On the performance of some bioinspired genetic operators in complex structures evolution

ABSTRACT

Indirect encoding methods demand operators that minimize the disruption of conventional operators, widely studied in direct encoding approaches. While some efforts have already been done in this direction, the growing field of Genetics sheds new light on the dynamics of the nucleic acids, and their implications in the evolution of life on Earth. Here we model basic mechanisms of gene duplication and horizontal gene transfer, presenting preliminary results of its application to L-systems evolution. The first interesting finding is that, in the particular simplified framework proposed, most of these operations are only slightly disruptive allowing the structures to evolve without loosing what has been gained in the past. Combining these operators with the traditional point-mutation, insertion and deletion generates interesting dynamics: cycles of genome expansion (duplication / transference) and further tuning (deletion / insertion / mutation) spontaneously emerge. Large populations of L-systems have been evolved to meet simple restrictions on their phenotypic readout. Two case-studies are described: (1) evolution under low selective pressure towards a target aspect, and (2) evolution of a form under changing conditions. Genotypic and phenotypic evolution is discussed, along with a fitness curve that is closer to punctuated equilibrium than to the traditional exponential shape.

Keywords

Indirect encoding, L-systems evolution, bracketed 0L-systems, bioinspired genetic operators, developmental methods, artificial ontogeny.

1. INTRODUCTION

Development is to evolutionary computation what cortical function is to artificial neural networks: a highly complex process that hides some of the deepest secrets of this computational paradigm. While Genetic Algorithms have been extensively developed and are applied in basically any field of engineering, the underlying principles that we find in these methods are not far from their original description by J.H. Holland [3]. The Simple Genetic Algorithm, operating over directly encoded solutions (typically binary numbers), has stood as the flagship for evolutionary computation.

Approaches using indirect (or implicit) encodings have been proposed for many years, but have always lacked the computational power of traditional genetic algorithms. The Genetic Programming paradigm proposed by J.R. Koza [5] is probably the most developed effort to evolve complex objects under an indirect (grammatical) scheme.

One of the main drawbacks, widely discussed in the field, of indirect encoding is the disruption of fitted solutions. Since the map between the genome and the phenotype is complex, small genetic changes are likely to propagate along the developmental process and bring about significant differences in the final outcome. How the genetic machinery of the DNA is configured so as to overcome this evolutionary problem is something that we are only just starting to understand. Crossover-related disruption is clearly solved by the speciation mechanism, a toolbox of methods to search for genetically-affine mates, while mutation is very much restricted by enzymatic error-correction molecules. This is not surprising, since evolved genetic mechanisms for altering or breeding new genetic molecules have gone through the natural selection sieve, optimizing their performance. An intelligent strategy would be to mimic those biological processes that fit well with particular indirect encodings, so we can benefit from both: the expressive power of indirect encoding, and the search capacity of new bioinspired genetic operators. In this spirit, we discuss here the performance of gene duplication and horizontal gene transfer, non-standard genetic operators, when applied to a concrete indirect encoding approach: the evolution of Lindenmayer systems.

L-systems [7] constitute a powerful formal method to encode complex structures. The evolution of L-systems have been studied from many different points of view, from the user-oriented evolution proposed in [9], to the elaborated fitting of bracketed 0L-systems together with their readout system, described in [1]. In general, visual validation is enough to assess the quality of the evolutionary process, with little concern being given to the designing of new strategies or to the performance of traditional operators.

Gene duplication plays an important role in evolution [10, 12]. Duplication of genes, and the consequent generation of paralogous genes, is a sort of silent mutation, in the sense that it is evolutionary neutral, and therefore free from selective pressure. The new material that gene duplication provides could be considered junk DNA, since it simply adds redundant information to the genome, but in essence it is providing a substrate for coding new proteins and functions. Duplication has been modeled before in combination with traditional operators, as in [4], where a non-restricted duplication operator is used.

On the other hand, while sexual recombination has been extensively used and discussed in the field of genetic algorithms, it could be considered a particular case of the most general mechanism of horizontal gene transfer. Organisms, do indeed, exchange genetic material in many different ways. From prokaryotic viral transduction to eukaryotic interchange with parasites, there is evidence well reported that the genomic material can become contaminated. The endosymbiotic theory interprets genetic variation mainly as a result of horizontal transference of nuclear information [8].

