

Boolean Networks Design by Genetic Algorithms

Andrea Roli*, Cristian Arcaroli, Marco Lazzarini, Stefano Benedettini

DEIS-Cesena, *Alma Mater Studiorum* Università di Bologna, Italy

Abstract. We present and discuss the results of an experimental analysis in the design of Boolean networks by means of genetic algorithms. A population of networks is evolved with the aim of finding a network such that the attractor it reaches is of required length l . In general, any target can be defined, provided that it is possible to model the task as an optimisation problem over the space of networks. We experiment with different initial conditions for the networks, namely in ordered, chaotic and critical regions, and also with different target length values. Results show that all kinds of initial networks can attain the desired goal, but with different success ratios: initial populations composed of critical or chaotic networks are more likely to reach the target. Moreover, the evolution starting from critical networks achieves the best overall performance. This study is the first step toward the use of search algorithms as tools for automatically design Boolean networks with required properties.

1 Introduction

The design of complex systems is one of the main challenges in scientific and engineering disciplines. Model synthesis, identification and tuning, reverse engineering of biological and social networks, design of self-organising artificial systems are just some of the areas in which scientists are asked to face this issue. Such systems and models are mostly designed and tuned by means of automatic procedures, some of which can be ascribed to the class of *search methods*. A prominent example of these approaches are evolutionary computation techniques, for instance for designing robotic systems [1].

In this paper we present and discuss results of a preliminary study in the context of automatic design of Boolean networks via Genetic algorithms. Boolean networks (BNs) have been introduced by Kauffman as a model for genetic regulatory networks [2] and have been also studied as computational learning systems [3,4]. The first study on the evolution of BNs can be found in [5], in which the authors apply a simple evolutionary algorithm to evolve BNs with an attractor containing a target state. A follow-up of that seminal work is that of Lemke et al. [6], in which the fitness function accounts also for a desired attractor length. These studies are mainly an investigation of how evolution performs over BNs and raise interesting and fundamental questions on the search landscape structure and the evolutionary dynamics depending on network structural

* Corresponding author. Email: andrea.roli@unibo.it

characteristics. More recently, works addressing the evolvability of robustness in BNs have been presented [7,8,9]. In the same direction is a recent paper, [10], in which the global fitness function is defined as the sum of single functions, each related to a network parameter somehow linked to network robustness (e.g., number and length of attractors).

Despite the amount of analytical studies on the properties of BNs and their effectiveness in capturing fundamental genetic phenomena, little effort has been received so far concerning their synthesis. The availability of tools for automatic design of BNs would make it possible to design and tune BNs for applications in genetics, as genetic regulatory network models [11], and robotics, as multi-functional controllers.

This contribution is a first step toward the development of a family of tools for automatic design of BNs and discrete dynamical systems in general. We first introduce preliminary definitions and concepts in Sections 2 and 3; experimental settings and results are described in Section 4. We then conclude with an outline of the agenda for future research in Section 5.

2 Boolean networks

BNs have been firstly introduced by Kauffman [2] and subsequently received considerable attention in the composite community of complex systems. Recent advances in this research field can be mainly found in works addressing themes in genetic regulatory networks or investigating properties of BNs themselves [7,12,13,14].

A BN is a discrete-state and discrete-time dynamical system defined by a directed graph of N nodes, each associated to a Boolean variable x_i , $i = 1, \dots, N$, and a Boolean function $f_i(x_{i_1}, \dots, x_{i_{K_i}})$, where K_i is the number of inputs of node i . Often, K_i is chosen to be equal to a constant value K for every i . The arguments of the Boolean function f_i are the nodes whose outgoing arcs are connected to node i . The state of the system at time t , $t \in \mathbb{N}$, is defined by the array of the N Boolean variable values at time t : $s(t) = (x_1(t), \dots, x_N(t))$. The most studied dynamics for BNs is *synchronous*, i.e., nodes update their states in parallel, and *deterministic*. However, many variants exist, including asynchronous and probabilistic update rules [15].

In this work, we consider networks ruled by synchronous and deterministic dynamics. Given this setting, the network trajectory in the N -dimensional state space is a sequence of states composing a *transient*, possibly empty, followed by an *attractor*, that is a cycle of length $l \in [1, \dots, 2^N]$. States that belong to a trajectory ending at attractor A_i are said to be members of the *basin of attraction* of A_i . When BNs are employed as genetic regulatory network models, attractors assume a notable relevance as they can be interpreted as cellular types [16].

