Contextual Genetic Algorithms: Evolving Developmental Rules

Luis Mateus Rocha
Los Alamos National Laboratory, MS P990
Los Alamos, NM 87545, USA
e-mail: rocha@lanl.gov

In: *Advances in Artificial Life*. F. Moran, A. Moreno, J.J. Merelo, and P. Chacon (Eds.). Lecture Notes in Artificial Intelligence, Springer-Verlag. pp. 368-382.

Abstract. A genetic algorithm scheme with a stochastic genotype/phenotype relation is proposed. The mechanisms responsible for this intermediate level of uncertainty, are inspired by the biological system of RNA editing found in a variety of organisms. In biological systems, RNA editing represents a significant and potentially regulatory step in gene expression. The artificial algorithm here presented, will propose the evolution of such regulatory steps as an aid to the modeling of differentiated development of artificial organisms according to environmental, contextual, constraints. This mechanism of genetic string editing will then be utilized in the definition of a genetic algorithm scheme, with good scaling and evolutionary properties, in which phenotypes are represented by mathematical structures based on fuzzy set and evidence theories.

1. Introduction

The essence of GA's lies on the separation of the description of a solution (e.g. a machine) from the solution itself: variation is applied solely to the descriptions, while the respective solutions are evaluated, and the whole selected according to this evaluation [14]. A genetic algorithm "is primarily concerned with producing variants having a high probability of success in the environment" [19, page 35]. Nonetheless, one important difference between evolutionary computation and biological genetic systems, lies precisely on the connection between descriptions and solutions, between signifier and signified. In genetic algorithms the relation between the two is linear and direct: one description, one solution. While in the biological genetic systems there exists a multitude of processes, taking place between the transcription of a description and its expression, responsible for the establishment of an uncertain relation between signifier and signified, that is, a one-to-many relation.

> "The proteins encoded by [DNA] are [...] oxymorphic: their individual shapes are precisely unpredictable. So long as this is true, the genomic language, like our own languages, will not have a logical link between signifier and signified. This will not prevent its being read or understood; rather, it will assure that DNA remains a language expressing as full a range of meanings through arbitrary signifiers as any other language." [26, p. 70]

In other words, the same genotype will not always produce the same phenotype; rather, many phenotypes can be produced by one genotype depending on changes in the environmental context. If the effects of changing environmental contexts affecting gene expression within an individual can be harnessed and used to it's selective advantage in

a changing environment, then we can say that such an individual has achieved a degree of control over its own genetic expression. It is the objective of this paper to propose a computational scheme which may be able to achieve this degree of control. It will be further suggested, that the modeling of biological development may be linked precisely to GA's capable of evolving this extra degree of control.

To establish this one-to-many relationship between descriptions and solutions in GA's, I will propose an extra mechanism inspired by the edition of RNA in biological genetic systems. Section 2 will introduce some of the known mechanisms of RNA Editing. Section 3 will introduce semiotic model offering a theoretical framework for RNA editing. Section 4 will propose computational counterparts to RNA editing. Section 5 will discuss the utilization of such mechanisms regarding the problem of development. Finally, Section 6 will present one particular algorithm, which utilizes fuzzy set and evidence theory to introduce even higher levels of uncertainty to description/solution relations.

2. RNA Editing

The discovery of messenger RNA (mRNA) molecules containing information not coded in DNA, first persuaded researchers in molecular biology that some mechanism in the cell might be responsible for posttranscriptional alteration of genetic information; this mechanism was called 'RNA Editing' [2, 1986]. "It was coined to illustrate that the alterations of the RNA sequence (i) occur in the protein-coding region and (ii) are most likely the result of a posttranscriptional event" [3, page 16]. The term is used to identify any mechanism which will produce mRNA molecules with information not specifically encoded in DNA. Usually we will have insertion or deletion of particular bases (e.g. uridine), or some sort of base conversion (e.g. adenosine ÷ guanisine).

The most famous RNA editing system is that of the African Trypanosomes [3; 36]. The mitochondrial DNA of this parasite, responsible for sleeping sickness, "consists of several dozen large loops called maxicircles and thousands of smaller ones called minicircles." [27, page 132] At first, the minicircles were assumed to contain no genetic information, while maxicircles were known to encode mitochondrial rRNA. However, the maxicircles were found to possess strange sequence features such as genes without translational initiation and termination codons, frame shifted genes, etc. Furthermore, observation of mRNA's showed that many of them were significantly different than the maxicircles from which they had been transcribed. These facts suggested that mRNA's were edited posttranscriptionally.

It was later recognized that this editing was performed by *guide RNA's* (gRNA's) coded mostly by the minicircles, the strands of DNA previously assumed to contain no useful information [37; 4]. In this particular genetic system, gRNA's operate by inserting, and sometimes deleting, uridines. To appreciate the effect of this edition consider figure 1. The first example [3, p. 14] shows a massive uridine insertion (lowercase u's); the aminoacid sequence that would be obtained prior to any edition is shown on top of the base sequence, and the aminoacid sequence obtained after edition is shown in the gray box. The second example shows how potentially the insertion of a single uridine can change dramatically the aminoacid sequence obtained; in this case, a termination codon is introduced.

It is unclear how exactly gRNA's insert uridines into mRNA's; basically, the shorter gRNA strings base-pair with stretches of mRNA, and at some point will insert a number of uridines [33]. An interesting aspect of the gRNA/mRNA base-pairing is that it is more general than the Watson-Crick base-pairing found in DNA and RNA, it is more ambiguous since “uracils in mRNA can be specified by either guanine or adenine in gRNA” [36, page 36]

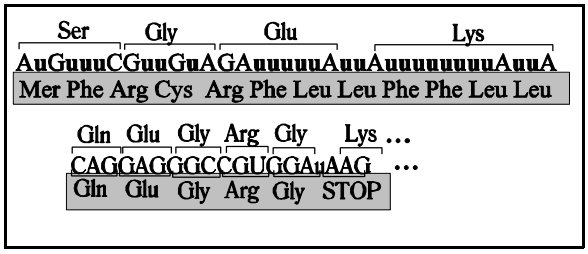

Figure 1: U-insertion in Trypanosomes

But even if the precise mechanisms of RNA editing are not yet know, its importance is unquestionable, since it has the power to dramatically alter gene expression: "cells with different mixes of [editing mechanisms] may edit a transcript from the same gene differently, thereby making different proteins from the same opened gene." [26, page 78] (one-to-many relations). It is important to retain that a mRNA can be edited in different degrees precisely according to the concentrations of editing operators it encounters. Thus, at the same time, several different proteins coded by the same gene may coexist, if all (or some) of the mRNA's obtained from the same gene, but edited differently, are meaningful to the translation mechanism.

