

Critical Kauffman networks under deterministic asynchronous update

Florian Greil, Barbara Drossel and Joost Sattler

Institut für Festkörperphysik, TU Darmstadt, Hochschulstraße 6,
64289 Darmstadt, Germany

E-mail: florian.greil@physik.tu-darmstadt.de

New Journal of Physics **9** (2007) 373

Received 27 June 2007

Published 18 October 2007

Online at <http://www.njp.org/>

doi:10.1088/1367-2630/9/10/373

Abstract. We investigate the influence of a deterministic but non-synchronous update on random Boolean networks, with a focus on critical networks. Knowing that ‘relevant components’ determine the number and length of attractors, we focus on such relevant components and calculate how the length and number of attractors on these components are modified by delays at one or more nodes. The main findings are that attractors decrease in number when there are more delays and that periods may become very long when delay times are not integer multiples of the basic update step.

Contents

1. Introduction	2
2. Relevant components	4
3. Loops with one delayed node	5
3.1. Even loops	6
3.2. Odd loops	6
3.3. Non-integer delays	7
4. Loops with multiple delayed nodes	7
4.1. Sequential update	7
4.2. Same delay, different phases	7
4.3. Different delay times	8
5. Two loops with a cross-link	8
5.1. Delayed node on first loop	9
5.2. Delayed node on second loop	10
5.3. General case	10
6. Loop with one additional link	10
6.1. Canalizing function at Σ	11
6.2. Function f_9 at Σ	13
7. Conclusion	13
Acknowledgment	13
References	14

1. Introduction

Random Boolean networks (RBN) are widely used as models for complex systems that consist of interconnected units that influence each other and have two states ('on' and 'off'). Kauffman [1, 2] used them as a simple model for gene regulation, but they can also be applied in a social and economic context [3, 4], for neural networks and protein networks [5]. Recently it was shown that the idealized representation of genes as Boolean units is sufficient to understand the essential dynamics of certain real gene regulation networks. In these cases, one need not include rate equations for the concentrations of the molecules involved in the processes in order to identify the sequence of steps taken by such a system [6]–[8].

A RBN is a directed graph consisting of N nodes and kN links between them. The nodes have values $\sigma_i \in \{0, 1\}$ and receive input from k other nodes, which are chosen at random when the network is constructed. Each node i has an update function f_i , which assigns to each of the 2^k states of its k input nodes an output 1 or 0. The update function of each node is chosen at random among all 2^{2^k} possible update functions. All nodes are usually updated in parallel according to the rule

$$\sigma_i(t+1) = f_i(\{\sigma_j(t)\}). \quad (1)$$

The assignment of connections and functions to each node remains fixed throughout the whole time evolution, the model is therefore referred to as 'quenched' [9].

The dynamics follows a trajectory $\vec{\sigma}(t) \equiv \{\sigma_1(t), \dots, \sigma_N(t)\}$ in configuration space which eventually leads to a periodically repeating sequence of configurations, called a *cycle*, as the state space is finite and the dynamics is discrete. Such a cycle is called an *attractor* if there is a set of transient states leading to it; these constitute the *basin of attraction*.

The dynamics can be classified [9] according to the way information spreads through the network.

1. In the *frozen phase*, all nodes apart from a small number (that remains finite in the limit of infinite system size) assume a constant value after a transient time. If in the stationary state the value of one node is changed, this perturbation propagates during one time step on average to less than one other node.
2. In the *chaotic phase* initially similar configurations diverge exponentially. Attractors are usually long, and a non-vanishing proportion of all nodes keep changing their state even after long times.

Critical networks are at the boundary between the frozen and chaotic phase [9] and neighboring configurations diverge only algebraically with time. Whether a network is frozen, critical or chaotic depends on the value of k and on the probabilities assigned to the different types of update functions. When all update functions are chosen with the same probability, networks with $k < 2$ are frozen, networks with $k = 2$ are critical and networks with $k > 2$ are chaotic [10].

The usual synchronous way of updating is not very realistic [11] as natural systems are rarely controlled by an external clock. It is known that properties of attractors for synchronous dynamics can differ from those for asynchronous dynamics. This is generally known for cellular automata, where part of the self-organization is closely tied to the synchronous updating [12]. Also for RBNs, the dynamics changes considerably when other updating schemes are chosen. For instance, when small stochastic deviations from synchronous update are introduced, only part of the attractors survive [13, 14]. For an update rule where the next node to be updated is determined at random, the number of attractors becomes a power law of the network size [15], in contrast to the superpolynomially large attractor numbers found for parallel update [16].

