

**Proceedings of the Fifth Annual
Western Computer Graphics Symposium**



**Silver Star Mountain
Vernon, British Columbia
March 28–30, 1993**

Artificial Evolution of Plant Forms

Cam MacKenzie and Przemyslaw Prusinkiewicz

University of Calgary
2500 University Drive N.W.
Calgary, Alberta T2N 1N4

1 Introduction

The purpose of this paper is to outline a technique for simulating the evolution of plants. The problem is addressed using genetic algorithms (GA's) [3, 5] operating on L-systems [12], with fractal dimension of the resulting structures serving as the fitness criterion. In physical terms, fractal dimension correlates with the "bushiness" or "fullness" of a plant. A plant with low fractal dimension (close to one) may have insufficient surface area to capture ample amounts of light. On the other hand, a plant with high fractal dimension (close to two in the case of planar structures) may be excessively bushy, causing substantial self-shading. Ultimately, plants which are bushy to the extent that they shade other plants but not themselves should be given evolutionary preference.

The idea of simulating evolution of plants captured by mathematical models was first explored by Niklas [9, 10]. He considered the abilities to capture light, sustain vertical growth, and to reproduce, as the main factors guiding plant evolution. According to these criteria, optimal plants would be tall branching structures capable of capturing an abundance of light. Qualitatively, the combination of fractal dimension and plant height represents the same evolutionary features as those considered by Niklas.

Computer simulation of evolutionary processes was also investigated by Wilson [16], who applied GA's to production rules describing the development of simple multicellular organisms. Wilson's approach involved a

formally defined fitness function. In contrast, Dawkins [1] simulated the evolution of insect-like “biomorphs” using a computer program to generate options, from which a human operator selected forms that were to survive and reproduce. A similar technique was applied by Sims [13] and Todd and Latham [15] to create abstract forms for artistic purposes. Smith [14] originated the application of genetic algorithms to L-systems, and introduced fractal dimension as a fitness criterion. The present paper is a continuation of his work. Our experiments involving chromosomes of fixed length (Section 2) rely on the original formulation of L-systems by Hogeweg and Hesper [4], making it easier to compare the evolved structures with their results. In addition, a different (more natural) class of L-systems is chosen for experiments involving chromosomes of variable length (Section 3).

2 Fixed Length Codings

Hogeweg and Hesper [4] described an experiment in which they searched a restricted class of bracketed deterministic 2L-systems, trying to identify those which generated plant-like structures. The successful structures were selected by a human operator from the set of designs presented by a computer. In this section we want to discover whether similar structures can be obtained without human intervention. The class of L-systems considered by Hogeweg and Hesper is suitable for this investigation since:

- it is rich in plant-like branching structures, and
- all productions follow the same format, and are therefore easy to code for use in genetic algorithms.

The productions defined by Hogeweg and Hesper are of the form:

$$a_l < a > a_r \rightarrow \chi,$$

which means that symbol a produces string χ if and only if a has left context a_l and right context a_r . The successor χ may include brackets, delineating branches in a tree structure. The following four production types are allowed:

1. $a_l < a > a_r \rightarrow 0$
2. $a_l < a > a_r \rightarrow 1$
3. $a_l < a > a_r \rightarrow 11$
4. $a_l < a > a_r \rightarrow 1[1]$

where $a_l, a, a_r \in \{0, 1\}$.

The axiom (starting symbol of the L-system) is 1, and any missing context is implicitly assumed to be 1. Specifically, the left context is missing at the root, and the right context is absent at branch endings. There are eight possible predecessors: $0 < 0 > 0$, $0 < 0 > 1$, ..., and $1 < 1 > 1$. Consequently, any L-system in the considered class can be specified as an ordered list of successors. For example, the list $0.11.1.1.0.1[1].1.0$ represents an L-system with the productions $0 < 0 > 0 \rightarrow 0$, $0 < 0 > 1 \rightarrow 11$, $0 < 1 > 0 \rightarrow 1$, etc.

In order to apply genetic algorithms to these L-systems, we represent the production successors by single character codes: $0 \leftrightarrow a$, $1 \leftrightarrow b$, $11 \leftrightarrow c$, $1[1] \leftrightarrow d$. Thus, the system $0.11.1.1.0.1[1].1.0$ is represented by string $X = acbbadba$. We think of X as a chromosome providing a genetic description of an organism in terms of a sequence of genes. Depending on their position in the chromosome, genes a and b cause the replacement of any symbol by a 0 or a 1, respectively. Gene c causes branch elongation by substituting a sequence of two 1 symbols for a single 0 or 1. Gene d initiates a new branch.