In this paper we will focus on the evolution of deterministic 0L-systems. It will be shown how an adequate use of the duplication operator can introduce neutral non-disruptive genetic alterations. This type of change, combined with traditional point mutation, will provide the genetic substrate for evolving new structures, which will help in fitting the individual to the landscape.

2. A SIMPLE INDIRECT ENCODING FRAMEWORK

Lindenmayer systems constitute a well-studied generative method. In the tradition of Chomsky's generative grammars, L-systems proceed by rewriting a string of symbols according to some production rules. We will consider the special case of D0L systems, where rules are not randomly applied, but deterministically. Such a grammar defines a single rule for each nonterminal, and rewrites all symbols in each production.

In the same way that a grammar represents a formal language, an L-system can be understood as an encoding mechanism to produce a complex object. While the rewriting process results in a string for a given derivation length, this string can be graphically interpreted. A drawing method like turtle graphics translates lengthy and incomprehensible chains of symbols into wonderful arborizations or classical fractal objects. According to this framework, we can think of the production rules as the genome of a complex graphical object, and the D0L scheme as the genetic expression mechanism that maps genotypes to phenotypes. Determinism is an important aspect of the framework, since biological genetics operates in a straightforward way, linking one genome to one developmental process, and then, to one particular organism. (Of course, there is an environmental bias in the epigenesis, but the environment is not considered in our model.)

These concepts will be illustrated with a concrete example. In the spirit of Prusinkiewicz and Lindenmayer's seminal work [11], lets assume a single nonterminal (G), corresponding to a line of a given length in the graphical object, and lets define a rule to rewrite it. Since the left side of the rule can only be the symbol G, it is the right-hand side that determines the dynamics of the rewriting process. An L-system like G→G+[G] generates the derivation

G ⇒ G+[G] ⇒ G+[G]+[G+[G]] ⇒

G+[G]+[G+[G]]+[G+[G]+[G+[G]]] ...

which maps to the developmental process depicted in Figure 1.

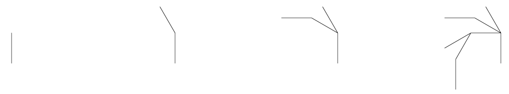

Figure 1. Representation of a developing L-system in a 3-steps derivation (axiom G on the left).

Since only one non-terminal symbol is allowed in the proposed D0L approach, the right-hand side string of the rule that rewrites this symbol will refer to the corresponding L-system, as well as to the structure that is derived after a pre-established number of productions (e.g., the string G+[G] will refer to the L-system G→G+[G] and to the structure generated by this L-system after a derivation of a fixed length, applied to the axiom G; typically the length will be 3, then the structure will be the graph resulting from the interpretation of the string G+[G]+[G+[G]]+[G+[G]+[G+[G]]]).

Considering that the starting direction of the axiom clearly favors verticality (i.e. generation of figures that are higher than wider: aspect ratios greater than 1), flat, horizontally distributed arrangements would be harder to find. Also, the individuals in the initial population will have short genomes, giving rise to poorly branched structures (for a small constant derivation length). Combined, these two objectives shape a non-trivial fitness function to make L-systems evolve; the target in our testbed will be searching for flatty, branched L-systems.

The aspect ratio of an L-system will be defined as the height over the width, relative to the bounding box of the structure that results from a fixed-length derivation of such L-system. Figure 1 shows the structure G−G+[GG+[+G]], obtained in a derivation of length 3, and the corresponding bounding box, with an aspect ratio of 1.69.

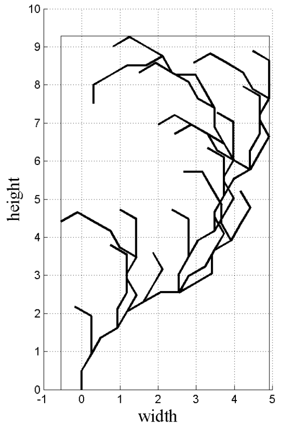

Figure 2. Bounding box of the structure G−G+[GG+[+G]].

