Complexity and stable evolution of circuits

S. Vakulenko[1] and D. Grigoriev[2]
[1]Institute of Mechanical Engineering Problems
St. Petersburg, Russia
[2]CNRS, Mathématiques, Université de Lille
59655 Villeneuve d'Ascq, France

Abstract

We consider the viability problem for random dynamical systems, in particular, for circuits. A system is viable only if the system state stays in a prescribed domain Π of a phase space. We assume that the circuit structure is coded by a code evolving in time. We introduce the notion of stable evolution of the code and the system: evolution is stable if there is a $\delta > 0$ such that the probability P_T to be in Π within time interval $[0, T]$ satisfies $P_T > \delta$ as $T \to \infty$.

We show that for certain large classes of systems, the stable evolution has the following fundamental property: the Kolmogorov complexity of the code cannot be bounded by a constant as time $t \to \infty$. For circuit models, we describe examples of stable evolution of complicated boolean networks for a difficult case when the domain Π is unknown.

We dedicate this paper to Professor Grisha Mints.
We admire the breadth of his interests.

1 Introduction

One of the main characteristics of biological systems is that these systems support their own life functions. In particular, a biological system tries to keep the values of the main characteristics of each cell – such as temperature, pressure, pH (acidity measure), concentrations of different reagents – within a certain domain of values that makes the biological processes possible. These domains of values are called *viability domains*, and the process of supporting the life functions – by keeping the values inside viability domains – is called *homeostasis*. The concept of homeostatis was first developed by the French physiologist Claude Bernard; it is now one of the main concepts of biology; see, e.g., [5].

The homeostasis process is notoriously difficult to describe in precise mathematical terms. At first glance, homeostasis is similar to the well-known and

well-studied notion of stability: in both cases, once a system deviates from the desirable domain, it is pushed back. However, a more detailed analysis shows that these notions are actually different:

- the usual mathematical descriptions of stability mean that a system will indefinitely remain in the desired state, for time $t \to \infty$, while
- a biological cell (and the whole living being) eventually dies.

This difference has been emphasized in a recent paper by M. Gromov and A. Carbone: "Homeostasis of an individual cell cannot be stable for a long time as it would be destroyed by random fluctuations within and off cell' ([13], p. 40).

One might argue that while individuals die, their children survive and thus, species remain. However, it turns out that the biological species are unstable too. This conclusion was confirmed, e.g., by L. Van Valen based on his analysis of empirical data; see, e.g., [23, 30]. Moreover, he concluded that the species extinction rate is approximately constant for all the species, so this species change is not just a problem for unfit species.

The species extinction does not necessarily mean complete extinction, it usually means that a species evolves and a new mutated better-fit species replaces the original one. From this viewpoint, the evolution is "stable" – in the sense that it keep life on Earth effectively functioning. However, as M. Gromov and A. Carbone mention, it is very difficult to describe this "stability" in precise terms: "There is no adequate mathematical formalism to express the intuitively clear idea of replicative stability of dynamical systems" ([13], p. 40). The problem of describing this idea in precise terms is called the *viability problem*.

Specifically, we need to formalize two ideas:

- First, that biological systems are unstable (in particular, under random perturbations).
- Second, that these systems can be stabilized by replication (evolution).

In this paper, we show that an important progress in solving both aspects of the viability problem can be achieved if we use the notion of Kolmogorov complexity. In our formalizations, we will use the basic concepts and ideas proposed by M. Gromov and A. Carbone [13], L. Van Valen [30], and L. Valiant [31, 32].

2 Systems under consideration

Let us describe the models that we will use to describe biological systems. Let n denote the number of quantities that characterize the current state of a given system. This means that the state of the system can be described by a tuple $u = (u_1, \ldots, u_n)$ consisting of the values of all these characteristics. The set H of all possible states of a system is therefore equal to $H = \mathbb{R}^n$.

The state of a biological system evolves with time. In practice, even when we have measuring instruments that "continuously" monitor the state of a biological system, we can only perform a finite number measurements in each time interval. So, in effect, we only know the values of the corresponding characteristics at certain moments of time. Thus, to get a good description of the observed data, it makes sense to consider discrete-time models.

Usually, there is a certain frequency with which we perform measurements, so we get values measured at moments T_0, $T_1 = T_0 + \Delta T$, $T_2 = T_0 + 2\Delta T$, ..., $T_t = T_0 + t \cdot \Delta T$, ... It is therefore convenient to call the integer index t of the moment T_t the t-th moment of time, and talk about the state $u(t)$, $t = 0, 1, \ldots$, at the t-th moment of time.

The state $u(t)$ of a system at a given moment of time affects the state of the system $u(t+1)$ at the next moment of time. The next state of the system is determined not only by its previous state: biological systems operate in an environment in which unpredictable ("random") fluctuations occur all the time. Let m be the number of parameters that describe such fluctuations; then, the current state of these fluctuations can be described by an m-dimensional vector $\xi(t) = (\xi_1(t), \ldots, \xi_m(t))$.

Once we know the current state of the system $u(t)$ and the current state $\xi(t)$ of all the external parameters that affect this system, we should be able to determine the next state $u(t+1)$. In other words, we consider the following dynamics:

$$u_i(t+1) = f_i(u(t), \xi(t)), \quad t = 0, 1, \ldots \tag{1}$$

with initial conditions $u_i(0) = \varphi_i$.

To specify evolution, we must therefore describe the transition functions f_i and the random process $\xi(t)$. We will do this in the following two subsections.

2.1 Transition functions

The transition functions f_i describe physical processes and thus ultimately come from physics. Most equations of fundamental physics – equations of quantum mechanics, electrodynamics, gravity, etc. – are partial differential equations with polynomial right-hand sides. Other physical phenomena are described by partial differential equations that use fundamental fields – i.e., solutions to the fundamental physics equations – as solutions. The resulting dependencies can be again used in the right-hand sides of other physics equations, etc. The resulting functions are known as *Pfaffian* functions; these functions are formally defined as follows (see [15]):

Definition 1.

