Evolution of Discrete Gene Regulatory Models

Afshin Esmaeili
Department of Computer Science
University of Calgary
2500 University Dr. NW
Calgary, Alberta T2N 1N4
a.esmaeili@ucalgary.ca

Christian Jacob
Department of Computer Science
Dept. of Biochemistry & Molecular Biology
University of Calgary
2500 University Dr. NW
Calgary, Alberta T2N 1N4
cjacob@ucalgary.ca

ABSTRACT

Gene regulatory networks (GRNs) are complex control systems that govern the interaction of genes, which ultimately control cellular processes at the protein level. GRNs can be represented using abstract models such as random Boolean networks (RBNs), where gene activities and their interactions are captured as nodes with associated Boolean functions, which receive activation or repressor signals from other nodes. We have developed an evolutionary model of gene regulatory networks using RBNs to study the dynamic behavior of these control systems.

We explore a range of different network parameters such as excess graph, sensitivity, basin entropy, number of attractors and maximum length of attractors in RBNs. We investigate the effects of mutations and crossover on the fitness of RBNs. We show that over the course of evolution, networks with a low level of damage spreading and a high tolerance to random perturbations can be produced. We also demonstrate that these networks are able to adapt to a range of different perturbations obtaining a high level of stability.

Categories and Subject Descriptors

I.6.5 Computing Methodologies [**Simulation and Modeling**]: Model Development

General Terms

Algorithms, Design, Experimentation

Keywords

Gene regulatory network; Random Boolean model; Systems biology; Evolutionary design

1. INTRODUCTION

One of the goals of modern developmental biology is to understand the transcriptional regulation processes that drive the genetic and, consequently morphological, changes in cells. Random Boolean networks (RBNs) have attracted attention as abstract models to understand gene regulatory networks and their time dynamics [11, 13]. RBN-based approaches have also been used to investigate and classify cellular automata [27, 26], neural networks [4, 8] and spin glasses [1]. Through mathematical and statistical models a wide range of RBN properties have been studied such as their topological compositions [11, 12, 10, 18, 22], and what constitutes biologically meaningful regulatory functions [20]. How random Boolean networks tend to react to perturbations [21] is related to the number and length of their state space attractors [24, 6].[1] In this same context of stability analysis, RBN criticality and scaling properties have been investigated [5, 19, 23]. Although most analytical research is performed on smaller-size networks, simulations of large RBNs have been undertaken as well [7].

In this paper, we focus on evolutionary aspects of genetic regulatory networks. RBN evolution and their capabilities to adapt to perturbations [2, 14, 17] as well as their relationship to biological mutations [3, 25] and evolutionary fitness landscapes [9] have been examined in previous work. However, none of these studies have utilized an evolutionary (optimization) system to analyze RBN properties and to generate network ensembles [12]. In this study we investigate the behavior of gene regulatory networks (GRNs) modeled by RBNs and focus on which network properties can be evolved.

The rest of our paper is organized as follows. Section 2 gives formal definitions of random Boolean networks and their associated state spaces. In Section 3 we briefly outline how to classify RBN dynamics. How we evaluate fitness of RBNs and evolve them is presented in Section 4. Our simulation results are presented in Section 5. Finally, in Section 6 we conclude the paper and discuss our further investigations into the evolution of RBN ensembles.

2. RANDOM BOOLEAN NETWORKS

We define a random Boolean network $RBN_{(N,K)} = (\mathcal{G}, \mathcal{F})$ as a graph, $\mathcal{G}$, together with a set of Boolean functions, $\mathcal{F}$ (Fig. 1). The parameters N and K are positive integers that define the number of nodes and the in-degree for each node, respectively. The gene interaction graph $\mathcal{G}$ represents the set of genes and how some of their products— those that act as transcription factors—regulate other genes. Hence, the transcription network $\mathcal{G} = (\boldsymbol{G}, \boldsymbol{E})$ is defined by its nodes,

[1] We will discuss this further in Section 2.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

$\boldsymbol{G} = \{g_1, \ldots, g_N\}$, which constitute the set of genes, and regulatory, directed edges $\boldsymbol{E}$ between these nodes (Fig. 1a):

$$\boldsymbol{E} = \{u \to v \mid u, v \in \boldsymbol{G}\}, \tag{1}$$

where $\forall w \in \boldsymbol{G} : \exists u_1, \ldots, u_K \in \boldsymbol{G}$:

$$u_1 \to w \in \boldsymbol{E} \wedge \ldots \wedge u_K \to w \in \boldsymbol{E}.$$