A special category of BNs that has received particular attention is that of Random BNs, which can capture relevant phenomena in genetic and cellular mechanisms and complex systems in general. Random BNs (RBNs) are usually generated by choosing at random K inputs per node and by defining the

Boolean functions by assigning to each entry of the truth tables a 1 with probability p and a 0 with probability $1 - p$. Parameter p is called *homogeneity* or *bias*. Depending on the values of K and p the dynamics of RBNs is *ordered* or *chaotic*. In the first case, the majority of nodes in the attractor is frozen and any moderate-size perturbation is rapidly dampened and the network returns to its original attractor. Conversely, in chaotic dynamics, attractor cycles are very long and the system is extremely sensitive to small perturbations: slightly different initial states lead to divergent trajectories in the state space. RBNs temporal evolution undergo a second order phase transition between order and chaos, governed by the following relation between K and p : $K_c = [2p_c(1-p_c)]^{-1}$, where the subscript c denotes the critical values [17]. Networks along the *critical* line have important properties, such as the capability of achieving the best balance between evolvability and robustness [7] and maximising the average mutual information among nodes [13]. Hence the conjecture that living cells, and living systems in general, are critical [18].

In this work we are interested in designing BNs such that the attractor they reach from a given initial state has a target length: this represents just one of numerous examples of requirements we may want a BN to satisfy. Nevertheless, since attractor length depends on the main properties of BNs, this goal enables us to address some of the most relevant issues in BN design.

3 Genetic algorithms

Genetic algorithms (GAs) belongs to the family of evolutionary computation methods and have been successfully applied as search techniques since several decades [19,20,21]. Inspired by Darwin’s theory of selection and natural evolution, a GA evolves a population of candidate solutions to a problem by iteratively selecting, recombining and modifying them. The driving force of the algorithm is selection, that biases search toward the *fittest* solutions, i.e., those with the highest objective function value. Algorithm 1 shows the basic structure of a GA.

Algorithm 1 Genetic Algorithm

```

 $P \leftarrow \text{GenerateInitialPopulation}()$ 
Evaluate( $P$ )
while termination conditions not met do
   $P' \leftarrow \text{Recombine}(P)$ 
   $P'' \leftarrow \text{Mutate}(P')$ 
  Evaluate( $P''$ )
   $P \leftarrow \text{Select}(P'' \cup P)$ 
end while

```

The function $\text{Evaluate}(P)$ computes the fitness of each individual of population P . The fitness function is positively correlated with the objective function, that quantifies the quality of a candidate solution and it is usually normalised

in the range $[0,1]$. In the next section, we detail the specific genetic algorithm used in our experiments.

4 Experimental analysis

The long term aim of this study is the definition and implementation of automatic procedures and methodologies for designing BNs and similar systems. The availability of such procedures would make it possible to perform inference of real genetic networks and to study the effects of evolution on simple genetic models [11]. Furthermore, a promising yet uninvestigated research area consists of using BNs to control autonomous systems: the same BN-controller can produce different behaviours, depending on the attractor it is traversing. The actual behaviours have to be encoded into a proper sequence of states, hence the need for a procedure for defining the network according to the requirements.

In this work, our goal is to investigate the possibility of evolving BNs by GAs so as to obtain a network able to reach an attractor of a desired period with a trajectory starting from a given initial state s_0 . The questions that we want to address are the following:

- (a) Is it possible to guide evolution in such a way to succeed in the goal? What is the probability of reaching the target? (i.e., how robust is the automatic design procedure?)
- (b) Are there differences across network parameters? Are there networks that are *easier* to evolve?
- (c) Which are the most difficult or the easiest targets to be reached?
- (d) What is the influence of GA parameters?
- (e) What are the effect of the evolution on networks structure?

In the remainder of this section we detail the experimental settings and report and discuss the experimental results.

