Algebraic Representation and Modeling of Evolutionary Innovation and Adaptation in Biological Systems

Igor Balaz[1] and Dragutin T. Mihailovic[1]

[1]Faculty of Agriculture, University of Novi Sad, Dositej Obradovic Sq. 8,
Novi Sad, Serbia
ibalaz@polj.uns.ac.rs

Abstract

Living systems are equipped with the coding system which enables them to autonomously determine set of agents for performing all functional tasks. Since scope of their functioning for given environment is entirely dependant on internally given structure of the coding system they are able to evolve both new traits (evolutionary innovation) and optimize existent ones (evolutionary adaptation) by means of mutations and different mechanisms of genetic rearrangements. In this paper we give a generalized mathematical framework for presenting evolution in living systems in terms of category theory, comprising both innovation and adaptation. On that basis we construct a simple computational model, where as an example we performed evolution of randomly generated coding sequences and analyze appearance of interaction networks and their evolution, as well as evolution of the coding sequence itself. We also demonstrate that evolved networks have some properties of metabolism-like systems.

Introduction

Basic mechanism of evolution in biological world is well known. All organisms are equipped with some form of coding sequences (RNA or DNA) which serve as a blueprint for synthesis of RNA and/or proteins, which in turn perform all functional tasks (interaction with environment, transformation of elements, synthesis of all necessary systemic structures). Their functional role depends on their ability to assimilate a segment of environment with an appropriate set of internal operations to produce reactions. Therefore, some form of shared interface between organism and its environment should exist. At the same time, coding sequences are subject to changes through generations due to various external or internal factors, and these changes can be reflected on phenotypic traits of an organism. Usually, phenotype changes are only variations of a given trait, but sometimes organisms can attain completely new properties. These changes are reflected on the overall reproductive success of an organism, as a measure of evolutionary success, which is relative category and depends on three factors: genotype of that organism, properties of the given environment and other organisms in the same population. In more abstract terms, in a given universe, an organism occupy subset of that universe (called niche), and possible scope of organism's place is genetically determined.

Currently, full formal treatment of evolvability is not yet achieved. Evolutionary adaptation is addressed through evolutionary computation (e.g. De Jong, 2006) but innovation has been scarcely touched. One of the reasons for that may be the need for more general formal setting in order to fully capture possibilities of appearance of new structures or mechanism. Some efforts have been made, within domain of topology spaces (Stadler et al. 2001; Shpak and Wagner, 2000) where importance of introducing genotype-phenotype separation was highly emphasized. However, in those works focus was mainly on analysis of topological configuration of state space.

Our aim here is to show that generalized framework for creating an evolvable system (both innovative and adaptive) for the given universe can be described as generation of a set of free objects in n generators from the monoidal subcategory where objects of the mother category are collection of all possible words over the given alphabet which constitutes the set of generators of the universe. The process consists of subsequent creation of equivalence classes where equivalence relations are only implicitly determined, so that their exact action depends on structure of objects on which they are applied. For the sake of simplicity in our elaboration we will limit ourselves only to the domain of strings and lattices. Resulting objects are ordered pairs of strings, which can be considered as functions in the given universe, by creation of enriched category. Overall, starting monoidal subcategory can be interpreted as mutation search-space, where objects are coding strings (DNAs), their substrings are transformed to functions which operate on a given environment and give rise to appearance of the network of interactions (metabolism). Therefore, one object of the monoidal subcategory coupled with the set of all functions derived from its structure constitutes one genotype, while the phenotype is here simply equal to the metabolic network. Together, they constitute an organism.

In the next section we will develop described framework and will point out some general requirements of genotype-phenotype mapping, in order to be functionally evolvable.

After that we will concretize previous notions by generating computational model and demonstrate its functioning for randomly generated coding sequences placed in randomly generated environment. Finally, in conclusions we point out some possibilities for further development and application of the given framework.

Mathematical Framework

If we denote some living system as L and its environment as E, in analogy to biological world, we can define three basic premises important for our task:

1. interaction of two systems L and E is based on transformation of elements of E by action of elements of L, called functional elements;
2. in order to interact, functional elements and environment must share some of their properties and such shared subset of properties should be general enough to serve as a representative of the environment;
3. generation of functional elements is determined internally, by the system L, through existence of some coding element(s) on which a sequence of equivalence relations is applied.