Each node therefore has K incoming connections (corresponding to the number of 1s in each row for a given gene in Figure 1a). Hence, each gene is modeled as being regulated by the transcription factors encoded in the K genes connected to it. Genes are treated as binary units, which can be either on (expressing their regulatory proteins) or off (no expression of gene product). Therefore, each node, g, at time step t can be in state $s_t(g) = 0$ (off) or $s_t(g) = 1$ (on). The actual inhibitory or activating regulatory effects of K genes, $g_1, \ldots, g_K$, on (the promoter of) a gene, g, is captured by a K-ary binary function $f_g \in B = \{b : \{0,1\}^K \to \{0,1\}\}$ such that (Fig. 1b)

$$s_{t+1}(g) = f_g(s_t(g_1), \ldots, s_t(g_K)). \tag{2}$$

Starting from a randomly assigned initial state for each node, we can now keep track of how these states change over time. The maximum possible *state space* $\mathcal{S}$ of a RBN is the set of all combinations of binary states over the N elements (Fig. 1c):

$$\mathcal{S}(RBN_{(N,K)}) \subseteq \mathcal{S}_N := \{0,1\}^N. \tag{3}$$

For a random Boolean network, $R = RBN_{N,K}$, starting from an initial configuration of randomly assigned states, $s_0(R) = (s_0(g_1), \ldots, s_0(g_N))$, we can observe the dynamics of R over time by looking at its *activity pattern*:

$$s_0(R), \ldots, s_t(R), s_{t+1}(R), \ldots, \tag{4}$$

where $s_t(R) = (s_t(g_1), \ldots, s_t(g_N))$, that is we step through the state space vectors of the network. We therefore have three different representations to capture such a network's topology, its regulatory functionality, and its time dynamics (Fig. 2):

- **Topology**: The topology of an RBN constitutes which gene products (or transcription factors) regulate the expression of which other genes, represented by the gene connectivity graph $\mathcal{G}$. For example, the $RBN_{(3,2)}$ network in Fig. 2a has three genes, with one self-regulatory node.
- **Functions**: The functions of an RBN describe how transcription factors regulate other genes. Figure 1b shows an example set of three Boolean functions that are assigned to the nodes in the RBN topology graph (Fig. 2a).
- **Activity Pattern**: Given some initial configuration, each node of the network is simultaneously updated. This results in a time progression of on-off states, which can be captured in an activity pattern plot (Fig. 2c). In these plots, the initial configuration, s_0, of the network is the leftmost column where a black square represents a one (gene on) and a white square stands for a zero (gene off). Of course, any of these activity plots only captures the dynamics of the RBN starting from a particular configuration, s_0, over a finite amount of time.
- **State Space Diagram**: The state space diagram of an RBN represents all the states, S_N, that a network can be in (Fig. 1c and Fig. 2b). Each of the 2^N nodes represents one configuration from S_N. In a state space graph we draw a connection from $s_t(R)$ to $s_{t+1}(R)$. Starting from $s_t(R)$, and applying all functions f_g simultaneously over all nodes, g, in $\mathcal{G}$, the next state configuration of R is $s_{t+1}(R)$. Columns that are side by side in an activity plot (Fig. 2c) end up being connected by an edge in the state space diagram. Consequently, the state space diagram gives an exhaustive picture of all possible dynamics for a given RBN.[2]

Genes	Gene 1	Gene 2	Gene 3
Gene 1	0	1	1
Gene 2	1	1	0
Gene 3	1	1	0

(a) Connections

x_1	x_2	f_1	f_2	f_3
0	0	1	1	1
0	1	1	1	1
1	0	0	1	0
1	1	0	1	1

(b) Function Set

Time (t)			Time (t+1)		
Gene 1	Gene 2	Gene 3	Gene 1	Gene 2	Gene 3
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	0
1	1	0	0	1	1
1	1	1	0	1	1

(c) State Space

Figure 1: Binary representation of a random Boolean network with size $N = 3$ and input degree $K = 2$. (a) The gene connectivity matrix determines which genes interact. (b) Each gene i is associated with a regulatory function f_i. (c) The activation dynamic over time can be captured in a state space table. The phenotypic representation for this network and its state space is illustrated in Fig. 2.

These complementary representations of random Boolean networks help to investigate the relationships between the network topologies, node functionalities, and the resulting activity dynamics over time.

[2]Obviously, the size of the state space presents a tremendous computational challenge. This is why we will restrict our inquiries within this paper to small networks with $N < 10$.