The branches of the L-systems are counted as the different segments that are read-out from the derived string, taking into account that long linear branches are made of several segments, and that overlapping segments are counted as one.

With these measures, a shape value will be computed for each individual that integrates how flat and dendritic the structure is. Equation (1) represents this relation:

$$s = b^{-\alpha} \frac{h}{w} \qquad (1)$$

where b is the number of distinguishable branches of a given individual, α is the weight given to the branching, h is the height, and w is the width of the bounding box the L-system lies in. This is to say: the aspect ratio (height over width) divided by the number of branches scaled by a weight. In our simulations $\alpha \in [0,1]$, and values around 0.5 typically resulting in a rapid convergence to complex structures that kept a horizontal distribution. The population of L-systems evolves so as to minimize this shape value, since the width must exceed the height, and the branching is dividing the aspect ratio in equation (1).

A target structure will be specified by a threshold shape value that the structure should meet. Any L-system with a shape value below this threshold will fit the needs, and the search will stop.

3. BIOLOGICALLY-INSPIRED GENETIC OPERATORS

Bracketing in L-systems allows our particular genetic approach the delimitation of areas of expression. Such areas can undergo duplications, resulting in genomes that develop indistinguishable phenotypes, while containing redundant genetic material. These additional copies constitute the genetic substrate for new structures.

Following this strategy, a duplication operator can be defined. Such operator rewrites a given bracketed expression into two concatenated copies of itself. For example, the duplication operator applied to the genome G–G+[GG+[+G]] might result into G–G+[GG+[+G]**[+G]**] or G–G+[GG+[+G]]**[GG+[+G]]** (in bold the duplicated material is represented). These new genomes develop into the same functional structure as the original one, as represented in a 2D graph hence we recall their indistinguishable nature. However, the phenotype of the duplicated one is not identical, since parts of the structure are replicated (but do overlap with the original ones). Future mutations (either alterations or additions) operating on this redundant parts of the genome will provide new structures.

The evolution of well-fitted L-systems in the proposed scenario demands a diversity of redundant genomes, potential progenitors of successful variants. This can be achieved by providing a large population that evolves under low selective pressure.

In our implementation, the selection of progenitors is made by assigning each individual a random number in the range [0, s_i] and selecting the individual with the higher number. s_i is a function of the fitness of the individual (equation (1)) and the overall selective pressure, given by equation (2),

$$s_i = 1 - p\left(1 - \frac{f_i - M}{m - M}\right) \qquad (2)$$

where M is the maximum shape value of the population, m is the minimum value, and $p \in [0,1]$ is the selective pressure (a value 0 makes all the individuals equally eligible, a value 1 keeps the original fitness distribution, and higher values makes the best-fitted individuals more and more eligible).

With this selection method, the next generation is made by selecting an individual of the current generation, and applying to it the genetic operators, that could alter its structure.

Three variants of the duplication mechanism have been modeled, where a bracketed expression (a segment, from now on) is copied:

- Random duplication: the copied segment is placed somewhere in the genome, without restrictions.
- Leveled duplication: the copied segment is placed somewhere in the genome at the same nesting level (...[...[...γ...]...]... → ...[...[...γ...]...]...[...[...γ...]...], where γ is a segment).
- Contiguous duplication: the copied segment concatenates the original one (e.g. ...γ... → ...γγ..., being γ a segment).

Horizontal gene transfer was modeled by allowing a genome to incorporate, in any location, a bracketed expression from another one.

Apart from duplication and gene transfer, three classical operators have been added:

- Insertion: a symbol or an empty segment ([]) is added in a random position.
- Deletion: a symbol or an empty segment is deleted from a random position.
- Point-mutation: a symbol in a random position is replaced by a different one (this operator does not affect bracketing).

The iterative application of these operators to an initial random population of L-systems will be shown to quickly converge to fitted individuals. The particular characteristics of the duplication operators give rise to interesting evolutionary properties, and make them appropriate for this kind of indirect encoding evolution.