4.1 Experimental settings

Experiments are run with networks of 100 nodes and $K = 3$. The initial state is chosen at random and the target attractor lengths are 1, 10, 50, 100, 500, 800. Networks composing the initial population are constructed by randomly assigning inputs, without self-inputs; Boolean functions are defined by assigning truth values biased by homogeneity values equal to 0.85 (ordered), 0.788675 (critical) and 0.5 (chaotic), in three different experiment series, respectively. However, Boolean functions homogeneity of single individuals can change during evolution because the initial distribution of 1s and 0s can be changed by the genetic operators. For efficiency reasons, the temporal evolution of each network is simulated for at most 1000 steps: if an attractor is not reached in this limit, a fitness value of 0 is returned. The individuals of the GA are encoded as a tuple of N binary vectors of size 2^K , each defining the Boolean function of a node.

Table 1. Summary of experimental parameter values. All the possible combinations of the values reported have been tested.

N	K	p	attractor length	population size	number of generations	mutation / crossover rate	number of runs
100	3	0.788675	1	80	200	0.5 / 0.9	100
			10				
			50				
			100				
			500				
800							

Thus, only Boolean functions of a network are evolved and the connections are kept constant. The recombination operator is a one-point crossover and mutation is a single-variable flip. Both operators are applied chromosome-wise. The fitness function is defined as $F(net) = (1 + |l - l_t|)^{-1}$, where l is the length of the attractor the individual network reached and l_t is the target length. The remaining parameters of the GA have been chosen as reported in Table 1, in which a summary of experimental parameter values is provided. All the possible combinations of the values reported have been tested.

BNs have been simulated with a BN simulator developed by the group of Artificial Intelligence and Complex Systems of DEIS-Cesena, further extended into the BN Simulation Toolkit [22] and the GA has been implemented with GAUL [23]. All experiments have been performed on a 2.4 GHz Intel Core 2 Quad with 4MB of cache and 2GB of RAM, running with Linux Ubuntu 8.10.

4.2 Performance comparison

We first discuss the results concerning the performance of each class of networks, addressing questions (a), (b) and (c). The first notable observation is that **for all** target attractor lengths and **for all** initial network classes the GA could find at least one network with maximal fitness in the 100 independent runs. This result means that all three classes of networks can be evolved to successfully reach the target. To assess the robustness of the process, we compare the fraction of successful runs at each generation of the algorithm, i.e., we estimate the *success probability at generation t* , defined as the probability that a network with maximal fitness is found at generation $t' \leq t$. The corresponding plots are depicted in Figures 1, 2. Results for attractor lengths of 1 and 10 are omitted, because the fraction of runs achieving maximal fitness reaches the 100% right in the initial population or after few generations.

We first note that the performance achieved with initially ordered networks is considerably lower than that of critical and chaotic ones. This can be ascribed to the fact that ordered networks are not very likely to have long attractors. Anyway, the search process performed by the GA is still able to find a network with the desired attractor length. The case of critical and chaotic networks has