- *By a* Pfaffian chain*, we mean a sequence of real analytic functions*

$$f_1(x), f_2(x), \ldots, f_r(x)$$

defined on $\mathbb{R}^n$ *which, for every* $j = 1, \ldots, r$*, satisfy a system of partial differential equations*

$$\frac{\partial f_j}{\partial x_k} = g_{kj}(x, f_1(x), \ldots, f_j(x)), \quad j = 1, \ldots, n,$$

with polynomials g_{kj}*.*

- *For each Pfaffian chain, the integer* r *is called its* length*, and the largest of the degrees of polynomials* g_{kj} *is called its* degree*.*
- *A function* $f(x)$ *is called* Pfaffian *if it appears in a Pfaffian chain.*

It is known that Pffafian functions satisfy many important properties; in particular:

- the sum and the product of two Pfaffian functions f_1 and f_2 of lengths r_i and degrees d_i are again Pffafian functions, of length $r_1 + r_2$ and degree $d_1 + d_2$;
- superpositions of Pfaffian functions are also Pfaffian.

Results from the theory of Pfaffian functions and the powerful computational tools that are based on these results are described in [10, 12, 15].

So, in this paper, we consider dynamical systems (1) with Pfaffian functions f_i. The class of such systems will be denoted by **Kh**:

Class Kh. *This class consists of the systems (1) for which* f_i *are Pfaffian functions.*

We will also consider several subclasses of this class, subclasses which are known to be useful in applications. Two of these subclasses are related to the fact that when the fluctuations are small and/or the deviation of the state from a nominal state is small, we can expand the dependence f_i into Taylor series and keep only the first few terms (or even only the first term) in this expansion – because higher-order terms can be safely ignored. In this case, we end up with a polynomial (or even linear) dependence.

Usually, the deviation of the state of a biological system from its nominal state can be reasonably large, so terms which are quadratic in this dependence cannot be ignored; however, random fluctuations can be small. When the random fluctuations are so small that we can only keep terms which are linear in ξ, we get the following class which is well studied in control theory:

Class Kl. *This class consists of the systems (1) in which the transition functions* f_i *have the form* $f_i(u, \xi) = g_{0i}(u) + \sum_{k=1}^{m} \xi_k g_{ki}(u)$*, with polynomial* g_{ki}*.*

When the fluctuations are larger and their squares can no longer be ignored, we get a more general class of systems:

Class Kp. *This class consists of the systems (1) in which the transition functions f_i are polynomial in u and ξ.*

Comment. Here, **l** in **Kl** stands for *linear* (meaning linear dependence on ξ), while **p** in **Kp** stands for *polynomial*.

Another important class comes from the situation when a random fluctuation simply means selecting one of the finitely many options.

Class Kr. *Let us assume that we have a finite family of maps $u \to \tilde{f}^{(k)}(u) = (\tilde{f}_1^{(k)}(u), ..., \tilde{f}_n^{(k)}(u))$, $u \in \mathbf{R}^n$, where $k = 1, \ldots, m'$. Assume that $f_i = \tilde{f}_i^{(k(t))}(u) + \lambda \cdot g_i(u, \xi)$, where $\lambda > 0$ is a parameter, $\tilde{f}_i^{(k)}$ and g_i are Pfaffian, and $k(t)$ is a random index: at each moment t we make a random choice of k with probabilities $p_k \geq 0$, $p_1 + p_2 + ... + p_{m'} = 1$ (these choices at different moments of time are done independently).*

In the particular case when all the maps $u \to \tilde{f}^{(k)}(u)$ are contractions and $\lambda = 0$, we obtain so-called *iterated function systems*; see, e.g., [14].

It is important to mention that the class **Kh** contains many neural and genetic *circuit models*. Genetic circuits were proposed to take into account theoretical ideas and experimental information on gene interaction; see, e.g., [11, 18, 21, 26]; see [25] for a review. In this paper, we consider the following model

$$u_i(t' + \tau) = \sigma\left(\sum_{j=1}^{N} K_{ij}(t')u_j(t') + h_i - \xi_i(t')\right), \quad u_i(0) = x_i, \tag{2}$$

where $t' = 0, \tau, 2\tau, \ldots, d \cdot \tau$, $i = 1, 2, \ldots, N$, d and N are positive integers, $\tau > 0$ is a real parameter, and $x = (x_1, \ldots, x_N)$ is an initial condition. It is usually assumed that the function σ is a strictly monotone increasing function for which $\lim_{z \to -\infty} \sigma(z) = 0$ and $\lim_{z \to \infty} \sigma(z) = 1$. Such systems have interesting applications to biology, e.g., to the morphogenesis problem [29].

Circuits (2) can simulate all Turing machines [16]. Also, they can generate all (up to topological equivalency) kinds of structurally stable semiflows with discrete time [28].

We want to restrict ourselves to Pfaffian systems. The functions σ used in practical applications are Pfaffian functions of length 1; moreover, they are solutions of a differential equation $\sigma' = P(\sigma)$, where P is a polynomial for which $P(0) = 0$, $P(1) = 0$, and $P(z) > 0$ for all $z \in (0, 1)$. Thus, in this paper, we will consider only such functions σ. Even with this Pfaffian limitation, we can still get both above-mentioned universality properties: with respect to Turing machines and with respect to topological behavior.

2.2 Assumptions on random processes ξ

In this paper, we assume that the fluctuations $\xi_i(t)$ are:

- *Markov processes*, i.e., that the probabilities of different values of $\xi(t)$ depend only on the previous values $\xi_i(t-1)$, and
- are *strong*, in the sense that there is a positive probability to move into a close vicinity of any state.

Formally, this assumption of strong fluctuations can be described as follows.

For $\delta > 0$ let $V(\theta, \delta)$ denote the δ-neighborhood of a point $\theta \in \mathbf{R}^m$.

Assumption 1. *Assume that $\xi_i(t)$ are Markov processes with discrete time, $t = 0, 1, 2, \ldots$. Assume that for each $\theta, \delta > 0$, all positive integers $t > t_0$, and each starting point θ_0 the probability that the process ξ is in the neighborhood $V(\theta, \delta)$ at the moment t is positive:*

$$\mathrm{Prob}(\xi(t) \in V(\theta, \delta) \,|\, \xi(t_0) = \theta_0) > c(\delta) > 0,$$

where a constant $c(\delta)$ is uniform in t, t_0.

Mathematically, this assumption is one of the versions of ergodicity of the Markov process. This assumption holds for many stochastic processes that are used in modeling biological phenomena.