In contrast to stochastic updating schemes, *deterministic* updating schemes that are not fully synchronous have been investigated less. In [17], different asynchronous updating schemes are compared numerically for small network sizes with the conclusion that ‘the number of attractors does not depend too much in the synchronicity of the updates, as they depend on their determinism’. One obvious way to implement such an updating scheme would be to assign a time delay to each link between two nodes. This can be motivated biologically, since the expression of genes is not an instantaneous process, but the transcription of DNA and transport of enzymes may take from milliseconds up to a few seconds. However, a network with different delay times can be mapped on a network with synchronous update by introducing a chain of nodes with unit delay times between nodes with longer delays. For this reason, we want to focus in this paper on a different type of asynchronous deterministic networks, namely on those with a ‘quenched randomness’ in updating times. This means that some nodes are less frequently updated than others. Such networks represent an intermediate step between parallel and fully stochastic updating rules. Quenched randomness in update intervals can be motivated by the necessity for each node to have a *refractory time*, i.e. a time during which a node cannot respond to an input signal because it is still processing or recovering from the previous input signal. While a realistic implementation of update delays and refractory times would lead to more complex rules, using fixed update intervals for each node is an instructive and important

step in the investigation of not fully synchronous networks. To each node i we assign a delay time τ_i . The value σ_i of node i is updated in time intervals τ_i , and each node must be assigned an initial ‘phase’ $\varphi_i < \tau_i$ (i.e. the time until the first update). The model is referred to as a deterministic random Boolean network (DRBN). The system is deterministic as the succession of network states $\vec{\sigma}(t)$ is entirely defined by the initial condition $\vec{\sigma}(0)$ and the initial phases $\{\varphi_i\}$. The case of parallel update, the so-called classical RBN (CRBN), is a special DRBN with all $\tau_i \equiv 1$, $\varphi_i \equiv 0$. The size of the state space Ω changes from $|\Omega|_{\text{CRBN}} = 2^N$ to $|\Omega|_{\text{DRBN}} = \prod_i 2 \cdot \tau_i$, when all τ_i are integers.

The outline of the rest of this paper is the following: first, we review the concept of relevant components (section 2). This shows that the most frequent relevant components are simple loops, and less frequent are collections of loops with additional links within and between them. In the subsequent sections, we therefore study simple loops with one delayed node, simple loops with several delayed nodes, two loops with a cross-link and one delayed node and a loop with one additional link and a delayed node (sections 3–6). In the conclusion, we discuss the consequences of our findings for the entire network, which is composed of several relevant components.

2. Relevant components

It has proven useful to classify the nodes of a RBN according to their behavior on attractors [16], [18]–[23].

1. The state of *frozen nodes* becomes constant after some time. Interestingly, the nodes that become frozen are the same nodes most of the time, and they constitute the *frozen core*. The frozen core is identified by starting from nodes with constant functions and by iteratively identifying nodes that become frozen due to frozen inputs. Networks without frozen functions can also develop a frozen core [24], however, the mechanism is different. In critical networks, the frozen core comprises all but a proportion $\sim N^{-1/3}$ of nodes.
2. *Relevant nodes* are non-frozen and have different dynamics on different attractors, and they influence at least one relevant node [25]. They determine the attractors, and the number of relevant nodes scales in critical networks as $N^{1/3}$ [22].
3. Finally, there are the *irrelevant nodes* which are not frozen but are slaved by the relevant nodes.

The non-frozen nodes of critical networks essentially form a $k = 1$ network. This means that typically all but one input of a non-frozen node are frozen. In critical networks, relevant nodes are arranged in $\mathcal{O}(\ln N)$ components, most of which are simple loops. Typically, there is only one component that is not a simple loop but has μ nodes with two relevant inputs [26]. There exist two possible components with $\mu = 1$, and then more complicated components with $\mu > 1$. If $\mu = 1$, there are either two loops with a cross-link or a loop with an additional link, see figure 2. Two loops with a cross-link occur twice as often as a loop with an additional link [26].

It is important to realize that the classification of nodes into frozen, relevant and non-frozen irrelevant nodes is independent of the updating scheme. To which class a node belongs, depends only on the topology of the network and on the functions assigned to the nodes. The update time of a node does not influence the question whether a node freezes if some of its inputs are

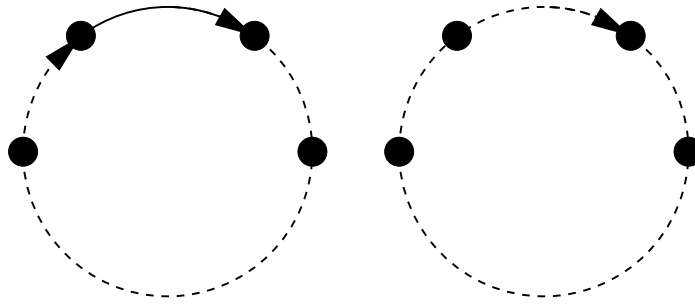


Figure 1. The two possible simple loops. The dashed lines depict a \oplus -coupling, the solid lines a \ominus -coupling. The left one is an odd, the right one an even loop.

frozen. For this reason, the relevant nodes determine the long-term dynamics of RBNs for the Nay updating scheme. The dynamics of the outgoing trees is slaved by the relevant parts, and the frozen nodes are obviously irrelevant. Thus, we will in the following study the properties of the relevant components. What we learn about their dynamics can directly be applied to the dynamics of the entire network.