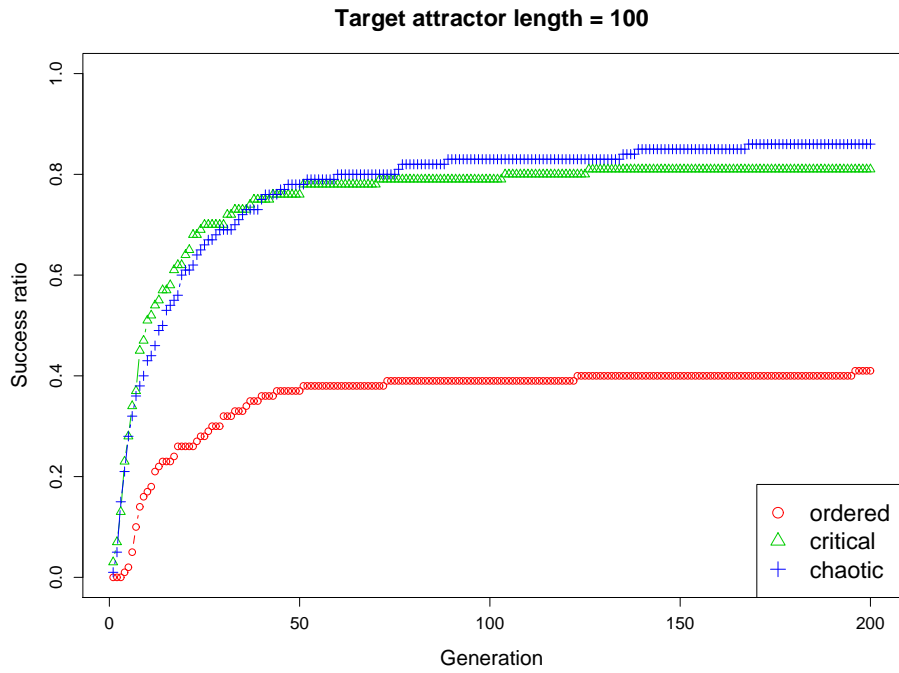
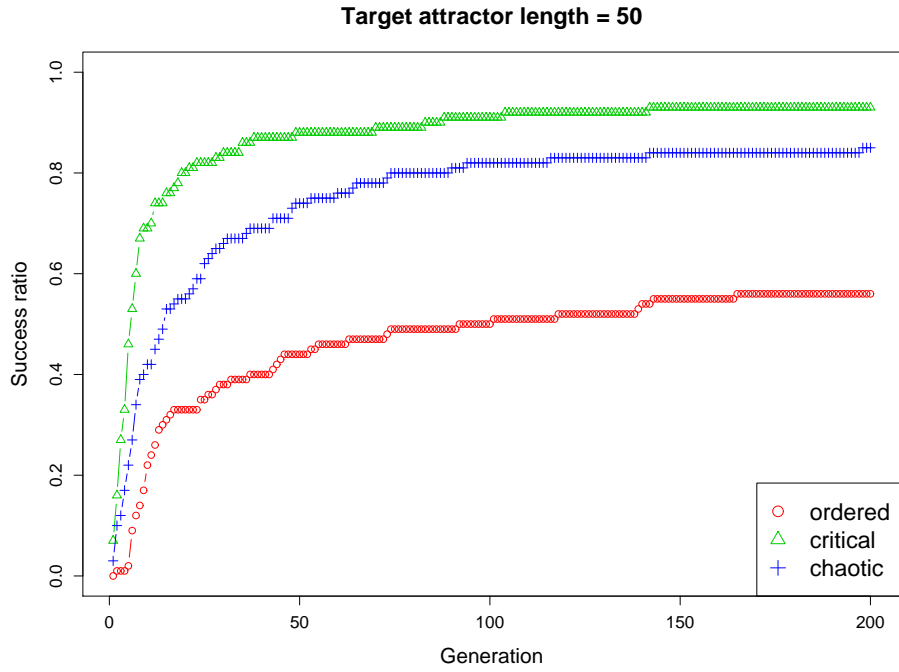


Fig. 1. Success ratio vs. generations. The comparison is made among the three initial network classes. Target attractor lengths equal to 50 and 100.

