Stability of the Kauffman Model

Sven Bilke and Fredrik Sjunnesson[1]

Complex Systems Division, Department of Theoretical Physics
University of Lund, Sölvegatan 14A, S-223 62 Lund, Sweden
http://www.thep.lu.se/complex/

Abstract:

Random Boolean networks, the Kauffman model, are revisited by means of a novel decimation algorithm, which removes variables that cannot be relevant to the asymptotic dynamics of the system. The major part of the removed variables have the same fixed state in all limit cycles. These variables are denoted the stable core of the network and their number grows approximately linearly with N, the number of variables in the original network. The sensitivity of the attractors to perturbations is investigated. We find that reduced networks lack the well known insensitivity observed in full Kauffman networks. We conclude that, somewhat counter-intuitive, this remarkable property of full Kauffman networks is generated by the dynamics of their stable core. The decimation method is also used to simulate large critical Kauffman networks. For networks up to $N = 32$ we perform *full enumeration* studies. Strong evidence is provided for that the number of limit cycles grows linearly with N. This result is in sharp contrast to the often cited $\sqrt{N}$ behavior.

---

[1]{sven,fredriks}@thep.lu.se

1 Introduction

Boolean networks were introduced by Kauffman [1,2] as simplified models of the complex interaction in the regulatory networks of living cells. The binary variable σ_i encodes the activity of the effective "gene" i; expressed or not expressed. Depending upon the initial state, the system evolves to one of possibly several limit cycles. In the biological picture, the different limit cycles are interpreted as different cell types. One of Kauffman's motivations for investigating these networks was the idea that the structure of genetic networks present in nature is not only determined by selection. Rather, a good fraction of the network functionality is inherent in the ensemble of regulatory networks as such. In fact, he found an ensemble of critical Boolean networks "on the edge of chaos" that captures some features observed in nature. These Boolean networks show a remarkable stability; in most cases small perturbations of the state of the network do not change the trajectory to a different limit cycle. This is desirable in the biological interpretation since stability of genetic regulatory networks against small fluctuations is a crucial property. Another striking observation is that the number of limit cycles for the critical Boolean networks grows as a square-root of the system size [1, 2]. This is an analogy to multicellular organisms, where it is found empirically that the number of cell-types also grows approximately as the square-root of the genome-size.

The model also exhibits analogies [3] with infinite range spin glasses [4]. In the framework of an annealed approximation [5], some of the previous numerical observations concerning a phase transition between a *frozen* and a *chaotic* phase in the model could be understood. The average limit Hamming distance d_h, the number of bit-wise differences between two random configurations, was used as an order parameter. In the frozen phase one has $d_h = 0$ for infinite systems whereas in the chaotic phase one has $d_h \neq 0$. The parameter driving the transition is the probability that two different inputs to a Boolean variable σ_i give raise to different values. In [6] the annealed approximation was extended to provide distributions for the number and the length of the limit cycles. Also, good agreement between the results from the annealed approximation and the numerical calculations was demonstrated in the chaotic phase. An alternative order parameter s, the fraction of variables that are stable, i.e. evolve to the same fixed state independently of the initial state, was introduced in [7]. These stable variables are said to constitute the stable core of the network. In the infinite size limit one has $s = 1$ in the frozen phase, whereas $s \neq 1$ in the chaotic phase. In [8] the concept of relevant variables was introduced. A variable σ_i is *not* relevant, if it is stable and/or no variable's state depends on σ_i. The relevant variables are of interest since they contain all information about the asymptotic dynamics of the network, i.e. the number of limit cycles and their cycle lengths.

In this work we focus on the stability of the Kauffman model and how this property is related to the stable core of the network. The probability that inversion of a single

variable will make the system end up in a different limit cycle is known to be small and approaches zero for large networks. However, we find that if the network is reduced to its relevant variables, this probability is drastically raised and increases slightly with the system size.

