Evolutionary Control of L-system Interpretation

Dan Ashlock
Mathematics Department
Iowa State University,
Ames, Iowa 50011
danwell@iastate.edu

Kenneth M. Bryden
Mechanical Engineering
Iowa State University
Ames, Iowa 50011
kmbryden@iastate.edu

Abstract

An L-system or Lindenmayer system consists of a grammar and an interpreter. The grammar contains an axiom, usually a short string, that the grammar expands into a long, complex string. The interpreter then renders the string into an object. The first use of L-systems was to provide morphological models of plants. In this exploratory initial study we use an evolutionary algorithm to evolve interpreters for L-systems. The interpreter is a graphics turtle. For a given L-system the evolutionary algorithm tunes the turtle’s parameter to cause it to drive in a constrained area of the Cartesian plane. Multiple L-systems and planar regions are given. In some cases a startlingly small number of optima are located indicating a relatively simple fitness landscape.

I. Introduction

An L-system or Lindenmayer system [2], [6], consists of two parts. The first is a grammar consisting of an axiom and a collection of replacement rules. The L-system creates a sequence of strings starting with the axiom by applying the rules. This application is called an expansion. Rules are applied simultaneously to every character in a string to generate the next string. An example of such a grammar, together with the first four expansions beyond the axiom, is given in Figure 1.

Fig. 1. An L-system grammar and the first few resulting strings.

| axiom: | A | A |
| rules:  | ABA | AB |
| A→AB   | ABAAB | AABABA |

L-systems are typically hand designed to allow the interpreter to yield a desired appearance. The hand design touches on both the L-system’s grammar and interpreter. This is probably due to the difficulty of writing an adequate fitness function to capture the notion of “plantlike” or even “beautiful” that is the typical goal of an L-system. Human-in-the-loop evolution has been performed [3] to create interesting L-systems. In this study simple L-systems system interpreters will be evolved. The goal is to cause a graphics turtle to draw within a restricted area of the plane. The L-system interpreter selection via evolution will be a completely automatic process.

Evolveable Interpreters

A graphics turtle may be given commands to turn and advance with or without drawing. The turtle used in this study take the L-system in Figure 1 and assign the meaning “turn 45 degrees” to A and “advance 0.22 units” to B then Figure 5 shows the result of the interpretation of the first 400 characters of the fourteenth expansion of the L-system.

Fig. 2. An interpretation of an L-system with a graphic turtle.
will always draw when it advances. There will be up to two angles available for turning and up two distances available for advancing. We will use an evolutionary algorithm to select these angle and distance parameters. The study encompasses L-systems with one turn angle and one advance, two turn angles and one advance length, and two turn angles and two advance lengths. A set of parameters for a turtle based L-system interpreter will be called a turtle controller.

We will reward a given turtle for drawing within a specified area of the plane called a drawing arena. Results are presented for two drawing arenas. The first drawing region is the annulus:

\[ \{(x, y) : 0.05 \leq (x - 0.5)^2 + (y - 0.5)^2 \leq 0.2\} \]  \hspace{1cm} (1)

while the second is the square:

\[ \{(x, y) : 0.2 \leq x, y \leq 0.8\} \]  \hspace{1cm} (2)

Both drawing arenas are proper subsets of the unit square with corners (0, 0) and (1, 1). Images of the two arenas are shown in Figure 3. When evaluating fitness the turtle starts inside the drawing arena. The initial placement is uniform. For the annular region the turtle starts at (0.2, 0.5), on the square the turtle starts at (0.5, 0.5). Since L-systems can be continuously scaled by scaling the turtle(s) fundamental drawing lengths there is no problem adapting the interpreters to different sized regions.

