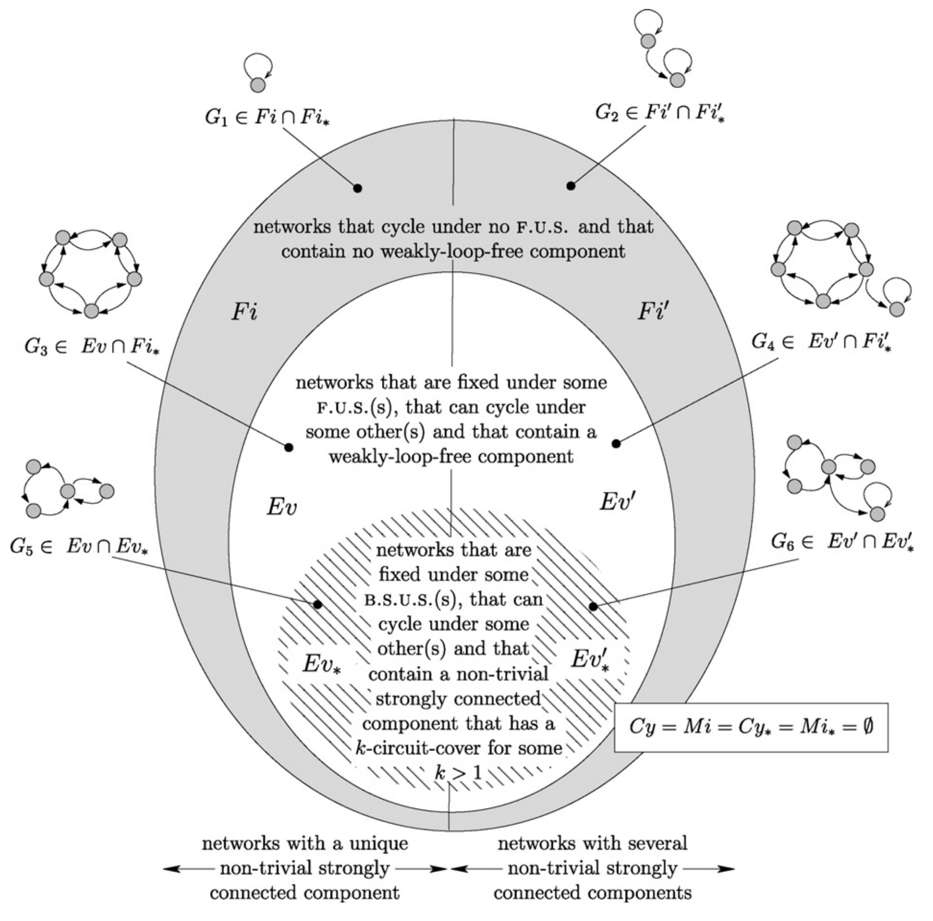


Fig. 5. Classes of disjunctive networks. F.U.S. stands for *fair update schedule* and B.S.U.S. stands for *block-sequential update schedule*. For a class C defined in Table 1, C_* denotes the class with the same definition except that B.S.U.S.s are considered instead of F.U.S.s. Networks that do not belong to Ev_* (resp. Ev'_*) belong to Fi_* (resp. Fi'_*). Digraphs G_1 to G_6 are examples of networks corresponding to each section of the diagram. In these examples, all non-trivial strongly connected components containing a loop-node are unable to cycle whatever the update schedule (by Proposition 4). $G_3 \in Fi_*$ and $G_4 \in Fi'_*$ follow from Proposition 1, $G_3 \in Ev$ and $G_4 \in Ev'$ from Proposition 4. By Proposition 2, G_5 satisfies $\eta(G_5) = 1$ but can still cycle under a B.S.U.S. by Proposition 2. The same is true for the top component of G_6 so $G_5 \in Ev_*$ and $G_6 \in Ev'_*$.

component of H with no in-coming arcs in H. C is not weakly-loop-free. Either it contains a loop-node $i \in V_C \subseteq V_H$ which must take state 1 in some configuration of $\mathcal{A}$ (otherwise, its state is fixed to 0 contradicting the definition of H). In this case, in $\mathcal{A}$, i is fixed in state 1. Either there exists $i \in V$ such that $V_C \subseteq N_G^-(i)$. In this case, node $i \in V$ (such that $V_C \subseteq N_G^-(i)$) also is fixed in state 1. Indeed, in each configuration $x \in \mathcal{A}$, $\exists j \in V_C$, $x_j = 1$ (otherwise all nodes in V_C would be in state 0 and would remain so by definition of C). Thus, in $\mathcal{A}$, whenever i is updated, it takes state 1. In both cases, the strongly connected graph G contains a node that is fixed in state 1 in $\mathcal{A}$. Lemma 1 implies that $\mathcal{A}$ must correspond to the fixed point $1^{|V|}$. □

Let us again restrict our attention to block-sequential update schedules. For any of the six classes C defined in Table 1, let C_* be the class of disjunctive networks that is defined similarly to C except that only block-sequential update schedules are considered. Then, by the results of Section 2, replacing the classes C by C_* in Theorem 1, points (i) and (ii) remain true. As for point (iii), it can be replaced, for instance, by the following claim that derives from Proposition 3: $Ev \cup Ev'$ is the set of disjunctive networks in which there is a non-trivial strongly connected component with a k-cycle-cover for some $k > 1$. Fig. 5 gives two examples (G_3 and G_4) of digraphs that do not cycle for any block-sequential update schedule but that do for some other fair update schedules. Thus, the network classes satisfy the following strict inclusions:

$$Ev_* \subsetneq Ev, \qquad Ev'_* \subsetneq Ev', \qquad Fi \subsetneq Fi_* \quad \text{and} \quad Fi' \subsetneq Fi'_*.$$

Fig. 5 sums up the results concerning all classes introduced in Table 1 for fair and block-sequential update schedules.