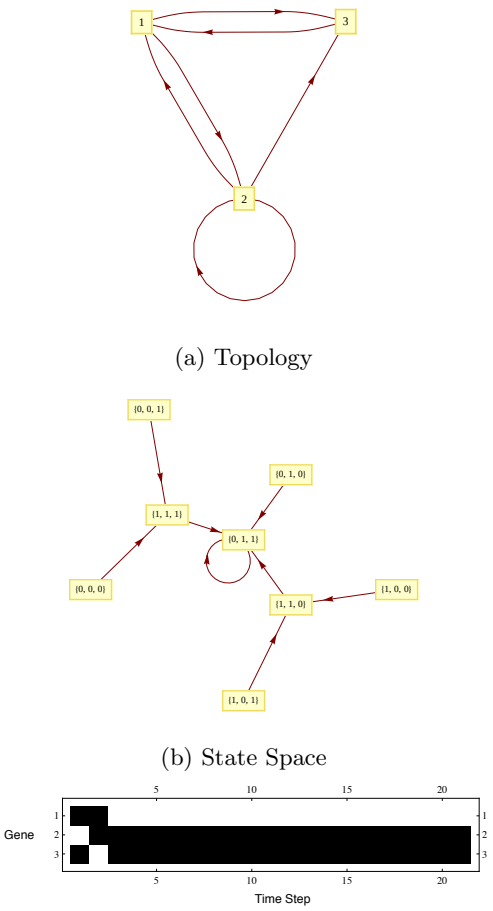
3. ANALYSIS OF RBN DYNAMICS

One desired key feature of RBNs is their ability to stabilize or 'settle' after an initial period of activity. Since there is a finite number of states (2^N) in the state space of an RBN, eventually a given state must be repeated. These state cycles are referred to as *attractors* [11]. Point attractors, the simplest kind of attractors, have a cycle length of one, where a state is its own successor state. The set of states that flow into an attractor is called the *basin of attraction*. Attractors and their basins partition the state space of an RBN, ranging from subgraphs of size 1 to size 2^N. Figure 2b shows state $\{0, 1, 1\}$ as a single attractor of length one. All states eventually end up at this attractor.

The temporal dynamics of RBN activity patterns can be categorized to fall into three distinct regimes [11]:

- In the *ordered regime* a network shows the highest stability. Networks with input degree $K = 1$ exhibit a high level of stability and are considered ordered networks. In an ordered regime many elements settle into fixed states.
- In the *chaotic regime* networks are mostly unstable. These types of networks have an input degree $K > 2$. The state of elements fluctuates and no steady state is reached.
- The *critical regime* can be described as the transition between the ordered and chaotic regime. Critical or edge-of-chaos networks ($K = 2$) seem to be most related to biological systems.

In this paper, we demonstrate how to utilize an evolutionary algorithm to generate ensembles of networks that not only are representative of a particular regime, but also share certain properties, such as increased resistance to perturbation, topological features (connection density), or state space characteristics (number of attractors, attractor lengths). Generating RBNs through an evolutionary system with specific fitness criteria makes these dynamic networks more readily available to study their properties and their interrelationships.



(a) Topology

(b) State Space

(c) Activity Pattern

Figure 2: Three (phenotypic) representations for an RBN of size 3 and input degree 2 based on the representation in Figure 1.

4. EVOLUTION OF RBNS

Here we describe a set of features that we use to evaluate random Boolean networks regarding their topology, their assigned regulatory node functions, their state spaces, and their activity patterns.

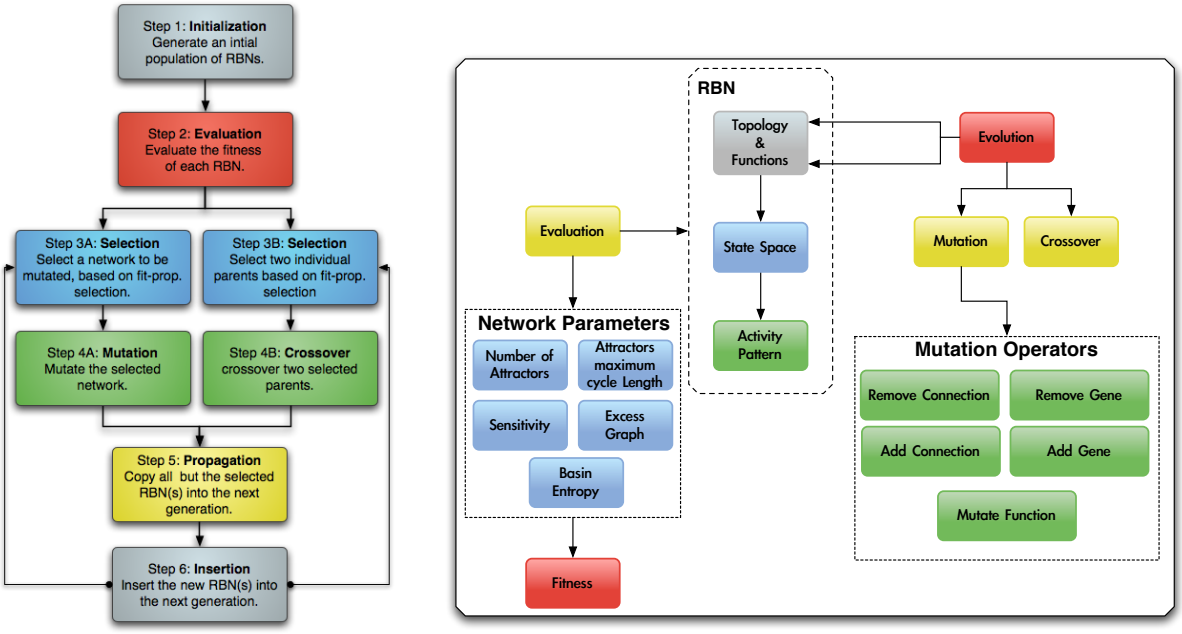
4.1 Assessing RBN stability and criticality

For the results we discuss in this paper we focus on the evolution of networks with increased stability in the critical and chaotic regime. Ordered networks are less interesting in this context, as we are ultimately interested in biologically plausible regulatory networks. We use the following features to help us characterize stability aspects of random Boolean networks, which have been used in previous work (see Section 1).