4. SIMULATION RESULTS

We have studied the functioning of biological gene duplication and horizontal transfer on the simplified framework described in section 2: evolution of one-symbol bracketed 0L-systems. In what follows we will consider populations as finite languages over {G, –, +, [,]}, where each string represents an L-system (being it the consequent of the rule that defines it). The axiom was always G, structures obtained after derivations of length 3 were considered, and branching influenced only discretely (α = 0.2 – branching does not affect the fitness if α = 0). Simulations have been run over reasonably big populations (300 individuals) and two levels of selective pressure: low (10^{-4}), where different species coexist for hundreds of generations, and high (0.5), where the initial random population rapidly homogenizes into variants of a single species.

4.1 First case-study: searching for fitted structures

The first example shows the evolution of an L-system poorly fitted to the environment in the initial population. As illustrated in Figure 3 (phenotypes) and Table I (genotypes), structures increase their complexity by acquiring and tuning new genes. While the initial structure does not improve for the first nine generations, the rest of the population tries different options: empty bracketed expressions are incorporated by some genomes, which eventually become filled with random symbols inserted into them. These new genes can be transferred, and subsequently duplicated and mutated, in this primordial soup. By generation 10, an ancestor in our genealogical tree collects one of these sequences from another one, to the cost of a significant degradation in its fitness. This horizontal transfer, along with two duplications and a new transfer have significantly extended the genomic material by generation 24. At this point, evolution plays a different game: altering the unused genes that have been conquered. Two point mutations make the shape value shrink to 1/3 of its original value (see figure 3). From that point, another cycle of genome extension takes place (generations 31 to 55), followed by new rewritings: point-mutations and deletions (generations 55 to 75). The target fitness is finally reached by generation 84 (bottom-right structure in Figure 3).