Whatever mathematical representation we chose for L and E, in the most general sense both of them can be regarded as free objects generated over some alphabets, which are in turn members of category of sets, *Set*:

$$\mathbb{C} \underset{\mathbf{F}}{\overset{\mathbf{U}}{\rightleftarrows}} Set\,. \qquad (1)$$

where $\mathbf{F}$ is free functor, $\mathbf{U}$ is its right adjoint, forgetful functor, while $\mathbb{C}$ is category of all algebraic structures generated by $\mathbf{F}$. If we take some $X \in Set$, then $\mathbf{F}(X)$ is free object, while X can be defined as a set of generators of the $\mathbf{F}(X)$. In order to keep things as simple as possible, we will neglect all notions of dynamics of metabolism and existence of any control mechanism, which will demand definition of additional restrictions on chosen structures. Also, from the environment we expelled all other "organisms" and consider environment as an inert space without internal dynamics. Therefore, we will only define two alphabets, X_E and $X_L \subseteq X_E$, and corresponding monoidal categories generated by $\mathbf{F}$: $(M_L, \bullet, e)$ and $(M_E, \bullet, e)$ where M_L and M_E are categories of all strings generated by corresponding generators (objects are strings, mappings are inclusion maps), bifunctor $\bullet$ is binary operation of string concatenation, while e is identity element, in this case, empty string. Following our analogy with coding elements in living systems, $(M_L, \bullet, e)$ can be interpreted as universe of all possible coding strings while $(M_E, \bullet, e)$ can be regarded as universe of all possible structures in the environment. At this stage, $(M_L, \bullet, e)$ is simply a subcategory of $(M_E, \bullet, e)$ with no additional properties. However, applying premises we postulated at the beginning of this section, this simple monoidal category will be transformed into category of function over the given universe.

Since by premise 3, we demand that some structure should serve as the coding element we can choose any $d \in M_L$ and construct slice category M_L / d where objects are mappings in M_L with d as the codomain ($a \xrightarrow{\alpha} d, b \xrightarrow{\beta} d, \ldots$), while mappings are given by $f : (a \xrightarrow{\alpha} d) \to (b \xrightarrow{\beta} d)$ such that $\beta \circ f = \alpha, \alpha \circ f = \beta$. Therefore, M_L / d is category of all strings which are substrings of the string d. Description of structure of objects in categories M_L / d and $(M_L, \bullet, e)$ can be regarded only as a specialization, but these two categories are structurally different. Category M_L / d is not monoidal category since closeness under operation $\bullet$ is violated. Loosing its monoidal character, category M_L / d is also expelled from dynamics provided by bifunctor $\bullet$ which in practice means that by changing d, M_L / d should be reconstructed *de novo*. However, certain stability of M_L / d can be achieved if d is part of some equivalence class. Living systems provided such stability by existence of rewriting-like systems of gene expression where linear order of elements of chains is preserved while several points of reduction are performed (e.g. basis for DNA transcription are triplets, genetic code is degenerate and amino acids can be sorted into groups with similar chemical reactivity). In other words, process of gene expression generates additional structure on the DNA in the following manner. If we represent gene expression as the functor $\mathbf{G}$ which is full and faithful but not embedding, from category $(M_L, \bullet, e)$ to monoidal category $\mathbf{At}$ generated from some $X_A \subseteq X_E$, then category $\mathbf{At}$ will preserve internal string order but exact reconstruction of its source in $(M_L, \bullet, e)$ could not be realized. In other words, since $\mathbf{G}$ is bijective only on hom-sets, but is not injective on objects, its object function g have section such that for

$$a \underset{s}{\overset{g}{\rightleftarrows}} \mathbf{G}(a) \qquad (2)$$

equation $g \circ s = 1_{G(a)}$ is valid but $s \circ g = 1_a$ cannot hold (it does not have retraction). In that case objects of $(M_L, \bullet, e)$ are naturally, by functor $\mathbf{G}$, separated into disjoint union of n sets. If we denote set of objects of $(M_L, \bullet, e)$ as S_{M_L} then relation $R \subseteq S_{M_L} \times S_{M_L}$ naturally defined by g:

$$R = \{(d, d') \in S_{M_L} \times S_{M_L} \,|\, g(d) = g(d')\} \qquad (3)$$

is congruence relation, and by $[d]_{\mathbf{G}}$ we will denote equivalence class of elements of $(M_L, \bullet, e)$ with respect to functor $\mathbf{G}$. Therefore, in order to provide some degree of stability when facing with mutations, sufficient formal requirement is that expression mechanism from genotype to phenotype reduces number of elements along the process. However, structures created during expression should be able

to interact with environmental structures and perform some action upon them, in order to be functional. This notion leads us to our second basic premise: two systems should share some of their properties, thus creating an interface.

Formally, interface between two different objects can be introduced quite simply. It is enough to postulate segments of structures as visible to each other which is basically in focus of control theory and agent based systems. From algebraic perspective it raises an interesting problem of general mathematical properties of interfaced objects which should be fulfilled in order to be able to perform transformations, either mutual or governed by one of interacting systems. However, here we will omit that question and will take most simple approach by creating interface within the same group of mathematical structures, performed by the functor **G**.

