Complex Systems **10** (1996) 157–165

The Complexity of Computations in Recurrent Boolean Networks

Sergey A. Shumsky
P. N. Lebedev Physical Institute,
Leninsky pr.53, Moscow, Russia

Abstract. This paper considers the amount of information processed in boolean automata networks with random interconnections. The *complexity* of computations is defined as the number of logical switchings during the run. The distribution of automata types in the network gives rise to a distribution of computational complexity with various initial conditions. We calculate this complexity distribution and find the limits of complexity in the individual algorithms.

1. Introduction

The theory of computational complexity deals with the quantitative aspects of the numerical solution of different classes of problems. It was initially formulated in regard to sequential algorithms [1, 2]. Later, rapid progress in the technology of integrated circuits spurred the theory of complexity of parallel computations in boolean networks (e.g., [3] and the review lectures by Rabin and Cook in [4]).

Most of the present literature on this subject analyzes feedforward network architectures as in [5–7], where calculation is performed by sequential layers of logical elements. The absence of feedback loops guarantees getting a result in a number of steps not exceeding the number of layers. Thus, the computational complexity of these networks is proportional to the number of boolean elements they contain. The major drawback of a feedforward architecture is that each element may change state only once during a run. On the contrary, each element may switch its state an arbitrary number of times in recurrent networks with feedback loops. It is natural to associate the computational complexity in recurrent networks with the total number of logical switchings in the network in the course of a computation. The purpose of this paper is to examine how this quantity depends on the *component basis* of the network, or the set of boolean automata from which the network is assembled.

This work complements previous studies of the convergence of computations in recurrent boolean networks [8–11]. In particular, the criterion

for convergence of network dynamics defined through the statistical properties of its component basis is presented in [11]. The mean complexity was also found. Here this result is extended, calculating the whole distribution function for the computational complexity for problems with various initial conditions. Our approach is formulated within the scope of Karp's program for studying the statistical properties of algorithms [4].

This paper is organized as follows. Section 2 formulates our model and fixes notation. Section 3 presents a hierarchical classification of network ensembles, which induces a chain of approximal descriptions of dynamical features of recurrent networks. This justifies the stochastic dynamics of logical switchings in networks, which facilitates the use of the theory of branching processes. In Section 4 the well-known results from the latter theory immediately give the desired distribution of computational complexity in a network ensemble. In Section 5 we derive constraints for the distribution of computational complexity in the individual networks from our network ensemble. Section 6 summarizes and concludes the paper.

2. Model description

The problem of computational complexity implies a formal definition of the computational equipment, that is, the computer. The *computer* is a tunable device for calculating the *functions*:

$$\boldsymbol{\Phi} : \Omega \Rightarrow \Omega_0,$$

mapping the set of the problems Ω to the set of the solutions Ω_0. The calculation proceeds according to the *algorithm* $\boldsymbol{\phi}$ constituting a phase portrait in the phase space of the computer consistent with the desired mapping:

$$\boldsymbol{\phi} : \Omega \Rightarrow \Omega, \qquad \lim_{n\to\infty} \boldsymbol{\phi}^n = \boldsymbol{\Phi}.$$

The process of tuning the phase portrait of a computer is called *programming*. The computational properties of any computer are determined by the *ensemble* of algorithms $\{\boldsymbol{\phi}\}$, which can be programmed on it. Constraints imposed by the programming induce a measure $\mu(\boldsymbol{\phi})$ in the set of algorithms, which determines the statistical properties of computations performed by such a computer. This ensemble approach corresponds to the concept of probabilistic algorithms developed in [12].

In the present paper, each algorithm constitutes a network of N boolean (two-state) automata with parallel updating that implement the mapping:

$$x_i \Rightarrow \phi_i(x_i; x_{j_{i1}}, \ldots, x_{j_{iK}}); \qquad 1 \le i, j \le N,$$

or in vector notation: $\mathbf{x} \Rightarrow \boldsymbol{\phi}(\mathbf{x})$, with $\mathbf{x}, \boldsymbol{\phi} \in \{\pm 1\}^N$. The programming represented by this model constructs network configurations containing N boolean automata from the infinite set of basic boolean automata, that is referred to as the *component basis*. Each automaton updates its state according to some boolean function ϕ of its own state and the states of its K binary

inputs at the current time step. Constraints, imposed by the component basis of the computer will be represented by the probability $\mu(\phi)$, associated with *each* possible boolean function. These are the only restrictions posed by the programs in the present paper. This means that all algorithms may be constructed by a probabilistic procedure in which the automata at all network sites are chosen independently:

$$\mu(\boldsymbol{\phi}) = \prod_{i=1}^{N} \mu(\phi_i), \tag{1}$$

and all permutations of interconnections between them are equiprobable:

$$\mu(\phi_i(\mathbf{x})) = \mu(\phi_i(\pi\mathbf{x})), \qquad (\pi x_j = x_{\pi j}, \pi \in \{S_N | \pi i = i\}). \tag{2}$$

