Limits to Expression in Genetic Programming
Lattice-Aggregate Modeling

Jason M. Daida
The University of Michigan
Artificial Intelligence Laboratory and the Space Physics Research Laboratory
2455 Hayward Avenue
Ann Arbor, Michigan 48109-2143

***Abstract*—This paper describes a general theoretical model of size and shape evolution in genetic programming. The proposed model incorporates a mechanism that is analogous to ballistic accretion in physics. The model indicates a four-region partition of GP search space. It further suggests that two of these regions are not searchable by GP.**

I. Introduction

Are there taboos to what can be expressed in the solutions derived under genetic programming (GP)?

While current theory has yet to offer a definitive answer, let alone proof of whether limits do or do not exist, there have been clues in the literature that point to the existence of taboos. Most notably, work in the evolution of size and shape has indicated that GP generates solutions that have an affinity towards particular sizes and shapes (e.g., [1-3]). Moreover, these findings suggest that this affinity is significant, which greatly curtails what can be generated if a solution is not of these shapes and sizes. Still, the literature suggests that it should still be possible to do so, even though it becomes increasingly difficult to generate a solution the further away that solution is from these preferred sizes and shapes.

My research group and I have been investigating the factors that contribute towards making a problem GP-hard. We have hypothesized that one of these factors is the structure that is implicit in a tree representation. In the process of doing so, we have discovered regions in the search space that are possibly taboo to GP. The purpose of this paper, then, is to describe a model of GP that isolates the consequences of structure, and to describe these possible regions of taboo.

II. Lattice-Aggregate Model

This section describes the proposed model in the context of previous work upon which the model has been based, followed by mathematical outlines of how the model works.

A. Background

At its heart, the proposed lattice-aggregate model is a rewriting system that is applied to a set of positive integers that bijectively map to locations on a circularly symmetric lattice. While trees in general are recursively defined in terms of a finite set [4], trees in GP usually have nodes that are associated with some type of programmatic content and are typically implemented in a manner that facilitates computation. While such implementations of trees are essential towards making GP operational, we have hypothesized that such implementations could obscure the salient mechanisms that affect the dynamics of GP. For that reason then, the contents of the model's trees are reduced to nil; only locations of the nodes remain.

The proposed model is analogous to Witten and Sander's model for diffusion-limited aggregation [5]. Similarities to Witten and Sander's model include the following:

- *Initial conditions presuppose the existence of a nucleating center.* In Witten and Sander's model, there is a nucleating center upon which subsequent growth occurs. In the proposed model, there is a nucleating center that includes the root node, also upon which subsequent growth occurs.
- *Growth occurs by randomly occurring collisions.* In Witten and Sander's model, growth occurs when a (random walk) particle collides and subsequently sticks to some random location on the perimeter of the nucleating center. In the proposed model, a particle sticks to what corresponds as a random leaf of a nucleating tree.
- *Model presupposes a lattice.* In Witten and Sander's initial model, a four-connected square lattice was presupposed, upon which particles traveled. In the proposed model, it can be shown that a lattice is also presupposed, upon which growth also occurs.

The proposed model differs from Witten and Sander's diffusion-limited model in several ways:

- *Ballistics of particles does not matter.* What matters to the proposed model is that the sites for growth on a nucleating center are selected at random—how a particle gets there is not of concern. For Witten and Sander's initial model, as well as many of Sander's subsequent models, the path by which a particle takes to get to a nucleating center is paramount.[1]
- *The particles used are not pixels.* The primary unit of growth in the proposed model is a set. Trees are abstracted into sets of positive integers, whereupon the value of each integer is bijective to a location on a lattice.
- *The type of lattice is non-rectangular.* The lattice that underlies the proposed model is a three-connected, circularly symmetric grid, which is distinct from the four connected, square grid of Witten and Sander's initial model.

[1] The type of growth that is characterized by Witten and Sander's model is known as *ballistic accretion.*

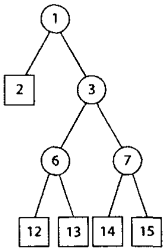
0-7803-7282-4/02/$10.00 ©2002 IEEE

The proposed lattice-aggregate model also shares several features with Lindenmeyer systems [6]. In particular, the proposed model can be expressed in terms of a stochastic OL-system (see [7]). However, this alternative representation is reserved for a later work.