- The *number of attractors* (μ) of a RBN is represented by the number of cycles in the associated RBN state space graph (Fig. 2b). A network could have a single point attractor where all other states are in its basin of attraction. However, this would only make a fast stabilizing network, if the average basin path length is relatively short. Otherwise, if the attractor basin is wide, chaotic networks seem to share such state space topologies. On the other end of the scale, if all states were part of a single attractor cycle, the network would most likely not be considered stable either. So the *length of attractors* (δ) does play a role in assessing a network's tendency to settle into stable patterns.
- *Network sensitivity* ($\bar{S}$) is defined as the number of nodes that change their state in response to a perturbation of a single randomly selected node. The less sensitive a given network is to perturbations, the more stable we consider the network to be. A gene g_i in the network with its corresponding Boolean function f_{g_i} and k_i inputs[3], that take on the value 1 for any of its possible input vectors with probability p_i has expected sensitivity [21]:

$$\bar{s}_i = 2k_i p_i(1 - p_i) \quad (5)$$

[3]This formula is for generalized random Boolean networks, where the in-degree k_i can vary from node to node.



(a) (b)

Figure 3: (a) Schematic of the evolutionary algorithm used to explore RBN ensembles. (b) Evaluation and Variation of Random Boolean Networks: We evaluate the networks with respect to their topologies, node functions, their associated state space, and selected activity patterns. Variations (mutations) are only performed on the topology and function level.

The expected network sensitivity is then normalized with respect to network size N:

$$\bar{S} = \frac{\sum_{i=1}^{N} \bar{s}_i}{N}. \tag{6}$$

- The *excess graph* measure (e) is defined as the difference between the number of edges and the number of nodes in a network $RBN_{(N,K)} = ((\boldsymbol{G}, \boldsymbol{E}), \mathcal{F})$ [16]:

$$e = |\boldsymbol{E}| - N. \tag{7}$$

The higher the excess graph number, the easier it is for a perturbation to move along the nodes in the network. As any edge (connection) implies an output from one node propagated as the input to another node, the more interconnecting edges a network has the more nodes depend on each others' state at any particular point in time. A higher degree of connectedness (i.e., the higher the excess graph number) increases the chance of any perturbating damage to be spread throughout the network. Thus network stability increases for lower excess graph numbers.

- The *basin entropy* (h) of a RBN R is defined as [15]:

$$h(R) = -\sum_{a} w_p \ln w_p, \tag{8}$$

where w_p denotes the weight of an attractor a which is defined as the total number of states in the state cycle of the attractor (its cycle length γ plus all of its basin states, β) normalized by the total size of the state space:

$$w_p(a) = \frac{\gamma(a) + \beta(a)}{|\mathcal{S}(R)|}. \tag{9}$$

Krawtiz and Shmulevich [15] introduce the basin entropy as a measure of complexity of information that a system is capable of storing. They also show that highest entropy is obtained by networks that are in the critical regime. Thus basin entropy can be used to monitor network criticality.

We will now describe what evolutionary algorithm we use to 'breed' RBNs and how we have incorporated the above RBN features into our fitness evaluations.

4.2 Evolutionary variation of RBNs

Starting from an initial, randomly generated set of RBNs, we want to apply an evolutionary algorithm scheme (Fig. 3a) to 'improve' these networks further with respect to the fitness criteria discussed above.

4.2.1 Mutation

We apply a variety of mutations—on both the RBN topology and the function set—and a crossover operator on the RBN topology, which we describe in this section. Figure 3b gives an overview of our variation scheme. We mutate either the topology of the regulatory network, or change the functions that govern the regulatory effects. In more detail, here are the mutation operators that we use, which all operate on the topology graph $\mathcal{G}$ of a $RBN_{(N,K)} = (\mathcal{G}, \mathcal{F})$ and on the functions assigned to its nodes:

- *Add a connection*: An edge is added to the RBN topology graph, $\mathcal{G}$, between two randomly selected genes in the network.
- *Remove a connection*: An edge within $\mathcal{G}$ is randomly selected and deleted from the graph.
- *Add a gene*: A new node (gene) is added to $\mathcal{G}$. Edges from K randomly selected genes in the network (including the added node) are connected to the newly inserted gene.
- *Remove a gene*: A randomly selected gene in the network is removed, including all connections associated with this node.
- *Mutate the Boolean function of a gene*: The Boolean function for a randomly selected gene in the network is mutated by randomly picking a new Boolean function for the gene from the set of the 2^{2^K} possible functions. The Hamming distance between the newly picked function and the old function can therefore be anywhere between zero (picking the same function again) and N (picking a function where all the bits are inverted).[4]

There is also a weight w associated with each mutation operator (Fig. 4). The weights are assigned based on how expensive (costly) each operation is. This gives us the chance to reflect biological constraints. For example, it is more costly to create a new gene (add a gene node) than adding or eliminating a transcription factor (add/delete edge). Therefore, we consider removing a connection much less costly than removing a gene. By removing a gene from the network one has to delete all the connections from and to the gene. The same idea holds when we look at adding a gene and adding a connection. In natural gene transcription networks removing a gene happens more frequently than adding a new gene. In order to integrate a completely new gene to an already established gene network many more steps are required. Usually, this also increases the complexity and affects the stability of the regulatory system, which may reduce this network's chance to pass through evolution's selection filters.