3. The hierarchical classification of computers

In accordance with the previous section, each computer is associated with a network ensemble $\{\boldsymbol{\phi}\}$ characterized by a probabilistic measure $\mu(\boldsymbol{\phi})$. This section develops a procedure for step by step extraction of useful information about the statistical properties of computations. This approach makes use of a hierarchical classification of computers similar to that used for classification of cellular automata [13], but utilizing temporal rather than spatial correlators.

Let the class of nth order, defined by the probabilistic function $P(\mathbf{x}_0 \dots \mathbf{x}_n)$, represent all the computers with the same set of probabilities,

$$P(\mathbf{x}_0 \dots \mathbf{x}_n) \equiv \sum_{\{\boldsymbol{\phi}\}} \mu(\boldsymbol{\phi}) \delta_{\mathbf{x}_1, \boldsymbol{\phi}(\mathbf{x}_0)} \cdots \delta_{\mathbf{x}_n, \boldsymbol{\phi}(\mathbf{x}_{n-1})},$$

to find a n-vector sequence $\mathbf{x}_0 \dots \mathbf{x}_n$ in the phase portrait of related network ensembles. Since $\sum_{\mathbf{x}_n} P(\mathbf{x}_0 \dots \mathbf{x}_n) = P(\mathbf{x}_0 \dots \mathbf{x}_{n-1})$ this classification constitutes a hierarchy of more and more comprehensive ensemble descriptions. The last level Ω specifies the whole measure $\mu(\boldsymbol{\phi})$ where $\Omega = 2^N$ is the phase volume of the network.

The above hierarchy induces a chain of approximal descriptions of the statistical properties of phase trajectories in a given ensemble. Namely, each nth-order class may be associated with a n-step Markov process. This generates phase trajectories at random, subject to the conditional probabilities:

$$P(\mathbf{x}_0 \dots \mathbf{x}_{n-1} \to \mathbf{x}_n) = \frac{P(\mathbf{x}_0 \dots \mathbf{x}_n)}{\sum_{\mathbf{x}_n} P(\mathbf{x}_0 \dots \mathbf{x}_n)}, \tag{3}$$

taking into account n previous states of the trajectory. So defined, a n-step Markov process generates trajectories with the same temporal correlations, up to order n, as the original phase trajectories in the ensemble, $\{\boldsymbol{\phi}\}$.

For our needs, one may confine the consideration to a two-step Markov process, since this approximation grasps the existence of the fixed points (representing solutions for the problems), describing the relaxation dynamics

of the network. For simplicity we will limit ourselves to *uniform* ensembles, where the Markov dynamics is uniform over the phase space. This means that the dynamics depend only on the relative position of the points: $P(\mathbf{x}_0\mathbf{x}_1) = P(|\mathbf{x}_0 - \mathbf{x}_1|)$, $P(\mathbf{x}_0\mathbf{x}_1\mathbf{x}_2) = P(|\mathbf{x}_1 - \mathbf{x}_0|, |\mathbf{x}_2 - \mathbf{x}_1|, |\mathbf{x}_2 - \mathbf{x}_0|)$, etc.[1] Here the distance between the two phase points (in the Hamming sense) is the number of automata changing their states when the system steps from one point to the other. Thus, the dynamics of uniform networks is formulated in terms of the number of automata that switch states, and is suited ideally for our needs.

The one- and two-step Markov dynamics provide respectively the spectrum of vector lengths W_m in the phase portraits of our ensemble, and the conditional probability P_{ml} that the vector of length l follows the vector of length m. For a formal derivation of these quantities see [11, 16]. Here we limit ourselves to intuitive arguments and retain the approximate Markov dynamics.