B. Sketch of Model

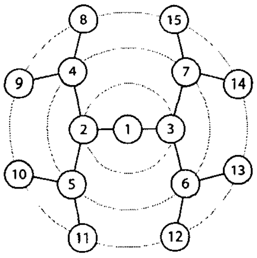
The following sketch outlines the model that has been used to describe the evolution of shape and size for a standard genetic programming system. The particular sketch that is given here is valid for binary trees for depths 0 – 26 (presuming that the root node of a tree is at depth 0).

Let A be a set of positive integers that correspond to the numbered nodes of a binary tree **T**. The numbering scheme for this tree is such that the parent of node k is node $\lfloor k/2 \rfloor$, and that the children of node k are nodes $2k$ and $2k + 1$. A binary tree may subsequently be represented in terms of its nodes' locations, with its structure being implicit in those locations.[2] For example, it is fairly straightforward to show that the following tree is equivalent to the set $A = \{1, 2, 3, 6, 7, 12, 13, 14, 15\}$:



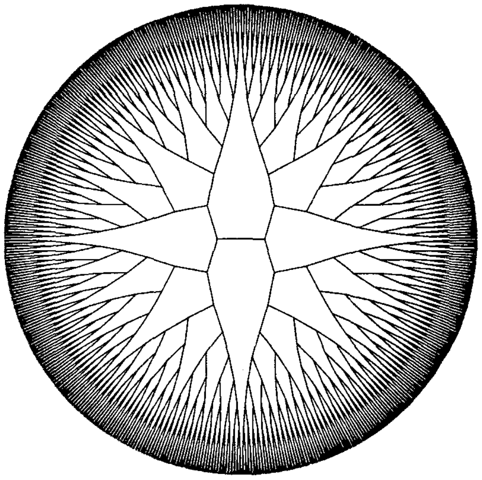
(1)

A particular consequence of numbering nodes in this manner is that the locations of both internal nodes and leaves are absolute. It is therefore possible to construct a lattice in which each number corresponds to a position on this lattice. For example, the following lattice shows the locations of the first fifteen positive integers:



(2)

The lattice for 2047 nodes looks like the following:



(3)

Let the root node be defined at depth $d = 0$. Assuming that T has $d > 0$, it can be shown that a set A can be decomposed into two mutually exclusive, non-empty sets J and K such that

- Set J corresponds to the internal nodes of T
- Set K corresponds to the leafs of T

We define a set B to correspond to a subtree of T. Note that set B is a function of k, whereupon $k \in A$. The smallest possible subtree, a leaf, can be described as

$$B^1 = \{k\}. \tag{4}$$

The next smallest subtree, a parent and two nodes, can be described as

$$B^3 = \{k, 2k, 2k + 1\}. \tag{5}$$

For the purposes of modeling GP behaviors for depths 0 – 26, we arbitrarily define B^5, B^7, B^9, and B^{11} to correspond to 5-, 7-, 9-, and 11-node subtrees, respectively.

$$B^5 = \{k, 2k, 2k + 1, 4k + 2, 4k + 3\}. \tag{6}$$

$$B^7 = B^5 \cup \{8k + 4, 8k + 5\}. \tag{7}$$

$$B^9 = B^7 \cup \{16k + 10, 16k + 11\}. \tag{8}$$

$$B^{11} = B^9 \cup \{32k + 20, 32k + 21\}. \tag{9}$$

Note that the particular selection of elements for B^5, B^7, B^9, and B^{11} is arbitrary. What each of these sets has in common, however, are that each corresponds to a minimal binary tree.

Now let $k \in K$. We can then represent the growth of a tree by B^5 as

$$A' = A \cup B^5(k) = A \cup \{2k, 2k + 1, 4k + 2, 4k + 3\}. \tag{10}$$

Likewise, we can do the same for B^7, B^9, and B^{11}.

[2] Note that this numbering scheme is similar to that of a complete binary tree, as defined by Knuth in [4]. However, unlike a complete binary tree, the locations given by this numbering scheme are not assumed to be sequential.