3. Loops with one delayed node

The simplest relevant component is a loop consisting of nodes with $k = 1$ incoming edges and Boolean functions which either copy (\oplus) or invert (\ominus) the previous node's value. A constant function within a loop will freeze the whole loop, and can therefore not occur in relevant loops. A loop with n inversions can be mapped bijectively on to one with $(n - 2)$ inversions by replacing two \ominus with two \oplus and by inverting the values of all nodes between these two couplings. It is therefore sufficient to distinguish loops with an even or an odd number of inversions, and we call them 'even' and 'odd' loops, respectively. When discussing even loops, we consider loops with only \oplus -functions. Irrespective of the updating scheme, there are two fixed points for such an even loop, namely $\vec{\sigma} \in \{0, 1\}$. To odd loops we assign one \ominus -function, the connection to a node with this function is called 'twisted'. In contrast to the even loop it has no fixed points. For synchronous update, the shortest attractor of an odd loop has period 2, with alternating 1s and 0s. If no node is delayed, expressions for the number of attractors for loops consisting of a prime number of nodes can easily be written down, because the pattern of zeros and ones traveling around the loop is repeated only after one (even loop) or two (odd loop) full rotations (except if we have an even loop at one of its fixed points or an odd loop on the cycle of period 2). The number of cycles on an even loop of N nodes is then $2 + (2^N - 2)/N$ and on an odd loop it is $1 + (2^N - 2)/2N$. If N is not prime, shorter cycle lengths exist, which are divisors of N . Let us now introduce a delay in the loop. Let node 1 be the delayed node, i.e. we choose a value $1 < \tau_1 \equiv \tau^* \in \mathbb{N}$, while $\tau_i = 1$ for $i > 1$. This means that one gene needs much longer to be expressed than all the others. Since node 1 remains at the same value for τ^* time steps, node 2 receives τ^* times the same input, leading to blocks of size τ^* traveling around the loop. When the head of a block arrives at node N , node N will have the value of this block for τ^* time steps, and during one of these steps node 1 will be updated. In the following, we will consider even and odd loops separately.

3.1. Even loops

An even loop with one delayed node has two fixed points, just as the loop with no delayed node. The other attractors are characterized by blocks of length τ^* traveling around the loop. Assume node 1 is updated at time 0. A block with the value of node 1 will start traveling around the loop, and the head of the block will arrive at node N at time $N - 1$. The next update of node 1 will be at time $T = \lfloor (N - 1 + \tau^*) / \tau^* \rfloor \cdot \tau^*$, where the Gaussian brackets $\lfloor x \rfloor$ denote the largest integer less or equal x . The value of node 1 becomes the same as at time 0, and the same block travels around the loop again. The same consideration can be made for all starting times that are multiples of τ^* , leading to the result that the state of the loop is repeated every T time steps.

After the transient time $N - 1$, the loop has reached an attractor that contains $\xi = T / \tau^*$ blocks, each of which either has only values 0 or values 1, and with the blocks that contains node 1 and N being shorter than the other ones, if N is not a multiple of τ^* . Every attractor corresponds to a pattern of blocks traveling around the loop. We can now make use of the results obtained for even loops without delays by replacing states of nodes with states of blocks. If the number of blocks is a prime number, $\xi \in \mathbb{P}$, the length A_{\oplus} of the attractors is identical to T ,

$$A_{\oplus} = T = \left\lfloor \frac{N - 1 + \tau^*}{\tau^*} \right\rfloor \tau^* = \xi \cdot \tau^*, \quad \forall \xi \in \mathbb{P}. \quad (2)$$

For $\tau^* \geq N$ there is only $\xi = 1$ block containing all node and the fixed points are the only attractors.

The number of different attractors, ν_{\oplus} , can be calculated from the number of different patterns, 2^{ξ} . Including the 2 fixed points this leads to

$$\nu_{\oplus} = \frac{2^{\xi} - 2}{\xi} + 2, \quad \forall \xi \in \mathbb{P}. \quad (3)$$

For $\xi \notin \mathbb{P}$ the pattern of blocks can have a period that is a divisor of ξ , because the same pattern can appear already after less time than a full rotation. Therefore the mean attractor length is shorter. Since each pattern of blocks is part of an attractor, the number of attractors increases compared to loops with a prime number of blocks.

3.2. Odd loops

Without loss of generality, we assign the twisted edge of the odd loop to be in front of the delayed node. As in the synchronous case, there are no fixed-point attractors for odd loops. Let again node 1 be delayed and updated at time 0. We can now proceed in the same way as for the even loop by considering blocks. At time T , node 1 will have the opposite state as the original one. After time $2T$, node 1 returns to its original state, which implies that the loop returns to its original state after $2T$ time steps, if it is on an attractor. If all nodes are identical initially, a single domain wall travels around the loop, and after T time steps all nodes are again identical, but with the opposite state. The shortest attractor has a period $2\tau^*$, and it has alternating blocks. If the number of blocks $\xi = T / \tau^*$ is a prime number, all other attractors have the period $2T$, and the number of different attractors is

$$\nu_{\ominus} = \frac{2^{\xi} - 2}{2\xi} + 1. \quad (4)$$

If $\xi \notin \mathbb{P}$, the number of attractors increases as the length of some attractors is shorter. If $\tau^* \geq N$, there is only one attractor with period $2\tau^*$.