2.3 Evolution

The system (1) is well suited to describe the dynamics of a single individual. Individuals belonging to different species s may have different dynamics. So, a more accurate way to describe the dynamics is to use the equation $u_i(t+1) = f_i(u(t), \xi(t), s)$, where s describes the species.

In biology, different species and subspecies can be characterized by their DNA, i.e., by a sequence of symbols. Without losing generality, we can always encode the 4-values language of DNA codons into a binary code, so we can assume that s is a finite binary sequence.

In mathematical terms, we consider a discrete (finite or countable) set S with $N(S) \leq +\infty$ elements s. We assume that all elements of the set S are binary strings, i.e., that $S \subseteq S_\infty$, where S_∞ denotes the set of all possible finite binary strings.

The fact that the transition functions f_i depend not only on $u(t)$ but also on s can be described by saying that we extend our original phase space H of all the states u to a larger space $H \times S$.

In addition to dynamics within a species, we also have to take into account the possibility of mutation, when s changes. We assume that these transitions follow a Markov chain with transition probabilities $p_{s's}(u)$ to go from s' to s; in line with the biological applications, we take into account that the probability of different transitions (mutations) may depend on the state u.

To take into consideration that only states from a certain set $\Pi \subseteq \mathbb{R}^n$ (called the *viability domain*) are viable, we use the following standard construction: We introduce, formally, an absorbing state a such that $p_{as}(u) = 0$ for each $s \neq a$. If u leaves the viability domain Π, then the system automatically reaches this absorbing state a.

So, our model is defined by:

1. a family of random dynamical systems $u_i(t+1) = f_i(u_i(t), \xi(t), s)$, $i = 1, 2, ..., n$, corresponding to different binary strings $s \in S$;
2. a set $\Pi \subseteq \mathbf{R}^n$;
3. a Markov chain $\mathbf{M}$ with the state space $S \cup \{a\}$ and the transition matrix $\mathbf{W}(u)$ with entries $p_{s's}(u)$ (the transition probability from s' to s depending on u) such that $p_{as}(u) = 0$ (if $s \neq a$).

About dependence of f on $s \in S$ we assume the following. Consider a class $\mathcal{C}$ of dynamical systems (1) with f depending on parameters $r \in \mathcal{P}$, where $\mathcal{P}$ is a set of possible values of the parameters. We assume that the set $\mathcal{P}$ is equipped with a measure ν.

For example, if the functions f_i are defined by a sequence of polynomials (as is the case when f_i are Pfaffian functions), then $\mathcal{P}$ is the set of all tuples of coefficients of all these polynomials, and as ν, we can select the standard Lebesgue measure on this set.

It is reasonable to assume that parameters r are random functions of s. To describe these random functions, we need to introduce, for every natural number l, a probability measure on the set of all possible mappings α from binary strings of lengths $\leq l$ to the set $\mathcal{P}$. It is reasonable to make the following assumption:

Assumption 2. *For every set $A \subseteq \mathcal{P}$ of ν-measure 0, for every integer l, and for every string s of length $\leq l$, the probability that the parameters $\alpha(s)$ are in A is 0:*

$$\mu_l(B_l(s)) = 0, \text{ where } B_l(s) \stackrel{\text{def}}{=} \{\alpha : \alpha(s) \in A\}.$$

For systems from the class $\mathcal{P}$, denote by $P_\Pi(v, r, t)$ the conditional probability that in the next moment of time the system will still be viable ($u(t+1) \in \Pi$) under the condition that its previous state is $u(t) = v \in \Pi$ and that the previous value of the parameters was $r(t) = r$.

Definition 1. *We say that a class of system $\mathcal{P}$ from the general class* **Kh** *is* generically unviable *in Π, if there exists a function $\kappa(r) > 0$ for which*

$$\sup_{u \in \Pi, t = 0, 1, 2, ...} P_\Pi(u, r, t) = 1 - \kappa(r) \tag{3}$$

for ν-almost all values of the parameter r.

This means that at every step there is a non-zero probability $\geq \kappa(r) > 0$ of moving into an unviable state – and since the unviable state is absorbing, we are guaranteed to eventually move into an unviable state.

For every viable state $u_0 \in \Pi$ and for every integer T, by $P_T(\Pi, u_0)$ we denote the conditional probability that $u(t) \in \Pi$ for all $t = 1, 2, \ldots, T$ under the condition that $u(0) = u_0$.

Definition 2. *We say that the evolution is* stable *if there exists a real number $\delta > 0$ for which $P_T(\Pi, u_0) > \delta$ for all integers $T > 0$ and all states $u_0 \in \Pi$.*

If such a real number δ does not exist, we say that the evolution is *unstable*.

3 Main result

For an arbitrary Turing machine F and for every string s, by $K_F(s)$ we denote the Kolmogorov complexity of the string s relative to F, i.e., the shortest length of the program (= initial configuration) on F for which F generates s [17].

In the following text, by a *Kolmogorov complexity* of a string s, or simply *complexity* (for short), we mean $K_F(s)$ for a fixed Turing machine F.

Theorem 1. *Let F be an arbitrary Turing machine. Consider a class of generically unviable systems. Assume that the Markov chain* **M** *and the system (1) generate strings s with a priori bounded Kolmogorov complexities $K_F(s)$ relative to F. Then, for almost all mappings $\alpha : s \to r(s)$ of strings s to the parameters r, the evolution is unstable and the corresponding system is not viable: $P_T \to 0$ as $T \to \infty$.*

Comment 1. Instead of K_F, one could take any function K' satisfying the following property: for any n there exists finitely many strings s with $K'(s) = n$.

Comment 2. In this analysis, we only consider the Kolmogorov complexity of the *codes* s, and not of the states u themselves. Complexity of the states can also be studied for systems similar to **Kh**; see, e.g., [29].

Comment 3. Theorem 1 says that the evolution is unstable for almost all mappings α, but it does not tell us whether the stable evolution is possible for *some* functions α. The existence of stable evolutions is analyzed in Section 6 for the case of *circuits* – i.e., systems of type (2).

4 Viability and unviability

It is difficult to determine when a given class of systems is generically unviable. Under Assumption 1 on ξ, a natural way for proving unviability is to consider ξ as a control and to use methods from control theory; see, e.g., [19, 24]. This

reduction to control leads to complex attainability and controllability problems. We will describe several results that can be thus obtained.