As we implied in introduction, object part of the functor **G** decomposes objects of M_L into ordered pairs whose structure is determined by arrangement of attributes introduced by the functor **G**. It can be done in *n* steps, depending on the chosen model. If we follow analogy with the natural world, then we can construct 2-step process. The first one is defined simply by identifying all permutations of strings of fixed length created over the alphabet X_L, grouping them into m disjoint subsets, and declaring equivalence relation θ over each subset. Then, subsets are equivalence classes and strings $s \in S_{M_L}$ can be mapped into corresponding quotient strings s/θ.

Since by functor **G** each element of X_E is equipped with some attributes it raises following structure. Let us denote set of all words over the alphabet X_E as *T*, set of all attributes as *M*, and set of attribute values as *J*, then formal context is (T,M,J,I) where I is a ternary relation $I \subseteq T \times M \times J$ which unites objects with corresponding attributes. Further, if $O \subseteq T$ is set of all functional elements, $W \subseteq M$ is set of their attributes and $K \subseteq J$ is set of attribute values for W, such that $O' = \{w \in W \,|\, (\forall t \in O), tIw\}$ and $W' = \{t \in T \,|\, (\forall m \in W) tIw\}$, then concept of the context (T,M,J,I) is triple (O,W,K) where $O' = W$ and $W' = O$. Since for a given context a number of different concepts can be defined, we will denote set of all possible concepts as $\mathbf{B}(T,M,J,I)$. If we take $\subseteq$ as an order relation, then $(\mathbf{B}(T,M,J,I);\leq)$ is concept lattice where nodes are concepts of the given context. Finally, we will demand that set of attributes M is created as union of languages created over *n* alphabets. Elements of alphabets we will denote as generator attributes, and all other words will be called derived attributes. Reasons for that will be clear shortly.

Since elements of s/θ are also equipped with some attributes, they are characterized by specific $I_s \,|\, o \in s/\theta$ and are associated with the mapping $\tau : s/\theta \to (W,\leq)$, $((o_1,o_2,...,o_n) \to (\{w_1 \times j_1\}_1,\{w_2 \times j_2\}_2,...,\{w_n \times j_n\}_n))$. Clearly $(W,\leq)$ is not a chain anymore since there are no defined order relations among members of the set M by the mapping τ which preserves only order generated at the original coding string. However, following our analogy with natural systems, we can define some relations among attributes themselves. Keeping things as simple as possible we can for example take only one alphabet $D \subset M$ and define equivalence relation ϑ over all words of the *D*-language. Resulting posets $s/(\vartheta \circ \theta)$ now represents "folded" functional elements, where order of remaining attributes determine interface. Referring back to our second basic premise, we demand that in order to interact, functional elements should reduce environment on the basis of existence of shared properties. Here, it means that interface is formed on the basis of existence of some $V_p \times L \subseteq W \times K$ where $V_p \subseteq W, L \subseteq K$, and V_p is actually a subset of remaining generator attributes on the $s/(\vartheta \circ \theta)$. Our questions are: (i) what are the meaningful constraints for determining V_p within our framework, and (ii) what is the position of the concept generated by the $s/(\vartheta \circ \theta)$ within the $(\mathbf{B}(T,M,J,I);\leq)$. Since V_p is naturally designed to be a filter for representing environment we can postulate that it should be a part of majority of concepts of the $\mathbf{B}(T,M,J,I)$. Choosing some obscure attributes will promptly lead the system to evolutionary or functional dead end. Further, concept generated by the $s/(\vartheta \circ \theta)$ represents interface for that functional element and its upper bound is exclusively composed of concepts with derived attributes and represents place where environmental objects suitable for functional transformation can be found.

Finally, referring back to our first premise, $s/(\vartheta \circ \theta)$ should also govern determination of some function over the "visible" part of environment. Since determination of exact function is highly dependant on the chosen model, at this stage is only possible to point out general requirements which should be fulfilled in order to autonomously generate functions within given framework. Our strategy is to reconstruct possible relations from already generated structures, and then to group them into small number of isomorphic representatives, according to the structure of V_p. We will start with some basic notions.

Any finitary relation R can be defined as a couple $R = (D(R), C(R))$ where $D(R)$ is collection of nonempty sets $X_1,...,X_k$ which are called domains, while $C(R) \subseteq X_1 \times ... \times X_k$ can be denoted as the figure of *R*. Since, number of possible relations which can be constructed from the set A, equals $2^{|A|^2}$ (Robinson, 2003), our aim is to postulate some restriction rules and to find some route to grouping them together. Reconstructed relation should satisfy following axioms:

1. Function: for each element of $D(R)$, is assigned a unique element of $C(R)$;
2. Identity: for every object *a*, there exists relation $\mathrm{id}_a : a \to a$ such that for every relation $f : x \to y$, $\mathrm{id}_y \circ f = f = f \circ \mathrm{id}_x$;

3. Associativity: if f, g, h are relations, then $h \circ (g \circ f) = (h \circ g) \circ f$ should always hold;
4. Limit: if we have some set of relations of shape J, then a diagram of shape J is functor $F : J \to C$. A cone of the diagram is an object K of C together with family of morphisms $\varepsilon_x : K \to F(x)$ such that for every morphism $f : x \to y$, $F(f) \circ \varepsilon_x = \varepsilon_y$. A limit of the diagram is a cone (K, ε) such that for any other cone (L, ϕ) there exists a unique morphism $u : L \to K$ such that $\varepsilon_x \circ u = \phi_x$ for all x in J. Valid relations are only those that have limit.