3.3. Non-integer delays

Let us now consider the case that τ^* is not an integer, but a rational or a real number. Real numbers can be approximated by a series of rational numbers, and we therefore consider the case $\tau^* = r/s$ with two incommensurate integers r and s , with $r > s$. During r time steps, s blocks emerge from node 1, part of them of length $\lfloor \frac{r}{s} \rfloor$, part of them of length $\lfloor \frac{r}{s} \rfloor + 1$. When the first block reaches node 1, the same sequence of blocks will emerge again only if node 1 is in the same phase at its next update as it was at its first update. Otherwise, the pattern of blocks will be changed at each circulation around the loop, until it starts repeating again after s (or $2s$ for an odd loop) circulations (or a divisor of it). It follows that for irrational values of τ^* the dynamics never become exactly periodic but are quasiperiodic. Of course, for values $\tau^* > N$, the only attractor is a fixed-point (for an even loop) or a state with only one domain wall (for an odd loop).

4. Loops with multiple delayed nodes

Next, we consider loops with multiple delayed nodes and integer delay times $\tau_i \in \mathbb{N}_{>0}$. Loops with rational values $\tau_i = r_i/s_i$ can be mapped on those with integer values of τ_i by measuring time in units of the inverse of the least common multiple of all s_i . In the following, we will first look at two special cases before we focus on the general case where updates may occur in any order.

4.1. Sequential update

We choose $\tau_i \equiv N$ and update the nodes in the order in which they occur on the loop, i.e. $\varphi_i = i - 1$ (connection-wise (cw) update) or $\varphi_i = N - i$ (counter-connectionwise (cc) update).

For cw-update, all nodes of an even loop have the same value, after every node has been updated once. Thus, we have two fixed-point attractors consisting of the two homogeneous configurations, $\vec{\sigma} \in \{\vec{0}, \vec{1}\}$. For an odd loop, the attractor has a single domain wall that travels around the loop, and the period of the attractor is $2N$.

For cc-update, node j is updated before node $j - 1$. Therefore, N update steps give the same result as 1 update step in the case of parallel update, and the results of 3 can be taken over.

4.2. Same delay, different phases

We now choose again $\tau_i \equiv N$, but we update the nodes in any order, i.e. the values of φ_i are some permutation of the numbers 0 to $N - 1$. There are two classes of nodes, according to whether a node is updated before or after its predecessor. Nodes that are updated after their predecessor have after N time steps the same state as their predecessor. Such nodes and their predecessor are therefore part of the same *effective node*. The number N^* of effective nodes is identical to the number of nodes that are updated after their predecessor. Let us give an example, for instance $N = 6$ and the updating order

$$\{\varphi_i\} = \{5, 0, 3, 1, 2, 4\}.$$

If we specify only whether a node is updated before (b) or after (a) its predecessor, this can be written as

$$\{a, b, a, b, a, a\},$$

leading to $N^* = 2$. Once we have identified the effective nodes, we can map N time steps on such a loop with sequential update on one time step on a loop of size N^* with parallel update. All results concerning attractor numbers and lengths obtained for loops with parallel update can then be transferred to loops with sequential update.

4.3. Different delay times

We now consider the general case where the delays τ_i and the phases φ_i can take any integer value. In order to determine whether the initial state of a given node influences the attractor, we proceed in the following way: we fix the state of this node, let us say, to 1, and we evaluate to which nodes this 1 propagates with time. In order to make sure that later on all 1s on the loop will be due to this initial 1, we set all other nodes to 0 and choose an even loop. When the chosen node is updated before its successor, the 1 is lost, and the initial state of this node does not affect the attractor. If the node is updated after its successor, the 1 has moved to the successor and is not yet lost. Next, we check whether the successor is updated before the 1 that is now there can propagate further. During the course of time, the 1 may spread to become a block of larger size, which continues to change its size with time. Now we consider the loop at times which are a multiple of $\tau^* = \text{lcm}(\tau_i)$. At these times, the phases of all nodes are the same as at the beginning. We wait until either the original node has again state 1 or until all 1s are gone. In the first case, the 1 will survive forever, and the initial state of the chosen node will consequently affect the attractors. In the second case, the chosen node does not affect the attractors.

If we repeat this procedure for every node, we will know the initial state of how many nodes N^* will affect the attractors. We only consider these ‘relevant’ nodes from now on, and we consider them only at times that are multiples of τ^* . Let m be the number of relevant nodes through which each block moves during τ^* time steps. If m and N^* have no common divisor, we order the N^* nodes in the sequence in which they are visited if the system is only considered at times that are multiples of τ^* . Then we have mapped the task of finding the attractor number and length on a loop of size N with any rational delay times of the task of finding the attractor number and length in a loop of size N^* . If m and N^* have a common divisor l , we can map our system on l loops of length N^*/l with parallel update and thus find the number and lengths of attractors.

For irrational delay times, the mapping on integer delays cannot be performed. Nevertheless, one can determine whether a 1 at a given node will eventually become a block that is so large that it will never vanish. If all delays have irrational ratios, there will eventually come a moment where all nodes are updated cw, and from then on there is only one block left.