If the concentrations of editing operators can be linked to environmental contexts, the concentrations of different proteins obtained may be selected accordingly, and thus evolve a system which is able to respond to environmental changes without changes in the major part of its genetic information (genome size optimization). One gene, different contexts, different proteins. This may be precisely what the trypanosome parasites have achieved: control over gene expression during different parts of their complex life cycles.

> "Space is clearly not a problem for mammalian nuclear DNA, so the [previous] rationale is not so obvious for the [editing mechanisms of mammals]. Also there, however, we see one gene encoding two proteins. In mammalian genomes, gene duplication followed by separate evolution of the two copies would be a more obvious way of producing closely related proteins in regulatable amounts. RNA editing, however, does provide the opportunity to introduce highly specific, local changes into only some of the molecules. [...] It could be reasoned that somehow this would be more difficult to achieve via gene duplication, since independently accumulating mutations would make it harder to keep the remainder of the two sequences identical" [3, p. 22]

Thus, RNA editing may be more than just a system responsible for the introduction of uncertainty (one-to-many relations), but also, and paradoxically, a system that may allow the evolution of different proteins constrained by the same genetic string. In other words, even though one gene may produce different mRNA's (and thus proteins), the latter are not allowed heritable variation since they are always constrained by the gene from which they are edited, and which is ultimately selected and transmitted to the

offspring of the organism. We can see RNA Editing, especially in the case of gRNA's, as a case of co-adaption of two distinct systems: the stored genetic information (e.g. maxicircles) and the contextual editors (e.g. minicircles), also stored in DNA, but independent and meaningless to the larger semantic loop of the genetic code.

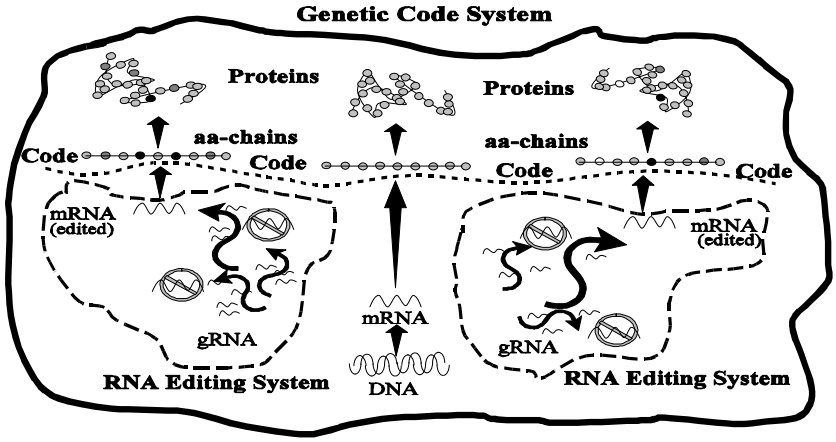

Figure 2: Co-adaptation of the RNA Editing and Genetic Code Systems

The dependent evolution of one gene and several contexts, as expressed by Rob Benne in the previous quote, may allow the introduction of highly specific, local (contextual) changes, more effectively than the independent evolution of several genes. If all of the different expressions were allowed different genes, they would evolve separately not only increasing the size of the genome, but also, possibly, making it harder to maintain coherent, multicellular, phenotypes as well as coherent developmental processes. For instance, the editing of several genes of the *Trypanosoma Brucei* is developmentally regulated [36] which may be of evolutionary advantage for these parasites [35]. Though in the course of evolution editing was partially or completely eliminated in many lineages of eukaryotic organisms containing mithocondria, by reverse transcription of partially edited mRNA's, it may be useful for the development of parasitic adaptations as is the case of the developmental regulation of editing in *T. Brucei* [35].

The role of RNA editing in the development of multicellular organisms has also been shown to be important, Lomeli at al [21] have discovered that the extent of RNA editing affecting a type of receptor channels responsible for the mediation of excitatory postsynaptic currents in the central nervous system, increases in rat brain development. As a consequence, the kinetic aspects of these channels will differ according to the time of their creation in the brain's developmental process.

3. A Theoretical Model: Evolving Semiotics

Semiotics concerns the study of signs/symbols in three basic dimensions: syntactics (rule-based operations between signs within the sign system), semantics (relationship

between signs and the world external to the sign system), and pragmatics (evaluation of the sign system regarding the goals of their users) [23]. The importance of this triadic relationship in any sign system has been repeatedly stressed by many in the context of biology and genetics [39; 24, 25; 26]; in particular, Peter Cariani [6] has presented an excellent discussion of the subject. We can understand the semiotics of the genetic system if we consider all processes taking place before translation (from transcription to RNA editing) as the set of syntactic operations; the relation between mRNA (signifier) and folded amino acid chains (signified), through the genetic code, as the implementation of a semantic relation; and finally, the selective pressures on the obtained proteins as the pragmatic evaluation of the genetic sign system. Jon Umerez [38] has discussed the importance of the code in the establishment of this genetic semiotics by developing, in the context of Artificial Life, Howard Pattee's notion of semantic closure [24,25]: the idea that only organisms capable of controlling their own syntactic operations and semantic relations are capable of open-ended functional creativity or evolution. Natural selection defines the pragmatic evaluations imposed on evolving semantically closed organisms.

Until now, the semiotics of DNA has been considered strictly unidirectional: DNA stands for proteins to be constructed. In other words, the symbolic DNA encodes (through the genetic code) actions to be performed on some environment. Naturally, through variation and natural selection (pragmatic evaluations) new semantic relations are created which are better adapted to a particular environment, however, real-time contextual measurements are not allowed by this unidirectional semiotics. If in addition to symbols standing for actions to be performed, the genetic sign system is also allowed a second type of symbols standing for environmental, contextual, measurements, then a richer semiotics can be created which may have selective advantage in rapidly changing environments, or in complicated, context dependent, developmental processes.