Chromosomes are subject to the *genetic operations* of mutation and crossover, which control the transfer of genetic material from parents to offspring. Mutations consist of random gene modifications and make it possible to recover prominent genetic information that may otherwise be permanently lost from the gene pool. Crossover is used to combine genetic information from both parents in an attempt to produce a child with a higher fitness value.

Once a string representing the modified structure has been generated, it is interpreted geometrically using the following two assumptions:

- symbols 0 and 1 denote line segments of unit length, and
- the branches are issued alternately to the left and to the right, using a constant branching angle.

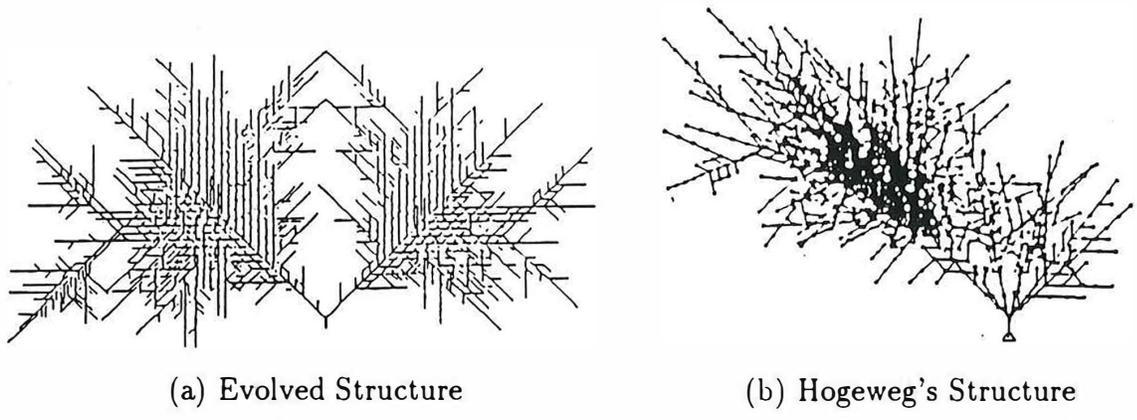


Figure 1: Two structures represented by chromosome 0.11.0.1.0.1[1].0.0

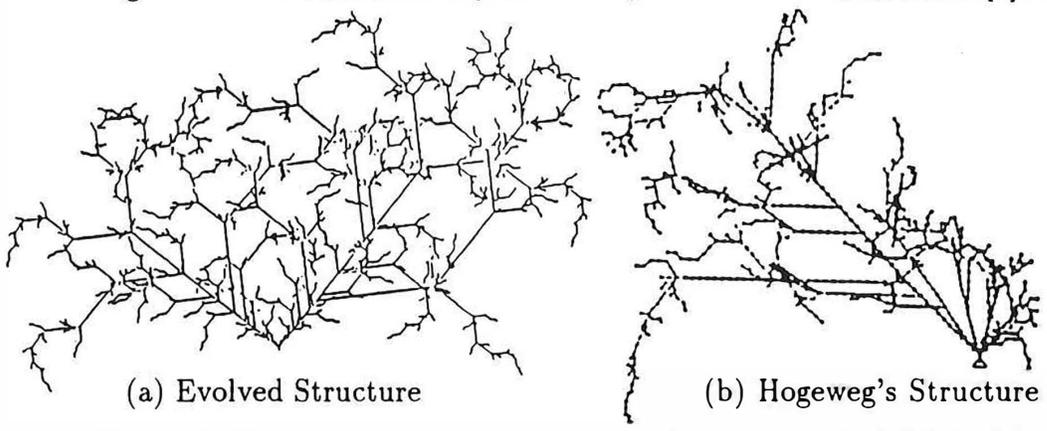


Figure 2: Two structures represented by chromosome 0.11.0.0.0.1[1].1.0

The fractal (Hausdorff-Besicovitch) dimension of the resulting figures are defined as the value of the parameter D which satisfies the inequality [8]:

$$0 < \lim_{r \rightarrow 0} N(r)r^D < \infty,$$

where $N(r)$ is the number of regularly shaped objects, such as squares with side length r , that are required to cover the figure. The estimation of D proceeds by covering the figure with a succession of objects of decreasing size r . The accuracy of the method depends largely on the quality of the coverings [2, pages 11-17]. The actual method employed in this research produced estimates containing less than 2% error [7]. Figures 1 and 2 compare two of the evolved structures with those discovered by Hogeweg and Hesper. The corresponding L-systems are identical; images differ because the figures

presented by Hogeweg and Hesper [4] are projections of three-dimensional structures, whereas our evolved structures are two-dimensional.