To facilitate this study we introduce a decimation method that removes variables that *cannot* be relevant by inspection of transition functions and network connectivity. The resulting reduced network contains *all* relevant variables and possibly some irrelevant ones. Since all relevant variables are included it will have exactly the same asymptotic dynamics as the original network even though the total number of variables is drastically reduced. We find that resulting number of variables is close to the true number of relevant variables. This indicates that properties of the stable core can mostly be understood by the comparatively trivial interactions detected in the decimation procedure.

The decimation procedure can also be used to reduce the bias in the estimate of some observables like for example the number n_c of limit cycles. Different from earlier works we do not observe a $\sqrt{N}$ scaling, but rather a linear growth of n_c with the system size.

2 The Kauffman Model

A random Boolean network is essentially a cellular automaton with N binary state variables σ_i. These evolve synchronously according to the transition functions $f_i(\{\sigma\})$, which are chosen randomly at time $t = 0$ and are then kept fixed. In the Kauffman model f_i are constrained to depend on at most K different randomly chosen input variables:

$$\sigma_i(t) = f_i[\sigma_{v_i^1}(t-1), \ldots, \sigma_{v_i^K}(t-1)], \tag{1}$$

for every variable σ_i. The integers $\{v_i^1, \ldots, v_i^K\}$ define the input connections to variable σ_i.

The transition function f_i maps each possible combination of input signals to Boolean output values. These output values are independently set to *true* or *false* with probabilities p and $1 - p$ respectively. This makes some functions independent of some or all of its K input variables. Furthermore, depending on K and p, a finite fraction of the state variables σ_i are not used by any of the transition functions.

The random Boolean network is a deterministic system. Given the state variables at some time, the future trajectory of the σ_i is known. The volume of the state space is finite, therefore all trajectories must posses a limit cycle. Besides the stability of the system the number of limit cycles, the length distribution of the cycles and transient

trajectories are well established observables for this model. In numerical simulations it is in general not possible to probe the models whole state space, except for very small systems. The volume of the $\{\sigma\}$ state space grows exponentially and the number of graphs $\{f_i\}$ even super-exponentially. The commonly used strategy for exploring this model therefore contains two approximations.

1. A small fraction of all possible networks is used as a representative ensemble.

2. For each network only a subset of the state space is probed.

Point 2 introduces a systematic bias to the number of limit cycles since not all of them will be found. In the results section we will re-analyze the number of cycles after decimation of irrelevant nodes. This allows *full enumeration* of state space for up to $N = 32$. In this way we get an improved estimate for the scaling of the respective observables with the system size.

3 The Decimation Procedure

It is well known that some variables in a Kauffman network evolve to the same steady state independently of the initial configuration. These *stable* variables are clearly irrelevant for the asymptotic behavior of the network. The same holds for those variables that do not regulate any other variable, i.e. no transition functions is dependent on them. In fact, as pointed out in [8], for a variable to be relevant it has to be unstable and regulate some unstable variables that in turn regulate others and so on.

By definition, in the frozen phase the fraction of stable variables goes to unity as N goes to infinity. Therefore, a large fraction of the variables are likely to be stable even for finite N. Since the irrelevant variables includes all stable variables, a considerable part of a network does not affect the asymptotic dynamics at all. The process of identifying the irrelevant variables can be divided into two separate steps. Firstly, the stable variables are identified. Secondly, the variables that do not regulate any unstable variables are identified.

Identifying stable variables is in principal easy, but computationally demanding. In [8] this was done by performing simulations of the dynamics of the system and monitoring which variables were in the same state in all probed limit cycles. However, finding all limit cycles essentially means that all 2^N possible states have to be probed, which is possible only for very small networks. Since a variable that is stable within the probed limit cycles may change state within some of the unprobed limit cycles, searching a fraction of state space will in some cases overestimate number of stable variables.

Here we introduce an alternative method, which by pure inspection of the connectivity and the transition functions of a network identifies variables that *must* be stable. The basis for our approach is that transition functions dependent on no input variables give a constant output, i.e. the corresponding variable is stable.