5. Two loops with a cross-link

Now we consider a complex component consisting of a loop with N_1 nodes connected to a loop with N_2 nodes, see figure 2(a). The node Σ is the one with two inputs and its input nodes are labeled G_1 and G_2 . The first loop is either odd or even, the second loop can without loss of generality be chosen such that it has only \oplus -couplings, except at Σ . We insert no nodes between

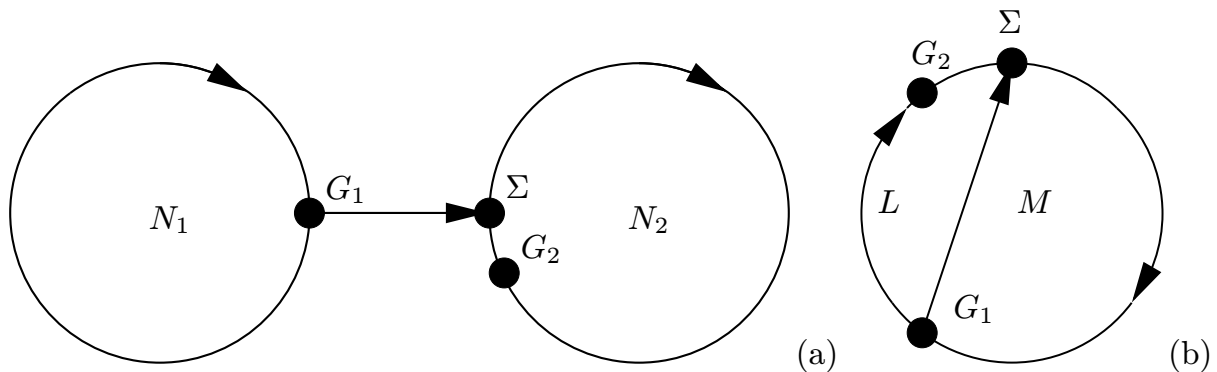


Figure 2. The complex components considered in this paper: (a) the $\circ-\circ$ -component consists of a loop with N_1 nodes connected to a loop with N_2 nodes. (b) the \circ -component is a loop of $N = L + M + 2$ nodes having an additional link.

Table 1. Boolean functions used in the $\circ-\circ$ -component for node Σ . The second letter in the label of the canalizing functions stands for ‘homogeneous’ or ‘inhomogeneous’.

Inputs		Reversible	Canalizing functions	
G_1	G_2	f_{rev}	f_{ci}	f_{ch}
0	0	1	1	0
0	1	0	0	1
1	0	0	1	1
1	1	1	1	1

G_1 and Σ , as a system with m nodes on the cross-link can be mapped on a system with a direct link by connecting node number m (counted clockwise from G_1) directly with node Σ .

We consider only the nontrivial cases where the coupling function of node Σ is a function that responds to both of its inputs. If we take into account that certain of these functions can be mapped on to each other by inverting the states of all nodes or by inverting the states on the first loop, we end up with three functions that are truly different. They are shown in table 1.

In the following, the results by Kaufman and Drossel [26] for components under synchronous update will be generalized to components with one delayed node.

5.1. Delayed node on first loop

If the first loop has a delay τ^* , the value of G_1 can change only at times that are multiples of τ^* , and the pattern of change is repeated after the attractor period of the first loop. If the first loop is on a fixed-point, the second loop can be considered as an independent loop with a function at Σ , which depends on the fixed-point value of the first loop. We therefore consider here the case that the first loop is not on a fixed-point but provides a periodic input of period p_1 to Σ , with blocks of size τ^* of identical bits. The second loop then behaves like an independent loop where the Boolean function at Σ changes after τ^* steps, and where the pattern of changes is repeated periodically with period p_1 .

If $f_{\Sigma} = f_{\text{rev}}$, the second loop switches between truth and negation, for the (in)homogeneous canalizing function f_{ch} (f_{ci}) the second loop changes between truth (negation) and constant value

$\sigma_\Sigma = 1$. The attractor period is at most $2p_1N_2$. More detailed results for possible attractors of such a system under synchronous updating can be found in [26], the only difference being that p_1 is now related in a different way with N_1 . An interesting finding is that for the homogeneous canalizing function, the second loop becomes frozen on the value 1 if p_1 and N_2 have no common divisor. Furthermore, for the inhomogeneous canalizing function, the first loop enslaves the second loop and imposes its period on it, if p_1 and N_2 have no common divisor.

5.2. Delayed node on second loop

We now proceed to the case where a single delayed node is on the second loop. The first loop behaves in the same way as for simple (synchronously updated) CRBN-loops. We focus on the sequence of values of the delayed node. We assign the delay to the node Σ : a system with the delayed node m nodes after Σ can be transformed into a system with the delay at Σ by rotating the first loop m nodes counterclockwise.

Node Σ responds to the input from G_1 only every τ^* time steps. Let us denote with p_1 the period of the sequence of values of G_1 every τ^* time steps, which is the period of the input sequence to Σ generated by the first loop at those times where Σ is updated. Let $\xi = \lfloor (N_2 - 1 + \tau^*) / \tau^* \rfloor$ denote the number of blocks of the second loop.

All results for the attractors on two loops with a cross-link and no delay can now taken over by replacing N_2 with ξ , by replacing nodes with blocks, and by taking τ^* as the time unit. In particular, for a reversible function f_Σ , the largest period is $p_1\xi\tau^*$. A homogeneous canalizing function f_Σ freezes the second loop on the value 1 if p_1 and ξ have no common divisor. Furthermore, for an inhomogeneous canalizing function, the first loop enslaves the second loop and imposes its period (times τ^*) on it if p_1 and ξ have no common divisor.