3 Variable Length Codings

The L-systems presented in the previous section were restricted to a class that could be represented by chromosomes of constant length. In this section we remove this restriction by considering a class of DOL-systems with two non-identity productions, admitting successors of variable length. Specifically, we assume that productions have the form:

$$\begin{aligned} X &\rightarrow a_1 \circ a_2 \circ \dots \circ a_n \\ Y &\rightarrow b_1 \circ b_2 \circ \dots \circ b_m, \end{aligned}$$

where each symbol a_i and b_j denotes an element of the set (gene pool):

$$P = \{X, Y, [+X], [-X], [+Y], F, [-Y], [+F], [-F], +F, -F\}$$

and $n, m \in \{1, 2, \dots, 10\}$.¹

The structures generated by these L-systems are visualized using the turtle interpretation [11, 12]. For example, the symbol F causes the turtle to draw a line segment of unit length, whereas symbols + and - turn the turtle left and right, respectively.

The eleven elements of the set P are coded using characters a, b, \dots, k . For example, the production $X \rightarrow [-X]FFF + F[+Y]$ is represented by an "X-chromosome" *dfffje*.

To illustrate the application of genetic operators, consider chromosome X (chromosome Y is handled identically). Suppose a crossover operation is to occur between two parents A and B. A crossing point is randomly chosen within the parent with the shortest chromosome. The first child receives genes from parent A until the crossover site is reached, with the remainder of its genes copied from parent B. The second child is created in an analogous way (see Figure 3). Chromosomes are also subject to mutations which may either:

¹In general, we could consider L-systems with more than two productions, each with its own gene pool.

Parents	Children
8 2 6 10 4	8 2 4 4 9 1 5
3 7 4 4 9 1 5	3 7 6 10 4

Figure 3: Example of crossover involving variable length chromosomes (crossover site = 2)

1. replace an existing gene by a randomly selected member of the gene pool,
2. insert a gene, or
3. delete a gene.

Figure 4 shows a sample “genealogical tree”, illustrating the progression of generated forms resulting from the simulated evolution.

4 Conclusions

Figures 1, 2, and 4 show the emergence of a wide variety of realistic plant forms in a process of artificial evolution based on a measurable fitness criterion. Thus, genetic algorithms operating on L-systems are suitable for simulating evolutionary processes on an abstract level. Our choice of genetic algorithm as the mechanism for generating plant forms is motivated by the appealing analogy between the formal operations of mutation and crossover and their real-life counterparts. It is an open problem, however, whether other algorithms could search the space of possible forms more efficiently, yielding a larger number of interesting forms in less time.

We are currently experimenting with fitness criteria other than fractal dimension. For example, Figure 5 depicts three models of primitive land plants evolved using *photosynthetic efficiency* (ratio between plant surface area exposed to light and total surface area) as the fitness criterion. The rightmost tree in the image is the most primitive plant. Exhibiting a closely knit crown, it is unable to capture much light. The leftmost tree evolved later. It has a vertically planated branching pattern, making it capable of intercepting more light than the previous forms. Boasting a horizontally planated branching structure, the tree in the foreground required longest to

