Fixed-points in Random Boolean Networks: The impact of parallelism in the Barabási–Albert scale–free topology case ☆

P. Moisset de Espanés[a,c], A. Osses[a,b,*], I. Rapaport[a,b]

[a] *Center for Mathematical Modeling (UMI 2807 CNRS), FCFM, Universidad de Chile, Beauchef 851, Santiago, Chile*
[b] *Departamento de Ingeniería Matemática, FCFM, Universidad de Chile, Beauchef 851, Santiago, Chile*
[c] *Center of Biotechnology and Bioengineering, FCFM, Universidad de Chile, Beauchef 851, Santiago, Chile*

Abstract

Fixed points are fundamental states in any dynamical system. In the case of gene regulatory networks (GRNs) they correspond to stable genes profiles associated to the various cell types. We use Kauffman's approach to model GRNs with random Boolean networks (RBNs). In this paper we explore how the topology affects the distribution of the number of fixed points in randomly generated networks. We also study the size of the basins of attraction of these fixed points if we assume the α-asynchronous dynamics (where every node is updated independently with probability $0 \leq \alpha \leq 1$). It is well-known that asynchrony avoids the cyclic attractors into which parallel dynamics tends to fall. We observe the remarkable property that, in all our simulations, if for a given RBN with Barabási–Albert topology and α-asynchronous dynamics an initial configuration reaches a fixed point, then *every* configuration also reaches a fixed point. By contrast, in the parallel regime, the percentage of initial configurations reaching a fixed point (for the same networks) is dramatically smaller. We contrast the results of the simulations on Barabási–Albert networks with the classical Erdös–Rényi model of random networks. Everything indicates that Barabási–Albert networks are extremely robust. Finally, we study the mean and maximum time/work needed to reach a fixed point when starting from randomly chosen initial configurations.

Keywords: Random Boolean Networks, fixed points, attractors

1. Introduction

A *Random Boolean Network* (RBN) is a model of a GRN. This model was introduced by Kauffman in 1969 [34] and it corresponds to a directed graph composed by N genes (nodes) where each of these genes can be either expressed (state 1) or not expressed (state 0). Each

☆This work was partially supported by CONICYT via Basal in Applied Mathematics, Fondecyt 1130061 (I.R.), Núcleo Milenio Información y Coordinación en Redes ICM/FIC RC130003 (I.R.), Basal-FB0001 (P.M.), and Fondecyt grant numbers 1151512 and (A.O.).

*Corresponding author

Email addresses: pablo.moisset@gmail.com (P. Moisset de Espanés), axosses@dim.uchile.cl (A. Osses), rapaport@dim.uchile.cl (I. Rapaport)

gene v receives K randomly chosen genes as input (K nodes pointing to v). In other words, the dynamics is defined *locally*. Kauffman proposed to choose, independently for each node v, any of the 2^{2^K} possible functions $f_v : \{0,1\}^K \to \{0,1\}$ with equal probability. Finally, a global dynamics $f : \{0,1\}^N \to \{0,1\}^N$ was defined by applying the local functions *in parallel* (sometimes this is referred to as *synchronous updates*). Kauffman found, through simulations, the existence of a phase transition at $K = K_c = 2$ (from an ordered phase to a chaotic phase). He also suggested that the number of attractors grows exponentially with N in the chaotic phase, proportional to N in the ordered phase, and proportional to $\sqrt{N}$ at the critical value $K_c = 2$.

Kauffman's model (and some natural generalizations) motivated a lot of work carried out mainly by physicists. They tackled problems arising from Kauffman's definitions both analytically and numerically: the critical in-degree value [10, 20], the number of attractors [13, 16, 20], the length distribution of the attractors [5, 12], etc. Robustness with respect of different perturbations has been studied. For example in [26, 54] they consider noise in function evaluation, while in [37, 35, 44, 53] they consider node state perturbations. In [52] a negative correlation between modularity (a topological property) and robustness was found. Oscillating behavior and the period of the oscillations is related to the existence and length of rings in the networks [46]. The fact that regulatory networks arose from an evolutionary process imposes some topological properties on them, as it is described in [41]. Evolution of robustness under selective pressure is described in [45] and approaches to design robust networks can be found in [36, 40, 48]. In [50] the effect on robustness of assigning update functions to nodes based on local topological properties is analyzed. All these publications assume synchronous updates. In [8] other deterministic update schedules are analyzed.

From the biologists' point of view, a fundamental challenge of the post-genomic era is the possibility of simulating the dynamics of *real* genetic networks. The enormous amount of available data of molecular interactions within the cell made it possible to examine critically the original RBN model. The first observation was that the topology assumption was inadequate. Contrasting the uniform topology assumed in the original RBNs, it has been shown that real genetic networks exhibit a *scale-free* topology [4, 14, 25, 43]. In such topologies a small fraction of the genes are highly connected whereas the majority of the genes are poorly connected [2, 51]. Therefore, RBNs dynamics with scale-free topology started to be intensively studied [6, 15, 29, 32, 42]. As a good example of topological features influencing the dynamics of a GRN, it is shown in [46] that oscillating behavior and the period of the oscillations are related to the existence and length of rings in the network.

Another criticism of Kauffman's model was that nodes were updated in parallel. Experimental results confirmed a rather intuitive fact: that genes transition between expressed and non-expressed states at different times [18, 22]. Informally, there is no global clock that allows transition to happen only at ticks. Therefore, RBNs with scale-free topology and asynchronous dynamics is a natural model to be analyzed [17]. We would also like to point out that asynchrony in the classical RBN model has also been studied in [28, 30, 39, 46, 49].

Our contribution. *Fixed points* are fundamental states in any dynamical system. In the case of GRNs they correspond to stable gene expression profiles associated to the various cell types. This interpretation has been used for modeling europhil differentiation [31], expression patterns of the segment polarity genes in *Drosophila melanogaster* [3], flower organ specification in *Arabidopsis thaliana* [7], etc.

The goal of this paper is to study the existence of fixed points in RBNs with Barabási–Albert topology. Notice that a state being a fixed point does not depend on the timing (i.e. synchronous

vs. asynchronous) of the update rule. In fact, in a fixed point every gene is in a stable state with respect to its local input. Therefore, there is no way to change such global configuration. In that sense, a fixed point is a very robust object of a network, despite the fact that its basin of attraction *can change* depending on how the update rule is implemented.

Once the topology of the network is (randomly) generated some nodes will have only outgoing arcs. These were called *source nodes* by Albert [4] and we will use the same terminology here. Since their states do not depend on the state of any other node we fix their values arbitrarily. In this paper we prove that, for every given assignment to the input nodes, the expected number of fixed points is exactly one (for *every* topology).

Now the main (and natural) question arises. Given a RBN, how can we actually *find* its fixed points? It is clear that testing all the 2^N configurations is impossible. For answering the question we come back to the issue of asynchrony. In fact, in 1994 Bersini and Detours studied an asynchronous version of the cellular automaton Game-of-Life [11]. They observed that the introduction of asynchrony modified the dynamics from a behavior with *long transients* to a behavior with fixed points. This is rather intuitive: asynchrony is a way to avoid the cycle attractors the deterministic (parallel) implementation tend to fall into. Roughly speaking, given a RBN we implement the α-asynchronous dynamics for different values of α ($0 \leq \alpha \leq 1$) [24]. This means that each gene is updated independently with probability α at each time step. When α varies from 1 down to 0 the dynamics evolves from the fully deterministic synchronous regime to a more asynchronous regime. When $\alpha = 0$ we choose randomly only one node at each step.