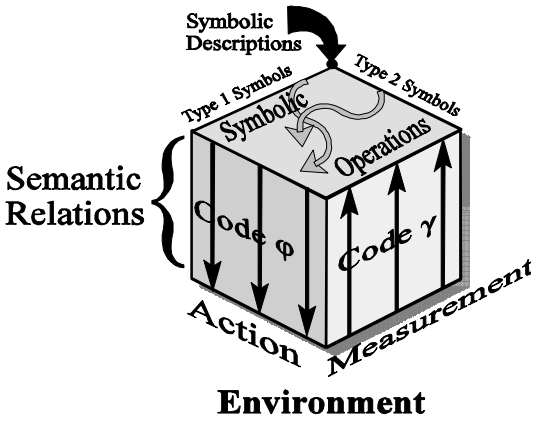

Figure 3: DNA Semiotics with two symbol types

Figure 3 depicts such a sign system. The top plane contains two different types of symbols which are combined in different ways (symbolic operations). *Type 1* symbols stand for actions through a *code f* (e.g. the genetic code) and *type 2* symbols stand for measurements through a different *code ?* which is being hypothesized here. The evidence presented in section 2 refers to genetic systems in which RNA Editing is used in different amounts according to different contexts (namely, different stages of a developmental process). We can think of DNA as a set of symbolic descriptions based on two types of symbols: *type 1* symbols will be expressed in mRNA molecules and will

stand for actions to be performed; *type 2* symbols will be expressed in gRNA molecules (or other editing mechanisms) and will stand for contextual observables. RNA editing can be seen as a set of symbolic operations performed with symbols of both types, resulting in symbols of *type 1* to be translated into actions by the genetic code.

Notice that *code ?* is proposed here as an abstraction referring to the set of mechanisms which will vary the concentration of editing agents *(type 2* symbols) according to environmental context. It is **not** expected to function as a proper genetic code. This issue has been dealt with in [29, 30] in the context of evolutionary systems and second order cybernetics.

4. Artificial Genetic Editing

In GA's, genes are substituted by strings of symbols taken from a binary vocabulary $\boldsymbol{V = \{0, 1\}}$ and called ***V-strings***. The genotype of an individual, referred to as its *symbolic description*, is the set of *V*-strings necessary to produce a phenotype or *solution alternative* [12]. The translation of symbolic descriptions into the space of solutions is performed by invariant formal rules which define a code for a particular application. In the following, a symbolic description is comprised of only one *V*-string.

Definition 1. *V* is a vocabulary with two symbols: $V = \{0,1\}$.

Definition 2. *S* is a *V*-string of dimension n : $S = s_1s_2s_3 \ldots s_n$, $s_i \in V$, $I = 1, 2, ..., n$. Let $\boldsymbol{S}^n$ denote the power set of *V*-strings of dimension *n*.

Definition 3. $\boldsymbol{P}(g) = \{S_i * I = 1, ...,n_p\}$, is a population of n_p *V*-strings at generation *g*.

Definition 4. $\boldsymbol{X} = X_1 \times X_2 \times ... \times X_d$ is a space of solutions, of dimension *d*, for a particular problem. X_i is the universal set of a relevant variable x_i, $I = 1, 2, ..., d$. f maps *V*-Strings *S* into solution alternatives $\boldsymbol{x}$. $f : \boldsymbol{S}^n \to \boldsymbol{X} * f(S) = \boldsymbol{x} \in \boldsymbol{X}$. This mapping establishes the translation rules between symbolic descriptions and solution alternatives: the code.

An individual is composed of a symbolic description, $S \in \boldsymbol{S}^n$, and a solution alternative, $x \in \boldsymbol{X}$. But the relation between *S* and *x* is not a result of direct application of the mapping f . Before *S* is translated into the space of solutions, it will possibly be altered through interaction with a different sort of string.

Definition 5. *U* is a vocabulary with three symbols: $U = \{0,1,(\}$.

Definition 6. *E* is a string of length *m* over the vocabulary *U*, or a *U*-string of dimension *m*:$E = e_1e_2e_3 \ldots e_m$, $e_i \in U$, $I = 1, 2, ..., m$. Let $\boldsymbol{E}^m$ denote the power set of *U*-strings of dimension *m*.

These *U*-strings will function as the editing agents of the population of *V*-strings. The length of *U*-strings is supposed much smaller than that of the *V*-strings: $m << n$, usually an order of magnitude. Maintaining the analogy with the RNA editing system of the Trypanosomes, *V*-strings can be referred to as *maxistrings*, and *U*-strings as *ministrings*. Here I will assume that the editing agents are constant, that is, the structure of the ministrings will be maintained through the successive generations of $\boldsymbol{P}$.

Definition 7. Let $\tilde{o}$ denote a finite family (ordered set) of *l U*-strings: $\tilde{o} = \{E_1, ..., E_l\}$.

Definition 8. For each family of *U*-strings, $\tilde{o}$, there exists an associated family of mappings $\ddot{o} = \{f_1, f_2, ..., f_l\}$. Each mapping f_i associates its respective *U*-string in $\tilde{o}$ with

a *V*-string, and produces another *V*-string: f_i: $\boldsymbol{E}^m \times \boldsymbol{S}^n \to \boldsymbol{S}^n$. The associated pair ($\tilde{o}$, $\ddot{o}$) = $\{(E_1, f_1), (E_2, f_2), ..., (E_l, f_l)\}$ is called a family of editors.

In other words, each editing ministring will have a function which is also dependent on the maxistring to be edited. This function will result in an edited maxistring, and thus specifies how a particular ministring edits maxistrings: when the ministrings match a portion of a maxistring, a number of symbols from the *V* vocabulary is inserted into or deleted from the (*V*-)maxistring. To introduce the sort of ambiguity the guanine-uracil base pairing allows the gRNA/mRNA duplex, the *U* includes an extra symbol '(', matching both '1' or '0' in *V*. Ministrings match more than one subsequence of maxistrings.