0-7803-7282-4/02/$10.00 ©2002 IEEE

Consequently, we can represent a stochastic model of tree growth for depths 0 – 26 as a recursive operation upon integer sets, namely

$$A' = A \cup B^{\boldsymbol{i}}(\boldsymbol{k}), \tag{11}$$

where $\boldsymbol{i}$ is a discrete random variable with sample space $S_B = \{5, 7, 9, 11\}$ and $\boldsymbol{k}$ is a discrete, uniformly distributed random variable with sample space $S_B = K$. It can be demonstrated that an appropriate probability distribution function corresponding to $\boldsymbol{i}$ entails the following relationship[3]

$$P(\boldsymbol{i} = 5) = 2\,P(\boldsymbol{i} = 7) = 4\,P(\boldsymbol{i} = 9) = 8\,P(\boldsymbol{i} = 11). \tag{12}$$

Example. Given $A = \{1, 2, 3, 6, 7, 12, 13, 14, 15\}$. Set A decomposes into $J = \{1, 3, 6, 7\}$ and $K = \{2, 12, 13, 14, 15\}$. Assuming that the second element of K and that B^5 have been chosen, $A' = \{1, 2, 3, 6, 7, 12, 13, 14, 15, 24, 25, 50, 51\}$. See Figure 1 for an example of both A and A' being mapped onto the lattice shown in (3).

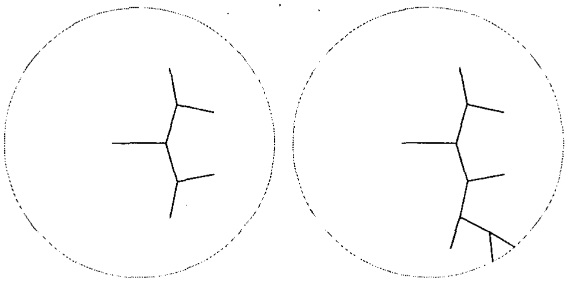


Figure 1. Sets A and A' are mapped onto the lattice of the proposed model. Left is set A; right, set A'. The gray circle corresponds to a tree depth of five.

C. Variations

There are several additional variations that need to be considered in the modeling of tree growth in GP. The first set of variations assists in identifying the upper and lower density bounds of tree growth, while the second set of variations address methods of population initialization.

The density of a set A can be defined as follows:

$$density \equiv \frac{\mathrm{N}(A)}{2^{\lceil \log_2(\max(A)+1)\rceil} - 1}, \tag{13}$$

where $\mathrm{N}(A)$ is the number of elements in A and $\max(A)$ identifies the maximum value in A. This definition corresponds to a ratio that is the number of nodes of a tree that is normalized by the number of nodes in a full tree of identical depth.

To identify the upper density bound, equation (11) can be restated as

$$A' = A \cup B^3(\boldsymbol{k}). \tag{14}$$

Equation 13 corresponds to the case whereby tree growth is entirely determined by three-node subtrees. Note that if k were instead deterministic such that all $k \in K$ is selected for replacement by B^3, the resulting tree would approach being full.

To identify a lower density bound, equation (11) can be restated as

$$A' = A \cup B''(\boldsymbol{k}), \tag{15}$$

where B'' is the least dense set of those sets B that are used in modeling growth. It is assumed that density for sets B are determined at $k = 1$. For the proposed model for depths 0 – 26, the set that is least dense is B^{11}.

It is possible to modify equation (11) to account for varying methods for population initialization. While such modifications have been done to model Koza's ramped half-and-half for depths 2 – 6, the exposition of these modifications have been left to a future paper.

III. Determination of Search Space Boundaries

The specified model was subsequently used to derive boundaries in the size-shape search space of trees from depths 0 – 26. This derivation consisted of four steps, namely:

- Used Monte Carlo methods to sample the proposed lattice-aggregate model corresponding to equation (11).
- Extracted depth and number of nodes from each sample.
- Computed the cumulative distribution of the numbers of nodes per tree for trees that correspond to a depth d for $d = \{0, 1, 2, \ldots 26\}$.
- Determined isopleths in size-shape space that correspond to contours of constant distribution.