5.3. General case

Now we consider the case of multiple (integer) delays on both loops. Let the first loop have a period p_1 and the second loop (if even and decoupled from the first loop) a period p_2 . The general system of two interconnected loops without any delay has been studied in [26]. There, it was shown that the attractor length lies between p_1 and $2p_1N_2/g$, where $g = \text{lcm}(p_1, N_2)$. We can conclude that now the attractor length lies between p_1 and $2p_1p_2/g$, where g is the greatest common divisor of p_1 and p_2 .

For a homogeneous canalizing function $f_\Sigma = f_{\text{ch}}$, the second loop is frozen if p_1 and p_2 are incommensurable. The longest attractors occur for reversible functions, $f_\Sigma = f_{\text{rev}}$.

6. Loop with one additional link

The other complex component with one node with two inputs is a loop with $N = L + M + 2$ nodes and one additional link. We call again the node with two inputs Σ and its inputs G_1 and G_2 , see figure 2(b). The link from G_2 to Σ can be treated as a direct link: a system with $n < L$ nodes in the additional link can be mapped on to a system with a direct link by connecting node $(M + 1 + n)$ to Σ (if we neglect delays). We consider five update functions at Σ , compare table 2 (we now use the common decimal representation as identifiers for the functions). The other canalizing or reversible functions yield the same result, one only has to invert the output values of the truth table. Without loss of generality, all other functions in the loop are copy functions.

Table 2. Boolean functions used in the \otimes -component for node Σ . The names for the functions are the decimal representation of the corresponding column of outputs, for instance $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 = 13$. Function f_9 is reversible and the other functions are canalizing.

G_1	G_2	f_1	f_2	f_{13}	f_{14}	f_9
0	0	1	0	1	0	1
0	1	0	1	0	1	0
1	0	0	0	1	1	0
1	1	0	0	1	1	1

Let there be one delayed node in the component. We distinguish two cases according to the position of the node with update period $\tau^* > 1$.

1. The delayed node lies on the first $M + 1$ nodes (including G_1). Without loss of generality the delay can be shifted to node Σ .
2. The delayed node is in the chain of nodes between G_1 and Σ and can then be shifted to G_2 .

In the first case, the component can be reduced to a network of effective nodes by looking at the network only every τ^* time steps. Each effective node corresponds to a block of τ^* nodes which are at the same state for $t \bmod \tau^* = 0$. The results for the synchronous case (as studied in [26], section 4) hold for the effective variables \tilde{N} , \tilde{M} , $\lceil x \rceil$ denotes the smallest integer greater or equal to x :

$$\tilde{N} = \left\lceil \frac{N}{\tau^*} \right\rceil, \quad \tilde{M} = \left\lceil \frac{M+2}{\tau^*} \right\rceil - 2. \quad (5)$$

In the following we will study the second case, where G_2 is the delayed node.

6.1. Canalizing function at Σ

If there is a canalizing function at Σ , we can classify the attractors in at least two types. In the following we will present analytical considerations for these types of attractors and compare the analytical predictions with the numerical results.

The first type of attractors is obtained by requiring that G_2 does never have its canalizing value at the moment when Σ is updated. (The canalizing value is 0 for f_{13} and f_2 and 1 for f_1 and f_{14} .) In this case, the loop consisting of the $M + 2$ nodes from Σ to G_1 is an even loop for f_{13} and f_{14} and an odd loop for f_1 and f_2 . Just before node Σ is updated, the state of node G_2 and the state of all nodes that will in $n\tau^*$ time steps determine the state of G_2 must have a value such that Σ never has its canalizing value (n is any positive integer). For functions f_{13} and f_{14} , this condition fixes the entire component at the same value if $M + 2$ and τ^* have no common divisor. If their greatest common divisor g is larger than 1, only the value of every g th node on the loop of length $M + 2$ is fixed by this condition. The number of attractors of the first type is therefore that of an even loop with $(M + 2)(g - 1)$ nodes and with no delays. For functions f_1 and f_2 , the condition that G_2 does never have its canalizing value can only be satisfied if τ^*/g is even. The number of attractors of the first type is then that of an odd loop with $(M + 2)(g - 1)$