Theorem 2. *Assume that we have a system (1) from the class* **Kp**, *with polynomials $f_i(u,\xi)$ of positive degree d, and with $m \geq 2$. Assume also that the viability domain Π is bounded, i.e., $\Pi \subseteq B_R$ for a ball B_R of some radius R. Then for ν-almost all polynomials f_i, the system (1) is stochastically not viable, i.e., $P_T \to 0$ as $T \to \infty$.*

Comment. Moreover, for almost all tuples of polynomials f_i, there exists a value $\kappa(f)$ for which

$$P_\Pi(u,f,t) \leq 1 - \kappa(f) \tag{4}$$

for all $u \in \Pi$ and t.

We present a proof of this theorem at the end of this section. Before that, let us consider other types of systems. Systems from the Class **Kr** can be both viable and not viable. Indeed, let Π be a bounded set.

Example 1. Suppose that, for some $R > 0$, for every u, the range of the map $\xi \to g(u,\xi)$ contains the ball $B_R = \{g : |g| < R\}$. Then one can show that for a sufficiently large $\lambda > 0$, the corresponding system is not viable.

Example 2. Let us consider situations when the functions g_i are uniformly bounded and $m' = 1$. Then, by the definition of m', the corresponding dynamical system $u(t+1) = \tilde{f}(u(t))$ is deterministic (not random). In particular, we can consider the case when this system has an attractor consisting of hyperbolic equilibria points, and that this attractor is contained in the viability domain Π. Then one can show [4] that if the initial state $u(0)$ is sufficiently close to the attractor, then, for sufficiently small values $\lambda > 0$, the corresponding system $u(t+1) = \tilde{f}(u(t)) + \lambda g(u(t),\xi(t))$ is viable.

Let us prove Theorem 2. We start with the following preliminary lemma.

Lemma 1. *Let Π be a compact set. Consider a system of polynomial equations*

$$g_i(u) = 0, \quad i = 1,\ldots,N, \tag{5}$$

where g_i are polynomials, and the number of equations N is greater than the number of variables n. Then the probability that this system has a solution $u_* \in \Pi$ is equal to 0.

Proof. This lemma easily follows from the resultant theory; see, e.g., [33].

Proof of Theorem 2. Since the set Π is bounded, there exists a real number $R > 0$ such that if $|u| > R$ then $u \notin \Pi$. For systems of class **Kp**, one has $f(u,\xi) = \sum\limits_{l:|l|<d} h_l(u)\xi^l$, where $l = (l_1,\ldots,l_m)$ is a multi-index, $h_l(u)$ are polynomials of u, $|l| \stackrel{\text{def}}{=} l_1 + \ldots + l_m$, and $\xi^l \stackrel{\text{def}}{=} \xi_1^{l_1} \cdot \ldots \cdot \xi_m^{l_m}$. Consider a finite tuple

$a = (a_1, a_2, \ldots, a_m)$, where a_j are different positive numbers. Set $\xi_j = a_j \cdot z$, and let $z \to +\infty$.

Suppose $|f(u, \xi)| < C$ for all $\xi(z)$, where $C > 0$. Then one can conclude that $h_l(u) = 0$ for all l for which $|l| < d$. The equations $h_l(u) = 0$ form $\geq n \cdot (d + 1)$ polynomial equations with n unknowns u_i.

Now we apply Lemma 1 and conclude that since, in general, such a system has no solutions, in general, the values $|f(u, \xi)|$ are not bounded as $z \to \infty$. Thus, if $u(t) \in \Pi$, for some ξ we have $|u(t+1)| > R$ and consequently $u(t+1) \notin \Pi$. The theorem is proven.

5 Proof of Theorem 1

First, let us show that a stable evolution is possible only when the code length is unbounded in time.

Indeed, suppose that the lengths $\mathrm{len}(s)$ of all the codes s are a priori bounded by an integer l. The number of such codes is bounded, and thus, due to Assumption 2, for almost all maps α,

$$\min_{s:\mathrm{len}(s)\leq l} \kappa(r(s)) > \kappa_0 > 0. \tag{6}$$

Indeed, one can observe that the set of all maps α for which $\kappa(r(s)) = 0$ for some string s of length $\leq l$ is contained in the finite union of the sets $B_l(s)$ of measure 0: $\mu_l(B_l(s)) = 0$. Then, since our process is a Markov one, according to Assumption 1, the probability $P_T(\Pi)$ for $\xi(t)$ to be in Π at time moments $0, 1, \ldots, T$ is smaller than $(1 - \kappa_0)^T$, and we conclude that the evolution is unstable. This proves the theorem for the case when all strings have a priori bounded length.

Let us note now that the lengths $l(s)$ of the strings of the relative Kolmogorov complexity $K_F(s)$ not exceeding K are a priori bounded – since there are only finitely many such strings: $l(s) < N_K$ for some N_K. Therefore, all strings of complexity $< K$ are contained in a finite set $\mathcal{B}_K$ of binary strings. The theorem is proven.

Comment. It is worth mentioning that while for every Turing machine F and for every integer K, there exists an upper bound on the length of all the strings s with $K_F(x) \leq K$, this upper bound is not always effectively computable. For example, for a universal Turing machine F, the impossibility of an algorithm computing such upper bounds follows from the well-known theorem of Rabin [22].

6 Stable evolution of circuit population

In this section, we show that for circuits, stable evolution is possible.

Specifically, we consider an evolution of a family ("population") of circuits (2). To simplify our analysis, we consider the boolean case, when the values $u_i(t)$ are always 0s or 1s. In this case, σ is the step function, i.e., $\sigma(z) = 1$ for $z > 0$ and $\sigma(z) = 0$ for $z \le 0$.

We also assume that for every time t, there is a positive integer $b(t)$ called *connection intensity*. For every i and j, the value $K_{ij}(t)$ is equal either to $b(t)$ or to $-b(t)$ or to 0. In other words, the quantity $u_j(t')$ either "excites" the value $u_i(t)$ at the next moment of time t, or inhibits it, or does not affect this value.

We assume that $h_i = \left(m_i + \dfrac{1}{2}\right) \cdot b$, where m_i are integers, and that the number N, in general, changes with time: $N = N(t)$.