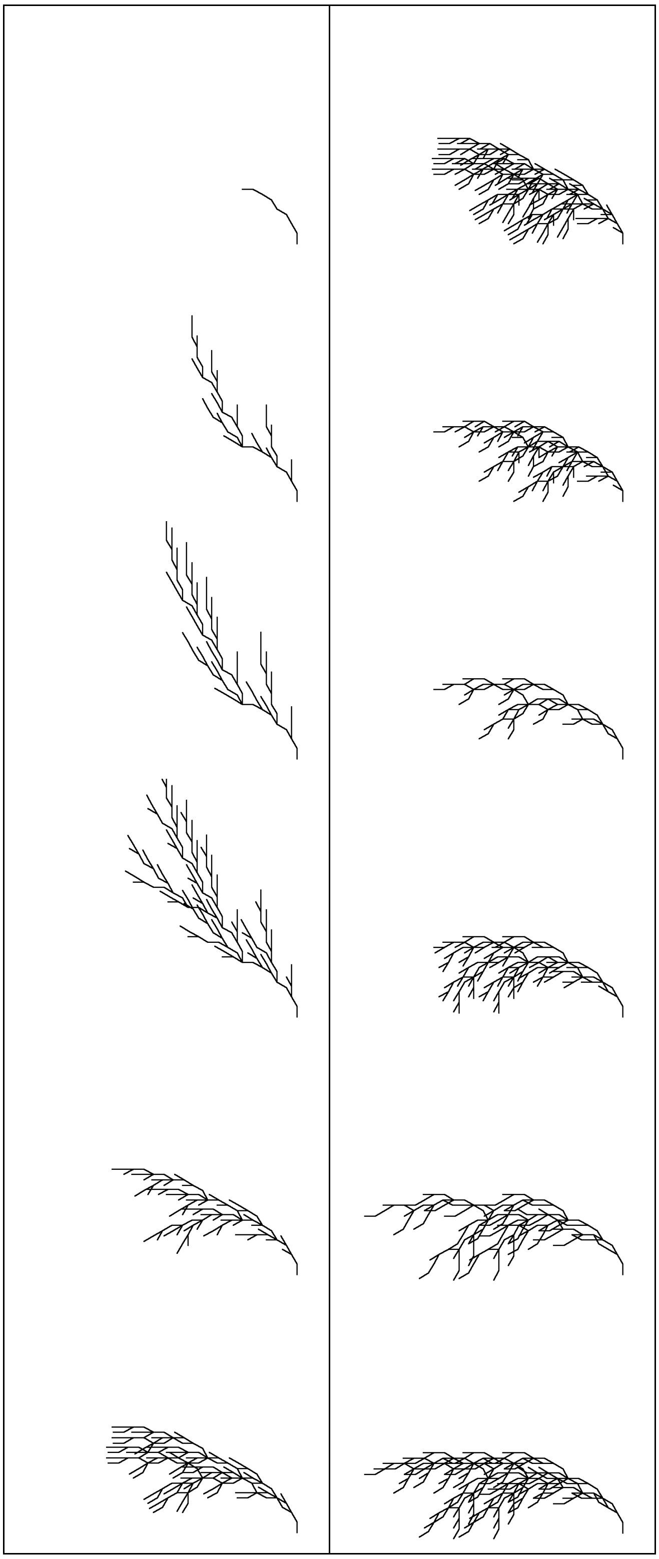

Figure 3. Phenotypes of the genomes in Table I (generations 1, 10, 22, 26, 27, 31, 34, 60, 65, 70, 75 and 84, from top to bottom and from left to right – indistinguishable phenotypes not shown).

Table I. Evolving genotypes of L-systems. Generation (left column), genotype (center) and shape value (right). Bold style indicates duplications, transference and insertion, and symbols inside a rectangle are about being deleted or mutated.

#	genotype	*s*
1	G+G−	.66
10	G+G−**[GG]**	.79
22	G+G−[G**[GG]**G]	.76
23	G+G−[G[GG] **[GG]**G]	.76
24	G+G−[G**[G]** [GG] [$\boxed{G}$G]G]	.76
26	G+G−[$\boxed{G}$[G] [GG] [+G]G]	.53
27	G+G−[+[G] [GG] [+G]G]	.22
31	G+G−[+[G] [GG] [+G**[−G+]**]G]	.19
34	G+**[+G]**G−[+[G] [GG] [+G[−G+]]G]	.18
52	G+[+G]G−[+[G] **[G]** [GG] [+G[−G+]]G]	.18
55	G+[+G]G−[+[G] [$\boxed{G}$] [GG] [+G[−G+] **[−G+]**]G]	.18
57	G+[+G]G−[+[G] [] [$\boxed{G}$G] [+G[−G+] [−G+]]G]	.18
60	G+[$\boxed{+}$G]G−[+[G] [] [G] [+G[−G+] [−G+]]G]	.15
65	G+[GG]G−[+[G] [] [G] [+G[$\boxed{-}$G+] [−G+]]G]	.17
70	G+[GG]G−[+[G] [] [G] [+G[G+] [−G+]]G]	.15
75	G+[GG]G−[+[G] [] [G] [+G[G+]**G**[−G+]]G]	.12
84	G+[GG]G−[+[G] [] [G] [+G[−G+] [G+]G**[−G+]**]G]	.10

As can be appreciated in Figure 4, this evolution has a fitness curve that does not follow the traditional exponential shape, but evokes instead a punctuated equilibrium, with long plateaus and fast transitions.

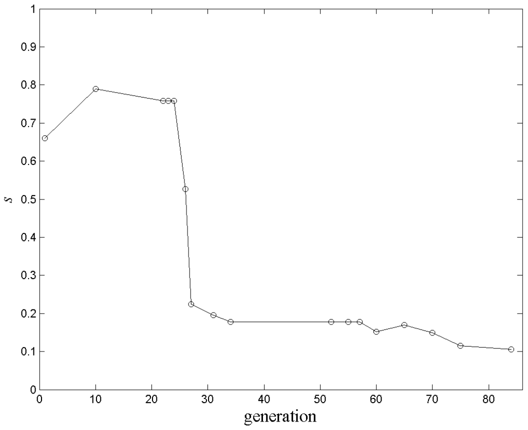

Figure 4. Evolution of the shape value (circles represent the new genotypes shown in Table I). Fitness has significantly improved in two different periods: generations 24 to 27, and 55 to 75.

The evolution shown in this example is prototypical of a big number of evolutions that has been studied. In all these simulations low selective pressure has allowed the population to elaborate a library of genes that can be shared and tested in different organisms. The little disruption introduced by most of the operators is also a key to stabilizing a set of different species in each run.