Our simulations show that RBNs with Barabási–Albert topology for which there exist fixed points *every* initial configuration converges to a fixed point when $0 \leq \alpha < 1$. On the other hand, when $\alpha = 1$, the percentage of initial configurations that reached a fixed point varies greatly from one network to another. In some cases the percentage is close to 0. In average (considering all the networks we use) the percentage is $\sim$28.9%. It's worthwhile to note that the tendency to fall into a fixed point is interesting from different points of view. Biologists would be interested in the implications of choosing different update rules in their models, while the problem of finding fixed points is of independent interest for a theoretician. If anything, our experiments show that asynchronous updates provide a very robust heuristic to find fixed points. This contrasts with a more analytic approach, such as the algorithms presented in [16, 21].

In order to find properties which could be associated exclusively to the topology of the network we compare the results of the simulations on Barabási–Albert networks with the results on the classical Erdös–Rényi model of random networks [23]. The main difference with the Barabási–Albert topology is that here we generated networks for which the percentage of initial configurations converging to a fixed point is close to 0. Finally, we study the mean and maximum time/work needed to reach a fixed point when starting from randomly chosen initial configurations.

Therefore, a remarkable and distinguishable dynamical property arise on RBNs with Barabási–Albert topology: robustness of convergence under asynchronous update. This fact could provide some insight about why such topologies are ubiquitous in GRN. Furthermore, asynchronous updating could be a natural mechanism present in GRNs in order to avoid cyclic dynamics [49]. We limited our study to Barabási–Albert topologies, which are a particular case of small–world networks. This, combined with additional choices we made for the boolean network generation algorithm, resulted in scale–free GRNs which are in the ordered phase regime. We observe that changing from a synchronous ($\alpha = 1$) to an asynchronous($\alpha < 1$) update rule dramatically increases the "stability" of the networks in terms of the fraction of trajectories that reach eventually a fixed point. Because of reasons we will state in Section 2.2, we choose parameters to generate

Erdös–Rényi topologies that produce GRNs lying on the boundary between the chaotic and ordered phase regions. These networks also become more stable when switching from synchronous to asynchronous updates, but the effect is not as marked as in the Barabási–Albert networks. We defer considering other families of small–world topologies and scale–free exponents that induce GRNs in the chaotic phase regime to future research.

2. Network model

A *Random Boolean Network (RBN)* corresponds to a directed graph composed by N genes (nodes) where each of these genes can be either expressed (state 1) or not expressed (state 0). We will refer to the nodes of a RBN as $v_1, v_2, \ldots, v_N$. We define K_i^{in} as the in-degree of node v_i and K_i^{out} as the out-degree of node v_i. Every zero in-degree node is called a *source node*, while every non-zero in-degree node is called an *internal node*. A *configuration* of the network is a vector $\mathbf{s} \in \{0,1\}^N$ that associates a binary state to each of the nodes.

2.1. Dynamics

We assign to each gene v_i a *local transition rule* $\phi_i : \{0,1\}^{K_i^{in}+1} \to \{0,1\}$. Informally, the value of ϕ_i depends on the state of the K_i^{in} input nodes together with the state of v_i itself. source nodes always remain in the same state. More precisely, if $K_i^{in} = 0$, then $\phi_i(0) = 0$ and $\phi_i(1) = 1$. For each internal node v_i we construct randomly its local transition function as follows. Call k the in-degree of v_i (i.e, $k = K_i^{in}$). There are 2^{2^k} possible functions of the form $f : \{0,1\}^k \to \{0,1\}$. A straightforward approach is to choose one of these functions from a uniform probability distribution. Nevertheless, before selecting a function, we should rule out those which do not strictly depend on all of its arguments (otherwise we would not be respecting the network topology). To define this concept precisely, we say that a function $f : \{0,1\}^k \to \{0,1\}$ *strictly depends on its arguments* iff for all $j \in \{1, 2, \cdots, k\}$, there exist $x_1, x_2, \cdots, x_{j-1}, x_{j+1}, \cdots, x_k \in \{0,1\}$ such that $f(x_1, x_2, \cdots, x_{j-1}, 0, x_{j+1}, \cdots, x_k) \neq f(x_1, x_2, \cdots, x_{j-1}, 1, x_{j+1}, \cdots, x_k)$. We fix function ϕ_i by selecting one (randomly) among all those functions strictly depending on all of its arguments.

The global update rule is characterized by a real parameter $\alpha \in [0,1]$. We denote the global transition rule by $\mathbf{\Phi}_\alpha : \{0,1\}^N \to \{0,1\}^N$, and we define it through the following protocol:

1. **Select** each internal node independently with probability α. We call *selected nodes* this set of randomly chosen internal nodes. For the special case $\alpha = 0$ we select randomly *a single* internal node (i.e, the set of selected nodes is a singleton).
2. **Update** in parallel all the selected nodes (applying the local transition rule in all the nodes belonging to such set). Do not change the state of the other nodes (input nodes and non-selected nodes).

Let $\mathbf{s}_t \in \{0,1\}^N$ be a configuration of a RBN at time $t \in \mathbb{N}$. A *stochastic trajectory*, starting from the initial configuration $\mathbf{s}_0$, is the sequence $\mathbf{s}_0, \mathbf{s}_1, \mathbf{s}_2, \ldots$, where $\mathbf{s}_i = \mathbf{\Phi}_\alpha(\mathbf{s}_{i-1})$.

The parameter α can be thought of as a measure of parallelism in the update process. The strictly sequential-random policy rule is captured by $\alpha = 0$, where only one internal node is updated at each step. Similarly, by making $\alpha = 1$, we represent the full parallel-deterministic policy rule, where all the internal nodes are updated at each step. If $\alpha = 1$ we call the resulting trajectory the *deterministic trajectory* of the system. A configuration $\mathbf{s}$ is a *fixed point* of a RBN

iff $\Pr\{\mathbf{\Phi}_\alpha(\mathbf{s}) = \mathbf{s}\}$=1. This is equivalent to say that $\mathbf{\Phi}_1(\mathbf{s}) = \mathbf{s}$. Therefore, $\mathbf{s}$ is a fixed point *regardless of the choice of* α.

We measure time simply by counting the applications of $\mathbf{\Phi}_\alpha$. Therefore, the time to go from $\mathbf{s}_0$ to $\mathbf{s}_T$ in a trajectory is T. This notion of time neglects the fact that the computational effort to evaluate $\mathbf{\Phi}_\alpha$ depends on α. The expected number of ϕ's to be evaluated is αN_I where N_I is the number of internal nodes. Thus, we define the *work* to go from state $\mathbf{s}_0$ to $\mathbf{s}_T$ to be αT for all $\alpha > 0$. For the special case $\alpha = 0$ we define the work as $\frac{1}{N_I}T$.

When considering the dynamics of the network with $\alpha = 1$, it is clear that the trajectory will eventually become cyclic. Note that if the system reaches a fixed point, the length of the period is 1. Both the time to enter the cycle and the period are bounded by 2^{N_I}, where N_I is the number of internal nodes. However, if $\alpha < 1$, the idea of cycle is insufficient to describe a trajectory that does not reach a fixed point. Instead, we have a set of configurations which are visited infinitely often. This greatly complicates the numerical determination of any eventual convergence to a fixed point. We therefore select a somewhat arbitrary time horizon to interrupt our simulations.