At every moment t, the situation can be described by a directed graph $(V(t), E(t))$ whose vertices correspond to components u_i: $V(t) = \{1, 2, \ldots, N(t)\}$, and where there is an edge $(i \to j) \in E(t)$ if and only if $K_{ij}(t) \neq 0$. Each graph represents a single circuit.

Each step of the evolution of an individual circuit consists of one of the following changes in the graph and in the corresponding values K_{ij}:

1. the graph (V, E) stays the same;
2. one adds a node to V;
3. one adds an edge $i \to j$ to E with a new weight K_{ij};
4. one change a weight K_{ij}; when the new value of K_{ij} is 0, this change deletes an edge $i \to j$.

Steps 2–4 will be called *mutations*. We will assume that mutations occur with a given probability $\mu > 0$.

Based on these individual changes, we can perform the following changes in the population:

- First, at each time step we can simultaneously change many circuits in the population, by performing changes 1–4 on different circuits.
- We can also replicate (make copies of) some circuits and delete ("destroy") some other circuits.

We consider a population consisting of $X(t)$ random circuits (2) of different structure and different depths $d(t)$. We will describe now the set Π. In this description, we will use several ideas from [31]. Suppose that a circuit $Circ_j$, a member of the population, survives at the moment t if and only if it gives a correct output $y(x)$ as an answer to a boolean input x: $y = f(x, t)$, where $f(x, t)$ are given boolean functions depending on t. The output y is the final state of some node: $y(x) = u_1(\tau \cdot d)$, where $u_i(t)$ are computed by the formulas (2) starting with $u(0) = x$. The whole population survives if it contains at least one circuit.

We suppose that f are a priori unknown: to survive, circuits should "learn" correct answers. So, in effect, we are dealing here with the notions from the learning theory [32] – but in a different context.

Suppose that the correct answers are defined by a special piecewise constant sequence of boolean functions

$$f(x,t) = f_j(x), \quad N(t) = N_j \quad t \in ((j-1)\cdot T_e, j\cdot T_e], \tag{7}$$

where T_e is a positive number (the "length" of the j-th evolution stage) and $x = (x_1, x_2, \ldots, x_{N_j})$. Here we also assume that each function f_j belongs to certain class $\mathcal{C}$ of boolean circuits (2) (naturally, the values N, K, and d can depend on j). Assume that the parameter τ is small enough; thus, we should not take into account the time $\tau \cdot d$ of the circuit reactions.

The problem can be interpreted as a problem of adaptive behavior of a large growing population of evolving circuits under the challenge of a "random environment". Let us now formulate our assumptions about this environment.

Suppose that at the j-th evolution stage, the values x are chosen randomly by a probability distribution $P_j(x)$ on the set Ξ of all possible inputs x. We assume that each circuit obtains the values generated by the same distribution P_j and that the values corresponding to different circuits are independent.

We say that the circuit (2) is *correct* if, whenever the noise is turned off ($\xi(t) = 0$), this circuit returns a correct answer for every input x. For each pair of functions f and f', we can define the probability of error

$$Err(f, f') \stackrel{\text{def}}{=} \text{Prob}\{f(x) \neq f'(x)\},$$

where the probability in the right hand side is defined with respect to P_j. We can then define, for every j, the probability

$$Err_j \stackrel{\text{def}}{=} \inf_{f \neq f_j, f \in \mathcal{C}} Err(f_j, f), \tag{8}$$

and $\delta_j \stackrel{\text{def}}{=} Err(f_j, 0)$, where 0 denotes a trivial circuit with output 0.

Here, two drastically different situations are possible:

A *Passive environment:* in this case, all the distributions P_j are the same, $P_j = P$. In this case, the environment does not actively interact with the circuits.

B *Active environment*, an environment that tries to create as many difficulties as possible to the circuit population. This may correspond to a predator-prey-type interaction, when a predator tries to learn the prey's behaviour and vice versa. In this case, the probability distributions P_j can be different. (Here, interesting situations appear when for large j, the probabilities corresponding to the distributions P_j are not computable in polynomial time.)

Our objective is to show that a stable evolution is possible. We will show that for the above-described population of circuits, a stable evolution is possible – provided that the circuit growth satisfies some natural conditions. These conditions are listed below.

R1 *We assume that for all time moments* $t = 1, 2, \ldots,$

$$Res(t) = \sum_C \sum_{i,j,(i,j)\in E_C(t)} |K_{ij}^C(t)| < Poly(t), \tag{9}$$

where the first sum ranges over all circuits involved in the population, and $Poly(t)$ *is a polynomial.*

This assumption means that, within each time interval $[0, T]$, $t = 1, 2, \ldots, T$, the evolution process can only use resources whose total amount is bounded by a polynomial of T.

R2 *There exists a value* $\beta > 0$ *such that the noises* $\xi_i(t)$ *corresponding to different* i *and* t *are independent identically distributed (i.i.d.) random quantities for which, for each* $a > 0$*, we have*

$$0 < P(|\xi_i(t)| > a) < \exp(-\beta \cdot a). \tag{10}$$

R3 *The population size is polynomially bounded:* $X(t) < Poly(t)$.

Our main assumption about the functions f_j can be described as follows. Let us assume that a conditional relative complexity of the correct outputs increases slowly in some reasonable sense; for example, we can assume that

$$f_{j+1} = g(f_j, f_{j-1}, \ldots, f_1, x), \quad g \in \mathcal{C},$$

$$d(g) = depth(g) \le d_{\max}, \quad Comp(g) < K_{\max}, \tag{11}$$

where $d_{\max}$ and $K_{\max}$ are constant (independent on j), and $Comp(g)$ denotes a circuit complexity, i.e., the number of elementary steps necessary to construct g.

Let us first formulate a simple lemma showing that sometimes one can survive without learning.

Lemma 2. (survival without learning) *If the series* $\sum_{j=1}^{+\infty} \delta_j$ *converges, then for every value* $p_0 \in (0, 1)$ *there exists a circuit population that survives with the probability* $\ge p_0$*, i.e., for which* $P_T > p_0$ *for all* T.

Proof. Take X identical circuits with $\xi_i = 0$. For every input, each circuit generates 1. For such individual circuits, the probability P_T to survive within

the time interval $[0, T]$ is then equal to $P_T = \prod_{j=1}^{T} (1 - \delta_j)^{T_e}$. By taking into account that $\delta_j \to 0$ as $j \to \infty$, we conclude that as $T \to \infty$, the values P_T are bounded from below by some value $\kappa > 0$.