As stated above, some transition functions are independent of all their input variables, i.e. they are constants. This means that the corresponding variables will be stable (after the initial time step) and a transition function that is dependent on such variable will receive a constant signal. By replacing the *stable* input variable with the corresponding *constant* value, the number of input variables is reduced. For each replaced input variable to a transition function, that functions input state space is reduced by a factor 1/2 and within this subspace it may be independent of yet other input variables. If in the end even this rule become a constant, the corresponding variable is stable (after a transient time), and can be replaced by a constant. Therefore, we have to repeat this procedure until no more stable variables are found. We summerize the method as follows:

1. For every transition function, f_i, remove all inputs it does not depend upon.

2. For those f_i with no inputs, clamp the variable σ_i to the corresponding constant value.

3. For every f_i, replace clamped inputs with the corresponding constants.

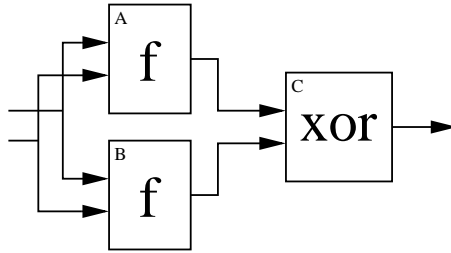4. If any variable has been clamped, repeat from step 1.

Figure 1: A and B have arbitrary but identical transition functions and C implements *xor*. Since A and B also have the same inputs their outputs will be identical. Thus, C will always output $false$, i.e. C is stable and can be removed. A and B are now non-regulating and can be removed too.

It is clear that our method sometimes does not find all stable variables. We see an example of such a situation in Fig. 1. Here the inputs to a function are coupled logically and hereby confined to a subspace of possibilities. Within this subspace the otherwise unstable variable is stable. The figure illustrates just one of the possible couplings between inputs.

Once the stable variables are identified and removed from the network the non-regulating variables can be removed iteratively. Since our method keeps all relevant variables the resulting network will have exactly the same asymptotic dynamics as the original network.

4   Results

Let us start by analyzing the size of the stable core as a function of the system size N. In Fig. 2 the the size of the stable core N^*, identified by the decimation procedure described above, is shown. Each data point is averaged over 10^4 instances of networks. For comparison the size of the stable core N^+ as estimated by the method used in [8] is also plotted. The latter procedure is based on observations of the dynamics of the full network and identification of nodes acquiring the same constant value independently of the start-configuration. Since only a small part of the state space can be probed in practice, the number N^+ is biased to overestimate the true size η of the stable core. On the other hand, our decimation procedure underestimates η because some configurations, like the one depicted in Fig. 1, which may lead to stable variables are not identified. Therefore, we have $N^+ \leq \eta \leq N^*$.
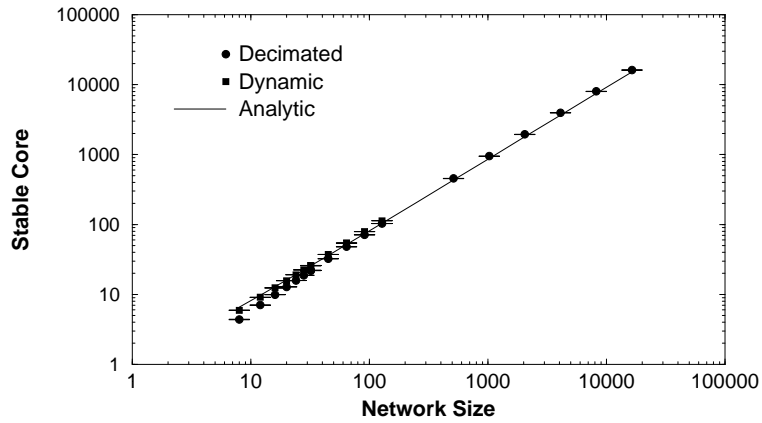

Figure 2: The size of the stable core – the number of variables going to the same constant value independently of the start-configuration – as a function of the network size for $K = 2$. The number of relevant variables estimated with our decimation procedure (circles), the observation of the dynamics [8] (squares) and Eq. (2) and (3) (full line) are in very good agreement.