Notice that, given an initial configuration $\mathbf{s}_0$, the set of configurations reachable from $\mathbf{s}_0$ by repeated applications of $\mathbf{\Phi}_1$ is a subset of the set of configurations potentially reachable from $\mathbf{s}_0$ by repeated applications of $\mathbf{\Phi}_\alpha$ when $\alpha < 1$. In other words, the stochastic trajectory can visit a larger portion of the state space than the deterministic one.

A remarkable feature of selecting the local update functions randomly, as we did, is that we can predict, in a statistical sense, the number of fixed points in the network. This is a particular case of a fact stated in [47] (Equation 35), where they analyze the expected number of attractors of arbitrary length L. We include our own derivation to make the paper self-contained. Besides, since we are using asynchronous updates, cases where $L > 1$ are irrelevant. This allows us to write a much shorter and elementary proof.

Consider an arbitrary configuration $\mathbf{s} \in \{0,1\}^N$. What is the probability of $\mathbf{s}$ to be a fixed-point? First notice that $\mathbf{s}$ is a fixed point if and only if $\mathbf{s}_i = \phi_i(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_{k_i}})$ for every $i \in \{1, \ldots, N\}$, where $v_{i_1}, \ldots, v_{i_{k_i}}$ are the input nodes of v_i.

It follows from the definition of the ϕ_i's functions, that

$$\Pr\{\phi_i(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_k}) = 0\} = \Pr\{\phi_i(\mathbf{s}_{i_1}, \ldots, \mathbf{s}_{i_k}) = 1\} = \frac{1}{2}$$

for every internal node v_i. The reason for that is the following: if a function ϕ_i strictly depends on all of its arguments then the complementary function (the one where we replace 1's with 0's and 0's with 1's) also does. On the other hand, $\Pr\{\phi_i(\mathbf{s}_i) = \mathbf{s}_i\} = 1$ for every source node v_i. Therefore, the probability of $\mathbf{s}$ to be a fixed-point is 2^{-N_I}, where N_I is the number of internal nodes.

Let $X_\mathbf{s}$ be the Boolean random variable that equals 1 if the configuration $\mathbf{s}$ is a fixed point and 0 otherwise. The expected number of fixed points is

$$\sum_{\mathbf{s}\in\{0,1\}^N} \mathbb{E}(X_\mathbf{s}) = \sum_{\mathbf{s}\in\{0,1\}^N} 2^{-N_I} = 2^{N-N_I}$$

Note that $N - N_I = N_E$ is the total number of source nodes of the network. If we fix all these source nodes to some arbitrary value in the initial configuration, then the expected number of fixed points goes down to 1. In fact, in that case, the number of effective different configurations is $2^{N-N_E} = 2^{N_I}$ and the expected number of fixed points becomes $2^{N_I-N_I} = 1$ instead of the original 2^{N-N_I}.

Until this point, neither the definitions nor the theorem we just described make assumptions about the topology of the RBN. Note that the theorem is only about the expectation of the number of fixed points, and gives no further information about the probability distribution of them. As we will see later on, the topology of the network does influence this distribution. In the following subsection we will describe two families of topologies that have been studied in the literature.

2.2. Topology

We start describing here a process by which we construct a *directed BA network* with N nodes and average in/out-degree equal to k. Let N_0 and k be positive integers such that $k \leq N_0 \leq N$. The process starts from a directed clique with N_0 nodes (i.e, $N_0(N_0 - 1)$ arcs). Call these nodes $v_1, v_2, \ldots, v_{N_0}$. The process now involves $N - N_0$ growth stages, numbered $N_0 + 1, N_0 + 2, \ldots, N$. At each stage, a single node is added to the network.

Call v_i the node added at stage i. We will also add k edges to the growing network. We toss a fair coin and proceed as follows:

1. In the case of heads we add k edges pointing from k different nodes in $\{v_1, \ldots v_{i-1}\}$ towards v_i. These k nodes are selected randomly following a *preferential attachment rule* such that the probability of v_j to be selected is proportional to $K_j^{out} + 1$.
2. In the case of tails we add k edges pointing from v_i to k different nodes in $\{v_1, \ldots v_{i-1}\}$. These k nodes are selected randomly following a *preferential attachment rule*, such that the probability of v_j to be selected is proportional to $K_j^{in} + 1$.

We will refer to the process just described as the *BA algorithm* because it is based on previous work by Barabási and Albert [9]. Following [32], we started with a clique of size $N_0 = 5$ and average in/out-degree $k = 2$ to create the topology using the BA method. If a RBN has a topology created by the BA algorithm, we will call it a *BA network*. An interesting property of BA networks is that the scale–free exponent is 3, regardless of the value of k. This exponent has an important influence on the dynamics, as was shown in [5].

The BA method, as defined here, never creates an edge from some node to itself. The same assumption is present in [9] and [32] and is pervasive in the literature. It is noteworthy that the theorem proved in Subsection 2.1 does not require a special topology. Therefore, it still applies to networks where auto-regulation is present. Thus, for given values for the input nodes, the expected number of fixed points would still be 1. However, it is conceivable that the probability distribution of the number of fixed points will change.

In the next sections, we compare the results on BA networks with the classical Erdös–Rényi model of random networks [23] (random directed graphs). By doing so we intend to find properties which could be associated exclusively to the topology of the network. This type of comparison to assess the influence of topology is common in this field of research, see for example [29]. Random networks of this well-known Erdös–Rényi family can be easily generated. Let p be a real number in the $[0, 1]$ interval. Each potential link (v_i, v_j) is selected independently with probability p. Therefore, the expected number of links in the Erdös–Rényi network is $pn(n-1)$, where n is the number of nodes in the Erdös–Rényi network. We will call this process the *ER algorithm*. If a RBN has a topology created by the ER algorithm, we will call it an *Erdös–Rényi network* or *ER network*, for brevity.

Conceptually, to make such comparison "fair," we have to adjust the parameters used by the generation algorithm so the networks generated have some common statistical properties. It seems reasonable to preserve both the number of internal nodes and the average in-degree. The

first parameter determines the number of states the system can be in, after the state of the source nodes have been fixed. The average in-degree is a measure of connectivity and density. Formally, if we run the topology generation BA algorithm presented for BA networks with parameter N (recall that $N_0 = 5$ and $k = 2$) we will obtain a BA network G_{BA}. The average in-degree is $k = 2$. Call N_I the expected number of internal nodes of G_{BA}. Similarly, if we run the topology generation ER algorithm with parameters n and p, we will obtain a graph G_{ER}.

The problem we wish to solve is: Given N_I find n and p such that the expected in-degree of nodes in G_{ER} is 2 and the expected number of internal nodes of G_{ER} is N_I. Therefore,

$$p \cdot (n-1) = 2 \tag{1}$$

$$n \cdot (1 - (1-p)^{n-1}) = N_I \tag{2}$$

From Eqn 1, $p = \frac{2}{(n-1)}$. Substituting in Eqn 2:

$$n \cdot \left(1 - \left(1 - \frac{2}{(n-1)}\right)^{n-1}\right) = N_I \tag{3}$$

The only unknown is n, and although this equation is hard to solve in closed form, the right hand side behaves linearly in the asymptotic sense. Using simple calculus techniques, we can estimate the solution to Eq 3 as:

$$n = \left(N_I - 2 \cdot e^{-2}\right) \frac{1}{1 - e^{-2}} \tag{4}$$

The asymptotic approximation is so good that the rounding of n to an integer is the biggest source of error even for small values, say 10, of N_I. Therefore, for practical purposes, Eq 4 gives the exact answer. To use this formula you have to know N_I, though. In spite of N_I being determined by N, it is not straightforward to find out an explicit formula. To obtain an approximation, we can simply generate a suitable number of networks using the BA algorithm and estimate N_I as the average of the number internal nodes over all the generated graphs.