By the first three axioms, we narrowed down universe of possible relations to structure preserving morphisms. In that sense set A of elements which constitute upper bound of the $s/(\vartheta \circ \theta)$ can be defined as domain of some function $f_{s/(\vartheta \circ \theta)}$ while its codomain should be subset of all concepts consisting V_p. By the last axiom we demand that reconstructed relation at least satisfy condition of being some of universal constructions in abstract algebra (product/coproduct, pushout/pullback...). As it is obvious we put only very elementary constraints, just in order to keep the system consistent. However, at the same time we come very near to our goal. Now, all possible functions can be generalized to some of few universal constructions applicable to chosen model. For example, if objects are strings, two most basic structure preserving operations are string separation and string concatenation. Limits of these universal operations are product and coproduct. Therefore, suitable codomains for the set A can only be such that operations of separation and concatenation are reconstructed. Exact structure of functions, of course depends on structure of attributes on a given functional element. As a final step, we should preserve stability of determined modes of action. It can be easily done by choosing any subalphabet of V_p attributes and mapping groups of words to some of possible universal constructions.

In summary, functor **G** maps strings of the category $(M_L, \bullet, e)$ to the monoidal category $\mathbf{At} = (A, \bullet, e)$ where objects are ordered pairs, composed of upper bound of the concept generated by the $s/(\vartheta \circ \theta)$ as the first member of the element, while the second member is determined again by equivalence relations generated by the V_p over the set *M*, morphisms are inclusion maps, bifunctor $\bullet$ is binary operation of object concatenation, while e is identity element, in this case, empty string. Due to its monoidal character, and structure of its objects, **At** can readily be used as generator of abstract metrics over the environment, represented by some category Γ, by replacing hom-sets from Γ with objects from **At**. Then Γ is category enriched over **At** (or **At**-category) such that for each pair of object $x, y \in ob(\Gamma)$, where $ob(\Gamma)$ is collection of objects of Γ, hom-set $\hom(x, y) \in \mathbf{At}$, with preserved identity, composition and associativity.

Computational Model

In order to demonstrate functioning of the framework described above we built the model of it using the individual-based approach: population of "cells" consists of individual coding strings glued with corresponding network of transformations of environmental elements, the environment is composed of *n* number of different strings and interaction of each cell with environment is computed individually. Additionally, process of transformation of codes to functions is inspired by natural process of gene expression and formally is composed of two approaches: reduction by imposing equivalence relations at different levels, on which are applied some elementary notions of relation theory. As a result, functions are created in recipe-like manner. It enables free application of mutations over the coding sequence, without designed constraints on allowed number of functional elements, or scope of their domains/codomains.

Elements of the alphabet O	**Corresponding triplets**
A	GCT, GCC, GCA, GCG
R	CGT, CGC, CGA, CGG, AGA, AGG, AAA, AAG
H	AAT, AAC, CAT, CAC, CAA, CAG
D	GAT, GAC, GAA, GAG, TCT, TCC, TCA, TCG, AGT, AGC, ACU, ACC, ACA, ACG
S	UGT, TGC
W	GGT, GGC, GGA, GGG, CCT, CCC, CCA, CCG, TGG
I	ATT, ATC, ATA, TTA, TTG, CTT, CTC, CTA, CTG, ATG, TTT, TTC
Y	TAT, TAC
V	GTT, GTC, GTA, GTG

Table 1: Rules of transformation of triplets from coding strings to elements of the alphabet O

Coding strings were generated randomly as words over the given alphabet $X = \{A, T, G, C\}$. There were no additional structures on coding strings; they were composed only as segments of symbols. Any additional structure on them can only be implicitly imposed, as a result of mappings applied to them. Separation into genes, and their expression into functional elements was designed as a composition of three mappings: identification of "proper" substrings (genes), their translation into strings of symbols equipped with attributes and folding into functions guided by order of attributes.