4.2.2 Crossover

Another form of variation we apply is crossover, which is performed on the topologies of two networks as demonstrated in Figure 5. Row by row, the complete connection matrix for each network is transformed into a binary string. According to 1-point crossover a series of bits is swapped between the two strings. At this level the crossover is only performed on RBNs with the same N and K values. At this stage of our model, the mutation operator and crossover can not be combined, because once a network is mutated most likely the number of genes (N) and the input degree (K) is no longer preserved. The initial population always starts as a set of classical Boolean networks where all the genes in the network have the same input degree and all the graphs in the population have exactly the same size. But after even a single mutation the RBNs are no longer bounded to specific N and K values. This limits the ability of crossover

[4]We have also tried multiple bit-flip mutations on the Boolean function strings, but this never resulted in any noticeable effect on the functionality of the network.

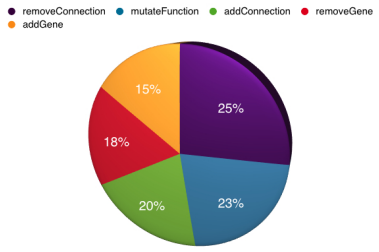


Figure 4: Distribution of weights associated with each mutation operator. The selection is weight proportionate. Thus, for example, *removeConnection* has the highest chance of being applied, then *mutateFunction* and so on. The displayed settings are used for our experiments as described in Section 5.

to be used in combination with mutation. We will therefore explore mutations and crossover separately (in Section 5).

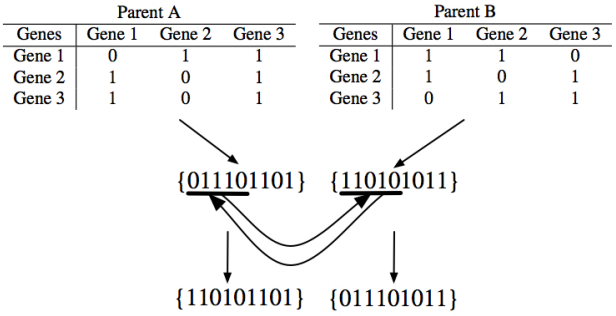


Figure 5: Genotypic details of crossover between two networks ($RBN_{(3,2)}$) which is performed on the corresponding gene connectivity matrices.

4.3 Evaluation and Selection

4.3.1 Fitness

The probability of an individual network to survive and reproduce a mutated version of itself into the next generation is determined by its fitness. In the context of this study, we are most interested in RBNs that display increased resistance to random perturbations. Thus, networks that are able to withstand perturbations will gain the highest fitness. We define RBN fitness to incorporate the number of attractors, the maximum length of its attractors, its sensitivity and the excess graph measure. Finally, we include the basin of entropy in order to move our evolved populations of RBNs towards the critical regime. Hence, the fitness Ω of a $RBN_{(N,K)}$ is defined as:

$$\Omega(RBN_{(N,K)}) = \frac{1}{\mu} + \frac{1}{\delta_{max}} + \frac{1}{\bar{S}} + e + h \qquad (10)$$

where μ represents the number of attractors, δ_{max} is the maximum attractor cycle length and $\bar{S}$ represents the network's sensitivity (all defined as in Section 4.1). Consequently, Ω enforces RBNs to have a small number of at-

tractors, a short maximum cycle length and low network sensitivity.

4.3.2 Selection

Following a general genetic algorithm scheme, we use fitness proportionate selection (roulette-wheel selection) to implement the filtering process from generation to generation. If Ω_i is the fitness of a network i in the population, its probability of being picked for mutation and subsequent placement into the next generation is:

$$p_i = \frac{\Omega_i}{\sum_{j=1}^{P} \Omega_j}. \quad (11)$$

Here P is the number of networks in the population. Figure 3a outlines the basic scheme of the evolutionary algorithm (EA) used in our study.

5. RESULTS

Figure 6 shows a representative evolutionary run with an initial population of five $RBN_{(8,7)}$ networks together with their state space diagrams. We evolve over 100 generations. For each network within a population, one of the five mutation operators is randomly selected; for each possible mutation (on the topology and functions) we change one third of the respective elements (nodes, edges, function strings, etc). The number in brackets below each RBN state space graph in Figure 6 is its fitness, $\Omega(RBN_{(8,7)})$. Initially, there are networks with long attractor cycles, but these mostly disappear by generation 100. One can observe a trend towards networks with single-point attractors, which are separated into sub-graphs. This means that these networks exhibit a higher level of stability, as perturbations won't spread over the entire network.