3. Experiments

We programmed a simulator using about 700 lines of portable ANSI C. We ran a number of pseudo-random experiments. The main goal was to study the influence of the parameter α in the dynamics of the networks. More precisely, we were interested in answering, *for a given network*, the following questions as a function of α:

1. Are there fixed points?
2. If yes,
   a) how many?
   b) what fraction of trajectories converge to a fixed point?
3. If we restrict the analysis to those trajectories that converged to a fixed point,
   a) what is the average time (number of iterations) until a fixed point is reached?

b) what is the average work (total number of operations) [1] until a fixed point is reached?

We setup the time horizon to 50000 iterations. This number seems to yield a robust determination of whether the network eventually reaches a fixed point or not. For the BA networks we ran the BA algorithm using parameters $N = 100$ and $k = 2$. We generated 50 networks. For each network we generated 1000 initial configurations. The states were generated randomly, but the values corresponding to source nodes were set to zero. More precisely, to generate the initial configuration $\mathbf{s}$ of a network N, if $K_i^{in} = 0$ then the i-th component of $\mathbf{s}$ was set to zero. Otherwise, the i-th component of $\mathbf{s}$ was generated randomly by tossing a fair coin.

Each one of the 50×1000 pairs network/initial configuration was used as a starting point for 11 simulations, each one using a different value of α. The values of α were $0, 0.1, ..., 0.9, 1$. We ran each one of the $50 \times 1000 \times 11$ simulations until the trajectory converged to fixed point or the time horizon was reached.

For the networks using the Erdös–Rényi topology we had to compute the input parameters for the ER algorithm (n and p). To estimate N_I, we generated 10000 graphs using the BA algorithm, with $k = 2$ and $N = 100$. Then we divided the total number of internal nodes by 10000, which yielded $N_I = 62.745$. By using Eq 4, we determined $n = 72$. From Eq 1 we obtained $p = 0.028196$.

With the two parameters, we repeated the process we used for the BA networks: We created 50 random networks and, for each one, we generated 1000 initial conditions. We also used the same values for α and the same time horizon for the simulations. We verified that the BA algorithm indeed produces graphs with vertex degrees that follow a power law probability distribution. Figure 1 shows the relative frequency of each degree, for the 50 BA networks. The diagram, in log-log coordinates, shows a high similarity with a straight line. This is what we expect from a power-law distribution. The noise for the higher vertex degrees is not surprising, as highly connected vertices are rare and hence the population size is small.

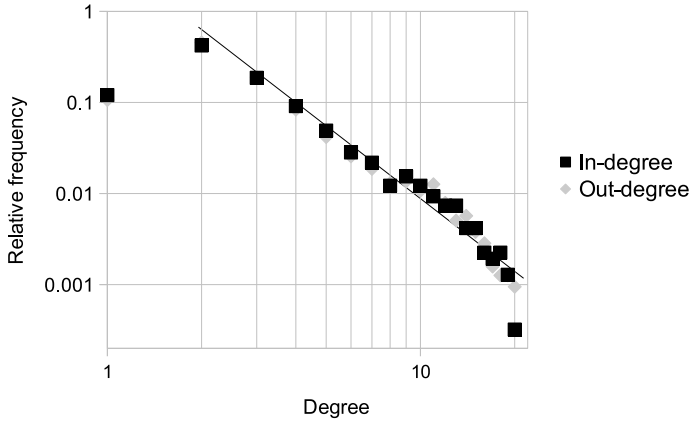

Figure 1: Relative frequency of vertex degrees in the 50 BA networks

[1] Notice that, for $\alpha = 0$, the amount of work per iteration is minimal (only 1 node is updated). On the other extreme, when $\alpha = 1$, the amount of work per iteration is maximal (all the internal nodes are updated in parallel in each iteration).

4. Results and analysis

4.1. Number of fixed points

For both the BA and the Erdös–Rényi families we computed the number of fixed points found per each network and we summarize the results in Figures 2 and 3. The vertical axis of the histograms represent relative frequency over the 50 networks tested. Note that in 15 out of the 50 generated BA networks, no fixed point was found. More precisely, for all α, none of the 1000 trajectories was absorbed into a fixed point within the 50000 iterations used as time horizon. The number of ER networks for which no fixed-point was found is 23.

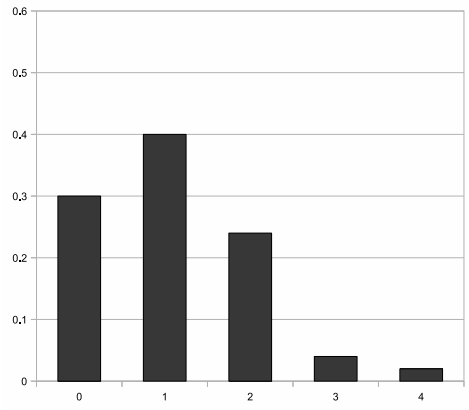

Figure 2: Frequency of number of fixed points for BA networks.

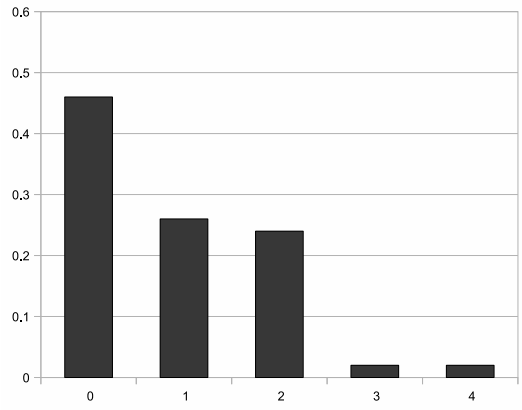

Figure 3: Frequency of number of fixed points for ER networks.

We proved analytically in Section 2 that the expected number of fixed points for any network is 1. Unfortunately, the actual distribution is hard to compute and therefore we use histograms as a base for a numerical approximation. We can see in Figure 2 that for BA network the most likely number of fixed points is 1. If we use the relative frequencies to approximate probabilities, then the estimated expected number of fixed points is 1.08, which is close to the theoretical prediction.

By contrast, for the Erdös–Rényi model, we can see in Figure 3 that the distribution gets more skewed and 0 becomes the most likely value. The estimation of expected number of fixed points in this case yielded 0.88. This estimation is close to 1 if we consider we are using only 50 networks and the number of potential fixed points is about 2^{N_I}, with $N_I \approx 63$, as we described in Subsection 2.2.

4.2. Fraction of converging trajectories

We begin by focusing our analysis on those 35 BA networks for which we were able to prove the existence of fixed points (by finding them). Recall that we used 1000 initial configuration per network. If $\alpha < 1$ (i.e, $\alpha = 0, 0.1, \ldots, 0.9$) out of the 350000 stochastic trajectories we computed, 100% of them reached a fixed point. Despite the fact that for all these 35 networks we were also able to find at least one converging trajectory when $\alpha = 1$, the situation in this case changed dramatically. In fact, out of the 35000 deterministic trajectories we simulated for those 35 networks, only $\sim$28.9% of them reached a fixed point. Notice also that the percentage of deterministic trajectories that reached a fixed point varied greatly form one network to another.