4. Conclusion

Despite the apparent simplicity of disjunctive networks and although their underlying structure are very tightly related to their dynamical behaviour, allowing a certain liberty in the updating mode yields a new level of difficulty in the analysis of their dynamics. In this paper, we have classified networks into four classes in both the cases of fair and block-sequential update schedules. We have showed that all networks can loose their limit cycles with a certain update schedule. We have also characterised those networks that may cycle for either a fair update schedule or a block-sequential update schedule. In the more general case of fair update schedules, we can determine in polynomial time whether a disjunctive network whose structure is known is able to cycle or not. When we restrict our attention to just block-sequential update schedules, Proposition 3 gives a characterisation of the networks that cycle under these updatings. It does so by means of the network structure only, allowing us to bring our study of dynamics under different updating modes back to a planer study of a digraph property (that of having a k-cycle-cover for some $k > 1$) that is not without similarities with the *Feedback-Arc-Set* problem. One question remains unanswered, however: can we determine in polynomial time if a disjunctive network is able to cycle for a certain block-sequential update schedule? A related problem is whether or not having a k-cycle-cover, induces having a 2-cycle-cover, *i.e.*, can every cycling network cycle in a limit cycle of period 2 ($\eta(G^s) = 2$)? Further, in [16], Reidys considers only update sequences such that each node is updated at least once and only one node is updated at once. Under these conditions, they show that the property of keeping the same set of periodic configurations is independent of whether or not we allow nodes to be updated several times in a sequence. It would be interesting to determine if Reidys's result remains true under the conditions we chose to study where parallel updatings are allowed, and more, generally, now that we are able to discriminate between a network that cycles and one that doesn't, to describe more precisely the limit dynamical behaviour of a disjunctive network and how its periodic configurations are distributed in its attractors.

Acknowledgments

This work has been supported by Fondecyt 1100003, Anillo ACT-88, CMM-BASAL, Universidad de Chile and CNRS as well as by the Ixxi project Maajes.

References

[1] J. Aracena, E. Fanchon, M. Montalva, M. Noual, Combinatorics on update digraphs in Boolean networks, Discrete Appl. Math. 159 (2011) 401–409.
[2] A.N. Boettiger, M. Levine, Synchronous and stochastic patterns of gene activation in the drosophila embryo, Science 325 (2009) 471–473.
[3] R.A. Brualdi, H.J. Ryser, Combinatorial Matrix Theory, Cambridge University Press, Cambridge, 1991.
[4] O. Colón-Reyes, R. Laubenbacher, B. Pareigis, Boolean monomial dynamical systems, Ann. Comb. 8 (2005) 425–439.
[5] J. Demongeot, A. Elena, S. Sené, Robustness in regulatory networks: A multi-disciplinary approach, Acta Biotheor. 56 (2008) 27–49.
[6] J. Demongeot, E. Goles, M. Morvan, M. Noual, S. Sené, Attraction basins as gauges of the robustness against boundary conditions in biological complex systems, Plos One 5 (2010).
[7] A. Elena, Robustesse des réseaux d'automates booléens à seuil aux modes d'itération. Application à la modélisation des réseaux de régulation génétique, PhD thesis, Université Joseph Fourier, Grenoble, 2009.

[8] E. Goles, Comportement oscillatoire d'une famille d'automates cellulaires non uniformes, PhD thesis, Université scientifique et médicale de Grenoble, France, 1980.
[9] E. Goles, M. Noual, Block-sequential update schedules and Boolean automata circuits, in: Proc. of Automata, 2010, in: Discrete Math. Theor. Comput. Sci., 2010, pp. 41–50.
[10] E. Goles, L. Salinas, Comparison between parallel and serial dynamics of Boolean networks, Theoret. Comput. Sci. 396 (2008) 247–253.
[11] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Natl. Acad. Sci. USA 79 (1982) 2554–2558.
[12] A. Jarrah, R. Laubenbacher, A. Veliz-Cuba, The dynamics of conjunctive and disjunctive Boolean network models, Bull. Math. Biol. 72 (2010) 1425–1447.
[13] S.A. Kauffman, Metabolic stability and epigenesis in randomly constructed genetic nets, J. Theoret. Biol. 22 (1969) 437–467.
[14] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, J. Math. Biol. 5 (1943) 115–133.
[15] H.S. Mortveit, C.M. Reidys, Discrete, sequential dynamical systems, Discrete Math. 226 (2001) 281–295.
[16] C.M. Reidys, Sequential dynamical systems over words, Ann. Comb. 10 (2006) 481–498.
[17] É. Remy, P. Ruet, D. Thieffry, Graphic requirements for multistability and attractive cycles in a Boolean dynamical framework, Adv. in Appl. Math. 41 (2008) 335–350.
[18] A. Richard, Negative circuits and sustained oscillations in asynchronous automata networks, Adv. in Appl. Math. 44 (2010) 378–392.
[19] F. Robert, Discrete Iterations: A Metric Study, Springer-Verlag, Berlin, 1986.
[20] R. Thomas, Boolean formalisation of genetic control circuits, J. Theoret. Biol. 42 (1973) 563–585.
[21] R. Thomas, On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations, Springer Ser. Synergetics 9 (1981) 180–193.
[22] Q. Wang, J. Huang, X. Zhang, B. Wu, X. Liu, Z. Shen, The spatial association of gene expression evolves from synchrony to asynchrony and stochasticity with age, Plos One 6 (2011).