In analogy to the natural world, as a unit of reading of coding string we choose triplets (how changing complexity and strategy of reading influence dynamics of evolution will be presented elsewhere). Again, in analogy to the natural world, we determine rules of transformations of triplets into elements of the alphabet *O = {A,R,H,D,C,I,W,Y,V}* which is reduced version of list of amino acids. We analyzed their chemical properties and grouped similar ones into only one

representative. In order to keep their natural ratio, we assemble their coding triplets under representative groups (Table 1). As genes we identify those substrings which start with ATG tripled and end with TAG, TGA or TAA triplet. In order to optimize procedure, we neglected all sequences translated into strings shorter that 10 characters. At the same time, for members of the alphabet *O* we define formal context by introducing the set of many-valued attributes $M = \{C, H, I, K, \#\}$, and the set of attribute values $J = \{+, -, 0, 1, 2, 3, k+, k-\}$. Together they constitutes many valued context (O, M, J, I) where I is ternary relation $I \subseteq O \times M \times J$ which unites objects with corresponding attributes in accordance to the Table 2.

Translated strings to the alphabet *O* are "folded" in accordance to the attribute *H* such that all elements where H value is 1, constitute equivalence class. On that basis, reduced quotient string is formed and it will be regarded as an active place. After that, procedure for determination of domains of active place and mode of action is activated.

Having in mind that each function can be represented as subset of the set of ordered pairs, we determined functioning of folded strings creating following duple. First element, which determine domain of the function, is defined as a set of strings such that for all strings exists substring which is equal to the structure of the C-index in the single domain of the active place. Determination of the second element is based on the structure of the K-index at the domain. We simply determine mode of action as k- when there is prevalence of k- values and vice versa. Exact action is defined as string separation at the beginning of the matching region for the k-, and concatenation of all strings recognized by all domains at the single functional element, for k+. It is clear that action of k+ can only be performed if there are more than one domains. Therefore, second element of the duple is defined by performing K-derived action.

Environment is composed of *n* number of randomly generated binary strings composed only of C-attributes. Since our focus is possibility for evolution in principle, we did not determine any constraints regarding spatial distribution or concentration of substances, or their internal structure. When examining population of cells we suppose they share the same environment. It means that after each interaction, newly generated strings are placed into shared environment uniformly available to all cells. Interaction of cells with environment is performed by searching given environment for members of the domain of each functional element. If some environmental element is recognized, it is transformed into product(s) according to the K-derived action for given functional element. As a result, interaction networks are created. In order to analyze them, we used Python-based package, NetworkX (Hagberg, et al. 2008). As main indicators of evolution of networks we used number of connected components, and diameter of components. After completing interaction with environment, fitness value was calculated for each individual cell using formula: $fit = nd^{ac} + en$ where *nd* denotes total number of reactions which can be performed in given environment, *ac* is number of autocatalytic chains, while $en = ob * 100 / uk$ where *ob* is number of different molecules transformed by the cell in given environment and *uk* is total number of molecules in given environment. According to calculated fitness values, some cells were removed, while some were duplicated, keeping the number of cells in population constant. Detailed procedure depends of particular experiment performed. The rules for the evolution were chosen in order to represent essential mechanisms of natural evolution: selection, chance and ability to cope with the environment. Since in this model cells do not reproduce autonomously, which should be used as a measure of their fitness, we designed fitness so to reflect cell's ability to survive, in terms of number of reactions and percent of environmental objects which cell can recognize and transform.

Elements of *O*	Attributes				
	#	C	H	I	K
R	0	+	0	3	k+
H	0	+	0	2	k-
D	0	-	0	3	k-
Y	0	-	0	3	k+
W	0	0	0	0	0
A	0	0	1	1	0
V	0	0	1	2	0
I	0	0	1	3	0
S	1	0	0	0	0

Table 2: Attributes of elements of the alphabet *O*. C-index and I-index are taken in accordance to Whitford [2005] so that C-index represents unified charge and polarity values, while I-index is derived from Van der Waals index and normalized for a real number scale in a range [0-3]. H and K values are taken from Copeland [2000] so that H-index represents hydrophobicity distribution, while K values are normalized pKa values transformed to the mode of reactivity, where k- denotes string separation, while k+ means string concatenation. Index # is introduced as a separator of active place into domains.

Finally, next generation was created by mutating existent coding sequences of each individual. Mutation rate was set to 1 mutation cycle / 100 coding bases. Mutation cycle consists of three defined possibilities, randomly chosen at each cycle repetition: (1) point mutation – one base is randomly chosen and replaced by some other random base; (2) deletion – randomly chosen sequence up to 10 elements is removed; and (3) insertion – randomly generated sequence of the same length is inserted at randomly chosen place within coding sequence.

Performed Experiments and Simulation Results

We performed two kinds of experiments. In the first one we randomly generate population of 20 coding strings of variable length, which was randomly chosen from the interval 500-1000 elements. Environment was composed of 100 randomly generated different elements, where maximum length of generated strings was set to 10. All cells were placed into the same environment, so products generated by one cell were available to all other members of the population. After each generation fitness was calculated for each cell and the

population was accordingly separated into two halves: "least fit" and "most fit". Half of the cells from the first group were randomly chosen, eliminated and replaced by the same number of cells, randomly chosen from the second group.