Figure 7 summarizes 40 evolution experiments, all started from twenty $RBN_{(7,6)}$ networks. Figure 7a represents the changes in the number of attractors over 100 generations. Due to the restricted size of the networks evolved ($RBN_{(7,6)}$), the number of attractors is relatively small. However this number decreases from 3 down to 2. This shows that by enforcing the fitness function to lower the number of attractors in subsequent generations, the evolutionary model in fact evolves RBNs toward networks with a lower number of attractors. As shown in Figure 7b the maximum attractor cycle length decreases from 8 nodes to 4 nodes. Network sensitivity and excess graph both decrease over the course of evolution as shown in Figures 7c and 7d, respectively. This decrease indicates that fewer nodes are affected as a result of a single random perturbation and also that damage spreading is minimized. Having lower excess graph entails that a given perturbation can not spread easily through out the network increasing the network's stability. Basin entropy (Fig. 7e) is also decreasing. This indicates a decrease in the complexity of the network. The high level of entropy is obtained at the boundary between the ordered and chaotic regimes. Any deviations from the boundary would either drive a network into the ordered regime (decreasing entropy) or into the chaotic regime (constant entropy). The decrease in the basin entropy indicates that evolution moves toward more stable networks. The overall fitness of individuals in each generation is plotted in Figure 7f. The evolutionary model increases the overall fitness of individuals. Figure 6a shows the five fittest individuals per population. After 100 generations the fitness is increased by about 300%. The average fitness of the best five individuals increases from 2.58 to 7.822.

As a second genetic operator we investigated crossover (Fig. 5). As crossover is performed on the connection matrix it requires two RBNs of the same size and input degree. This means that the excess graph value remains unchanged, as it is defined as the difference between the edge and node numbers in a network. The same argument applies to network sensitivity and entropy. Other network parameters, such as the number of attractors and attractor cycle length, did not show any significant change either, if crossover was the only genetic operator applied.[5]

6. CONCLUSION

In this study we investigated RBNs through evolutionary exploration of their topologies and state spaces. We demonstrated how networks can be evolved towards higher stability by taking the number of attractors, the length of attractors as well as their sensitivity, criticality (basin entropy) and excess graph into account. There are numerous other network parameters, such as average basin transient time and path diversity [22], and frozen and relevant nodes, which we will investigate further. All our experiments are implemented in Mathematica. For the complete set of notebooks to repeat our experiments discussed in this paper as well as further material see: `http://www.swarm-design.org/RBN`.

7. REFERENCES

[1] P. W. Anderson. Suggested model for prebiotic evolution: the use of chaos. *Proc. Natl Acad. Sci.*, 80:3386–3390, 1983.

[2] S. Bornholdt and T. Rohlf. Topological evolution of dynamical networks: Global criticality from local dynamics. *Phys. Rev. Lett.*, 84(26):6114–6117, Jun 2000.

[3] S. Bornholdt and K. Sneppen. Neutral mutations and punctuated equilibrium in evolving genetic networks. *Phys. Rev. Lett.*, 81(1):236–239, 1998.

[4] B. Derrida, E. Gardner, and A. Zippelius. An exactly soluble asymmetric neural network model. *Europhysics Letters (EPL)*, 4(167), 1987.

[5] B. Derrida and D. Stauffer. Phase transitions in two-dimensional kauffman cellular automata. *Europhysics Letters (EPL)*, 2(10):739–745, 1986.

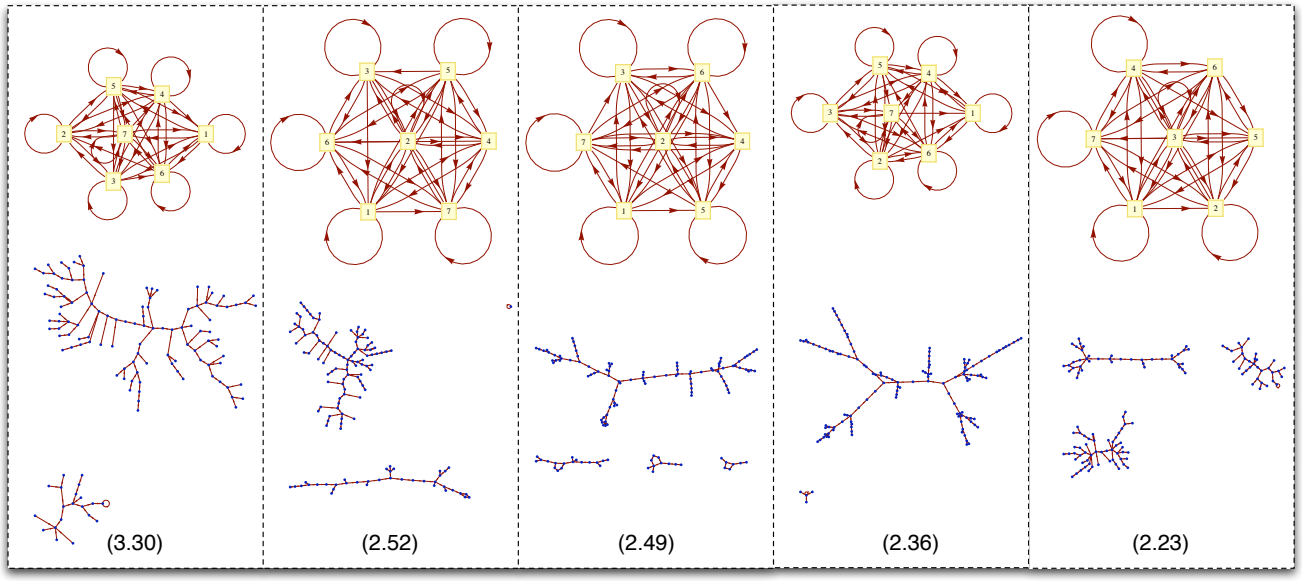
[6] B. Drossel, T. Mihaljev, and F. Greil. Number and length of attractors in a critical kauffman model with connectivity one. *Phys Rev Lett*, 94(8):088701, Mar 2005.