It is somewhat surprising to observe $N^+ \approx N^*$, which indicates that properties of the stable core, at least for $K = 2$, mostly can be understood by the comparatively trivial interactions detected in the decimation procedure. The probability s for a node to belong

to the stable core can be estimated by using Eq. (2) in [7]

$$s(t+1) = \sum_{k=0}^{K} s(t)^{K-k}(1 - s(t))^k \binom{K}{k} p_k, \tag{2}$$

where p_k is the probability that a transition function for given values of $K - k$ of its input variables is independent of its other k inputs. This equation describes the growth of the stable core with the time. At $t = 0$ only nodes which happen to have a constant transition function are stable. At later times non-constant transition functions, which receive inputs from stable nodes, can acquire a constant value. In [7] Eq. (2) was used as a self-consistency equation for infinite systems, i.e. letting $t \to \infty$. In a finite system, the iteration has to stop at some time T, which reflects a characteristic length in the network, the maximal distance a signal can flow before it reaches all nodes. The length scale is set by the average distance (in number of links) a signal can travel. The signal pathway in a sparse directed random graph with only a few loops is approximately a branched polymer, where it is known (see e.g. [9]) that the average distance grows algebraically, i.e. $T \sim cN^\gamma$. We have fitted the constants c and γ numerically to our data for $K = 2$ and find $\gamma = 0.32(3)$.

After removing the $s(T)$ stable variables, the decimation procedure eliminates the leaves of the network, i.e. those nodes with out-degree $q(t = 0) = 0$. This changes the out-degree of the remaining variables. Therefore, this procedure is repeated until no more variables with $q(t) = 0$ are found. The fraction P_l of leafs, direct and indirect, can be estimated by the self consistent equation

$$P_l = \sum_{q=1}^{\infty} P(q|\tilde{N}, \tilde{K}) P_l^q, \tag{3}$$

where $\tilde{N} = N(1-s)$ is the number of variables after removing the η stable ones, $\tilde{K}$ is the average in-degree, an $P(q|\tilde{N}, \tilde{K})$ the distribution of the out-degree q given in Appendix B. We solved Eq. (2) and (3) numerically, the resulting graph is also shown in Fig. 2, which is in very good agreement with our numerical results.

One of the important features of Kauffman's model is the intrinsic stability of critical Boolean networks. How does decimation affect this behavior? While the network decimation does not change the number and the length of limit cycles, the size of the basins of attraction has to be reduced because the state space is shrunken by orders of magnitude. To get a quantitative picture, we analyze the network stability with respect to the inversion of one randomly chosen variable, after the state trajectory has reached a limit cycle. The *error-probability*, i.e. the probability to end up in a *different* limit cycle compared to the undisturbed system, is shown in Fig. 3. If a limit cycle has not been found within 10^5 steps the network is discarded. For the full network we observe the well known stability, the probability to end up in a a different limit cycle asymptotically approaches
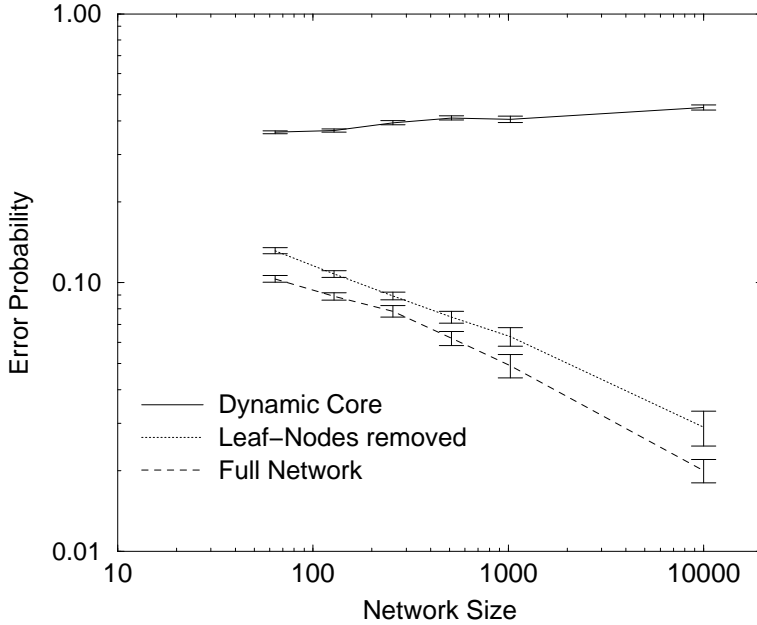
6