Definition 9. A *U*-string $E \in \boldsymbol{E}^m$, *matches* a substring, of size *m*, of a *V*-string, $S \in \boldsymbol{S}^n$, at position *k* if:

$$\rho_{k*1\#k\#n}: \begin{cases} s_{k\%i}=1 \text{ and } e_i=(1 \circ *) \\ s_{k\%i}=0 \text{ and } e_i=(0 \circ *) \end{cases} \forall_{i = 1,2,...,m}$$

Example of a family of mappings $\boldsymbol{f}: \boldsymbol{E}^m \times \boldsymbol{S}^n \to \boldsymbol{S}^n$. $\ddot{o}$ = {*Add_1*(*E*, *S*), *Del_1*(*E*, *S*}}. ***Add_1*** will add the symbol '1' at position *k*+*m*+1 if *E* matches *S* at position *k*; all string symbols in *S* from position *k*+*m*+1 to *n*-1 are shifted one position to the right (the symbol at position *n* is lost). ***Del_1*** will instead delete the symbol '1', if it is present at position *k*+*m*+1 when *E* matches *S* at position *k*; the string symbols are shifted in the inverse direction (the symbol at position *n* is randomly selected from *V*).

Definition 10. Let the *concentration* of a family of editors ($\tilde{o}$, $\ddot{o}$) be defined by $\div$ = $\{v_1, v_2, ..., v_l\}$, where v_i represents the average number of editors (E_i, f_i) per *V*-string of a population $\boldsymbol{P}$. If n_p is the number of *V*-strings in $\boldsymbol{P}$, then there will be $v_i \cdot n_p$ editors (E_i, f_i) randomly distributed by the n_p *V*-strings of $\boldsymbol{P}(g)$.

Figure 4 shows the operational layout of this genetic algorithm with string editing. Generally, we have a population $\boldsymbol{P}$ of n_p maxistrings, and a family of l editors with different concentrations. Before the maxistrings can be translated into the space of solutions $\boldsymbol{X}$, by the mapping f , they must "pass" through successive layers of editors, present in different concentrations. At each generation, the same number of editors (given by the concentrations) is randomly distributed over these layers. Thus, in the example of figure 4, editor 1 (E_1, *Add_1*) with a concentration of 0.5, will have $n_p/2$ copies of itself randomly distributed by the n_p positions of its layer; there will be on average 0.5 of such editors 'meeting' each maxistring. When an editor meets a

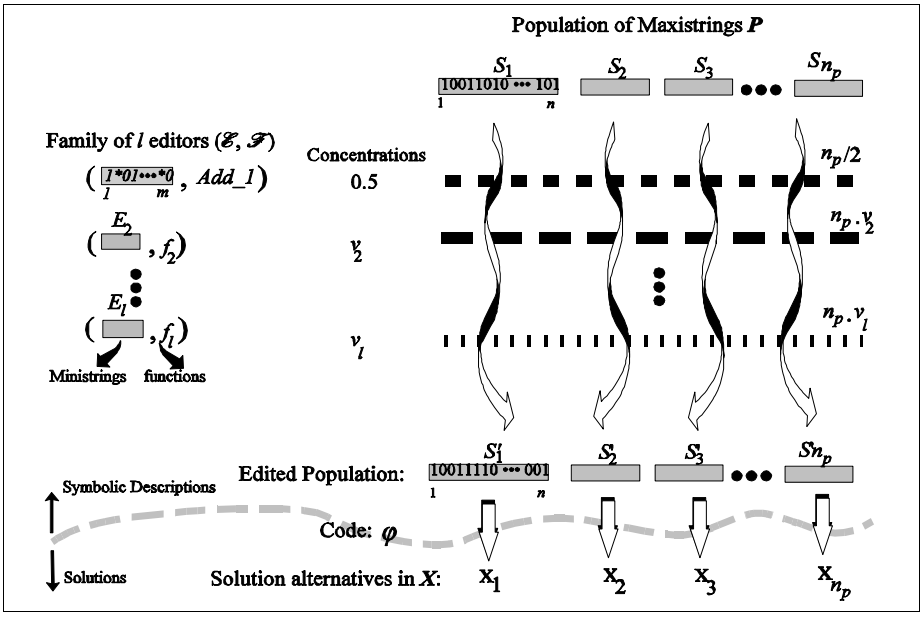

Figure 4: String Editing in a Genetic Algorithm

maxistring, and its ministring matches some subsequence of the maxistring, the editor's function is applied and the maxistring is altered.

5. Context and Development

5.1 Context

In biological genetic systems, RNA editing regulates gene expression; somehow, organisms have used the edition of mRNA molecules to their advantage, perhaps by linking it to environmental context. If a particular external event has the effect of changing the concentrations of editing agents in some genetic system, then those genes which are able to produce fit phenotypes in the different contexts will be selected. Notice that changing environmental context will not merely affect the concentration of editing agents, but also, potentially, the fitness landscape of the genetic system. Thus, the ability

to link changes in the environment with internal parameters such as concentrations of editing agents, gives organisms an adaptive advantage as gene expression can become contextually regulated. The idea is the introduction of the second kind of semantic relation leading to a second type of symbol described in section 3. The editing strings are now more than symbolic constraints, but are also semantically related to context variation through a (postulated) code ?.

Figure 5 shows precisely this kind of coupling between environmental context and the regulating effects of editor concentrations. Notice, at the bottom of the figure, the dependence of the fitness landscape of the solution alternative space ***X***, on environmental context. When the context changes, not only are the symbolic descriptions edited differently, but the solution alternatives are also evaluated differently. The inclusion of this extra level of semantic relations and pragmatic evaluations establishes the kind of genetic semiotics described in section 3.

Consider now two sets of concentrations $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ of our family of editors ($\tilde{\mathbf{o}}$, $\ddot{\mathbf{o}}$) linked respectively to two evaluation functions, *fitness1* and *fitness2*. When the first context is at play, we obtain a population of solution alternatives $\mathbf{X}_1$ which will be evaluated by *fitness1*; alternatively, when the second context is at play, $\mathbf{X}_2$ is evaluated by *fitness2*. Notice that both $\mathbf{X}_1$ and $\mathbf{X}_2$ are produced from the same population of symbolic descriptions ***P***. Those symbolic descriptions in ***P*** which tend to produce fit solution alternatives in $\mathbf{X}_1$ and $\mathbf{X}_2$ (evaluated by *fitness1* and *fitness2* respectively) will have a higher probability of being selected. This result will of course be dependent on the timing and sequence of application of contexts: if contexts are alternated rapidly, then it will be possible to have symbolic descriptions, with a high probability of selection in the population, which produce fit solutions in only one of the contexts; if contexts are maintained a bit longer before alternating, those symbolic descriptions that tend to produce fit solutions in both contexts will have a higher probability of selection; if the contexts are maintained too long, however, it will be more difficult to evolve symbolic descriptions able to survive in both contexts. These results follow Richard Levins [20, chapter 2] strategies of adaptation.