We present the percentages for each network in Figure 5. Every single trajectory that did not reach a fixed point ended in a cycle within the time horizon. For presentation purposes, we included the analogous diagram for $\alpha < 1$ in Figure 4.

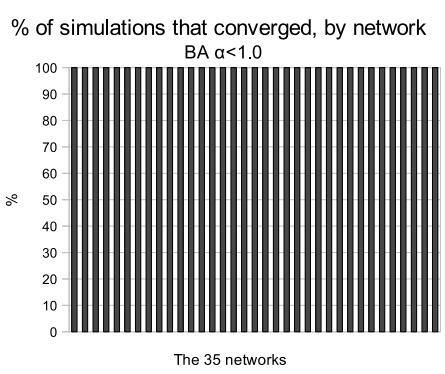

Figure 4: Percentage of simulations that converged when $\alpha < 1$ for each of the 35 BA networks with at least one fixed point.

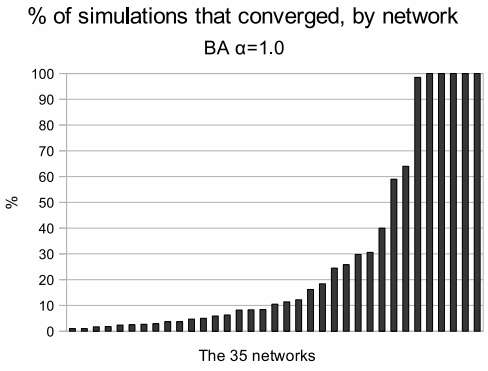

Figure 5: Percentage of simulations that converged when $\alpha = 1$ for each of the 35 BA networks with at least one fixed point.

We immediately notice that a higher proportion of simulations reached a fixed point when $\alpha < 1$ than when $\alpha = 1$. This is not unexpected as the non-determinism in the stochastic trajectories is a way to avoid the cyclic attractors the deterministic trajectories tend to fall into. The remarkable property is that, if for a given network a single trajectory with $\alpha < 1$ converged to a fixed point, then *all the trajectories* with $\alpha < 1$ also converged (although not necessarily to the same point). Also note that all the fixed point discovered through fully parallel updates were also discovered with the $\alpha < 1$ simulations. This indicates that stochastic updates can be used as a robust method to detect the existence of fixed points.

For the ER networks we computed the same statistics as we did for the BA networks, were applicable. There were 27 out of the 50 networks were we found at least a fixed point. Interestingly, only 21 of those fixed points were detected by deterministic trajectories. The breakdown of percentages of converging trajectories by network for the case $\alpha = 1$ is presented in Figure 7. Every single trajectory that did not reach a fixed point ended in a cycle within the time horizon. The breakdown of percentages by network for the case $\alpha < 1$ is presented in Figure 6. We notice immediately the difference between Figures 6 and 4: For some Erdös–Rényi networks with fixed point not all stochastic trajectories converged to a fixed point while all trajectories converged in the BA networks case, as we already described before.

To emphasize the differences in the qualitative behavior we look more closely at two particular networks. We selected one ER network for which the percentage of trajectories that converged (considering all $\alpha < 1$) was about 50%. Recall that 1000 initial configurations were used, 10 values of α were tried and one trajectory per α was simulated. That means that about 5000 out of 10000 trajectories converged to a fixed point for that particular network. The histogram in Figure 9 shows that the choice of the initial configuration changes the probability of reaching a fixed point. The heights of the bars represent numbers of initial configurations. The horizontal axis is the percentage of trajectories (starting from one particular initial configuration) that converged to a fixed point. For instance, for the ER network, we can deduce the following: 160

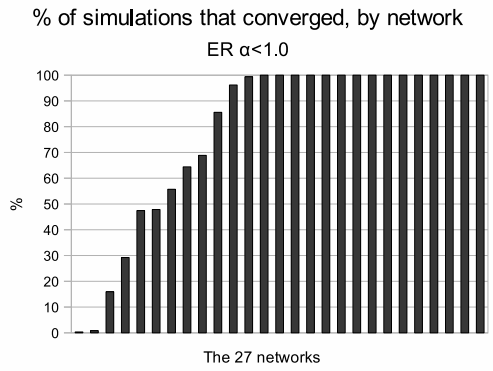

Figure 6: Percentage of simulations that converged for each of the 27 ER networks where a fixed point was found.

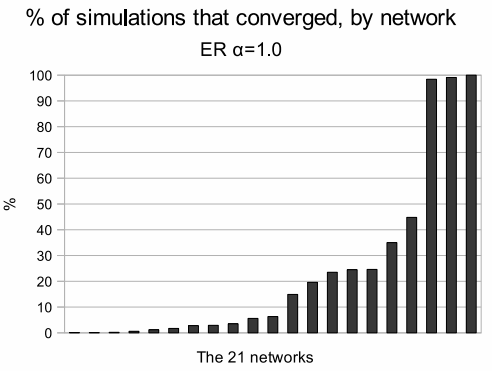

Figure 7: Percentage of simulations that converged for each of the 21 ER networks where a fixed point was found.

initial configurations converged in 40% of the simulations. Since for each initial configuration we ran exactly 10 simulations there is no need to associate intervals with bins. For contrast, we show in Figure 8 the analogous histogram for an arbitrary BA network with a fixed point.

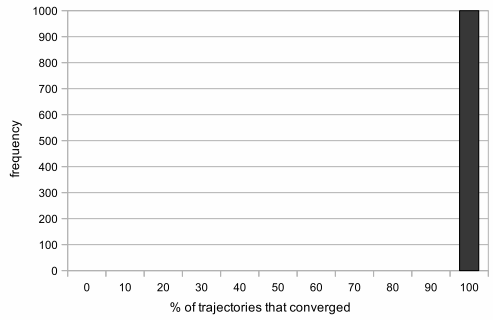

Figure 8: Histogram of number of initial configurations that lead to a converging trajectory for a single BA network which had at least one fixed point ($\alpha < 1$).

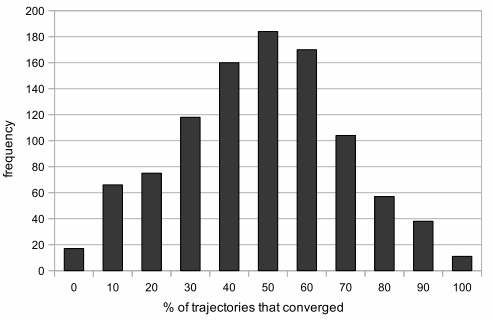

Figure 9: Histogram of number of initial configurations that lead to a converging trajectory for a single ER network which had at least one fixed point ($\alpha < 1$).

Another difference between the influence of parallelism when updating BA or ER networks is seen comparing Figures 2, 3 and 10. Figure 10 shows the difference in number of fixed points found depending on α being 1 or less than 1. The columns that describe the $\alpha < 1$ case represent the same values found in the histogram in Figure 3. The columns that describe the $\alpha = 1$ case in Figure 10 show that fewer fixed points were discovered. This, again, strongly suggests that using partially parallel updates is a sensible strategy to find fixed points. It also indicates that the set of discovered fixed points somehow depends on the choice of α. For BA networks, we also discriminated between $\alpha = 1$ and $\alpha < 1$ cases. For these networks though, in both cases the histograms were identical to the one in Figure 2. This indicates that the convergence to fixed points is fairly insensitive to the choice of α. This property shows a form of robustness of BA

networks.