If $\kappa < p_0$ we increase X until we get $\kappa \geq p_0$. The lemma is proven.

Theorem 3. *Assume that for some real number $\rho \in (0, 1)$, the functions f_j satisfy the conditions (11) and*

$$Err_j > \rho \tag{12}$$

for all j. Then there exist values μ and T_e for which there exists an algorithm describing evolution of circuits that satisfies the conditions **R1**, **R2**, *and* **R3**, *and for which $P_T > p_0 > 0$ for all $T > 0$.*

In other words, for this algorithm, the system remains stochastically stable for large time intervals.

This theorem can be interpreted as follows: stable evolution is possible even in severe conditions (when a single error leads to destruction) – if the rate of change of the environment complexity is bounded.

Proof. In this proof, we will use two lemmas from [6]. Recall that a *Bernoulli process* is a discrete-time stochastic process consisting of a sequence of independent random variables that take only two values: *success* and *failure*. For each integer M and real number $p \in (0, 1)$, we can consider an M-trial Bernoulli process, in which in each of the M trials, the probability of success is equal to p. Let us denote the total number of successes in all M trials by Y.

Lemma 3. *For $k < M \cdot p$, we have*

$$\text{Prob}\{Y < k\} \leq \frac{k \cdot (1 - p)}{M \cdot p - k} \cdot C_M^k \cdot p^k \cdot (1 - p)^{M-k}. \tag{13}$$

Lemma 4. *For $r > M \cdot p$, we have*

$$\text{Prob}\{Y > M \cdot p + r\} \leq \left(\frac{M \cdot p \cdot e}{r} \right)^r. \tag{14}$$

Since for our choice of K_{ij} and h_i we have $\min |K_{ij} \cdot u_j + h_i| \geq 0.5b$, we can prove the following useful lemma.

Lemma 5. *Let $y(x)$ be a circuit (2) of depth d and complexity K_{Max}, for which $K_{ij} \in \{b, 0, -b\}$ and $\xi = 0$. Let $\tilde{y}(x)$ be the same circuit with the noise ξ (which satisfies the condition* **R2***). Then*

$$\sup_x \text{Prob}\{y(x) \neq \tilde{y}(x)\} < \exp(-c \cdot \beta \cdot b), \text{ where } c = c(d, K_{Max}) > 0. \tag{15}$$

Let us now describe a circuit evolution and a population growth that satisfy the conditions **R1**, **R2**, and **R3**. We will proceed in three stages. Our estimates are obtained by induction.

Suppose that at $j = m$ the populations contain correct circuits that give correct answers with probabilities $p'_m = 1 - \exp(-c_1 \cdot m)$, where $c_1 > 0$.

Stage I. Generation of new circuits by random mutations. Consider the time interval $I_m = [m \cdot T_e, m \cdot T_e + K_{max}]$, where $T_e > C \cdot d_{max}$ for some large constant $C > 1$. Denote by $\tilde{x}$ a combined entry (x_0, x), where we use $x_0 = (f_m, f_{m-1}, \ldots, f_1)$ as an additional entry component. Using steps 1–4, we construct all possible circuits of complexity $\leq K_{max}$ and depths $\leq d_{max}$. Among them correct circuits may occur, i.e., circuits coinciding with $g(f_m, x)$. For $t \in I_m$, we set $b(t) = b_*$, where b_* is a large constant independent of m. Such a correct circuit can be obtained with the probability $p_c^+(b_*) \cdot \mu^K$, where $\mu > 0$ is the mutation probability and $p_c^+(b_*)$ is the probability that an incorrect circuit gives a correct answer.

We have already obtained the estimate $p_c^+(b_*) > \exp(-c_2 \cdot b_*)$. Denote $\kappa \stackrel{\text{def}}{=} (\mu \cdot p_c^+(b_*))^K$. Then one can expect that, after $K_{\max}$ steps, we will have at least $X_+ = 0.5\kappa \cdot X_m$ correct circuits, where X_m is the number of circuits at the moment $m \cdot T_e$. Indeed, using Lemma 3, one can prove the following result:

Lemma 6. *Consider the random number $Z_m \stackrel{\text{def}}{=} X_+(m \cdot T_e + K_{max} - 1)$ of correct circuits $X_+(t)$ at the moment $t = m \cdot T_e + K_{max} - 1$. If the parameter β is small enough, then the probability that $Z_m < 0.5\kappa \cdot X_m$ can be bounded by the following expression:*

$$\text{Prob}\{X_+(t) < 0.5\kappa \cdot X_m\} < \exp(-c_3 \cdot X_m), \tag{16}$$

where $c_3(\mu, p_c^+, K)$ is a positive constant that does not depend on X_m.

Stage II. Removing circuits and increasing b. The following $T_1 = T_e - K_{max} - 1$ time steps we do nothing, no mutations. Many circuits die, as a result of incorrect answers. On this stage, we increase the parameter b in these circuits (Step 4) by setting $b = b_2 = O(m)$. Denote

- by P_1^*, the probability that at the moment $t = T_e(m + 1)$, the number $\tilde{X}_+ = X_+(m \cdot T_e + K_{max} - 1)$ of correct circuits is smaller than the number $\tilde{X}_- = X_-(m \cdot T_e + K_{max} - 1)$ of incorrect ones: $\tilde{X}_+ < \tilde{X}_-$, and
- by P_0^*, the probability that there are correct circuits left, i.e., that $\tilde{X}_+ > 0$.

Lemma 7. *There exist values T_1, c_4, and c_5 for which the probabilities P_i^* satisfy the following inequalities for all m:*

$$P_0^* < \exp(-c_4 \cdot m), \quad P_1^* < \exp(-c_5 \cdot m). \tag{17}$$

Proof. The probability that a correct circuit survives after T_1 trials is larger than q^{T_1}, where $q > 1 - \exp(-c \cdot b_2)$. Thus, the probability that all correct circuits die is $(1 - q^{T_1})^{X_+} < \exp(-c \cdot X_m)$; since $X_m = O(m)$, we get the first estimate (17).