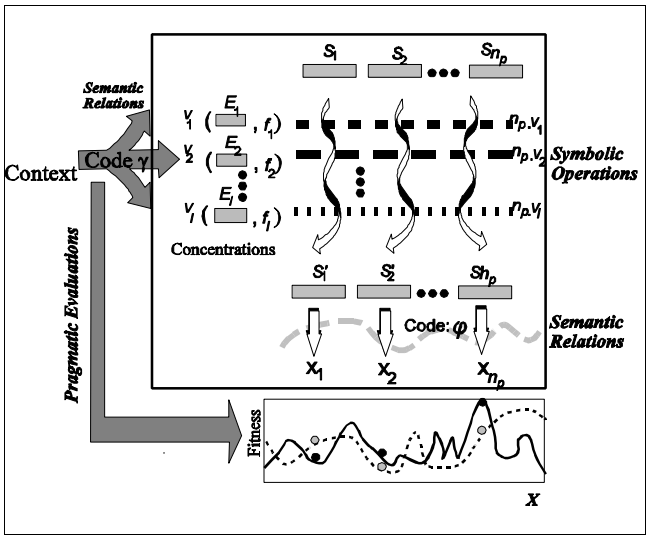

Figure 5: GA with editing parameters linked to context

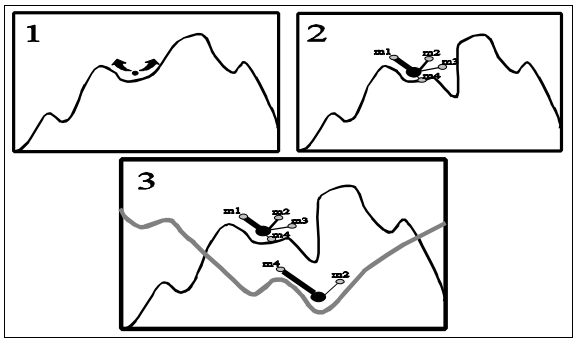

Figure 6: GA search (1); GA with edition search (2); Search in GA with edition linked to context (3)

Figure 6 shows the different searches of traditional GA's, GA's with edition, and GA's with edition linked to environmental context. In the first case, one solution alternative, directly obtained from a symbolic description, is evaluated in a fitness landscape. In the second case, a set of possible solution alternatives, where the dark spot represents the solution obtained from a symbolic description with no edition, and the lighter spots, connected to the center one by links with varying thickness, portraying the relative probabilities of certain edited solutions in a particular concentration of editing agents, represent all the possible solutions obtained by edition of the primitive symbolic description, is evaluated in a fitness landscape. In the third case there are two fitness landscapes evaluating the **different** clouds of edited solutions.

5.2 Development

Development refers to those processes taking over an organism once it is reproduced and which are responsible for the transformation of its form. Generally, artificial life models of development are based on Stewart Wilson's [40] ideas: a GA will encode "a *production system program* (*PSP*) consisting of a finite number of production (condition-action) rules [...] of the form: $X + K_i \Upsilon K_j K_k$. The *K*'s stand for cell phenotypes and *X* represents the local context". [40, page 159]. Basically, the symbolic descriptions of the GA code for a population of "mother cells", or "eggs". These "eggs" code for a specific PSP (a set of production rules) dictating how the "cell" develops into some multicellular aggregate, which is then evaluated for its fitness. The more fit aggregates will have the symbolic description of its "egg" reproducing with a larger probability in the population. These ideas have been used mostly to generate neural networks [16; 1; 13] or more generally sensorimotor control systems [for a good overview see 15]. Recently, the idea of encoding metabolic cycles in a genetic algorithm [17], represented by boolean networks for instance [11], which will then in certain conditions effect developmental steps has also been proposed. This approach aims at an increasing self-organization of the developmental PSP's.

Developmental cycles have been argued to offer an expanded universe of solution alternatives, that is, rather than precisely encoding a fixed number of parameters, more general rules are encoded which will themselves organize, and search a larger universe of alternatives. Thus development cycles come as a necessary solution for design problems affected by scaling constraints (such as neural networks). By the same token, we can expect developmental cycles in artificial life models to come up with more complicated morphologies arising through the interaction of several developmental rules

(PSP's) rather than direct encoding. Basically, the evolutionary advantage of these PSP's is the definition of a smaller search space which is then amplified through development into more complicated morphologies. We can also think that this reduced search space is more amenable for evolution since lower dimensionality spaces will have more valleys; if more morphology details have to be encoded then dimensionality is increased and the search becomes more difficult (see [9] for a discussion of these topis). Related to this is Conrad's tradeoff principle between structural programmability and evolvability [7, 10].

The several approaches vary in many ways, for instance, on the degree of context allowed in production rules of the various PSP's (e.g. how rules are applied depending on a cell's neighbors). Nevertheless, in all of them, the symbolic description-solution space relation is always certain. The production rules are primitives of the representational system and encoded in a one-to-one manner in the symbolic descriptions of the genetic algorithm. The more self-organizing approaches of Kitano [17] and Dellaert and Beer [11], seem to offer a way out for this one-to-one correspondence, but the wiring of the boolean networks (or metabolic cycles) is still encoded in the genetic algorithm in a one-to-one manner. The metabolic networks will then reach some state corresponding to a particular developmental rule; however, this correspondence, established by a **second** set of semantic relations, a simulation code (see section 6), is also completely certain. These systems are very powerful, and offer very interesting and sound approaches to modeling developmental cycles in artificial organisms, however, they do not aim at the understanding of how and why developmental stages arise in the first place through internal regulation of genetic expression.