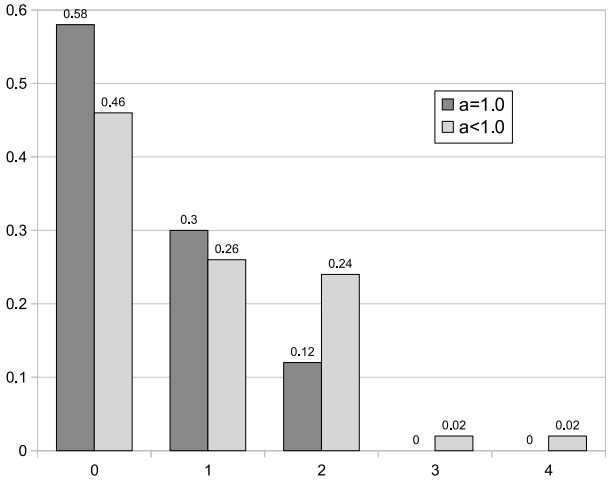

Figure 10: Frequency of number of fixed points for ER networks for the $\alpha = 1$ and $\alpha < 1$ cases.

4.3. Time and work until convergence

For BA networks, Figures 11 and 12 represent the mean and maximum time/work to reach the fixed point for *only* the simulations that reached it. This means, for $\alpha < 1$, each column gives the average and maximum over 35000 simulations, while for $\alpha = 1$ the values were computed over a smaller sample because only $\sim$28.9% of the runs ended before the time horizon was reached. This makes the values of the rightmost column of each figure somewhat incomparable with the others.

In Figure 11 we note how the average time changes with α. The qualitative behavior of the results are in part intuitive. If $\alpha = 0$ we expect the time to be high, since not much work is done per iteration. We also expected the time to increase when α approaches 1, because the system becomes more deterministic and tends to imitate the behavior of the $\alpha = 1$ case. Therefore, the cycles of the deterministic trajectories become metastable regions of the stochastic trajectories.

Note that the average number of iterations for $\alpha = 1$ is smaller than for all the values for $\alpha < 1$. The comparison is not fair, though, because the parallel update rule is not a reliable way to find fixed points. With respect to the amount of work, we can notice, as expected, that small values of α become very competitive (see Figure 12). One possible explanation for the parallel simulations outperforming the $\alpha < 1$ cases (neglecting the lack of robustness, of course) is as follows: The $\alpha = 1$ updates are prone to solve only the "simple instances" of the problem. That is, only very stable fixed points located close to the initial configurations are likely to be found.

To make a fair comparison between the $\alpha < 1$ and $\alpha = 1$ situations, we recompute the statistics but only for the cases where the deterministic model found a fixed point. This means, we considered only the pairs (network G, initial configuration $\mathbf{s}$) such that the deterministic trajectory of G starting from $\mathbf{s}$ reached a fixed point. Figures 13 and 14 show the results. From comparison against Figures 11 and 12 we notice that the average times of the simulations for

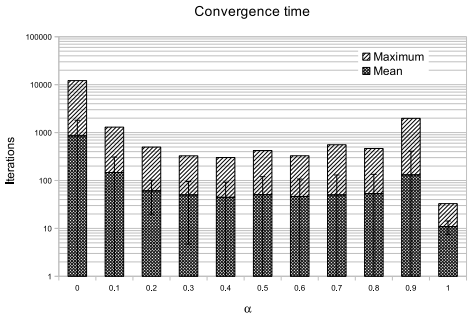

Figure 11: Convergence time for BA networks

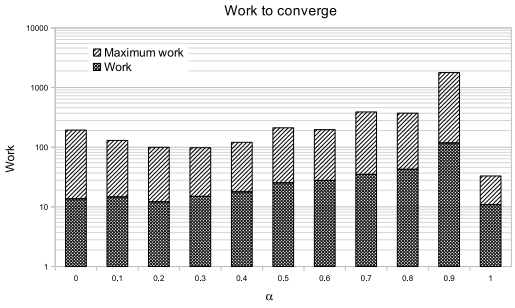

Figure 12: Convergence work for BA networks

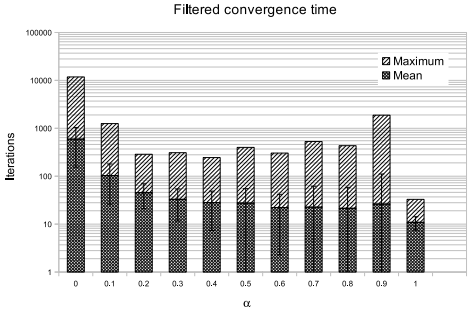

Figure 13: Filtered convergence time for BA networks

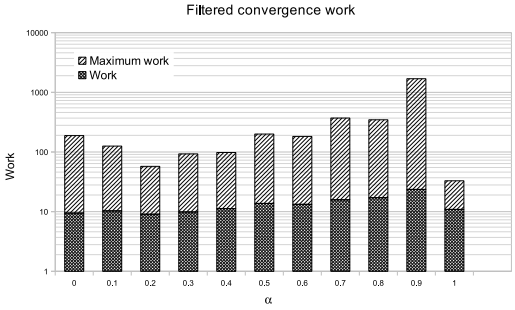

Figure 14: Filtered convergence work for BA networks

$\alpha < 1$ actually decreased. This supports the hypothesis of $\alpha = 1$ working mostly for simple instances of the problem. The differences in running time may not be considered substantial though and the problem requires further study.

For ER networks, Figures 15 and 16 are the mean and maximum times/work to reach the fixed point for *only* the simulations that reached it. This means, for $\alpha < 1$, each column gives the average and maximum over 27000 simulations, while for $\alpha = 1$ the values were computed over a smaller sample because only $\sim$24% of the runs ended before the time horizon was reached. In this case we did not compute the averages/maximums for the stochastic model restricting the networks and initial conditions to those that reached a fixed point with deterministic updates. There was no qualitative difference between the results for BA networks (Figures 11 and 12) and the times for ER networks. Data suggests that ER networks are slower to converge than BA systems.

4.4. Discussion

The mathematical concept of fixed points play a central role in Systems Biology. For example, as stated in [38], "Huang and coworkers [31] provided the first evidence that mammalian cell types might correspond to attractors of a high dimensional dynamical system."

We will make the case that RBNs, as defined here, have a behavior that resembles that of natural networks. In fact, it is well known that gene expression/repression changes according to

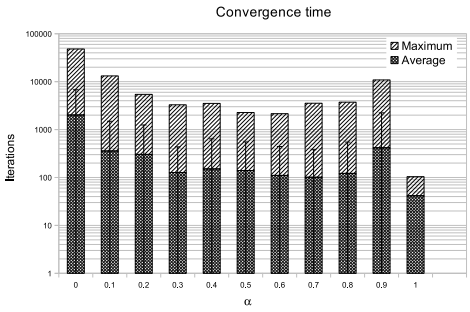

Figure 15: Convergence time for ER networks

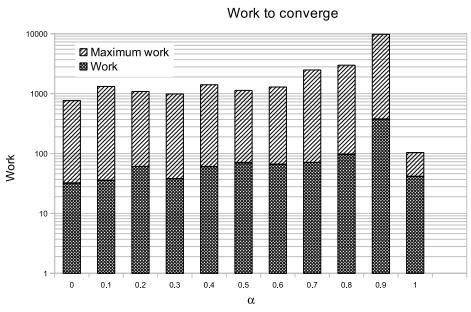

Figure 16: Convergence work for ER networks

external signals, such as temperature or availability of nutrients and/or oxygen. This can be modeled easily by setting the values of input nodes depending on the external signals. Considering only the case of rapid responses to external challenges, for example to achieve homeostasis, we will also assume that gene expression state is (after the transient) *mostly* function of the external signals. This is consistent with the assumtions in [27]. In terms of the model this means the fixed point should be function of the values of the input nodes. Although stochasticity sometimes plays an important role in cell dynamics [33], for purposes of discussion we will assume that for each set of external signals there should be one or a small number of fixed points. Another property is stability, meaning the fixed point should have a large basin of attraction. Another desirable feature is that the fixed point(s) should be reached "quickly." By quickly, we mean that the number of transitions should be small compared to the total number of possible configurations. Finally, the behavior of the model should not depend dramatically on the choice of α.