![Fig. 3. The two drawing arenas within the unit square.](image)

II. EVOLUTIONARY ALGORITHM DESIGN

The only difficult part of the system for evolving turtle controllers is the design of a fitness function. If the goal is to constrain the turtle to draw within a region of the plane by selecting a small number of turn angle and advance distance parameters for the turtle then there are several conflicting constraints. First, the turtle must be encouraged to move. A fitness function that rewards command executions within the specified area would encourage the evolution of immobility within the drawing arena. Second the turtle must be encouraged not to move outside of the drawing arena. Third, the penalties for moving outside of the arena must not be so harsh as to prevent progress.

The constraints listed above were identified by the process of evolving turtle controllers with fitness functions that ran afoul of those constraints. An ideal fitness function to address all these constraints is to compute distance drawn within the drawing arena. This encourages motion and does not reward drawing outside the arena. Immobile turtles are granted nil fitness and turtles that drive wildly across the boundaries receive lower fitness evaluations. The sole problem we have identified with this fitness function is that it requires a fairly intimate knowledge of the geometry of the drawing arena or exceedingly fine-grained computation to compute with precision.

As a compromise we offer the following. The \( k \)-approximate driving fitness for a turtle controller executing a fixed string of commands in relation to a given drawing arena is computed as follows. Each time the turtle advances, the line segment of that advance is divided into \( k \) sub-segments. The turtle’s fitness is increased by the length of each sub-segment that has both ends in the drawing arena. The fitness is the sum of sub-segment lengths with both ends in the arena over the execution of a given string of \( n \) turtle commands. As \( k \) increases this function converges to distance driven within the drawing arena. One more detail remains to construct the fitness function used.

In an initial population there is often a large number of essentially worthless structures. A common type of worthless turtle controller is one that drives not only outside of the drawing arena but off into the Cartesian plane outside of the unit square. To give evolution a push in the correct direction each turtle commands that is executed and leaves the turtle outside of the unit square (well outside the drawing arena) yields a fitness penalty of -0.1, added into the \( k \)-approximate driving fitness. This study uses \( k = 4 \). The penalized \( 4 \)-approximate driving fitness will be referred to as simply “fitness” hereafter.

The evolutionary algorithm used operates on a population of 120 turtle controllers each of which has two, three, or four real parameters. The number of parameters follows from the number of symbols in the L-system. The parameters are treated as atomic objects and stored as machine reals in an array data structure. The algorithm is steady state and uses single tournament selection with a tournament size of seven. A mating event consists of choosing seven controllers and then operating on the two best to produce two new controllers that replace the two worst. The algorithm performs 24,000 mating events each time it is run. A single point crossover of an array that stores angular parameters and then length parameters for the turtle controller is used. A single parameter is mutated after crossover in each new controller. Mutation consists of multiplying the parameter by a number of the form \( e^X \) where \( X \) is uniformly distributed in the range \([-0.1, 0.1]\).

The number of mating events was chosen in a preliminary study in which fitness appeared to plateau in most runs by the ten-thousandth mating event. The crossover operator is appropriate for the quite small genome size and runs with the crossover operator disabled produce similar results. The slightly unusual mutation operator is chosen to permit mutation to operate by scaling aspects of the system. Choice of tournament selection and population size is ad-hoc but
matches common values in other parameter optimization studies.

III. EXPERIMENTS PERFORMED

A set of 100 evolutionary runs were performed for all six combinations of the two drawing arenas with each of the three L-systems given in Table I. To create the string of turtle commands the successive strings of the L-system, starting with the axiom, were concatenated until the length exceeded 400 characters. The first 400 characters were then used as a fixed set of commands to the turtle with only the interpretation of those commands varying. The number 400 is large enough to capture the behavior of the turtle controller without being so long as to excessively slow fitness evaluation. The evolutionary algorithm searched for turn angles and advance lengths that yielded the highest fitness for a given string of turtle commands. The most fit turtle controller from each run was saved. The population mean, standard deviation, and maximum fitness were saved every 120 mating events.