If the editing system above is able to evolve developmental stages triggered by the internal control of the expression of symbolic descriptions, then we are moving towards utilizing the principle of natural selection not only at the level of the individual, but also at levels internal to the individual, namely through the evolution of semantic referents, for contextual information, in the genetic system.

> "More specific to GA's is the central question of representation. [...] The choice of system primitives (in the case of GA's, the features that comprise the genotype) is a decision that cannot be automated." [22, page 281]

The direct engineering of a relationship between descriptions and solutions allows only what Peter Cariani [5] has referred to as syntactic emergence, that is, the inability of a formal system to change its primitives and create new observables, and therefore respond with open-ended evolution. The kind of automation that Mitchell and Forrest refer above, would amount to the evolution of the semantic relationship between symbolic descriptions and solution alternatives itself, the representation issues above, and would therefore shed some light on the problem of the origin of symbols.

This is not what is pursued here, the direct semantic relationship of the GA will be given by the mapping f which is predefined from the beginning. The choice of primitives for this mapping, the code, is permanent. What can be utilized as a source of contextual input, is the editing system of the GA's presented above. Remember once again that this system is independent of code f, and is therefore only taking place at the syntactic level (symbolic descriptions) of the GA. However, the symbolic descriptions can be made to evolve with the editing constraints, tied up to environmental context, which become referents for this context. In other words, the aim is to evolve the

contextual semantic relations for type 2 symbols described in section 3. Thus, it is possible to evolve context referents for the rules of a PSP, rather than predefine them from the start, provided different sets of concentrations of editors are linked to different fitness functions. Also, since the solutions of the same symbolic descriptions in the various contexts are not allowed independent evolution, as only the "mother descriptions" are reproduced, the evolved rules will be more related than if evolved independently (with distinct descriptions), and have therefore the potential to evolve more coherent PSP's with shorter symbolic descriptions.

6. Physical Simulations and Fuzzy Developmental Rules

In artificial life, it is important to distinguish between the code of the GA (the mapping between symbolic descriptions and solution alternatives, f , or genetic code) and the code of a simulation. The latter refers to all the physical characteristics the modeler attributes to the solution alternatives of his or hers simulation. It is important to realize that this code is external to the GA and does not affect its search. Often, these distinctions are blurred in artificial life and evolutionary computation precisely because traditional GA's, due to their one-to-one mapping between symbolic descriptions and solution alternatives, do not distinguish between the two, or metaphorically, do not distinguish genotype from phenotype.

Naturally, in a computational realm all material aspects must be simulated and therefore a semantic relation is imposed which refers the simulation's symbols to the physical characteristics we desire to model. It is important to keep this in mind especially in the simulation of developmental cycles since these are defined on two stages: first the GA searches for a particular developmental program, and then this program is executed. The first stage depends on the GA's code (f), independently from the physical attributes of the simulation, while the second stage executes the program according to some simulated physics defined by the simulation code, from now on referred to as code ß.

If we are to utilize contextual GA's to tackle the problem of development, the primitives of the solution alternatives obtained should naturally code for all the characteristics needed to form the rules of a PSP, namely, phenotypic characteristics such as "cell thickness" as well as orders such as "divide in two", etc. However, there will be no coding of rules themselves, in particular, the context in which a rule should be applied, will not be a semantic primitive, but allowed to evolve from the coupling of the editing system of the GA to the external contexts. This is by no means achieved, or easy to achieve, it indicates a proposed research direction necessary to attain true evolution of development cycles in artificial life models.

6.1 Fuzzy Sets as Uncertain Physical States

Fuzzy sets may be ideal mathematical structures to characterize some simulated physical dynamics. For instance, the stable states of metabolic networks used for the definition of developmental cycles referred above [17, 11] can be represented by a fuzzy set in which the nodes of the network and their activation states are the elements of the set and their membership degrees respectively. More generally, the elements of a fuzzy set can refer to some desired physical attributes (through the simulation code ß) while their membership degrees can describe the degree to which such physical attributes are

present in a certain situation. In the context of developmental cycles, certain actions will be taken when certain elements have membership degree beyond a specified value.

To allow for a better representation of uncertainty, that is, if we desire the physical characteristics to observe in addition to fuzziness the two other recognized forms of uncertainty — nonspecificity and conflict [18] — then a more complicated set structure can be used. This structure is referred to as an *Evidence Set* [28, 31,32] and is based on the extension of fuzzy sets by utilizing *Evidence Theory* [34]. Basically, this structure formalizes the membership degree of an element in a set, with a finite number of weighted subintervals of [0,1]. A degree of membership in [0, 1] captures uncertainty in the form of *fuzziness*, an interval of membership introduces *nonspecificity*, and finally several competing intervals introduce *conflict*. The measurement of uncertainty in set structures is discussed in [31].

Evidence sets can be obtained through the operation of simpler fuzzy sets. Several operations for evidence sets have been defined in [28, 32]. Consider now a string of fuzzy sets, defined on some universal set *K*, and operations amongst them together with parenthesis which group the operations in the string in different ways:
$S = F_1$ q ($(F_2$ u $F_3)$ þ F_{n-2}) s F_{n-1} r F_n. Consider further that these fuzzy sets, F_i, are picked from a finite, small, family of possible fuzzy set shapes, and the operations are likewise picked from a small family of operations. Finally, a number of parenthesis is somehow randomly distributed over the string. Once a string is generated, it must be parsed in order to obtain an evidence set: parenthesis will have to be matched and operations performed. If a right (left) parenthesis is not matched all the fuzzy sets and operations to its left (right) are discarded. Thus, from an original string with *n* fuzzy sets, after parsing, we will obtain strings with 0 to *n* fuzzy sets.

Returning to our GA's, consider now that the edited strings obtained will code (through f) to such a string of fuzzy sets and operations. In other words, the solution alternatives of the GA will be fuzzy set strings which will be parsed and operated into evidence sets whose elements (of *K*) refer to some simulated physics through code ß. Since fuzzy sets capture only one form of uncertainty (fuzziness) and evidence sets capture three (fuzziness, nonspecificity, and conflict), we can metaphorically say that the fuzzy set strings "fold" from a one dimensional into a three dimensional uncertainty state. Figure 7 presents a scheme of this process.