Our simulations with asynchronous updates on BA networks showed all properties mentioned above. The theorem in Subsection 2.1, together with the histogram in Figure 2 show that the number of fixed points is, with high probability, close to one. The fact that all trajectories converged, as shown in Figure 4 indicated that the model is robust *regardless* of the choice of α, as long as it is less than 1. With respect to the speed of convergence, the average number of iterations to reach a fixed-point is small compared to the number of possible configurations, which was about 2^{63} in all the numerical experiments. Our results contrast with those presented in [19], were they show that the behavior of their asynchronous RBNs was highly sensitive to changes in the way or randomly picking up the nodes to be updated. This does not contradict our findings as their assigning of update functions to vertices and their algorithm to create the topology differs from ours. In particular, they were assigning to *every node* a generalization of cellular automata rule 126. An anonymous reviewer suggested that perhaps the difference in behavior between the BA and ER GRN could be explained by the former being in the ordered phase, and the later being in the chaotic phase. This is an interesting possibility. Since BA networks have a free scale exponent of 3 and, using the same notation as in [5], we chose update rules with uniform probability $p = 1/2$, our BA GRN are well in ordered phase. Our ER GRNs have a mean connectivity value $k = 2$. Therefore, by Equation (8) in [5], they are on the boundary between the ordered and chaotic phases. This could explain the qualitative differences in behavior and prompts for further research.

Of course, the goal of the model is to provide insight on biological processes. The most remarkable observation is that the three aforementioned properties appeared in the simulations

without requiring any kind of deliberate design. The network topologies, the dynamics and vertices to be updates were selected at random. Yet, the behavior was as desired, *as long as the network was BA and* $\alpha < 1$. This suggests that the prevalent scale–free topologies in natural GRNs [4] could be a major factor in robustness of living cells. Also note that updating nodes picked randomly is not the only way of adding non-determinism to a RBN. In [55], they considered the existence of noise while evaluating the (synchronous) update rules and they analyzed the shift among multiple basins of attraction resulting from this stochastic perturbation. They present the idea that some biological processes may rely on the relative stability of network states. It is conceivable that topological properties of the network affect the expected passage time (from one basin to another) just as it affects the distribution on the number of fixed points, so more study on this is necessary.

As an aside, finding fixed points of a Boolean network is equivalent to finding satisfying assignments to Boolean formulas. This is the well known *SAT* problem in Computer Science, and it is believed to be computationally hard if we consider the worst case running time [1]. However, the results shown in Figures 12, 14 and 16, which imply fast convergence, suggest that the asynchronous update rule provides us with an efficient heuristic to find fixed points or satisfying assignments.

5. Conclusions

We analyzed the tendency to reach a fixed point, the number of iterations and the work needed to converge to it for different families of Boolean networks and update policies. We summarize our results as follows:

a) Using partial parallelism (i.e, $\alpha < 1$), the likehood of the trajectory to reach a fixed point increases. This happens regardless of the topology.

b) For BA networks the choice of α is not very important from the convergence point of view, as long as it is less than one. We base this conclusion on the fact that we could not find a simple example of a network not converging to a fixed point provided that it has 1.

c) For BA networks the choice of α is not very important from the work point of view, as long as it is less than one. This follows from the analysis of the filtered case. The work is about the same as in the $\alpha = 1$ case, while full parallelism increases the likehood of being trapped in a cycle.

d) For ER networks the choice of α is relevant from the work point of view, when it is less than 1. The set of points discovered seems to be somehow dependent on the choice of α.

e) The combination of BA networks and using $\alpha < 1$ for the updates was enough to impose dynamical properties which are similar to those observed in nature.

References

[1] T. Akutsu, S. Kuhara, O. Maruayama, S. Miyano. *Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions.* Proc. Ninth ACM-SIAM Symp. on Discrete Algorithms, 1998, pp. 695702.

[2] R. Albert and A.-L. Barabási. *Statistical mechanics of complex networks.* Rev. Modern Phys. 74 (1) (2002) 47–97.

[3] R. Albert and H.G. Othmer. *The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster.* Journal of Theoretical Biology 223 (2003) 1–18.

[4] R. Albert. *Scale-free networks in cell biology.* Journal of Cell Science 118 (2005), 4947–4957.

[5] M. Aldana, S. Coppersmith and L.P. Kadanoff. *Boolean dynamics with random couplings.* In Perspectives and Problems in Nonlinear Science, Springer Applied Mathematical Sciences Series, Springer, Berlin 2003, 23–89.