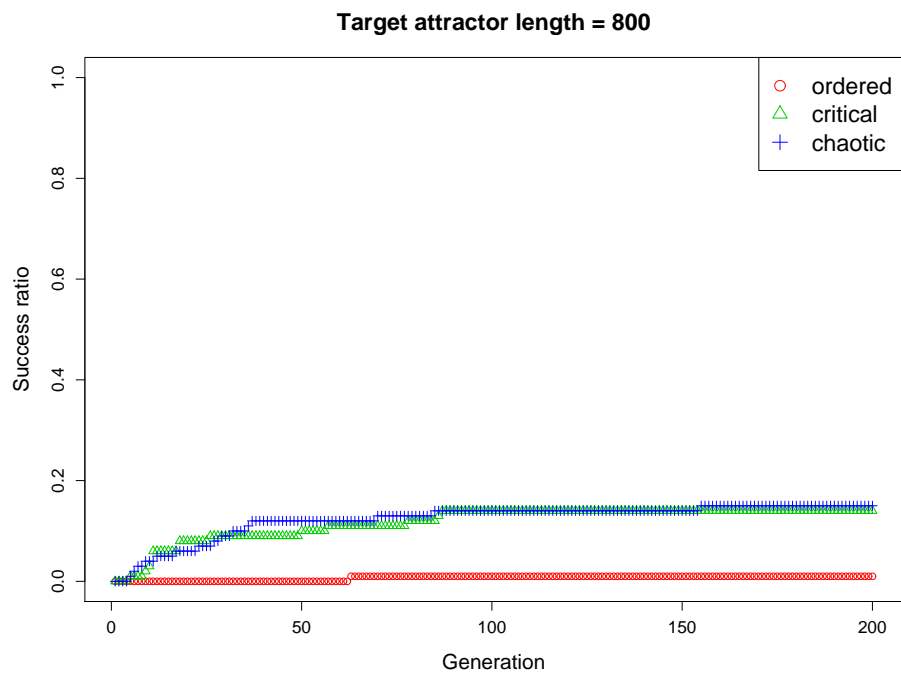
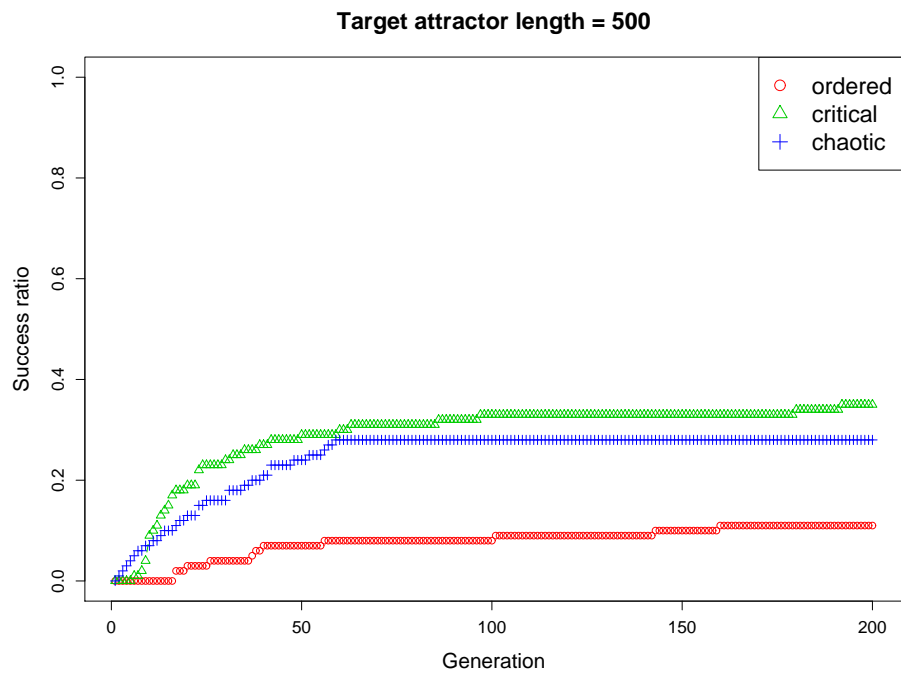


Fig. 2. Success ratio vs. generations. The comparison is made among the three initial network classes. Target attractor lengths equal to 500 and 800.

some subtleties which deserve to be outlined. First of all, we observe that the success ratio decreases as the attractor length increases. Moreover, in most cases critical networks dominate or are almost equivalent to chaotic ones, while for target attractor length equal to 100, initial chaotic networks seems to provide a better start to the GA. Both the phenomena can be explained by the combination of two factors. First: the cutoff imposed on simulation steps limits from above the networks attractor length, hence making it difficult to evolve networks with an attractor of length comparable with the maximal number of simulation steps because, if an attractor is not found, the corresponding fitness value is zero. Second: critical networks have usually many attractors, but of small length compared to attractor periods of chaotic networks, that can be exponential in the number of nodes. In a survey experimental analysis, we observed that for networks with 100 nodes and a maximal number of simulation steps of 1000, the median attractor length for critical networks is 6, while for chaotic ones is 130. Therefore, for a target length of 100, the fitness of individuals composing the initial population is likely to be higher in the case of chaotic networks than in critical ones. However, it is worth to be noted that critical networks can be anyway evolved to reach long attractors, despite their handicap in the initial population’s fitness. This could be a further evidence of their tendency of maximising adaptiveness. The study of the search space, that would provide insight into problem hardness, is subject of ongoing work.

4.3 Influence of GA parameters

The influence of mutation and crossover on search performance can shed light on the evolution characteristics of the different initial population classes and can answer question (d). Figures 3, 4 show a typical case¹ of algorithm performance in the three examined cases of mutation and crossover rates. From the plots we observe that the synergy of both mutation and crossover are crucial for the evolution of initially ordered and critical networks. Conversely, for chaotic networks, mutation is much more important than crossover.