4.2 Second case-study: fitting in a changing environment

Here we have studied the evolution of an unfitted complex structure (optimized for an aspect ratio close to 1) to a form that is optimized for flatty and branchy arrangements (generations 1 to 15), and its fast adaptation (generations 16 to 26) to an environment where high and narrow structures are favored. Since the initial population was homogeneous (no variability), it made no sense to expect speciation, then a high selective pressure was applied to speed up the simulation. Also, since the richness of possible genes to be transferred was small, a horizontal transfer was not used, as duplication have the same effect in this particular scenario.

The first row in Table II shows the original genome that is spread out along the initial population. In the second generation the genome has expanded with a random duplication (new genes in bold), and this same gene has been selected to be copied randomly in the fourth and fifth generations. Again, once the genome has reached a size that allows variability, deletions and point-mutations do the rest of the work: from generation 6 to generation 15, only small mutations take place, tuning the structure in the favored direction. Finally, the evolution meets the target shape value (0.07).

Having reached this point, the surprising fact is that an abrupt change in the fitness landscape was quickly solved by small modifications of the genome: deletions (mainly) and insertions. This rapid adaptation can be appreciated in Figure 5 (third and fourth rows) and Figure 6 (curve after the dotted line).

Table II. L-systems evolving in a changing environment. Bold style indicates duplications and insertion, and symbols inside a rectangle are about being deleted or mutated.

#	genotype	*s*
1	G+G-G[++G-GG][G]	.27
2	G+G-G[++G-GG**[++G-GG]**][G]	.13
4	G+G-G[++G-GG[++G-GG**[++G-GG]**]][G]	.10
5	G+G-G[**[++G**$\boxed{-}$**GG]**++G-GG[++G-GG[++G-GG]]][G]	.10
7	G+G-G[[++GGG]++G-GG[++G-GG[+$\boxed{+}$G-GG]]][G]	.10
9	G+G-G[[++GGG]++G-GG[++G-GG[+G-GG]]][$\boxed{G}$]	.08
13	G+G-G[[++GGG]++G-G$\boxed{G}$[++G-GG[+G-GG]]][-]	.08
15	G+G-G[[++GGG]++G-G[$\boxed{+}$+G-GG[+G-GG]]][-]	.07
16	G+G-G[[++GGG]+$\boxed{+}$G-G[-+G-GG[+G-GG]]][-]	.31
17	G+G-G[[++GG$\boxed{G}$]+G-G[-+G-GG[+G-GG]]][-]	.36
19	G+G-G[[+$\boxed{+}$GG-]+G-G[-+G-GG[+G-GG]]][-]	.44
21	G+G-G[[+GG-]+G-G[-+G-GG[+G-GG]]][-]	.62
23	G+G-G[[+GG-]+G-G[-+G-GG[+G-G**+**G]]][-]	.65
24	G+G-G[[+G$\boxed{G}$-]+G-G[-+G-GG[+G-G+G]]]**G**[-]	.72
25	G+G-G[[+G-]+G-G[-+G-GG[+G-G+G]]]G[-]	.93
26	G+G-G**G**[[+G-]+G-G[-+G-GG[+G-G+G]]]G[-]	1.0

This second example shows again that the evolution favors particular types of operators depending on the phase of adaptation the structure is in: duplications first, followed by deletions and insertions.