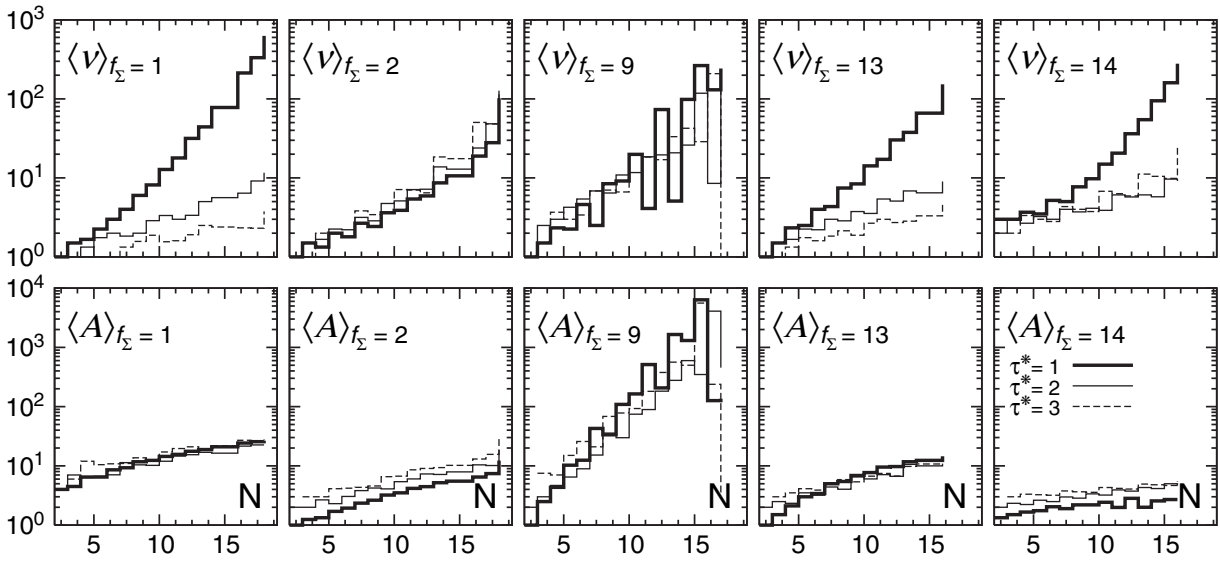


Figure 3. Results of the exhaustive state space search for the \varnothing -component with small system size, $N < 20$. The upper panel shows the mean attractor number $\langle \nu \rangle$ in dependence of the system size N while the lower shows the mean attractor length $\langle A \rangle$ for fixed delay τ^* and for all functions $f_{\Sigma} \in \{1, 2, 9, 13, 14\}$. In both cases the average was taken over all possible realizations which corresponds to an average over all L for a single delayed node. To keep the diagrams concise only some delays have been plotted.

nodes and with no delays. Compared to a component with no delays, this new type of attractors increases the mean attractor length if the canalizing function is f_{14} . Without delays, only very short attractors (apart from two fixed points) can occur [26] for f_{14} , and the increase in attractor length for values $\tau^* > 1$ is clearly visible in the exhaustive state space search shown in figure 3.

On all other attractors, G_2 has at least sometimes its canalizing value. The second type of attractors referred to above is obtained if $M + 2$ is a multiple of τ^* . Again, we can derive properties of these attractors from analytical considerations: if the loop of $M + 2$ sites consists of blocks of size τ^* , the dynamics can be mapped on that of an effective component with $\tilde{N} = \lceil \frac{N}{\tau^*} \rceil$ nodes and with $\tilde{M} = \frac{M+2}{\tau^*} - 2$ with no delay but time steps of length τ^* , and the results of [26] can be taken over. In addition to attractors consisting only of homogeneous blocks, further attractors can be constructed by realizing that only one bit in each block is the one that triggers node G_2 . The value of bits that do not trigger node G_2 does only matter when the block reaches G_1 : if at this moment Σ is not canalized by G_2 , an inhomogeneous block will not be homogenized, but copied or inverted to node Σ . An inhomogeneous block can therefore survive forever if the blocks that are at G_2 at the moment where the inhomogeneous block is at G_1 do not have their canalizing value. However, this implies that these non-canalizing blocks are copied to Σ from G_1 , which is only possible for f_{13} . Indeed, from [26] we know that a period $\tilde{M} + 2$ of attractors on the effective component is only possible for this function, unless \tilde{N} has special values.

Finally, let us look for nontrivial attractors that can occur even if $M + 2$ and τ^* have no common divisor. Let us choose the function f_2 . An isolated block of size τ^* of 1s in the initial

state will survive forever since there will be a 0 at G_1 while this block is at G_2 . In fact, there exist a multitude of such attractors where there is a 0 at the right position at distance L behind a block of 1 s. This explains why the number of attractors found numerically for $\tau^* > 1$ in the exhaustive state space search shown in figure 3 is larger for f_2 than for the other canalizing functions. The length of these attractors can be larger than $N + \tau^* - 1$, as can also be seen in figure 3.

6.2. Function f_9 at Σ

f_9 is a reversible function, i.e. if one of the inputs changes its value the output changes, too. If there are no delays, the dynamics is reversible and therefore all states are on cycles. There is 1 fixed point $\vec{\sigma} = 1$. A striking feature of the synchronous case is that cycles of the order 2^N exist [26].

The exhaustive numerical attractor search suggests that the maximal attractor length can be approximated by τ^* times the maximal attractor length in the non-delayed case for odd τ^* .

7. Conclusion

In this paper, we studied the influence of a deterministic but non-synchronous update on a critical Kauffman network by introducing node-based delays. The dynamics of critical networks can be derived from the dynamics on its relevant components, most of which are simple loops and some of which have a few nodes with two inputs. For this reason, we have studied in this paper the three simplest types of relevant components. Not surprisingly, delays typically increase the attractor lengths and reduce the attractor numbers. New types of attractors emerge in the presence of delays. The basins of attraction are naturally larger and thus the path to the attractor becomes more robust. If all delays are randomly chosen real numbers, loops are most likely to be frozen or on a single attractor. Similarly, more complex components with real delays should have far less attractors than for parallel update.