This process is shown in Figure 2 for 50,000 sets. Isopleths were generated for 99%, 75%, median, 25%, and 1% distributions. Note that given the relatively steep fall-offs in the distribution of sets in size-shape space, the 99% and the 1% isopleths do approximate boundaries that specify where trees do or do not occur in this search space.

A similar procedure was applied to determine isopleths for equations (14) and (15). Again, given relatively steep fall-offs in distribution, the 99% isopleth for equation (14) approximated the uppermost bound of search, while the 1% isopleth for equation (15) approximated the lowermost bound of search.

IV. Model Results and Predictions

Figure 3 summarizes the isopleths for equations (11), (14), and (15). The isopleths suggest the existence of at least four distinct regions for depths 0 – 26. These regions are as follows:

[3] This assumes that the comparison is with standard GP, in which the probability of selecting an internal node for crossover is uniform.

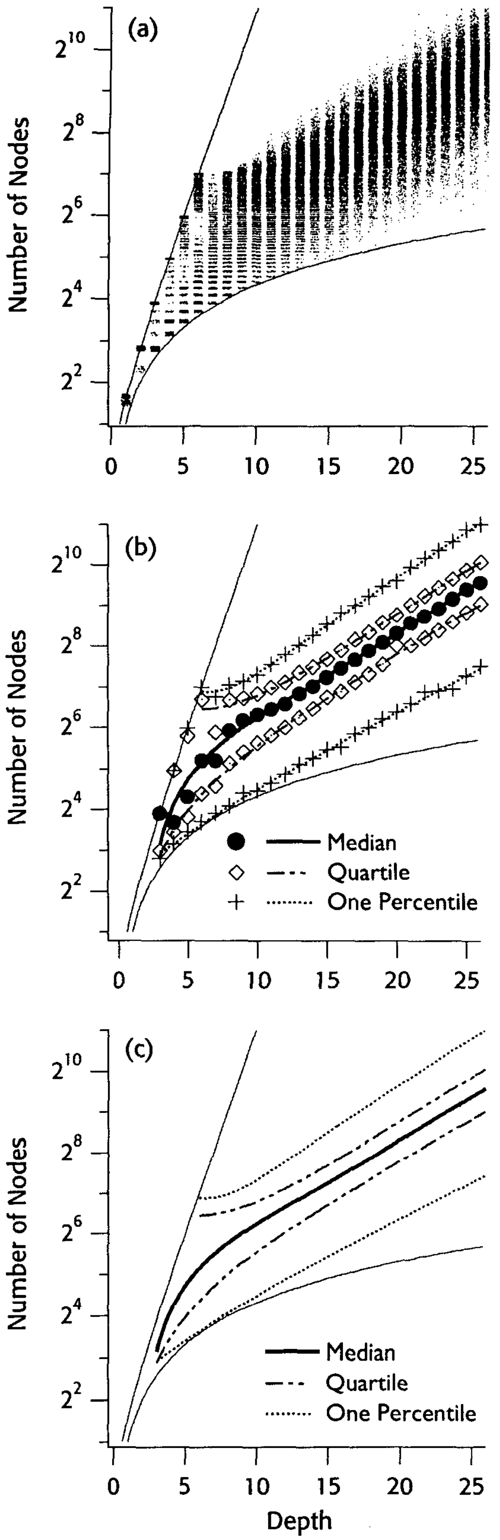


Figure 2. Derivation of isopleths of constant distribution for equation (11). Top plot shows the Monte Carlo results. Middle shows the numerical values of constant distribution, plus curve fits. Bottom shows just the curve fits.