In the second experiment our aim was to monitor evolution of a single cell lineage. Therefore, we randomly generated only one coding string of length randomly chosen from the interval 500-1000 elements. Environment was created as in the first experiment. In order to reduce complete randomness of the search space we performed "forced" evolution by creating generations according to the following procedure. After completing interaction with the environment, the string was multiplied into 10 copies and each of them was mutated. Interaction of mutants with environment was "virtual" in a sense that their interaction was performed separately of each other. Fitness for them was calculated like in the previous experiment and only one was randomly chosen from the "most fit" group to replace the original one.

Figure 1a, depicts growing of length of coding strings during the evolution, which is followed by increase in number of encoded functional elements at the same rate (results not shown). This is clearly governed by demands of the fitness value, where any increase in number of performed reaction is favorable. However, analysis of appeared interaction networks shows that underlying process of optimization also takes place. We searched each network for number of components where component is defined as its maximal connected subgraph where any two vertices are connected to each other by path. When number of components is 1, it means that the whole network is connected. Otherwise, network is divided into n disjointed subgraphs. For each component we also determined diameter defined as greatest distance between any pair of vertices. As it can be seen from Figure 1b, average number of components per cell in the population decreases with time and at n = 378, population became uniform in the sense that networks for all cells became fully connected. However, it takes additional 235 generations until population settled down to that value and remains uniform for next 200 generations. At the same time, diameter of the largest component slowly declined and for the last 200 generations oscillates around 9. Diameter is important characteristic which can indicate structural difference between non-biological scale-free networks, as opposed to metabolic networks (Jeong et al. 2000). For non-biological networks diameter increases logarithmically with the addition of new nodes (Barabasi and Albert, 1999) which, in this case would imply that increase in number of functional elements should lead to larger diameter of the corresponding interaction network. However, as Figure 2a depicts, diameter remains stable despite several-fold increase in number of nodes. Indirect confirmation can be seen in Figure 2b which show that after initial rapid increase in number of environmental elements suitable to transformation their number remains relatively stable, while total number elements is stabilized within first 30 generations. All of these facts indicate increased connectedness of substrates, leading to evolutionary stabilization of appeared interaction networks. Additionally, it clearly shows that networks evolved within described framework diverge from randomly generated ones.

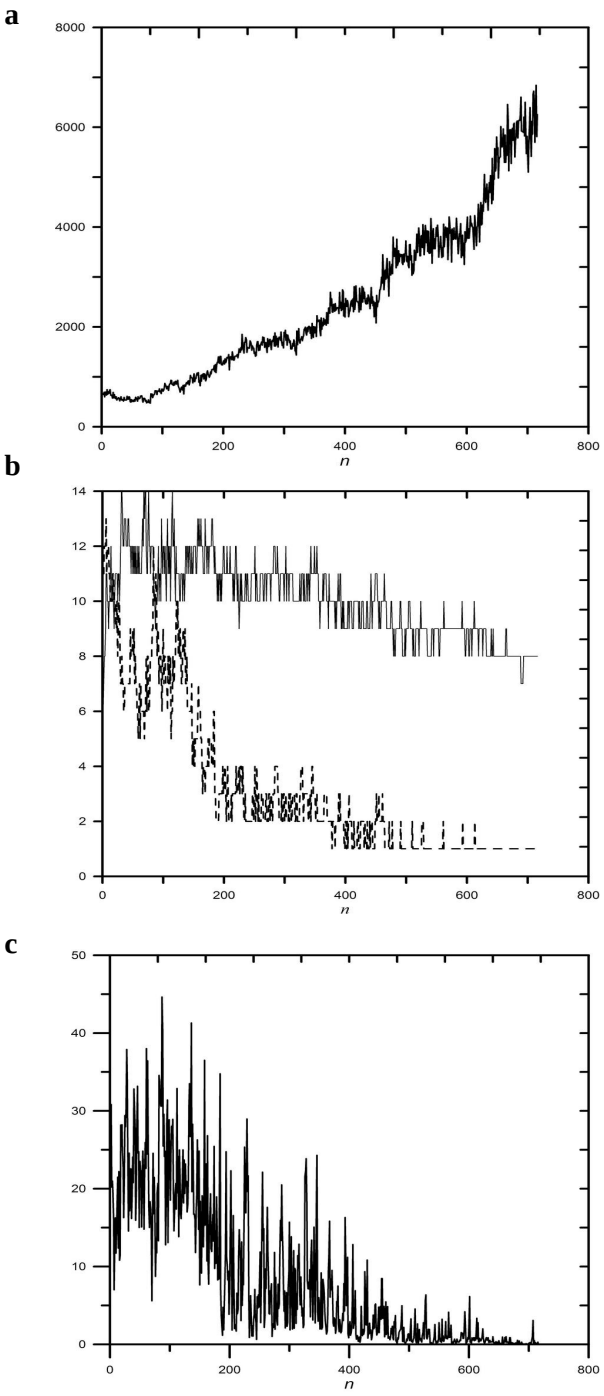

Figure 1. Results of the population experiment, where n is number of generations. Vertical axes represents: (a) average length of coding strings in population; (b) average number of connected components in interaction networks in population (dashed line) and average diameter of the largest component (solid line); (c) variance of number of connected components among cells in the population.