Figure 3: The probability to be pushed out from a limit cycle by the inversion of a randomly chosen variable. For the (full) Kauffman-network (dashed line) the error probability scales to zero for large lattices. This behavior is not changed if one does only a geometric reduction of leafs (dotted line). If the network is reduced to its relevant variables (full line) the tolerance against small fluctuations in the state space is completely lost.

zero for large lattices. By contrast, for the reduced network the error-probability grows slowly with the network size and the stability is essentially lost. This means, the tolerance against perturbations observed for Kauffman networks is mostly generated by the stable core: in most cases the perturbed signal is "lost" in the stable core and the full network remains unaffected. It has recently been argued [10] that the in-homogeneous, for example scale-free, *geometry* of real world networks is underlying the stability of these systems. Here we find the opposite: stability is primarily generated dynamically by the propagation (Eq. 2) of the stable core in the network *logic*. The homogeneous geometry plays only a secondary role: if just a geometric reduction of the network is performed, i.e. the leafs are removed (see the discussion of of Eq. (3)), the error-sensitivity is almost unchanged compared to the full network.

The decimation of constant variables from the network enables us to probe a much larger fraction of the state space for a given network. Therefore one may expect to get an better estimate for the number of limit cycles n_c, which with the commonly used method tends to be underestimated, because some limit cycles may have been missed due to the huge state space. By decimating the networks we can fully enumerate the state space for $N \leq 32$ and hereby get an unbiased estimate. For larger systems we use the standard
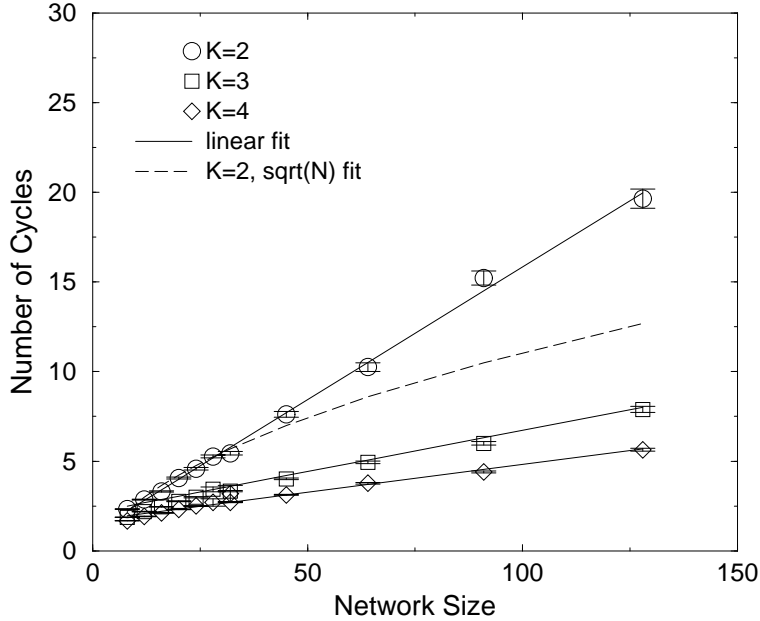
Figure 4: The number of limit cycles as a function of the network size for critical Boolean networks with $K = 2, 3, 4$ inputs. The solid lines connecting the data points are *linear* interpolations with $\chi^2/\text{D.O.F} = 1.5$ ($K = 2$), 1.6 ($K = 3$), 1.4 ($K = 4$). A $\sqrt{N}$ behavior fits the data much worse with $\chi^2/\text{D.O.F} = 11.5$ ($K = 2$), 8.6 ($K = 3$), 8.5 ($K = 4$). The dotted line represents the the best $\sqrt{N}$ fit for $K = 2$.