To make things more general, the fuzzy sets, F_i , define only shapes of membership as seen in figure 7. These shapes are then positioned and stretched over some pre-defined portions of the universal set *K*. This is a very important point since it eliminates any scaling problem of whatever physical attributes we wish to simulate. To explain this better, I must be a bit more formal. Consider that the universal set *K* of our fuzzy sets is divided into octants (eight portions of *K*). A fuzzy set shape can now be associated with a particular octant as well as with some width stretching over a number of octants. If we have eight possible fuzzy set shapes we only need 3 bits of information to express the shape, plus 3 bits to position it in an octant, and finally 2 extra bits can specify 4 possible widths for stretching the shape over *K*. This way, a fuzzy set can be specified by only 8 bits: 1 byte. Likewise for the fuzzy operations and parenthesis. 8 different operations are possible (3 bits). If we specify that an operation will carry with it a left or a right parenthesis one fourth of the time, we need 2 bits for each parenthesis (4 bits). With an

unused bit, an operation with parenthesis can be specified by one byte. A string with 8 fuzzy sets and 8 operations can be described by 16 bytes (128 bit long string), for **any** finite cardinality universal set *K*.

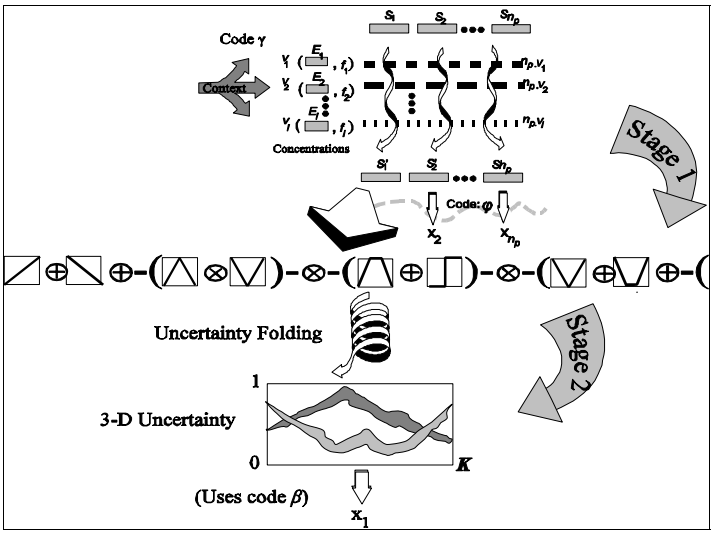

Figure 7: Two stage contextual GA coding for 3-D uncertainty fuzzy rules

Hence, the definition of the solution alternatives of the contextual GA in terms of fuzzy set strings is independent of the size of a particular physical simulation, that is, of the number of physical characteristics of our artificial organisms. Whatever the number of these characteristics, whatever the cardinality of *K*, the search space of the GA will be the same, namely, the one defined by the 128 bit long strings coding (through f) into fuzzy set strings. Nevertheless, and naturally, the size of *K* is relevant for other aspects of the simulation external to the GA. A larger *K*, will mean that the definition of an organism (by an evidence set) will require a finer tuning of the composing fuzzy set string, which may take longer for the GA to reach. In any case, the search space will remain constant, only more elaborate searches will be required.

So far the fuzzy set strings have been shown to increase the uncertainty description of a simulation, as well as to allow for a good scaling management. But they possess yet another important evolutionary advantage: a buffering mechanism for genetic mutation. Michael Conrad [1990] has developed the notion of genetic buffering as an important requirement for evolvability. Though mutation is required for evolution, it is also important that certain shapes may be resilient to changes which may potentially destroy an important physical functionality. As discussed above, the fuzzy set strings will be parsed according to its parenthesis. Consider the following parsing situation:

$$F_1 \text{¿} \quad F_2 \text{½} \quad (F_3 \text{Á} \quad \ldots F_7 \text{Ù} \quad F_8 \; \mu \; F_1 \text{¿} \quad F_2$$

all the fuzzy sets and operations to the right of the unmatched left parenthesis are discarded. This means that any bit to the left of the second fuzzy set is free to mutate without any effect on the final organism, except those few bits which may cause a matching parenthesis to occur to the left of F_2.

As a final note, crossover was not considered in this model precisely to not disrupt this kind of genetic buffering. Also, since eventually this kind of buffering is

transcended, usually with a dramatic change of form (a string with two functional fuzzy sets can suddenly become a string with, say, eight functional fuzzy sets), crossover seems to be unnecessary as a source of more variety.

7. Final Remarks

The most important characteristic of all the mechanisms here introduced is related to a conflict between introducing more variety and constraining, or buffering, this variety. Contextual editing allows for a variety, a cloud, of solution alternatives to effect the genetic search, however, this variety of alternatives is not allowed independent evolution and is constrained to an original symbolic description (which can be edited in different ways) ultimately reproduced. On another level, the uncertain fuzzy set strings, though introducing a large amount of variety in their parsing and uncertainty folding, also observe the kind of genetic buffering described earlier. It is believed that the right amounts of variety and constraint lie at the core of evolvability [8,9]. Only the implementation and testing of the proposed model will tell if it has the right amount of both. In addition, the inclusion of context in the genetic algorithm, or the limited evolution of a semantic relationship between editing mechanisms and contexts (the type 2 symbols in a semiotic relation), opens the way for the study of the emergence of developmental cycles triggered by contextual constraints.