The attractors of the entire network are obtained from the attractors of the relevant components by the usual considerations [22]: the number of relevant components is of the order of $\ln N$ and usually only the largest component (if any) is more complex than a simple loop. The length of an attractor of the entire network is the least common multiple of the attractor lengths on the components. The attractor number of the network is at least as large as the product of the attractor numbers of the components. The main conclusions obtained for components and summarized above can therefore be generalized to the entire network.

It will be interesting to see how these results are affected when nonrandom networks, such as real gene regulation networks, are considered. Clearly, they are not updated in parallel. Some networks, such as in budding yeast [7] appear to be very robust with respect to the introduction of delays. This means that their choice of connections and functions is such that the update sequence does not matter much. It remains to be seen if this is a general feature of all those networks that can be described by using a Boolean idealization.

Acknowledgment

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under contract no. Dr200/4-1.

References

- [1] Kauffman S A 1969 Metabolic stability and epigenesis in randomly constructed genetic nets *J. Theor. Biol.* **22** 437–67
- [2] Kauffman S 1969 Homeostasis and differentiation in random genetic control networks *Nature* **224** 177–8
- [3] Alexander J Mc 2002 Random Boolean networks and evolutionary game theory *Philosophy of Science Assoc. 18th Biennial Mtg*
- [4] Paczusi M, Bassler K E and Corral Á 2000 Self-organized networks of competing Boolean agents. *Phys. Rev. Lett.* **84** 3185–8
- [5] Kauffman S, Peterson C, Samuelsson B and Troein C 2003 Random Boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. USA* **100** 14796–9
- [6] Bornholdt S 2005 Less is more in modeling large genetic networks. *Science* **310** 449–51
- [7] Li F, Long T, Lu Y, Ouyang Q and Tang C 2004 The yeast cell cycle network is robustly designed. *Proc. Natl. Acad. Sci.* **101** 4781–6
- [8] Albert R and Othmer H G 2003 The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster* *J. Theor. Biol.* **223** 1–18
- [9] Aldana-Gonzalez M, Coppersmith S and Kadanoff L P 2003 Boolean dynamics with random couplings. *Perspectives and Problems in Nonlinear Science. A Celebratory Volume in Honour of Lawrence Sirovich* ed E Kaplan, J E Marsden and K R Sreenivasan (Berlin: Springer) pp 23–89
- [10] Derrida B and Pomeau Y 1986 Random networks of automata: a simple annealed approximation *Europhys. Lett.* **1** 45–9
- [11] Harvey I and Bossomaier T 1997 Time out of joint: attractors in asynchronous random Boolean networks *Proc. 4th Eur. Conf. on Artificial Life (ECAL97)* ed P Husbands and I Harvey (Cambridge, MA: MIT Press) pp 67–75
- [12] Ingerson T E and Buvel R L 1984 Structure in asynchronous cellular automata *Physica D* **10** 59–68
- [13] Klemm K and Bornholdt S 2005 Stable and unstable attractors in Boolean networks *Phys. Rev. E* **72** 055101
- [14] Klemm K and Bornholdt S 2005 Robust gene regulation: deterministic dynamics from asynchronous networks with delay *Phys. Rev. E* **055101**
- [15] Greil F and Drossel B 2005 Dynamics of critical Kauffman networks under asynchronous stochastic update *Phys. Rev. Lett.* **95** 048701
- [16] Mihaljev T and Drossel B 2006 Scaling in a general class of critical random Boolean networks *Phys. Rev. E* **74** 046101
- [17] Gershenson C 2002 Classification of random Boolean networks *Artificial Life VIII: Proc. Eight Int. Conf. on Artificial Life* ed R K Standish, M A Bedau and H A Abbass (Cambridge, MA: MIT Press) pp 1–8
- [18] Bilke S and Sjunnesson F 2002 Stability of the Kauffman model *Phys. Rev. E* **65** 016129
- [19] Socolar J E S and Kauffman S A 2003 Scaling in ordered and critical random Boolean networks *Phys. Rev. Lett.* **90** 068702
- [20] Samuelsson B and Troein C 2003 Superpolynomial growth in the number of attractors in Kauffman networks *Phys. Rev. Lett.* **90** 098701
- [21] Drossel B 2005 On the number of attractors in random Boolean networks *Phys. Rev. E* **72** 016110
- [22] Kaufman V, Mihaljev T and Drossel B 2005 Scaling in critical random Boolean networks *Phys. Rev. E* **72** 046124
- [23] Bastolla U and Parisi G 1998 The modular structure of Kauffman networks *Physica D* **115** 219–33
- [24] Paul U, Kaufman V and Drossel B 2006 Properties of attractors of canalizing random Boolean networks *Phys. Rev. E* **73** 026118
- [25] Bastolla U and Parisi G 1998 Relevant elements, magnetization and dynamical properties in Kauffman networks. A numerical study *Physica D* **115** 203–18
- [26] Kaufman V and Drossel B 2005 On the properties of cycles of simple Boolean networks *Eur. Phys. J. B* **43** 115–24