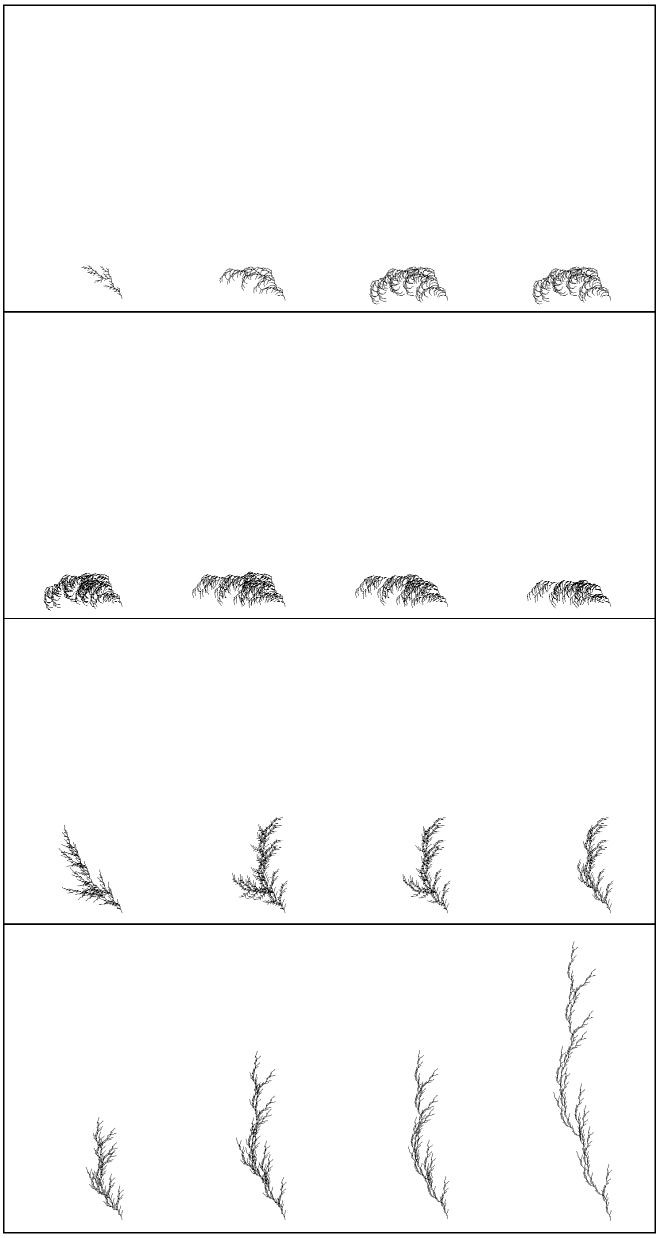

Figure 5. Phenotypes of the evolved genomes in Table II: regular to flatty and branchy (first two rows), and towards high structures (last two rows), from left to right and from top to bottom.

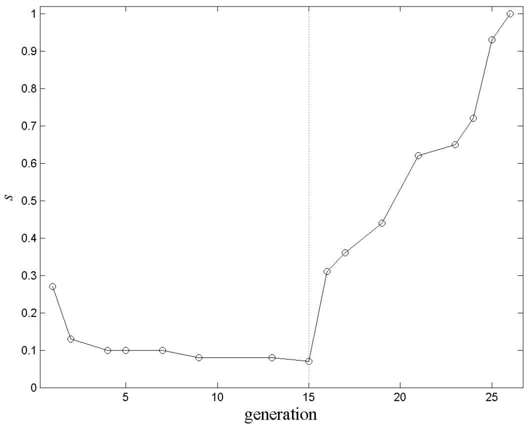

Figure 6. Evolution of the shape value for the two environments (delimited by a vertical dotted line).

4.3 Diversity

The two previous cases showed the extraordinary potential of these operators to generate a fabulous richness of forms by following an evolutive line. But we have also found that, under low selective pressure, an extraordinary collection of structures that followed the requirements can coexist as differentiated lines of search. Figure 7 illustrates some of these designs.