Since we are interested in the number of automata switchings, one should specify the tendency of the automata from the basic set to change their states. The first order approximation utilizes the probability q that an automaton will change its state for arbitrary values of its inputs. Thus, the probability that m automata in an ensemble in an arbitrary initial state will change states at the next step:

$$W_m = \left(\begin{array}{c} N \\ m \end{array} \right) q^m (1-q)^{N-m}, \tag{4}$$

represents the spectrum of vector lengths in the phase portraits of the ensemble. The generating function

$$\widehat{W}(s) = \sum_m W_m s^m = (1 - q + sq)^N$$

gives one the average vector length:

$$\overline{m} = \sum_m W_m m = \widehat{W}'(1) = Nq.$$

To calculate P_{ml}, one needs the probabilities, $\{q_k\}$, $0 < k \leq K$, that an automaton will change its state if k of its inputs have been changed (assuming $q_0 = 0$). For $K \ll N$, the probability that an automaton will change state, when there were m switchings in the network, is:

$$p_m = \sum_{k=1}^{K} \left(\begin{array}{c} K \\ k \end{array} \right) (m/N)^k q_k.$$

Thus, if the parameter of expansion is small enough, $Km/N \ll 1$, the probability of an automaton switching is proportional to the number of switchings in the network at the previous step:

$$p_m \approx q_1 Km/N.$$

[1] In uniform ensembles, both states of the automata play a similar role. The Kauffman NK-model [15] represents an example of an uniform ensemble. (See [16] for more details.)

The probability P_{ml} in this limit is given by the Poisson distribution:

$$P_{ml} \approx \exp(-\kappa m)(\kappa m)^l/l!, \qquad (Km/N \ll 1) \tag{5}$$

with $\kappa = Kq_1$. The corresponding generating function,

$$\hat{P}_m(s) \equiv \sum_l P_{ml} s^l = \exp[m\kappa(s-1)] \equiv [\hat{f}(s)]^m,$$

indicates the statistical independence of automata switchings in the network.

The condition $Km/N \ll 1$ may hold only in *diluted* networks with $K \ll N$. On the other hand, networks large enough to display converging dynamics *must* be diluted. Indeed, relaxation to equilibrium implies that the mean number of switchings at the next step κm must be less than the number of switchings m at the previous step. The convergence criterion $\kappa < 1$ limits the number of automata inputs for *computing* networks: $K < 1/q_1$. Since q_1 does not depend on N, $\lim_{N\to\infty} K/N = 0$.

4. Distribution of computational complexity in network ensemble

Assuming that the *typical* number of switchings $\widetilde{m}$ during a run is small enough, $K\widetilde{m}/N \ll 1$, one may use equation (5) to examine the relaxation dynamics in the network. This approximation facilitates the use of elegant mathematical tools from the theory of branching processes. Indeed, one may treat the relaxation dynamics of equation (5) as a branching process for logical switchings in a network. The probabilities for the number of offspring f_k of an individual switching is given by the Taylor expansion of the generating function $\hat{f}(s) = \sum_k f_k s^k$.

This interpretation immediately gives one the distribution of the total number of automata switchings during a run for one initial switching. According to the general theory [17], the corresponding generating function $\hat{F}(s)$ obeys the functional equation:

$$\hat{F}(s) = s\hat{f}(\hat{F}(s)).$$

The latter may be resolved with the help of the Lagrange theorem [18], which gives the expansion of any function of $\hat{F}(s)$. In particular, for $\hat{f}(s) = \exp[\kappa(s-1)]$ and m initial switchings, one obtains:

$$\begin{aligned} \hat{F}^m(s) &= \sum_{I\geq m} s^I F_m(I), \\ F_m(I) &= \frac{m}{I}\frac{(\kappa I)^{I-m}}{(I-m)!}\exp(-\kappa I) \\ &\approx \frac{m\exp[(1-\kappa+\ln\kappa)I]}{\kappa^m\sqrt{2\pi I^3}}, \qquad (I \gg m). \end{aligned} \tag{6}$$

Averaging equation (6) over the initial conditions results in the following distribution of *computational complexity* I of problems solvable by a given computer:

$$F(I) = \sum_{m=0}^{I} W_m F_m(I) \approx \frac{\widehat{W}'(\kappa^{-1})}{\kappa\sqrt{2\pi I^3}} \exp[(1-\kappa+\ln\kappa)I], \qquad (I \gg N). \tag{7}$$

One may easily check that the average computational complexity coincides with that found in [11]:

$$\overline{I} = \sum_{I} IF(I) = \sum_{m} W_m m\hat{F}'(1) = \frac{\overline{m}}{1-\hat{f}'(1)} = \frac{\overline{m}}{\gamma}, \tag{8}$$

with $\gamma \equiv 1-\kappa$ being the relaxational decrement of the network.