[7] H. J. K. Hawick and C. Scogings. Simulating large random boolean networks. *Res. Lett. Inf. Math. Sci.*, 11:33–43, 2007.

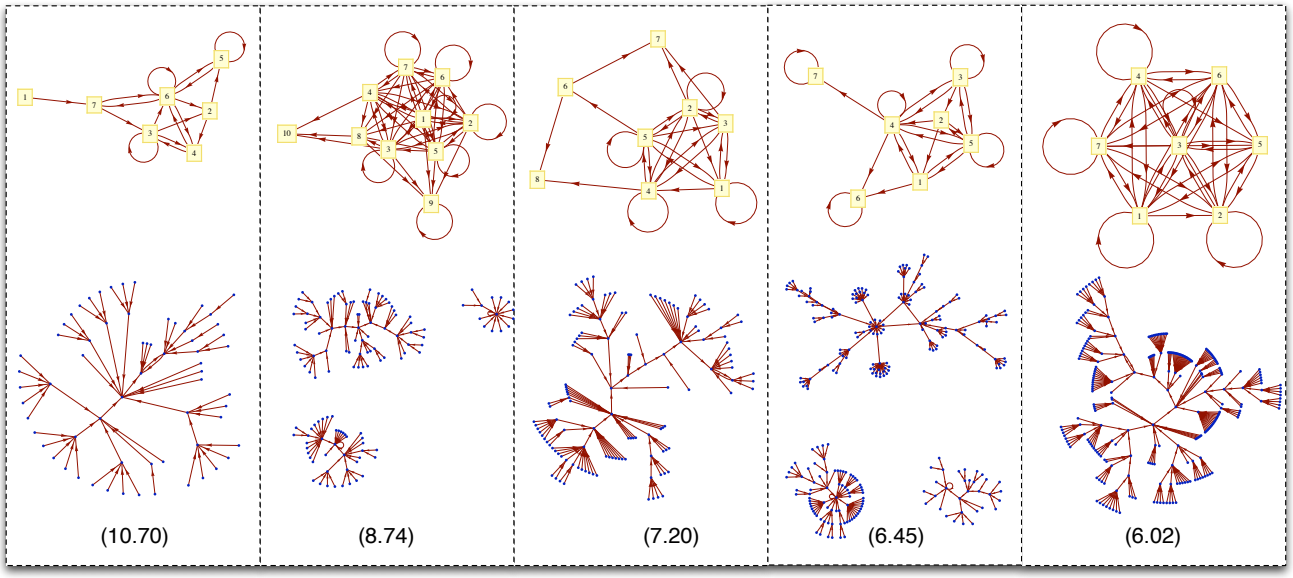
[8] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Natl Acad. Sci.*, 79:2554–2558, 1982.

[9] K. Iguchi, S. Kinoshita, and H. Yamada. Rugged fitness landscapes of kauffman models with a scale-free network. *Physical review E, Statistical, nonlinear, and soft matter physics*, 72(6 Pt 1):061901, Dec 2005.

[5] A mixture of crossover and mutations is problematic as well and requires further investigation.