4.4 Effect of evolution on Boolean function homogeneity

We conclude this analysis by comparing homogeneity distribution at the beginning and at the end of the search. These results should be taken *cum grano salis*, as evolved networks might not have the very same properties as the random initial ones and a complete answer to question (e) requires also to study the properties of Boolean functions as well as network dynamics. Nevertheless, since only Boolean functions are evolved and topology is kept constant, the evolution of homogeneity can still provide some insights into the effects of evolution on the initial BNs. The average homogeneity of the best individual in the initial and final populations are compared in Table 2, where statistically significantly

¹ Target attractor length equal to 100.

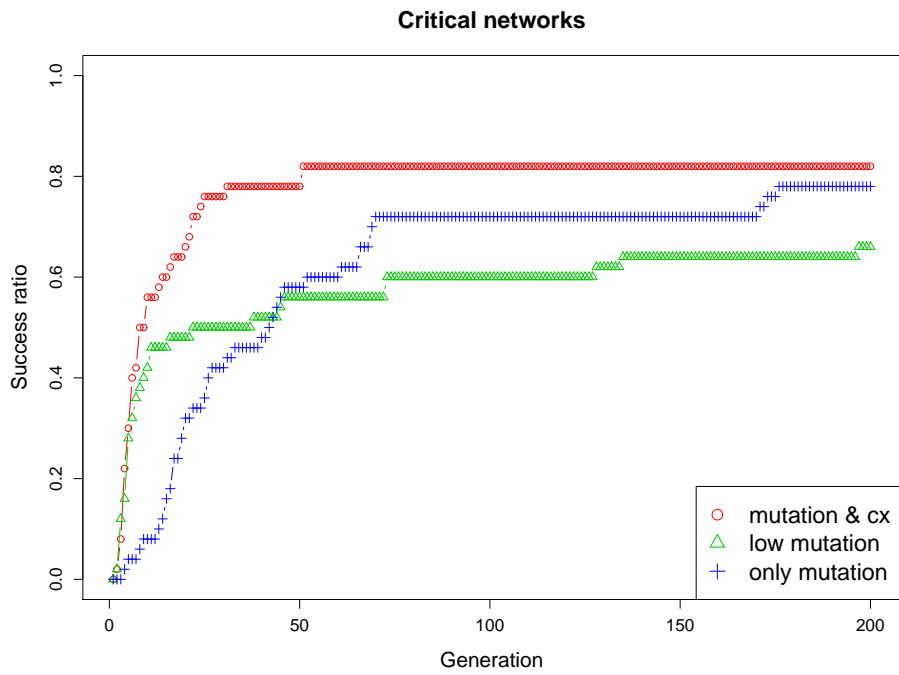
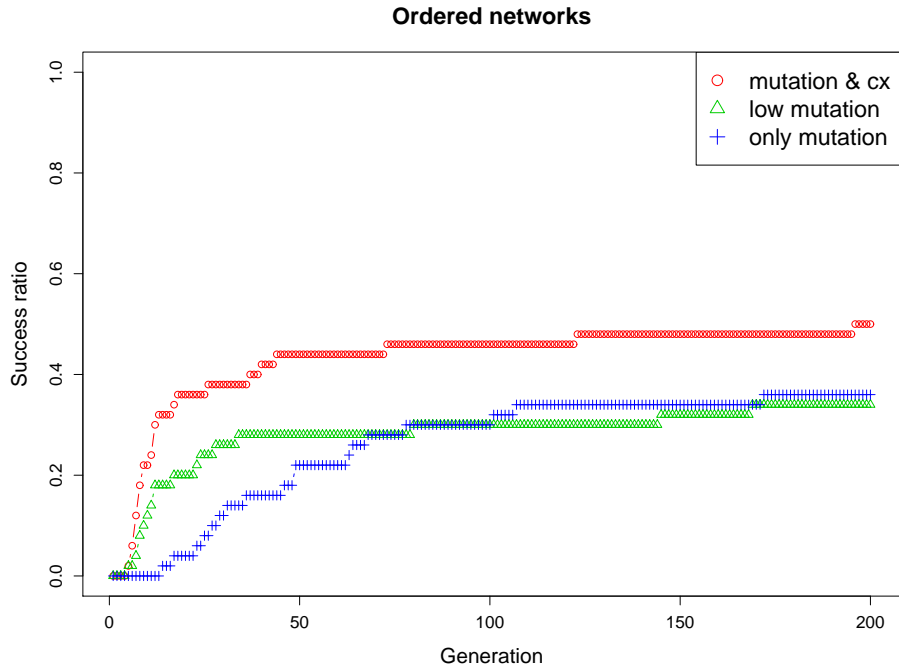


Fig. 3. Comparison of the impact of mutation and crossover on search performance. The case of ordered and critical initial network classes are reported.