Denote by Z the number of inputs $\tilde{x}$ among T_1 inputs $\tilde{x}$ for which all incorrect circuits give an incorrect answer $y(x) \neq f_{m+1}(x)$. We denote such inputs by x_{inc}. If an incorrect circuit C_{inc} obtains such an input, C_{inc} dies with a probability p_d close to 1: $p_d = 1 - \exp(-c \cdot b_2)$. The probability that a circuit obtains, as an input, some x_{inc} within T_1 independent inputs, is $p_1 = 1 - (1-\rho)^{T_1}$. Then by Lemma 4, the probability that the number of surviving incorrect circuits is larger than $6(1-\rho)^{T_1} X_m$, does not exceed $0.8^{O(X_m)}$. The number of the correct circuits will be close to $X_+ = c_9 \cdot X_m$, with a probability $> 1 - \exp(-c \cdot X_m)$, where c_9 depends on μ and b_2 but does not depend on T_1. This observation gives the second estimate (17) for sufficiently large values of T_1 large. This completes the proof of the lemma.

Stage III. Replications. We now come back to the design of the algorithm required in Theorem 3. Notice that it is not a priori known whether a given circuit is correct or not. However, one can investigate structures of circuits and one can find a group of circuits having the same structure. We preserve these circuits and remove all the others. Then, we replicate all the remaining circuits to obtain $X_{m+1} = X(t) = O(m)$ up to the moment $t = (m+1)T_e$. By Lemma 5, it is clear that for new noisy correct circuits, the probability of the incorrect output admits the upper bound $\exp(-c \cdot m))$, where $c > 0$ (we repress the noise by increasing $b(t)$ on Stage II; at the other stages $b(t)$ is a large constant b_* independent of m).

We notice that the probability to survive within I_m is larger than $1 - \exp(-c_1 m))$, $c_1 > 0$. The resources within I_m are $O(m|E(m)|) < O(m^3)$. This completes the proof of Theorem 3.

Theorem 3 has simple intuitive meaning. It describes survival with learning. To survive, the population should learn something about a boolean black box. It is a difficult problem, but the population can recognize a black box step by step, if the box's complexity increases "slowly" (i.e., according to (11)).

Example. An interesting example is given by the sequence of conjunctions

$$f_j = D_{\mathbf{i}_1} \wedge D_{\mathbf{i}_2} \ldots \wedge D_{\mathbf{i}_{k(j)}}, \tag{18}$$

where each $D_{\mathbf{i}}$ is a disjunction of some literals: $D_{\mathbf{i}} = \tilde{x}_{i_1} \vee \tilde{x}_{i_2} \ldots \vee \tilde{x}_{i_K}$, where $\tilde{x}_i$ is either x_i or $\neg x_i$, $i \in \{1, \ldots, N\}$. The integer K can be interpreted, biologically, as a redundancy parameter. The dependence of the number $k(j)$ on j can be increasing, decreasing, or non-monotonic (it depends on g in (11)). Notice that learning of (18) is hard for large N [31].

In the case of (18) the evolution stability is connected with the K-SAT problem, which for a few decades has been a focus of many important research activities; see, e.g.,, [1, 6, 9, 20]. We can consider our evolution as a "game" of population against an environment which becomes more and more complicated.

1. If the dimension N of inputs x is fixed, one can survive in a simple way (see Lemma 2) if $P_j = P$ and P is uniform. However, the survival probability may be exponentially small in N.

2. Assume now that $N = N(t)$ increases with time. Suppose that each new clause contains a new literal that is not used in the previous clauses. Then for passive environment with uniform P it is possible to survive without learning (Lemma 2). For active environments, Theorem 3 holds if the distributions P_{m+1} have the following property: $\mathrm{Prob}\{1 - \delta_0 > f_m(x) = 1\} > \delta_0$ with δ uniform in m for x chosen randomly according to P_{m+1}.

3. It is natural to assume that conjunctions (18) are constructed randomly, i.e., all indices $i_K \in \{1, 2, \dots, N(t)\}$ are chosen randomly (*random K-SAT*). For example, at each j we choose a random i, and we add, with probability p, certain L disjunctions to f_m.

In this situation, our problem looks complex, and it is related to the results on phase transitions in hard combinatorial problems [9, 20]. We consider this relation in our forthcoming publications. In this paper, we restrict ourselves to some simple observations.

If $K \leq 2$, one can expect that, inevitably, some new clause will be in a contradiction with previous ones; thus, for the passive case **A**, we can again use Lemma 2.

For $K > 2$ and active case **B**, it is possible that P_m are not computable in polynomial time $Poly(m)$. Indeed, to implement the algorithm from Theorem 3, we should have P_j satisfying the condition (12) (or, at least such that $Err_j > const \cdot j^{-n}$ for some $n > 0$). For large j, it is possible that the number N_j of solutions of the K-SAT problem corresponding to $f_j(x)$ is exponentially small in j; moreover, if $P \neq NP$ and j is a part of the input, there is no polynomial-time algorithm to find x such that $f(x) = 1$ [9, 20]. Nonetheless, even in such a situation, survival is possible if the population always preserves a trivial circuit.

It would be interesting to compare results from this section and Theorem 1 with the real biological situation. A discussion of the problems of species extinction and complexity growth can be found, for example, in [23]. A change of f_j can be interpreted as a variation in ecological conditions. It can be shown that, according to our model, such a change leads to a massive species extinction with an exponential rate (this fact is in good accordance with biological reality, see [23], Ch. 23).

7 Conclusion

There are two fundamental problems of mathematical biology: the morphogenesis problem (emergence of complex structures) and the evolution problem:

a why structures became more and more complicated and

b why Darwin's evolution could generate such structures within "short" time and with "bounded resources" [7, 23, 31].

Mathematical approaches to the first problem started with the seminal work of A. Turing [27]. Now, we can explain the emergence of complicated patterns and describe algorithms to obtain such patterns (for network and circuit models, see [21, 26, 28, 29]). However, the questions about the stability of such emergence are still open.

It seems that the second problem is even more mysterious. In our opinion, the key to this problem can be found in [13, 30]: all biological systems with fixed parameters are unstable, but evolution can stabilize them; in this case, according to our Theorem 1, the Kolmogorov complexity grows (on average). This fact explains why complexity increases in evolution.

Note that complexity here is the complexity of the *genetic code*; the relation between this complexity and the complexity of the organism complexity is not obvious. For some Pffafian models like (2), one can prove that the pattern complexity can be estimated in terms of the Pfaffian chain complexity. Thus, "complex" patterns can be obtained only by using sufficiently "complex" pfaffian models [29]. Probably, both pattern complexity and gene complexity are increasing during the evolution process [23].