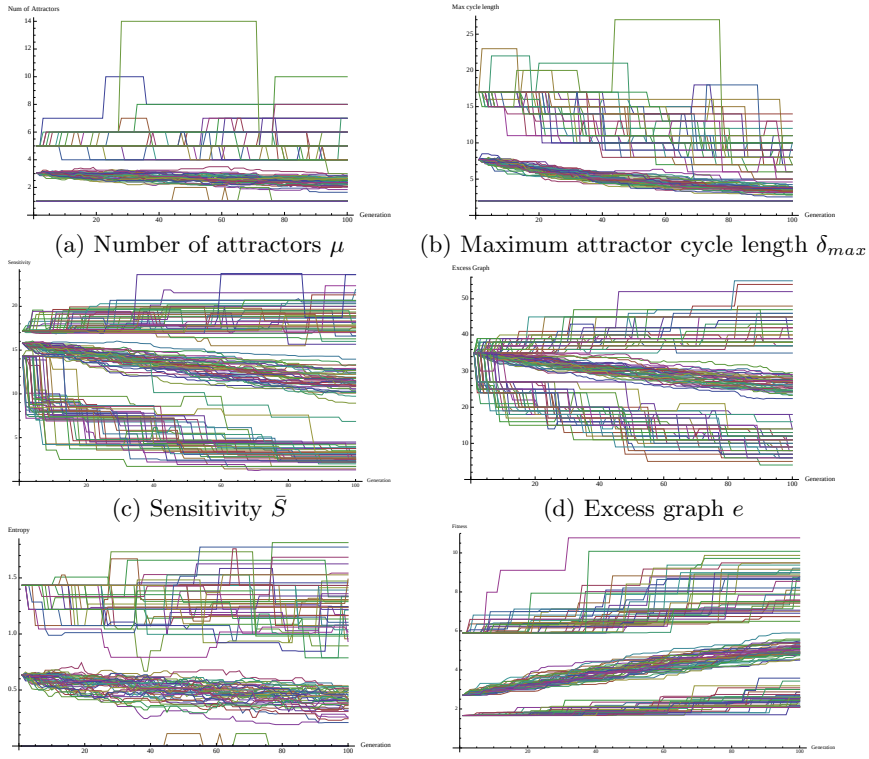


(a) Top 5 individuals in generation 1



(b) Top 5 individuals in generation 100

Figure 6: The topologies and their corresponding state spaces for the initial population (generation 1,$RBN_{7,6}$ networks) and last population (generation 100) is shown here for a typical evolutionary run. We use a population size of 10 and a mutation rate of 20%. The best 5 individuals (fitnesses in brackets) are shown for each generation. Note that the number of nodes N in the evolved networks does not change much for this example, that is $7 \leq N \leq 10$. The in-degrees vary within a wider range: $1 \leq K \leq 10$.



(a) Number of attractors μ (b) Maximum attractor cycle length δ_{max}

(c) Sensitivity $\bar{S}$ (d) Excess graph e

(e) Basin Entropy h (f) Overall fitness Ω

Figure 7: Evolution of $N = 7$, $K = 6$ RBNs over 40 runs. A population of 20 networks is evolved over 100 generations. The three separate groups of plots in each graph represent the best, average, and worst fitness, respectively from top to bottom.

[10] K. Iguchi, S.-I. Kinoshita, and H. S. Yamada. Boolean dynamics of kauffman models with a scale-free network. *J Theor Biol*, 247(1):138–51, Jul 2007.

[11] S. Kauffman. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, 1993.

[12] S. Kauffman. A proposal for using the ensemble approach to understand genetic regulatory networks. *J Theor Biol*, 230(4):581–90, Oct 2004.

[13] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[14] S. A. Kauffman and R. G. Smith. Adaptive automata based on darwinian selection. *Physica D*, 2(1-3):68–82, 1986.

[15] P. Krawitz and I. Shmulevich. Basin entropy in boolean network ensembles. *Phys Rev Lett*, 98(15):158701, Apr 2007.

[16] P. Krawitz and I. Shmulevich. Entropy of complex relevant components of boolean networks. *Physical review E*, 76(3 Pt 2):036115, Sep 2007.

[17] N. Lemke, J. C. M. Mombach, and B. E. J. Bodmann. A numerical investigation of adaptation in population of random boolean networks. *Physica A*, 301:589–600, 2001.

[18] M. T. Matache and J. Heidel. Random boolean network model exhibiting deterministic chaos. *Physical review E*, 69(5 Pt 2):056214, May 2004.

[19] T. Mihaljev and B. Drossel. Scaling in a general class of critical random boolean networks. *Physical review E, Statistical, nonlinear, and soft matter physics*, 74(4 Pt 2):046101, Oct 2006.

[20] L. Raeymaekers. Dynamics of boolean networks controlled by biologically meaningful functions. *Journal of Theoretical Biology*, 2001.

[21] I. Shmulevich and S. A. Kauffman. Activities and sensitivities in boolean network models. *Phys Rev Lett*, 93(4):048701, Jul 2004.

[22] A. Shreim, A. Berdahl, V. Sood, P. Grassberger, and M. Paczuski. Complex network analysis of state spaces for random boolean networks, 2007.

[23] J. E. S. Socolar and S. A. Kauffman. Scaling in ordered and critical random boolean networks. *Phys Rev Lett*, 90(6):068702, 2003.

[24] Z. Somogyvári and S. Payrits. Length of state cycles of random boolean networks: an analytic study. *Journal of Physics A: Mathematical and General*, 33(38):6699–6706, 2000.

[25] J. Watson, N. Geard, and J. Wiles. Towards more biological mutation operators in gene regulation studies. *BioSystems*, 76(1-3):239–48, Jan 2004.

[26] S. Wolfram. *Cellular Automata and Complexity*. Addison-Wesley, Reading, 1994.

[27] S. Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.