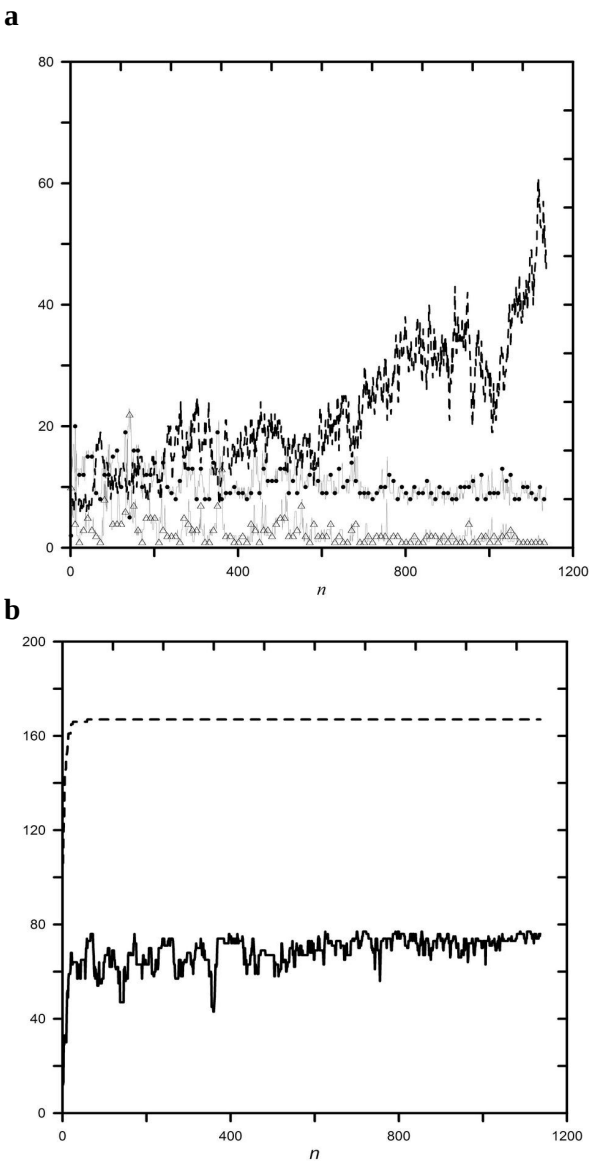

Figure 2. Results of the experiment where single cell lineage was followed through evolution; *n* is number of generations. (a) number of functional elements (dashed line), diameter of the network (circles), and number of components in the network (triangles); (b) total number of elements in the environment (dashed line) compared with number of environmental elements suitable to transformation by functional elements of the cell (solid line).

Another problem we investigated is the ability of the computational model to avoid being trapped in a fixed stable state. As it was pointed out by Conrad (1998), evolving system that gradually optimizes its traits can escape local stable state by increasing dimensionality of evolutionary search space. He termed such strategy as extradimensional bypass. In natural systems, the only mechanism to transform evolutionary search space is adding new observables by constructions of new sensors (Pattee, 1985). Each new sensor means opening possibility for functional existence of new observable in the environment which in turn means creation of additional state variable. Changing the set of state variables that characterize the system, at the same time means changing the structure of both: its functional space and its evolutionary search space. In order to examine the possibility of extradimensional bypass of our computational model, we first evolved one population of cells within one environment, and after 500 generations we replaced environment with the new one composed of 100 randomly generated different elements. Figure 3, depicts change of the fitness value along generations. A gap at n = 500 and relatively fast recovery indicate the ability of the model to extend its dimensionality when faced with new conditions. Therefore, settling into the stable state indicated by results shown in Figure 1, is directed by environmental fixedness that leads to evolutionary stagnation.