<table>
<thead>
<tr>
<th>L1:</th>
<th>L2:</th>
<th>L3:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Axiom:</strong></td>
<td><strong>Axiom:</strong></td>
<td><strong>Axiom:</strong></td>
</tr>
<tr>
<td>$\theta$</td>
<td>$\theta$</td>
<td>$\theta$</td>
</tr>
<tr>
<td><strong>Rules:</strong></td>
<td><strong>Rules:</strong></td>
<td><strong>Rules:</strong></td>
</tr>
<tr>
<td>$\theta \rightarrow \theta R$</td>
<td>$\theta \rightarrow \sigma R$</td>
<td>$\theta \rightarrow \sigma R$</td>
</tr>
<tr>
<td>$R \rightarrow \theta$</td>
<td>$\sigma \rightarrow \sigma R$</td>
<td>$\sigma \rightarrow \sigma R$</td>
</tr>
<tr>
<td>$R \rightarrow \theta$</td>
<td>$R \rightarrow \theta R \theta$</td>
<td>$R \rightarrow \theta \theta$</td>
</tr>
<tr>
<td>$S \rightarrow \sigma$</td>
<td></td>
<td>$S \rightarrow \sigma$</td>
</tr>
</tbody>
</table>

**TABLE I**
The L-systems used in this study. The symbols $\theta$ and $\sigma$ denote turn angles while $R$ and $S$ denote distances to advance.

IV. RESULTS

The behavior of the fitness traces for individual runs was typical for an evolutionary algorithm performing parameter optimization. Most runs exhibited a sharp initial increase in fitness followed by modest fitness increase and long static epochs. In some runs examples of “discovery”, characterized by a jump in the maximum fitness, appeared. An example of this behavior is shown in Figure 4.

In all six sets of evolutionary runs there was substantial success in constraining the turtle to operate inside the driving arena. In all cases solutions could be scaled to fit completely inside the drawing arena but such scaling would lower the fitness. The fitness function slightly emphasizes the importance of total distance traveled over that of staying in bounds.

When the L-system had the smallest number of symbols, L1, and hence the evolutionary algorithm was optimizing a smaller number of constraints, there were well defined optima. Figure 5 shows all the types of optima found in the runs using the first L-system on the annular drawing arena and gives the number of times each was discovered as well as the parameters for the turtle controller. Visually similar solutions have very similar parameter values. On the square drawing arena for L1 the evolutionary algorithm produced more distinct solutions, but no more than eight (depending on the pickiness of the person performing the classification). The groupings of solutions were less crisp on the square than on the annulus.

System L3 gives the evolutionary algorithm the most degrees of freedom. It also produced the most diverse collection of solutions for each of the drawing arenas. Solutions did, with some scatter, appear multiple times. Figure 6 shows four variations of one of the more common solutions found for L3 on the square drawing arena. A sampling of the diversity of L3 solutions on the annular drawing arena is given in Figure 7. It is interesting to note that the lower right solution shown in Figure 7, with a fitness of 11.13, is the only solution in its group of 100 runs with a fitness in excess of 10. The other 99 solutions located have fitnesses from 6.29 to 9.42. This solution thus appears to lie on a high hill that is relatively difficult to find. See Figure 10 for a visual sense of the degree

![Fig. 4. Mean and maximum fitness for a population with a ‘discovery’ or abrupt change in the maximum fitness.](image)

![Fig. 5. The four types of interpretations found using the two symbol L-system on the annulus. The number of appearances of these types, in reading order, is 29:32:38:1 within the 100 runs performed. The parameter values for these turtle controllers are $(\theta = 67.4^\circ, R = 0.145)$, $(\theta = 37.0^\circ, R = 0.101)$, $(\theta = 44.6^\circ, R = 0.113)$, and $(\theta = 94.9^\circ, R = 0.126)$ in reading order.](image)
Fig. 6. An example of four similar solutions found using \textbf{L3} with the square drawing arena.

Fig. 7. An example of four of the many solutions found using \textbf{L3} with the annular drawing arena. The fitnesses of these solutions in reading order are 8.92, 6.62, 7.64, and 11.13.