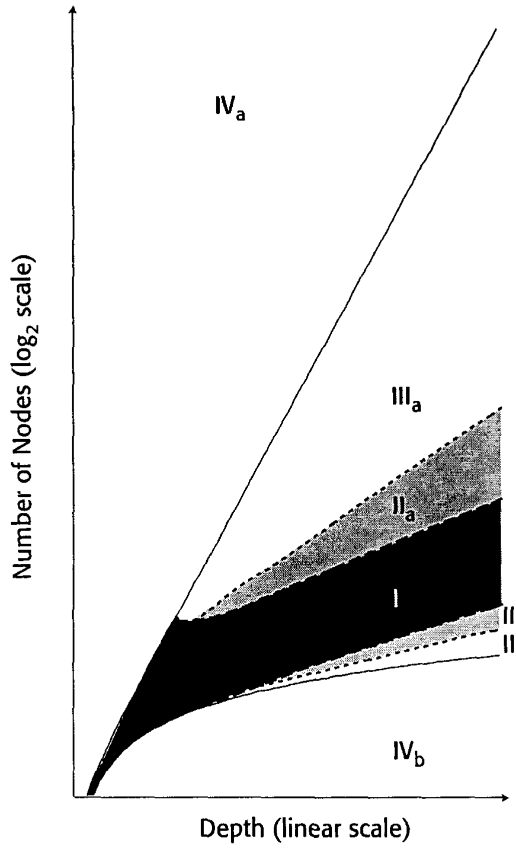


Figure 3. Predicted regions of search. There are at least four regions that the model predicts that ultimately limit where GP can search in size-shape space.

- *Region I.* This is the region where most solutions in standard GP occur (for binary trees). Full mixing of various size /shape subtrees in the derivation of solutions occurs here. The width of Region I is driven largely by population initialization.
- *Regions II.* These are the regions where increasingly fewer individuals appear the further away from Region I. Only partial mixing of size/shape subtrees occurs here, with mixing becoming non-existent towards the boundaries furthest away from Region I. Region II_a is delineated by the boundaries that are approximately located by the 99% isopleth for equation (14) and the 99% isopleth for equation (11). Region II_b is delineated by the boundaries that are approximately located by the 1% isopleth for equation (14) and the 1% isopleth for equation (11). The transition between Regions II and III is pronounced.
- *Regions III.* These are the regions where practically no individuals appear. Region III_a is delineated by the boundaries that are approximately located by the 99% isopleth for equation (14) and bound for full trees. Region III_b is delineated by the boundaries that are approximately located by the 1% isopleth for equation (14) and bound for minimal trees.

- *Regions IV.* These are the regions that are precluded from binary trees.

V. Comparison to Experimental Data

In earlier papers [8, 9], we published work on a tunably difficult problem in genetic programming that we have since named the binomial-3. We can use those empirical results as an instance to test against this paper's theoretical results. (Other empirical results could have also been used, but because of limited space, only the binomial-3 results were given.)

Figure 4 portrays several of the previously published data sets for tuning values of 1, 3, 10, 100, and 1000 (in order of increasing difficulty). Each dot represents a best-of-trial individual out of a population of 500; each graph represents the ensemble performance of 600 trials (i.e., a sampling of 30,000 individuals total per graph). In the left column of Figure 4 are the results of adjusted fitness versus the number of nodes; in the right, the results of number of nodes versus depth. The 99% and the 1% isopleths for equation (11) are superimposed on the graphs on the right.

In spite of significantly varying degrees of problem difficulty and wide variation in shapes and sizes across 3,000 statistically independent trials, better than 99% of all of the best-of-trial individuals fall in the area described as Region I and less than 1% in Regions II. No trials were found to be in Region III.

VI. Discussion and Summary

This paper has described a lattice-aggregate model for the purpose of describing the evolution of shape and size in genetic programming. It presumes nothing about the programmatic content associated with each node. It could also be argued that the described method of growth also presumes little, if anything about tree generation, manipulation, crossover, or mutation. The region boundaries that are indicated by the proposed model should apply to a broad range of problems with arity-2 functions at depths that have been examined by the model (i.e., depths 0 – 26). These results should also hold across various implementations and flavors of tree-manipulating systems, including those that are not of GP.[4]

It would be appropriate to say that the proposed approach is a structuralist one, which is distinct from those taken in previous work. There are three major departures:

- ***Structure produces its own behavior.*** The notion is not new and has occurred in other fields (e.g., see [12]). However, the notion is, perhaps, not as intuitive in an analysis of GP dynamics because programmatic content seems inextricably linked with the structure of that content. The proposed model offers sufficient explanatory power to account for a broad range of observed phenomena that has been obtained under a wide variety of domain-specific problems.