[6] M. Aldana. *Boolean dynamics of networks with scale-free topology.* Physica D 185 (2003) 45–66.
[7] E.R. Álvarez-Buylla, Á. Chaos, M. Aldana, M. Benítez, Y. Cortes-Poza, C. Espinosa-Soto, D.A. Hartasánchez, R.B. Lotto, D. Malkin, G.J. Escalera-Santos and P. Padilla-Longoria. *Floral morphogenesis: Stochastic explorations of a gene network epigenetic landscape.* PLoS ONE 3(11) (2008): e3626.
[8] J. Aracena, E. Goles, A. Moreira and L. Salinas. *On the robustness of update schedules in Boolean networks.* Biosystems 97, Issue 1, July 2009, 1–8.
[9] A.-L. Barabási and R. Albert *Emergence of scaling in random networks.* Science 286 (5439) (1999) 509–512.
[10] U. Bastolla and G. Parisi. *Closing probabilities in the Kauffman model: An annealed computation.* Physica D 98, 1–25 (1996).
[11] H. Bersini and V. Detours. *Asynchrony induces stability in cellular automata based models.* Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems: Articial Life IV, MIT Press, 1994.
[12] A. Bhattacharjya and S. Liang. *Power-Law distributions in some random Boolean networks.* Phys. Rev. Lett. 77, 1644–1647 (1996).
[13] S. Bilke and F. Sjunnesson. *Stability of the Kauffman model.* Phys. Rev. E 65 (2001) 016129.
[14] C. Christensena, A. Gupta, C.D. Maranasand and R Albert. *Large-scale inference and graph-theoretical analysis of gene-regulatory networks in B. Subtilis.* Physica A (2007) 796–810.
[15] Ch. Damiani, M. Villani, Ch. Darabos and M. Tomassini. *Dynamics of interconnected Boolean networks with scale-free topology.* Artificial Life and Evolutionary Computation, proceeding of WIVACE 2008, World Scientific Publication, 27–283.
[16] C. Daizhan and Q. Hongsheng. *A linear representation of dynamics of Boolean networks.* IEEE Transactions on Automatic Control, vol. 55, no 10, 2251–2258 (2010)
[17] Ch. Darabos, M. Tomassini and M. Giacobini. *Dynamics of unperturbed and noisy generalized Boolean networks.* Journal of Theoretical Biology 260-4, (2009) 531–544.
[18] E.H. Davidson, J.P. Rast, P. Oliveri, A. Ransick, C. Calestani, C.-H. Yuh, T. Minokawa, G. Amore, V. Hinman, C. Arenas-Mena, O. Otim, C.T. Brown, C. B. Livi, P.Y. Lee, R. Revilla, Al.G. Rust, Z. jun Pan, M.J. Schilstra, P.J.C. Clarke, M.I. Arnone, L. Rowen, R. A. Cameron, D.R. McClay, L. Hood and H. Bolouri. *A genomic regulatory network for development.* Science 295 (5560) (2002) 1669–1678.
[19] X. Deng, H. Geng, M.T. Matache. *Dynamics of asynchronous random Boolean networks with asynchrony generated by stochastic processes*. Biosystems 88(1–2):16–34, (March 2007).
[20] B. Derrida and Y. Pomeau. *Random networks of automata: A simple annealed approximation.* Europhysics Letters 1 (2) (1986) 45–49.
[21] E. Dubrova and M. Teslenko. *A SAT-based algorithm for finding attractors in synchronous boolean networks.* IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 8.5: 1393–1399, (2011)
[22] R. Edwards and L. Glass. *A calculus for relating the dynamics and structure of complex biological networks.* Advances in Chemical Physics, Vol. 132, John Wiley & Sons, New York (2006), 151–178.
[23] P. Erdös and A. Rényi. *On random graphs.* Publicationes Mathematicae 6 (1959): 290–297.
[24] N. Fatès, N. Schabanel, D. Regnault and É. Thierry. *Asynchronous behavior of double-quiescent elementary cellular automata.* Proceedings of LATIN 2006: 7th Latin American Symposium, LNCS 3887, 2006, 455– 466.
[25] J. J. Fox and C.C. Hill. *From topology to dynamics in biochemical networks.* Chaos 11 (4) (2001) 809–815.
[26] A. Garg, K. Mohanram, A. Di Cara, G. De Micheli and I. Xenarios. *Modeling stochasticity and robustness in gene regulatory networks.* Bioinformatics, 25(12), i101–i109. (2009)
[27] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel,M. Eisen ,G. Storz,D. Botstein and P. Brown. *Genomic expression programs in the response of yeast cells to environmental changes.* Molecular Biology of the Cell. 11(12) 4241–4257 (December 2000)
[28] C. Gershenson. *Updating schemes in random Boolean networks: Do they really matter?* Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, MIT Press, 2004, 238–243.
[29] S.F. Greenbury, I.G. Johnston, M.A. Smith, J.P. Doye and A.A. Louis. *The effect of scale-free topology on the robustness and evolvability of genetic regulatory networks.* Journal of theoretical biology, 267(1), 48–61, (2010).
[30] I. Harvey and T. Bossomaier. *Time out of joint: Attractors in asynchronous random Boolean networks.* Proceedings of the Fourth European Conference on Artificial Life (ECAL97), MIT Press, 199, 67–75.
[31] S. Huang, G. Eichler, Y. Bar-Yam and D.E. Ingber. *Cell fates as high-dimensional attractor states of a complex gene regulatory network.* Phys. Rev. Lett. 94: 128701 (2005).
[32] K. Iguchi, S Kinoshita and H.S. Yamada. *Boolean dynamics of Kauffman models with a scale-free network.* Journal of Theoretical Biology 247:1, (2007) 138–151.
[33] M. Kaern, T. C. Elston, W. J. Blake and James J. Collins. *Stochasticity in gene expression: from theories to phenotypes*, Nature Reviews Genetics 6, 451–464 (June 2005).
[34] S. Kauffman. *Metabolic stability and epigenesis in randomly constructed genetic nets.* Journal of Theoretical Biology 22:3 (1969) 437–467.

[35] S. I. Kinoshita, K. Iguchi, H. S. Yamada, M. Tokuyama, I. Oppenheim and H. Nishiyama *Robustness of attractor states in complex networks.* AIP Conference Proceedings, Vol. 982, 768–771, (February 2008)
[36] D. H. Le and K. Yung-Keun. *A coherent feedforward loop design principle to sustain robustness of biological networks.* Bioinformatics 29.5, 630–637, (2013)
[37] J. Lloyd-Price, A. Gupta and A. S. Ribeiro. *Robustness and information propagation in attractors of Random Boolean Networks.* PloS one, 7(7), e42018, (2012)
[38] B. D. MacArthur, A. Ma'ayan and I. R. Lemischka. *Systems biology of stem cell fate and cellular reprogramming.* Nature Reviews Molecular Cell Biology, 10(10), 672–681, (2009).
[39] B. Mesot and Ch. Teuscher. *Critical values in asynchronous random Boolean networks.* Advances in Artificial Life, ECAL2003, Vol. 2801 of Lecture Notes in Artificial Life (2003) 367–376.
[40] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon. *Network motifs: simple building blocks of complex networks.* Science, 298(5594), 824–827, (2002)
[41] Z. Nikoloski, P. May, and J. Selbig. *Algebraic connectivity may explain the evolution of gene regulatory networks.* Journal of theoretical biology 267.1, 7–14, (2010)
[42] R. Olfati-Saber. *Ultrafast consensus in small-world networks.* Proceedings of the 2005 American Control Conference, 2371–2378, (June 2005).
[43] C. Oosawa and M.A. Savageau. *Effects of alternative connectivity on behavior of randomly constructed Boolean networks.* Physica D 170 (2002) 143–161.
[44] T. P. Peixoto. *Redundancy and error resilience in Boolean networks.* Physical review letters, 104(4), 048701, (2010)
[45] T. P. Peixoto. *Emergence of robustness against noise: A structural phase transition in evolved models of gene regulatory networks.* Physical Review E, 85(4), 041908, (2012).
[46] P. Rohlfshagen, E.A. Di Paolo: *The circular topology of rhythm in asynchronous random Boolean networks*. Biosystems 73(2):141–52, (February 2004).
[47] B. Samuelsson and C. Troein. *Random maps and attractors in random Boolean networks.* Phys. Rev. E 72, 046112 (October 2005).
[48] C. Schmal,T. P. Peixoto and B. Drossel. *Boolean networks with robust and reliable trajectories.* New Journal of Physics, 12(11), 113054 (2010).
[49] B. Schönfisch and André de Roos. *Synchronous and asynchronous updating in cellular automata*, Biosystems, Volume 51, Issue 3, 123–143, (September 1999).
[50] S. Squires, A. Pomerance, M. Girvan and E. Ott. *Stability of Boolean networks: The joint effects of topology and update rules.* Physical Review E, 90(2), 022814, (2014).
[51] S.H. Strogatz. *Exploring complex networks.* Nature 410, 268–276 (2001).
[52] T. D. Tran and Y. K. Kwon. *The relationship between modularity and robustness in signalling networks.* Journal of The Royal Society Interface, 10(88), 20130771. (2013).
[53] K. Willadsen, J. Triesch and J. Wiles. *Understanding robustness in Random Boolean Networks.* Proceedings of the Eleventh International Conference on Simulation and Synthesis of Living Systems 694–701, (2008)
[54] K. Yung-Keun, J. Kim, and K. Cho. *Dynamical Robustness Against Multiple Mutations in Signaling Networks.* IEEE/ACM transactions on computational biology and bioinformatics 1–9, (November 2015).
[55] J.X. Zhou, A. Samal, A.F. d'Hérouël, Price ND, Huang S.: Relative stability of network states in Boolean network models of gene regulation in development. Biosystems 142–143: 15–24, (April–May 2016).