System \textbf{L2} produced the least diverse collection of solutions, somewhat at odds with a reasonable expectation that diversity would increase strictly with the number of parameters. This lack of diversity was manifested in both drawing arenas. In the square arena a single solution was located in all 100 runs. This solution is shown in Figure 8. Two solutions were found for the annular drawing arena. These are given, together with their relative frequency, in Figure 9.

Figure 10 shows a pair of histograms. The upper enumerates solutions at different fitness levels for \textbf{L1} on the annulus while the lower does the same for \textbf{L3}. The three groups of fitnesses in the upper histogram correspond to the three common solutions show in Figure 5. The single fitness entry near 17 is the odd solution that appeared once. Leaving out the solution discovered once, the number of times a solution was discovered was in inverse proportion to its fitness. This suggests that the fitness landscape for \textbf{L1} on the annular driving arena has an irregular and interesting topology. If you look at the uppermost cluster, corresponding to the first solution shown in Figure 5, there are two variations. The first has a smaller $R$, the second a larger $R$ with very similar values for $\theta$. They represent two peaks both near on another in fitness and with similar drawings. One, the least fit, stays almost completely inside the drawing arena. The other escapes the arena but, by driving a good deal farther, puts in more driving in the arena. These solutions are shown in Figure 11.

The second histogram indicates that there are many hills in the fitness landscape for \textbf{L3} on the annular driving area. Examining the fitness traces of these runs it seems that having a shift in the hill that a majority of the population is on is more common than in the landscape for \textbf{L1}. The presence of the fitness 11.13 outlier also indicates that the study done here could profitably be run for a much longer time to obtain a better idea of what optima exist. The goal of this study is not, however, to find the highest fitness turtle controllers. Rather it seeks to survey the diversity of solutions that drive mostly within the drawing arenas.
Fig. 10. The above histograms give the distribution of fitnesses for the best turtle controllers found in each run. The upper plot is for \textbf{L1} on the annular driving arena. The lower is for \textbf{L3} on the annular driving arena.

Fig. 11. The two major variations of the most fit solution for \textbf{L1} on the annular drawing arena.

V. DISCUSSION

The system presented here, which fits the parameters of an L-system interpreter to a given L-system grammar to cause a graphics turtle to drive within a specified driving arena, is not a difficult target for evolutionary computation. The fact that it is not hard to make a turtle simultaneously lay down a lot of ink and stay close to specified driving area is not \textit{a-priori} obvious. Since the turtle turns a couple hundred times during fitness evaluation a very small change in a fixed turning angle results in an enormous change in the turtles final behavior. Examining the four simulation shown in Figure 6 allows the reader to see the effect of very small changes in an angular parameter. This potential for extreme sensitivity to an angular parameter opened the possibility to make a very brittle system.

The picture drawn by a turtle does depend continuously on the angular and radial parameters. The picture also scales directly as the radial parameter. This means that the fitness landscape, at least if the sampling number \( k \) is large, depends continuously on the parameter values. For a given choice of angle(s), variation of the radius scales the drawing. Given that the turtle starts within the drawing arena this means that the algorithm can always optimize the radius to fit the drawing within the arena. This perhaps explains the relatively smooth functioning of the system.

Figure 12 shows examples of members of the initial population. All four examples have fitnesses below 2.0 and only the first stays mostly within bounds. These random turtles can be compared to the best-of-run solutions depicted elsewhere to give a sense of the degree to which the evolutionary algorithm is improving the quality of L-system interpretation.

In the results section examples were given that show there are high hills in the fitness landscape that are difficult to discover. This is a sense in which a fitness landscape can be interesting. A turtle controller uses a small number of parameters and uses each hundreds of times in a manner that causes their influence to accumulate. The interesting character of the fitness landscape is probably partly caused by this accumulation of the influence of each parameter. Even the most complex case studied, \textbf{L3} on the annular drawing arena, exhibited a “hidden” optima.