method with 1000 restarts on each of the reduced networks. Not unexpectedly we observe a small discontinuity in the curve at the point were the simulation scheme is changed. In Fig. 4 we plot n_c as a function of the network size N. We do not find the quite often cited $\sqrt{N}$ behavior for this observable. Rather, we find a linear growth with N. A possible explanation for the different results obtained in some earlier works may be the bias introduced by the standard method in combination with lower computational power.

5 Summary

The source of the remarkable error tolerance of critical Kauffman model is identified as the "dynamics of the stable core". While this seems to be a contradiction in terms, it quite nicely describes the percolation-like process, which underlies the propagation of the "stability" signal. Starting from the relatively few nodes with transition functions which do not at all depend on their inputs, the islands of frozen states grow in time by the interaction with the already stable nodes. This process is only limited by the finite

size of the system. A small fluctuation in the state of the system will most probably not propagate through the stable core and therefore in most cases has no effect. We demonstrate this by studying reduced networks , where most of the stable, irrelevant variables have been removed. The stability against small fluctuations for these networks is reduced by orders of magnitude and will probably go to zero for infinite networks. It is interesting to observe that these effects are mostly driven by the network *logic* and not by the network geometry.

For the identification of the relevant variables we have developed a decimation procedure, which is based on inspection of the networks connectivity and logic. The relatively simple procedure works surprisingly well. The results for the size of the stable core are in very good agreement with the values obtained by observing the dynamics of state-space trajectories in the full network [8].

As a by-product we use the reduced networks to get an improved estimate for the number of limit cycles as a function of the network size. We find that the number of limit cycles grows linearly with N, which is in sharp contrast to the square-root behavior reported by other groups. Even though this $\sqrt{N}$ behavior was an interesting analogy with multi-cellular organisms (with approximately $\sqrt{N}$ different cell types for genomes with genome size N), our result does in no way reduce the importance of Kauffman networks as an example of self-organized order.

A   In- and Out-degree distribution

The reduced number $\tilde{K}$ of inputs after the decimation described in Eq. (2) is the expectation value of the in-degree for the number of inputs from a non-stable variable. For two inputs in the original network we get:

$$\tilde{K} = \frac{1(1-s(T)) + 2(1-s(T))^2}{1-s(T)} = 1 + 2(1-s(T)). \qquad (A.4)$$

The out-degree distribution for a node in the random network can be understood by enumerating the number of ways the NK links can be distributed over this node and the $N-1$ remaining nodes, weighted by the corresponding probabilities to choose the

nodes:

$$P(q|N, K) = \left(\frac{1}{N}\right)^q \left(\frac{N-1}{N}\right)^{NK-q} \binom{NK}{q}. \tag{A.5}$$

References

[1] S. A. Kauffman, J. Theor. Biol. **22**, 437 (1969).

[2] S. A. Kauffman, *The Origins of Order* (Oxford University Press, 1993).

[3] B. Derrida and H. Flyvbjerg, J. Phys. A: Math. Gen. **19**, L1003 (1986).

[4] M. Mezard, G. Parisi and M. A. Virasoro, *Spin Glass Theory and Beyond* (World Scientific, Singapore, 1987).

[5] B. Derrida and Y. Pomeau, Europhys. Lett. **1**, 45 (1986).

[6] U. Bastolla and G. Parisi, Physica D **98**, 1 (1996).

[7] H. Flyvbjerg, J. Phys. A: Math. Gen. **21**, L955 (1988).

[8] U. Bastolla and G. Parisi, Physica D **115**, 203 (1998).

[9] J. Ambjørn, B. Durhuus, T. Jonsson, *Quantum Geometry* (Cambridge University Press, 1997).

[10] H. Jeong, S. P. Mason, Z. N. Oltvai, A.-L. Barabasi, Nature **407**, 651 (2000).