Equation (7) allows one to estimate the maximal complexity in a typical algorithm for a given computer. To this end, consider the probability that the complexity of all Ω problems in a given algorithm $\boldsymbol{\phi}$ does not exceed I_0: $\left(\sum_{I<I_0} F(I)\right)^{\Omega} \approx \exp\left(-\Omega \sum_{I>I_0} F(I)\right)$. This quantity rapidly increases from zero to unity in the vicinity of $I_{\max}$, defined by: $\sum_{I>I_{\max}} F(I) = \Omega^{-1}$. From equation (7) one finds:

$$I_{\max} \approx \frac{N\ln 2}{1-\kappa+\ln\kappa} \approx \frac{N2\ln 2}{\gamma^2}, \qquad (\gamma \ll 1). \tag{9}$$

One may use the theory of branching processes to examine the distribution of relaxation times as well [19]. The result for the mean convergence time reads:

$$\overline{T} \sim \gamma^{-1}\ln(\gamma\overline{m}), \qquad (\gamma\overline{m} \gg 1). \tag{10}$$

This allows one to estimate the limits imposed by our main approximation, equation (5). Indeed, from equations (8) and (10) it follows that a typical number of switchings in a single step during the relaxation process is

$$\widetilde{m} = \overline{I}/\overline{T} \sim \overline{m}/\ln(\gamma\overline{m}), \qquad (\gamma\overline{m} \gg 1),$$

thus confirming our initial assumption $K\widetilde{m}/N \ll 1$ for large enough N:

$$\ln(\gamma q N) \gg qK. \tag{11}$$

5. Distribution of computational complexity within algorithms

In section 4 we examined the distribution of computational complexity over the whole ensemble of algorithms implemented by a given computer. In other words, the computational complexity distribution in a *typical* algorithm. However, due to the exponentially large variety of algorithms in a network ensemble, it may contain a number of *nontypical* (characterized by exponentially small probability) algorithms as well. The question of interest in this section is what distribution functions can be found within single algorithms from a given network ensemble?

The correct treatment of this problem requires a measure of the variety of algorithms in a given network ensemble. This is the notion of *entropy*. Let the entropy H_n be the logarithmic measure of variety in a given n-order class, defined in section 3. This means that computers from this class may implement $\sim \exp(H_n)$ algorithms. H_n may be expressed through the mean entropy of the phase vectors h_n in the network ensembles from this class, $H_n = \Omega h_n$, where:

$$h_n = -\sum_{\mathbf{x}_0 \ldots \mathbf{x}_n} P(\mathbf{x}_0 \ldots \mathbf{x}_n) \ln \left(\frac{P(\mathbf{x}_0 \ldots \mathbf{x}_n)}{P(\mathbf{x}_1 \ldots \mathbf{x}_n)} \right). \tag{12}$$

According to information theory, this quantity measures the constraints imposed on the phase vector by the statistical properties of trajectories. Recall that there is a single trajectory starting from each phase point in the phase portrait, while there is a whole "tree" of trajectories which precede it. Thus, the entropy of a phase vector (equation (12)) in a *portrait* differs from that in a *trajectory*:

$$\tilde{h}_n = -\sum_{\mathbf{x}_0 \ldots \mathbf{x}_n} P(\mathbf{x}_0 \ldots \mathbf{x}_n) \ln \left(\frac{P(\mathbf{x}_0 \ldots \mathbf{x}_n)}{P(\mathbf{x}_0 \ldots \mathbf{x}_{n-1})} \right),$$

defined in [20].

Let $\mathcal{F}(I)$ be the distribution of computational complexity in a particular algorithm, while $F(I)$ denotes that in a network ensemble. In such an algorithm, $\Omega\mathcal{F}(I)$ problems have complexity I. Thus, the probability of such an algorithm is:

$$P\{\mathcal{F}(I)\} = \prod_I F(I)^{\Omega\mathcal{F}(I)} = \exp[\Omega \sum_I \mathcal{F}(I) \ln F(I)].$$