This study has looked at only six of an infinite number of cases of the basic problem. It is not clear how the system will scale with an increase in the number of parameters available to the turtle controller. A twenty symbol L-system with six turns and fourteen radial parameters might become a very smooth, simple surface.

The choice of the three L-systems used in the study was done by looking for L-systems that used all their symbols in all expansions of length 400 or greater. \textbf{L1} is the standard example
of an L-system used in most monographs on L-systems [5]. Other than this choice of a “traditional” L-system, the choice of L-systems in this study is ad hoc. A systematic survey of different types of L-system is desirable but is not possible in this initial exploration. The lack of diversity of solution for L2 on both drawing arenas suggests that L2 may be atypical.

The rule \( \sigma \rightarrow RR\theta \) is a potential culprit because it forces the system to deal with, in effect, two drawing lengths: \( R \) and \( 2R \).

The L-systems used in this study are the simplest type. Having one of the radial parameters represent not drawing while advancing would make the system’s drawings potentially more complex. It is intuitive that the fitness function used in this study would select to zero out such a radial parameter. Bracketed L-systems, discussed in the next section, can draw enormously more complex objects than the L-systems presented in this study. Attempting to evolve interpreters for bracketed L-system might yield a more challenging fitness landscape.

This study is part of an ongoing project to learn to control L-system. In [3] a system for evolving the grammatical portion of an L-system is presented. Other attempts to evolve L-system grammars appear in unpublished manuscripts. This study fills a gap by fitting an interpreter to fixed L-systems.

VI. NEXT STEPS

One reason that no attempt was made to present a systematic study of representative L-system grammars is that there is no obvious procedure for systemization. Given that we see particular optima multiple times we suggest a possible classification technique. For a given L-system, cluster best turtle-controller from a large number of runs. Compute the empirical probability of each type of turtle controller and use these distributions, both number of optima and their probabilities, as a type of signature for the L-system. Clustering these signatures would permit the identification of at least one type of classification system for L-systems.

The three L-systems samples in this study do not give sufficient basis for developing intuition about the scalability of the system. An L-system with more rules and symbols and a more complex drawing arena are both directions in which the system can be scaled. Additional examples would be required to assess the rate at which this task grows more difficult.

The exploration of bracketed L-systems [5] is a clear next step. A bracketed L-system incorporates two new symbols [ and ]. The open bracket pushes the state of the turtle on a stack and the closing bracket pops the turtle’s state. The use of a turtle-state-stack permits the L-system interpreter to draw branching structures. An interpretation of an L-system of this type is shown in Figure 13.

The rules in an L-system need not be deterministic. There can be a random or input driven choice of how to expand a given character in the grammar. Studying random or data driven L-systems is a natural generalization. In this case there is the potential to permit the turtle to create “data flakes” that form a type of visualization of input data.

The close connection between turtle controller parameters and appearance of the interpretation in this study suggests that adding a classification pipeline to the system may be practical. Applying simple clustering to the parameters and fitnesses in a given set of runs would permit automation of the process of locating hills in the fitness landscape.

Another direction that is natural to evolve this work is adding additional types of controls to the L-system command set. If some symbols control line thickness and style, color, or other parameters then evolutionary control of interpretation can start to more issues concerned with the appearance of an L-system interpreter. In [5] procedures for L-systems in three dimensions are discussed. The system presented here generalizes smoothly to three dimensions.

Evolution of L-system grammars has been a target of evolutionary computation to a greater degree than parameter setting for L-system interpreters. Both problems are interesting and the problem of evolving grammars is, at least at first glance, more challenging. Fusing the two is a possible direction for future work. Would simultaneous evolution of grammar and system parameters be more valuable or would stepwise evolution of grammar and then interpretation cyclically yield better results?
VII. ACKNOWLEDGMENTS

The authors would like to thank the Iowa State Virtual Reality Applications Center for its support of this research. We would also like to thank the members of the Iowa State Complex Adaptive Systems Program for many helpful discussions.

REFERENCES