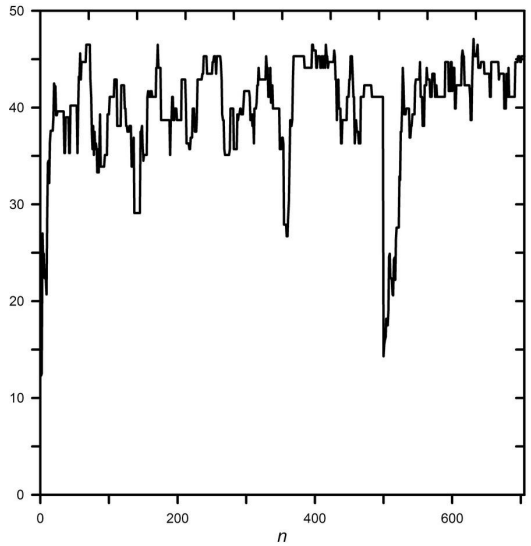

Figure 3. Results of the experiment where population of cells was successively evolved in two different environments; *n* is number of generations while vertical line shows fitness value. New environment is introduced after 500 generations.

Conclusions

We have created generalized mathematical framework for describing evolutionary systems in which appearance of phenotype (network generated by the interaction of the set of functional elements with the environment) is governed by expression of the coding sequence (genotype). Rules of expression were determined implicitly as successive determination and application of equivalence classes. In terms of universal algebra, described framework is actually process of freely creating algebraic structures. Therefore, depending on chosen rules for formation of equivalence classes, any mathematical object can be created. Comparison of such created objects can be performed by associating appropriate homomorphisms, which adds additional strength to the described framework. In the context of this paper it means that patterns of evolution can further be abstracted and analyzed

for different algebras, thus possibly revealing underlying mechanisms of evolutionary adaptation and innovation.

On the other hand, it is also a rich source of investigation of evolution within the single model. In this paper we confine ourselves only to pursuing analogy to existent natural world. However, dynamics of evolution can be easily investigated with different parameters: different modes of reading coding sequence, variations in chosen attributes and their characteristics, or cardinality of alphabets used.

Even within single model we described in this paper, some significant results are obtained. First, we show that initially created disjointed networks, during evolution tends to fall into single connected network which remains relatively stable under unchanged mutation pressure. Additionally, as opposed to random networks, diameter of evolved networks remains stable even when number of functional elements increase several-fold. Therefore, we think that they can be regarded as metabolism-like.

Finally, in order to enrich described system and raise it to the level of artificial cells, two additional aspects should be introduced into models derived from the framework. Firstly, in natural systems gene manipulation machinery is the product of that machinery and its evolution. Therefore, it would be necessary to allow interaction of obtained functions with the coding string in order to allow evolution of expression rules. Although the framework allows such reverse operations, we tried to keep presented models as simple as possible. Therefore, we didn't define any attributes over coding strings, thus keeping them out of the domain of derived functions. Secondly, notions of space (either metric or topological), quantity and control would add possibility for investigation of metabolism functioning, not just of its general structure, which is the case here.

Acknowledgements

This work was funded by the Serbian Ministry of Science and Technology under the project "Modeling and numerical simulations of complex physical systems", No. OI141035 for 2006-2010. We are also grateful to anonymous reviewers whose useful comments helped us to improve this manuscript.

References

Barabasi, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science* 286:509–512.

Conrad, M. (1998). Towards high evolvability dynamics. In Van de Vijver, G., Salthe, S., and Delpos, M., editors, *Evolutionary Systems*, pages 33-43. Kluwer Academic Publishers, Dordrecht, Holland

Copeland, R.A. (2000). *Enzymes-A Practical Introduction to Structure, Mechanism and Data Analysis*. Wiley-VCH, New York.

De Jong, K.A. (2006). *Evolutionary Computation: a unified approach*. MIT Press, Cambridge MA

Hagberg, A.A., Schult, D.A., and Swart, P.J. (2008). Exploring network structure, dynamics, and function using NetworkX. In Varoquaux, G., Vaught, T., and Millman, J., editors, *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15. Pasadena, CA USA.

Jeong, H., Tombor, B., Albert, R., Oltvai, Z.N., and Barabasi, A.-L. (2000). The large-scale organization of metabolic networks. *Nature* 407:651–654.

Pattee, H.H. (1985). Universal principles of measurement and language function in evolving systems. In Casti, J. L., Karlqvist, A., editors, *Complexity, Language, and Life: Mathematical Approaches*, pages 268-281, Springer, Berlin

Robinson, D.J.S. (2003). *An Introduction to Abstract Algebra*. Walter de Gruyter, Berlin

Shpak, M., and Wagner, G.P. (2000). Asymmetry of Configuration Space Induced by Unequal Crossover: Implications for a Mathematical Theory of Evolutionary Innovation. *Artificial Life* 6:25–43.

Stadler, B.M.R., Stadler, P.F., Wagner, G.P., and Fontana, W. (2001). The Topology of the Possible: Formal Spaces Underlying patterns of Evolutionary Change. *J. theor. Biol.* 213:241–274.

Whitford, D. (2005). *Proteins – Structure and function*. Willey & Sons, West Sussex.