Approximations similar to that used to obtain $I_{\max}$ in equation (9) lead one to the following limit on the possible distributions of computational complexity in network ensembles that belong to a certain n-order class with the entropy H_n:

$$-\sum_I \mathcal{F}(I) \ln F(I) < h_n. \tag{13}$$

Substituting the asymptotics from equation (7) into equation (13) one can see that in a network ensemble there exist algorithms with a *power* law decrease of the complexity spectrum:

$$\mathcal{F}(I) \propto I^{-\alpha}, \qquad (\alpha > 2),$$

while a typical algorithm such as equation (7) has an *exponentially* decreasing spectrum. In general, the asymptotics in equation (7) lead to the following limitation for the mean complexity of the algorithms:

$$\sum_I I\mathcal{F}(I) < 2h_n/\gamma^2, \qquad (\gamma \ll 1).$$

Using equation (8) and estimating $h_n \approx h_1 \sim \overline{m}$ according to [11] the latter result may be rewritten as:

$$\sum_I I\mathcal{F}(I) \lesssim \overline{I}/\gamma. \tag{14}$$

Thus in a network ensemble with mean computational complexity $\overline{I}$, there exist algorithms with mean computational complexity $\sim \gamma^{-1}$ times greater than $\overline{I}$.

6. Conclusions

The aim of this paper is to analyze the number of logical switchings during the computations in recurrent boolean networks with random interconnections. The natural measure of complexity of such computations generalizes the usual notion of complexity as the number of boolean elements in feedforward networks.

We have proposed a hierarchy of approximations, providing a more and more accurate treatment of the statistical properties of parallel computations. The present description of the relaxation dynamics corresponds to a second order approximation and is limited to a special class of uniform ensembles where both automata states are equiprobable.

The distribution of the computational complexity (equation (7)), as well as the limits on the mean complexity of individual algorithms (equation (14)) were found in the limit of extreme dilution of network interconnections ($K = o(\log N)$).

The present approach sheds some light on the information content of the dynamics of complex systems. It also facilitates the use of the methods of statistical dynamics for problems in computational complexity. We believe this is a novel and productive approach to computational complexity.

References

[1] J. Hartmanis and R. E. Stearns, "On the Computational Complexity of Algorithms," *Transactions of American Mathematical Society*, May (1965) 285–306.

[2] S. A. Cook and S. O. Aanderaa, "On the Minimum Computation Time of Functions," *Transactions of American Mathematical Society*, **142** (1969) 291–314.

[3] J. E. Savage, *The Complexity of Computing* (Wiley, New York, 1976).

[4] *ACM Turing Award Lectures. The First Twenty Years* (Addison Wesley, Reading, MA, 1987).

[5] A. Borodin, "On Relating Time and Space to Size and Depth," *SIAM Journal of Computing*, **6** (1977) 733–744.

[6] S. A. Cook, "Toward a Complexity Theory of Synchronous Parallel Computation," *L'Enseignement Mathematique*, (1981) 99–124.

[7] *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, edited by D. E. Rumelhart and J. L. McClelland, volumes 1 and 2 (MIT Press, 1986).

[8] K. E. Kűrten, "Critical Phenomena in Model Neural Networks," *Physics Letters A*, **129** (1988) 157–160.

[9] B. Derrida and Y. Pomeau, "Random Networks of Automata: A Simple Annealed Approximation," *Europhysics Letters*, **1** (1986) 45–49.

[10] S. A. Kauffman, "Requirements for Evolvability in Complex Systems: Orderly Dynamics and Frozen Components," *Physica D*, **42** (1990) 135–152.

[11] S. A. Shumsky, "Computational Properties of Boolean networks," *Complex Systems*, **8** (1994) 337–346.

[12] M. O. Rabin, "Probabilistic Algorithms," in *Algorithms and Complexity, New Directions and Recent Trends*, edited by J. F. Traub (Academic Press, New York, 1976).

[13] H. A. Gutowitz, "A Hierarchical Classification of Cellular Automata," *Physica D*, **45** (1990) 135–156.

[14] M. Kac, *Probability and Related topics in Physical Sciences* (Interscience, London–New York, 1959).

[15] S. A. Kauffman, "Metabolic Stability and Epigenesis in Randomly Connected Nets," *Journal of Theoretical Biology*, **22** (1969) 437–467.

[16] S. A. Shumsky, "Statistical Mechanics of Distributed Computations," unpublished.

[17] T. E. Harris, *The Theory of Branching Processes* (Springer-Verlag, Berlin, 1963).

[18] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration* (J. Wiley and Sons, Inc., New York, 1983).

[19] S. A. Shumsky, "Phase Portrait Characteristics of Random Logical Networks," *Journal of Moscow Physical Society*, **2** (1992) 263–281.

[20] A. Crisanti, M Falcioni, and A. Vulpiani, "Transition from Regular to Complex Behaviour in a Discrete Deterministic Asymmetric Neural Network Model," *Journal of Physics A: Mathematical, Nuclear, and General*, **26** (1993) 3441–3453.