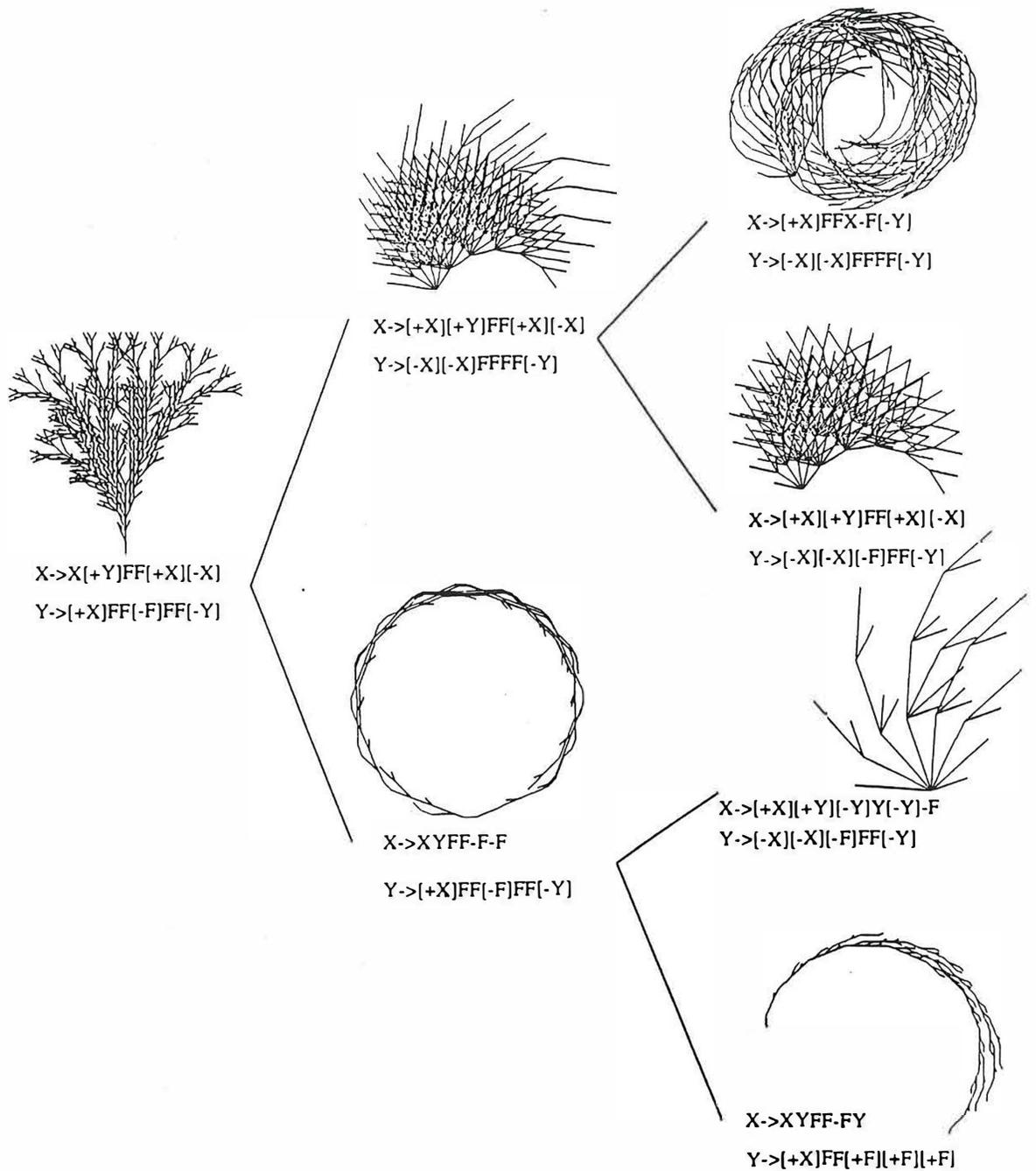


Figure 4: Genealogic tree of branching structures generated by L-systems with successors of different lengths. Fractal dimension of the final structure is equal to 1.33.



Figure 5: Simulated evolution of early land plants

evolve, but captures the most light. This progression of artificially evolved forms is similar to that obtained by Niklas.

An extension of the problems addressed in this paper involves the artificial evolution of particular plant species. A good solution would provide an answer to the long-standing inference problem of L-systems theory [6]: How can we automatically synthesize an L-system that captures the essential characteristics of a particular plant species?

References

- [1] R. Dawkins. *The Blind Watchmaker*. W.W. Norton and Company, 1987.
- [2] J. Feder. *Fractals*. Plenum Press, 1988.
- [3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.

- [4] P. Hogeweg and B. Hesper. A model study of biomorphological description. In *Pattern Recognition*, pages 165–179, 1974.
- [5] J.H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [6] H. Jurgensen and A. Lindenmayer. Inference algorithms for developmental systems with cell lineages. *Bulletin of Mathematical Biology*, 49(1):93–123, 1987.
- [7] C. MacKenzie. Estimating fractal dimension, 1991. Manuscript, Department of Computer Science, University of Calgary.
- [8] B.B. Mandelbrot. *The Fractal Geometry of Nature*. W.H. Freeman, 1983.
- [9] K. J. Niklas. Computer-simulated plant evolution. *Scientific American*, 254(3):78–86, March 1986.
- [10] K.J. Niklas. *Plant Biomechanics: An Engineering Approach to Plant Form and Function*. The University of Chicago Press, 1992.
- [11] P. Prusinkiewicz. Graphical applications of L-systems. In *Graphics Interface '86*, pages 247–253. CIPS, 1986.
- [12] P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants*. Springer Verlag, 1990.
- [13] K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, 1991.
- [14] R. Smith. Evolution of l-systems, 1989. Manuscript, Department of Computer Science, University of Regina.
- [15] S. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press, 1992.
- [16] S. W. Wilson. The genetic algorithm and biological development. In *Proceedings of the Conference on Genetic Algorithms*, pages 247–251, Cambridge, Massachusetts, July 1987.