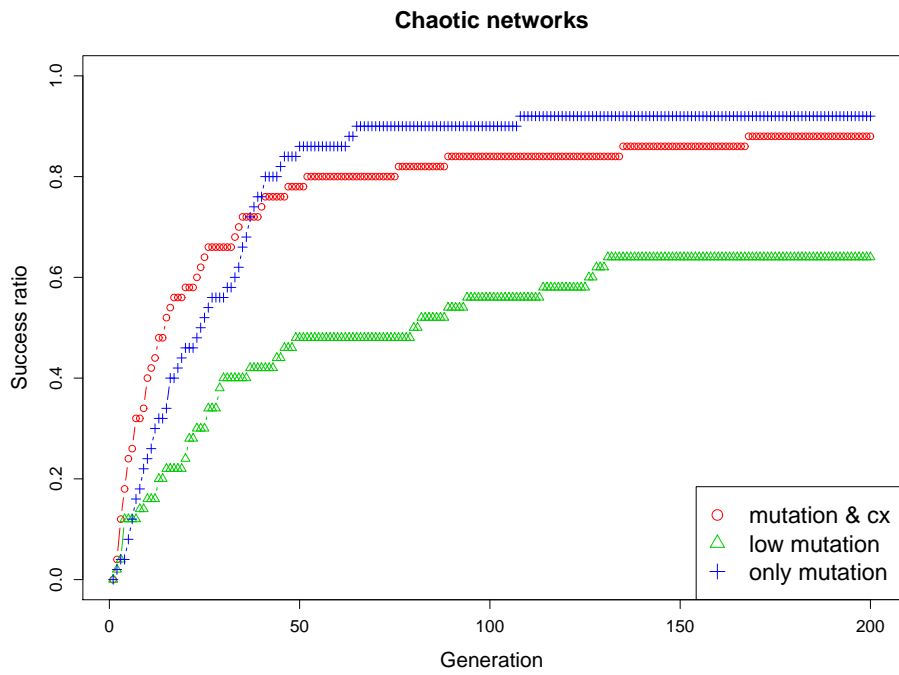


Fig. 4. Comparison of the impact of mutation and crossover on search performance. The case of initial chaotic network class is reported.

Table 2. Comparison of the average homogeneity of the best individual in the initial and final population. Significantly differing averages, i.e., those which passed the Wilcoxon test with confidence level 95%, are denoted by a star.

Target attr. length	Ordered networks		Critical networks		Chaotic networks	
	Best indiv. average homogeneity initial/final		Best indiv. average homogeneity initial/final		Best indiv. average homogeneity initial/final	
	Successful runs	Unsuccessful runs	Successful runs	Unsuccessful runs	Successful runs	Unsuccessful runs
50	0.8474/0.8369*	0.8447/0.8458	0.7807/0.7790	0.7895/0.7884	0.5023/0.4986	0.4943/0.5016
100	0.8456/0.8345*	0.8459/0.8380*	0.7844/0.7752*	0.7818/0.7822	0.5032/0.5034	0.4964/0.4996
500	0.8434/0.8285	0.8465/0.8365*	0.7860/0.7688*	0.7824/0.7735*	0.5019/0.5046	0.5010/0.5018
800	0.8325/0.7962	0.8463/0.8346*	0.7854/0.7770	0.7834/0.7700*	0.4951/0.5034	0.4997/0.5039

differing averages are denoted by a star.² We can observe a mild tendency of homogeneity decrease for ordered and critical networks, while the GA does not affect homogeneity in chaotic networks. The conclusion we can draw is that, in our experimental setting, the GA does not dramatically change the distribution of 0s and 1s, even if there are some clues suggesting that ordered and critical networks are more affected than chaotic ones and they are somehow pushed towards the chaotic region. However, a more detailed analysis is required before drawing strong conclusions on the effect of GA on network structure.