Evolution does not necessarily mean "improving". Ch. Darwin avoided the words "higher" and "lower". In fact, following D. Wandschneider ([7], Ch. 10), let us "compare the chance of survival of, say, infusoria with that of humans: risk increases with an increase in capability. A glance at inorganic structures makes this even clearer. The Alps are obviously characterized by considerable stability". Theorem 1 explains this paradox: the evolution of unstable structures has no goals and stability is not necessarily increasing: simply, if evolution stops, destruction is inevitable.

We think (following [31]) that the problem **b** can be correctly posed mathematically only by NP-hard ideas: short time means polynomially bounded, and energetic resources also should be polynomially bounded. Here we show that, at least in certain cases, such fast evolution is possible (even under severe restrictions, but these restrictions should evolve sufficiently slowly in time). One can hope that recent ideas on phase transitions in hard combinatorial problems (for example, [9, 20]) can help us understand efficiency of the Darwin evolution and the Red Queen law (extinction of species when the number of ecological restrictions become too large) [30].

Acknowledgments

The authors are thankful to Prof. O. Radulescu for discussions and to the referees for many useful remarks. One of the authors was supported by the grants RFBR 10-01-00627-a and CDRF NIH Grant RR07801.

References

[1] D. Achlioptas, Lower bounds for random 3-SAT via differential equations, Theor. Comp. Sci. 265 (2001) 159185.

[2] J. P. Aubin, A. Bayen A., N. Bonneuil and P. Saint-Pierre (2005) Viability, Control and Games: Regulation of complex evolutionary systems under uncertainty and viability constraints, Springer-Verlag.

[3] R. Albert and A. L. Barabási, (2002) Statistical mechanics of complex networks, Rev. Modern Physics, **74**, pp. 47-97

[4] R.Bhattacharya, M.Majumdar, Random Dynamical Systems. Theory and Applications., Cambridge, 2007

[5] W. B. Cannon: The Wisdom of the Body, W. W. Norton Co., New York, 1932.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms (Second Edition), MIT Press, 2001.

[7] V. Hisle and Ch. Illies (eds), Darwinism and Philosophy University of Notre Dame Notre Dame, Indiana, 2005.

[8] P. Erdos, A. Rényi, (1960) On the evolution of random graphs, Publ. Math. Inst. Hungarian Academy of Sciences, **5**, pp. 17-61

[9] E.Friedgut, Sharp thresholds of graph properties, and the k-SAT problem. J. Amer. Math. Soc.,12, 1999, no. 4, p.1017-1054.

[10] A. Gabrielov, N. Vorobjov, Complexity of computations with Pfaffian and Noetherian functions, in: Normal Forms, Bifurcations and Finiteness Problems in Differential Equations, 211-250, Kluwer, 2004.

[11] L. Glass and S. Kauffman, The logical analysis of continuous, nonlinear biochemical control networks, (1973) J. Theor. Biology, **34**, pp. 103-129

[12] D. Grigoriev, Deviation theorems for solutions of linear ordinary differential equations and applications to parallel complexity of sigmoids, (1995) St.Petersburg Math. J., **6**, pp. 89-106

[13] M. Gromov and A. Carbone, Mathematical slices of molecular biology, Preprint IHES/M/01/03, 2001.

[14] J. E. Hutchinson (1981). Fractals and self similarity" Indiana Univ. Math. J. **30**, pp. 713747.

[15] A. Khovanskii, Fewnomials, Translations of Mathem. Monographs, Amer. Math. Soc., **88**, 1991.

[16] P. Koiran and C. Moore, Closed-form analytic maps in one and two dimensions can simulate Turing machines. (1999) Theoretical Computer Science, **210(1)** 217-223

[17] Ming Li and P. Vitanyi, An Introduction to Kolmogorov Complexity and Its Applications, Second Edition Springer Verlag, 1997.

[18] A.Lesne. Complex networks: from graph theory to biology. (2006) Letters in Math. Phys., **78**, pp. 235-262

[19] C. Lobry, Une proprieté generique des couples de champs de vecteurs, (1972) Chechoslovak Mathematical Journal, **22 (97)** 230-237.

[20] S. Mertens, M. Mézard, R. Zecchina, Threshold values of Random K-SAT from the cavity method, Random Structures and Algorithms 28 (2006) 340-373

[21] E. Mjolness, D. H. Sharp and J. Reinitz, A connectionist Model of Development, (1991) J. Theor. Biol. **152**, pp. 429-453.

[22] M. Blum, A machine-independent theory of the complexity of recursive functions, (1967) J. Assoc. Comput. Machin., 14, , pp. 322-336

[23] M. Ridley, Evolution, 2nd ed. (Blokwell Scientific Publications Ltd, Oxford, 1996)

[24] H. J. Sussmann, A general theorem on local contollability SIAM J. Control and Optimization Vol. 25 N1, pp. 158 -194

[25] P. Smolen, D. Baxter, J. H. Byrne, Mathematical modelling of gene networks, Review in Neuron, **25**, (2000) 247-292.

[26] D. Thieffry and R. Thomas, Dynamical behaviour of biological regulatory networks, II.Immunity control in bacteriophage lambda, Bull. Math. Biology, **57**, (1995) 277-295.

[27] A. M. Turing, The chemical basis of morphogenesis, (1952) Phil. Trans. Roy. Soc. B, **237** pp. 37-72.

[28] S. Vakulenko, Complexité dynamique de reseaux de Hopfield, (2002) C. R. Acad. Sci. Paris Sér. I Math., t.335.

[29] S. Vakulenko, D. Grigoriev, (2006) Algorithms and complexity in biological pattern formation problems, Annales of Pure and Applied Logic **141**, pp. 421 -428

[30] L. Van Valen, (1973) A new evolutionary law, Evolutionary Theory, **1**, 1-30

[31] L. G. Valiant, Evolvability. Lect. Notes Comput. Sci., v. 4708, 2007, pp. 22-43.

[32] L. G. Valiant, A theory of learnable. (1984) Comm. ACM 27, pp. 1134-1142

[33] B.L. van der Waerden, Algebra, Volumes I, II. New York, NY: Springer (2003).