References

1. Belew, R.K. [1993]."Interposing an Ontogenic Model Between Genetic Algorithms and Neural Networks." In: *Adv. in neural information processing*. J. Cowan (Ed.). Morgan Kaufmann.
2. Benne, R. Et al ."Major transcript of the frameshifted coxII gene from trypanosome mitochondria contains four nucleotides that are not encoded in the DNA." *Cell* N. 46, 819-826
3. Benne, Rob (Ed.) [1993]. *RNA Editing: The Alteration of Protein Coding Sequences of RNA*. Ellis Horwood.
4. Blum, B, N. Bakalara, and L. Simpson [1990]."A model for RNA editing in kinetoplast mitochondria: "guide" RNA molecules transcribed from manicircle DNA provide the edited information." In: *Cell* No. 60, pp. 189-198.
5. Cariani, Peter [1991]."Some Epistemological Implications of Devices which Construct their own Sensors and Effectors." In: *Towards a Practice of Autonomous Systems, Proc. First European Workshop on Artificial Life*. F. Varela and P. Bourgine (Eds.). MIT Press. pp 484-493.
6. Cariani, Peter [1995]."Towards an evolutionary semiotics: the role of symbols in organisms and adaptive devices." In: *Proceedings of the International Seminar on Evolutionary Systems, Vienna 1995*. Stanley Salthe and Gertrudis Van de Vijver (eds). In Press.
7. Conrad, Michael [1974]."The limits of biological simulation." *J. The. Biology* Vol. 45, 585-590.
8. Conrad, Michael [1983]. *Adaptability*. Plenum Press.
9. Conrad, Michael [1990]."The geometry of evolutions." In: *BioSystems* Vol. 24, pp. 61-81.
10. Conrad, Michael [1993]."Adaptability theory as a guide for interfacing computers and human society." In: *Systems Research* Vol. 10, No. 4, pp. 3-23.
11. Dellaert, F. and R.D. Beer [1994]."Toward an evolvable model of development for autonomous agent synthesis." In: *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*. R. Brooks and P. Maes (Eds.).

MIT Press.
12. Goldberg, D. E. [1989]. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.
13. Gruau, Frédéric [1992]."Genetic Sythesis of boolean Neural Networks with a cell rewriting developmental process." In: *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks*. Whitley, L.D. and J.D. Schaffer. IEEE. pp 55-74.
14. Holland, John H. [1975]. *Adaptation in Natural and Artificial Systems.* U. of Michigan Press.
15. Husbands, P., I. Harvey, D. Cliff, and G. Miller [1994]."The use of genetic algorithms for the development of sensorimotor control systems." In: *(retrieved from the Internet)* .
16. Kitano, H. [1990]."Designing Networks using Genetic Algorithms with Graph Generation System." In: *Complex Systems* Vol. 4, pp 461-476.
17. Kitano, Hiroaki [1994]."Evolution of Metabolism for Morphogenesis." In: *Artificial Life IV: proc. 4th international workshop on the synthesis and simulation of living systems*. R. Brooks and P. Maes (Eds.). MIT Press.
18. Klir, George J. [1993]."Developments in uncertainty-based information." In: *Advances in Computers*. M. Yovits (Ed.). Vol. 36, pp 255-332.
19. Langton, Christopher G. [1989]."Artificial Life." In: *Artificial Life: SFI Studies in the Sciences of Complexity*. C.G. Langton (Ed.). Addison-Wesley. pp. 221-249.
20. Levins, Richard [1968]. *Evolution in Changing Environments: Some Theoretical Explorations.* Princeton Univ. Press.
21. Lomeli, H. et al [1994]."Control of kinetic properties of AMPA receptor channels by nuclear RNA Editing." In: *Science* Vol. 266. pp. 1709-1713.
22. Mitchell, M. and S. Forrest [1994]."Genetic algorithms and Artificial Life." In: *Artificial Life* Vol. 1, pp 267-289.
23. Morris, Charles W. [1946]. *Signs, Language, and Behavior.* G. Braziller, New York.
24. Pattee, Howard H. [1982]."Cell psychology: an evolutionary approach to the symbol-matter problem." In: *Cognition and Brain Theory* Vol. 5, no. 4, pp 325-341.
25. Pattee, Howard H. [1995]."Evolving self-seference: matter, symbols, and semantic closure." *Comm. and Cog. - AI* Vol. 12, nos 1-2 pp 9-27.
26. Pollack, R. [1994]. *Signs of Life: The Language and Meanings of DNA.* Houghton Mifflin.
27. Rennie, J. [1993]."DNA's New Twists." In: *Scientific American* March 1993.
28. Rocha, Luis M. [1994]."Cognitive Categorization revisited: extending interval valued fuzzy sets as simulation tools for concept combination." In: *Proc. of the 1994 Int. Conference of NAFIPS/IFIS/NASA*. . IEEE. pp 400-404.
29. Rocha, Luis M. [1995]."Contextual Genetic Algorithms and an Evolutionary Semiotics." In: *Proceedings of the Int. Seminar on Evolutionary Systems, Vienna 1995*. Stanley Salthe and Gertrudis Van de Vijver (eds.). In Press.
30. Rocha, Luis M. [1995]."Eigen-states and symbols." In: *Heinz von Foerster Festschrift (Special Issue)*. Ranulph Glanville (Ed.). *Systems Research* Vol. 12, No 3. (In Press).
31. Rocha, Luis M. [1995]."Computing Uncertainty in Interval Based Sets." In: *Advances in Interval Computation*. Vladik Kreinovich (ed.). Kluwer. (In Press).
32. Rocha, Luis M. [1995]."Evidence (interval based) Sets: Modelling Subjective Categories." In: *International Journal of General Systems* In Press.
33. Seiwert, S. D. and K. Stuart [1994]."RNA Editing: transfer of genetic information from gRNA to precursor mRNA in Vitro." In: *Science* Vol. 266 pp. 114-116.
34. Shafer, Glenn [1976]. *A Mathematical Theory of Evidence*. Princeton Unversity Press.
35. Simpson, L. and D. Maslov [1994]."RNA Editing and the evolution of parasites.". *Science* Vol. 264, pp. 1870-1871.

36. Stuart, K. [1993]."RNA Editing in mitochondria of african trypanosomes." In: *RNA Editing: the Alteration of Protein Coding Sequences of RNA*. R. Benne (Ed.). Ellis Horwood. 26-52.
37. Sturn, N.R., and L. Simpson [1990]."Kinetoplast DNA minicircles encode guide RNA's for editing of cytochrome oxidase subunit III mRNA." In: *Cell* No. 61, pp. 879-884.
38. Umerez, Jon [1995]."Semantic Closure: A guiding notion to ground Artificial Life." In: *Proc. of the ECAL 1995*
39. Waddington, C.H. [1972]. *Biology and the History of the Future*. Endinburgh University Press.
40. Wilson, S.W. [1988]."The genetic algorithm and simulated evolution." In: *Artificial Life: SFI Studies in the Sciences of Complexity*. C.G. Langton (Ed.). Addison-Wesley. pp. 157-166.