5 Conclusion and outlook to future work

In this work, we have presented and discussed results of the evolutionary design of BNs with a desired attractor length. We have shown that it is possible to find networks with such a property for every kind of initial class: ordered, chaotic and critical. Search performance starting from critical and chaotic networks is considerably higher than in the case of ordered networks. Another important outcome of the experiments is that, critical networks are a good start for GA for all the target attractor lengths tested, despite the low probability of finding long attractors in those networks. This work is just a first step in this research area. A future research agenda include: *(i)* relaxing the constraint of keeping constant the initial state, thus moving to stochastic search problems (as the enumeration of all possible initial states is impractical), *(ii)* evolving also network topology and exploring the use of different search algorithms, mainly metaheuristics and their hybrids. Finally, we are also planning to experiment with other targets, such as specific patterns in the attractors and combinations thereof, aiming at the design of networks with a desired landscape of attractors, each with a specific characteristic.

Acknowledgements

We thank Roberto Serra, Marco Villani and the anonymous referees for useful comments and suggestions.

² i.e., those which passed the Wilcoxon test with confidence level 95%.

References

1. Nolfi, S., Floreano, D.: Evolutionary robotics. The MIT Press (2000)
2. Kauffman, S.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press (1993)
3. Kauffman, S.: Antichaos and adaptation. *Scientific American* **265**(2) (1991) 78–84
4. Dorigo, M.: Learning by probabilistic Boolean networks. In: Proceedings of World Congress on Computational Intelligence – IEEE International Conference on Neural Networks, Orlando, Florida, USA (1994) 887–891
5. Kauffman, S.: Adaptive automata based on darwinian selection. *Physica D* **22** (1986) 68–82
6. Lemke, N., Mombach, J., Bodmann, B.: A numerical investigation of adaptation in populations of random boolean networks. *Physica A* **301** (2001) 589–600
7. Aldana, M., Balleza, E., Kauffman, S., Resendiz, O.: Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology* **245** (2007) 433–448
8. Szejka, A., Drossel, B.: Evolution of canalizing Boolean networks. *European Physical Journal B* **56** (2007) 373–380
9. Braunewell, S., Bornholdt, S.: Reliability of genetic networks is evolvable. *Physical Review E* **77** (2008) 060902–1–4
10. Esmaeili, A., Jacob, C.: Evolution of discrete gene regulatory models. In ACM, ed.: Proceedings of GECCO’08. (2008) 307–314
11. Kauffman, S.: A proposal for using the ensemble approach to understand genetic regulatory networks. *Journal of Theoretical Biology* **230** (2004) 581–590
12. Fretter, C., Drossel, B.: Response of boolean networks to perturbations. *European Physical Journal B* **62** (2008) 365–371
13. Ribeiro, A., Kauffman, S., Lloyd-Price, J., Samuelsson, B., Socolar, J.: Mutual information in random Boolean models of regulatory networks. *Physical Review E* **77**(011901) (2008)
14. Serra, R., Villani, M., Graudenzi, A., Kauffman, S.A.: Why a simple model of genetic regulatory networks describes the distribution of avalanches in gene expression data. *Journal of Theoretical Biology* **246** (2007) 449–460
15. Gershenson, C.: Introduction to random Boolean networks. In: Proceedings of ALife IX, Boston, MA (2004) 160–173
16. Huang, S., Ingber, D.: A non-genetic basis for cancer progression and metastasis: Self-organizing attractors in cell regulatory networks. *Breast Disease* **26** (2006,2007) 27–54
17. Derrida, B., Pomeau, Y.: Random networks of automata: a simple annealed approximation. *Europhysics Letters* **1**(2) (1986) 45–49
18. Nykter, M., Price, N., Aldana, M., Ramsey, S., Kauffman, S., Hood, L., Yli-Harja, O., Shmulevich, I.: Gene expression dynamics in the macrophage exhibit criticality. In: Proceedings of the National Academy of Sciences, USA. Volume 105. (2008) 1897–1900
19. Holland, J.: Adaption in natural and artificial systems. The University of Michigan Press, Ann Harbor, MI (1975)
20. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading, MA (1989)
21. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, Germany (1996)
22. Benedettini, S.: The Boolean networks toolkit. Available at: <http://booleannetwork.sourceforge.net> Viewed: September 2009.
23. <http://gaul.sourceforge.net/> Viewed: September 2009.