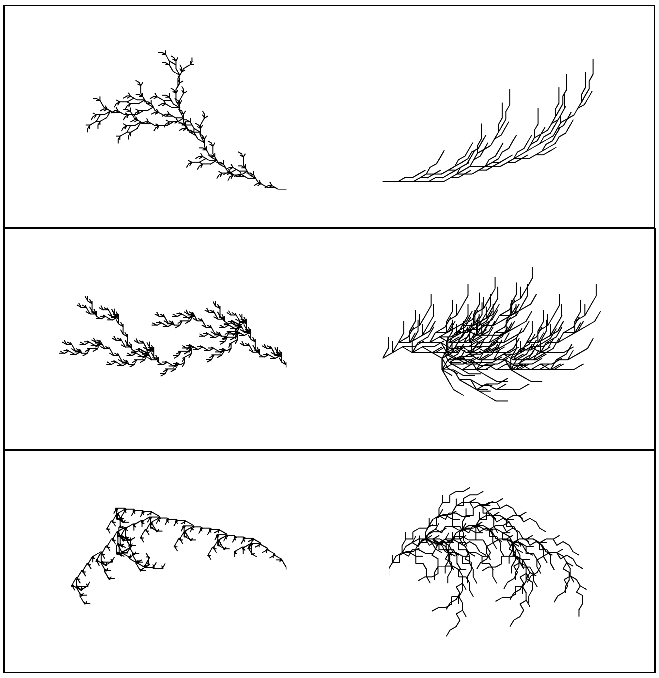

Figure 7. A collection of selected forms that meet the flatty and branchy criteria.

5. CONCLUSIONS

Indirect encoding is an evolutionary technique that demands additional understanding of biological developmental process, as well as modeling basic principles of genetics dynamics. Here we have studied the application of gene duplication and transfer in a framework of L-systems evolution.

The main consequence derived from the simulation results is the self-organized application of operators: firstly, the genomes tend to expand, through the application of duplications and horizontal transfer; then, after collecting enough genetic base, small mutations take place. This spontaneous synchronization is reflected in the evolution curve, which shapes the punctuated equilibrium that has been claimed to explain the stasis found in the fossil record [2].

One consequence found in this study, directly derived from the type of operators used, is the generation of junk DNA. Only the portion of the genome that has been tuned to the fitness landscape is relevant to the structure, while a significant part of it (sometimes most of it) is of no use at all. Substrings like `+-`, `[]` or consecutive segments have no effect on the read out of the L-system, and could be deleted. Its importance, instead, is evolutive, since these symbols can play an important role in the future.

The example of L-systems evolution described here could also find application in comparing structures. A type of edit distance (as defined in [6]) can be established between two different structures by determining the minimal number of operators that have to be applied to their representations (genomes) in order to transform one into the other.

The operators modeled here, and many other mechanisms of DNA dynamics (like transposition), demand an extensive study in different frameworks of indirect encoding evolutionary computation. Probably the knowledge obtained will be to a great extent linked to the definition of the particular framework, but it is possible that general principles can also be derived.

6. ACKNOWLEDGMENTS

This work has been partially funded by FP6 of the European Union under *BioEmergences* project. The Matlab code to represent 2D L-systems was based on the functions developed by the *Institut für Angewandte Analysis und Numerische Simulation* of *Universität Stuttgart* (Germany). Thanks also to Daniel Lobo for technical support.

7. REFERENCES

[1] Ashlock, D., Bryden, K. & Gent, S. (2006). Simultaneous Evolution of Bracketed L-system Rules and Interpretation. In *Proceedings of the 2006 Congress on Evolutionary Computation*, pp. 7403-7410.

[2] Eldredge, N. & Gould, S.J. (1972). Punctuated equilibria: an alternative to phyletic gradualism. In T.J.M. Schopf, ed., *Models in Paleobiology*. San Francisco: Freeman, Cooper and Company, pp. 82-115.

[3] Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. Michigan, NM: University of Michigan Press.

[4] Jacob, C. (1996). Evolution programs evolved. In *Parallel Problem Solving from Nature–PPSN-IV*, pp. 42-51. Berlin: Springer-Verlag.

[5] Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: The MIT Press.

[6] Levenshtein, V.I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Technical Report* 8.

[7] Lindenmayer, A. (1968). Mathematical models for cellular interaction in development, Parts I and II. *Journal of Theoretical Biology*, **18**:280-315.

[8] Margulis, L. & Sagan, D. (2002). *Acquiring Genomes: A Theory of the Origins of Species*. Perseus Books Group.

[9] McCormack, J. (1993) Interactive Evolution of L-System Grammars for Computer Graphics Modeling. In David Green and Terry Bossomaier (eds.) *Complex Systems: From Biology to Computation*. ISO Press, Amsterdam, pp. 118-130.

[10] Ohno, S. (1970). *Evolution by Gene Duplication*, Springer.

[11] Prusinkiewicz, P. and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. Springer-Verlag.

[12] Zhang, J. (2003). Evolution by gene duplication: an update. *Trends in Ecology & Evolution*, **18**(6):292-298.