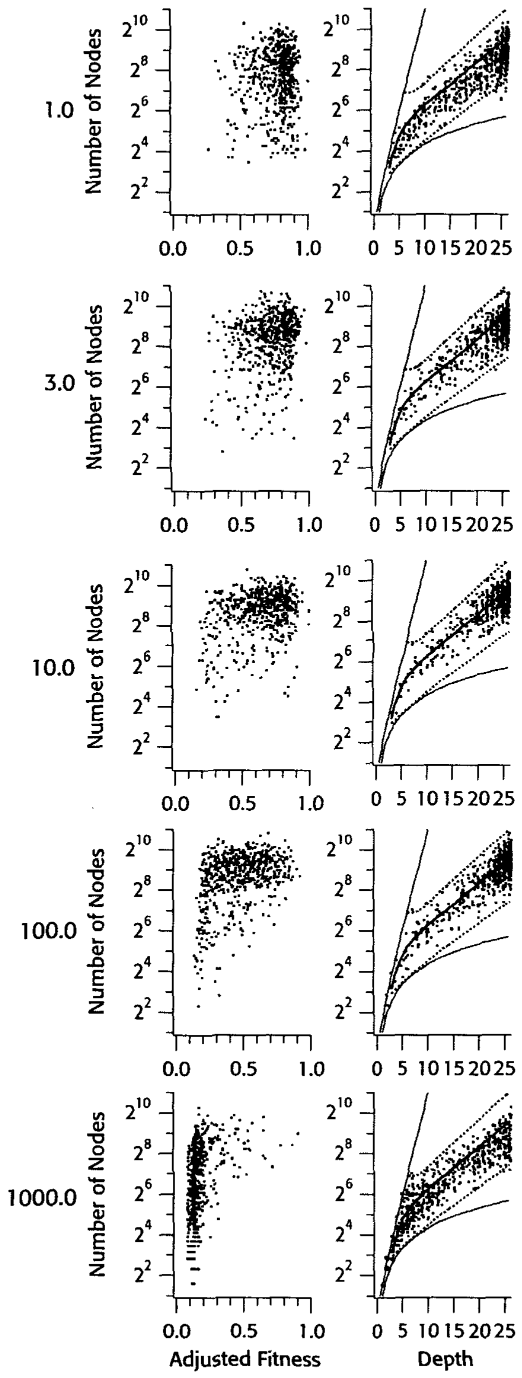


Figure 4. Comparison of proposed model with an instance of empirical data.

[4] The proviso, of course, being that the programmatic representation of an individual is not *articulated* (terminology mine). In an unarticulated representation, a tree *directly* represents a coding solution, as opposed to an articulated representation, in which a tree represents an intermediary. For example, Koza et al. [10] makes frequent use of articulated representations by employing Gruau's method of cellular encoding [11]. In this case, GP provides an intermediary program, which is then executed to manipulate graph structures that ultimately generate a final solution. In this case, the results of the lattice-aggregate model would likely apply to the intermediate, but not the final solution.

- *Structure is a predominant factor in determining where search occurs.* The use of a log scale in Figure 3 obscures, somewhat, just how small is the allowable search space in size and number of nodes, if only because of the use of a log scale. For example, there are 268,434,725 allowable combinations of size and number of nodes from depths 0 – 26. In this same depth range, Region I (where most search occurs) encompasses only 12,573 combinations. In other words, Region I represents only 0.005% of the entire allowable search space in size and number of nodes.
- *Content and manipulation of that content are secondary factors in the evolution of shape and size of individuals.* True, most of what occurs in this narrow "bottle" of size and shape has been of intense interest in the GP community. However, given the fairly significant effect of structure on the evolution of size and shape, one should (perhaps) scrutinize theories of code growth that presuppose fitness as a primary driver.

Not all work in GP presupposes that fitness causes code growth and that fitness is the primary driver in the evolution of size and shape. Langdon et al. [2], drew on early comparisons with Flajolet and Odlyzko's work on random trees [13]. Langdon has since amplified this comparison and has speculated that exponential growth is a primary driver [14]. However, there are three departures away from this other work, as well.

- *The proposed model is not designed to be a comprehensive means of generating random trees.* It does not use an efficient numbering scheme (cf. [4]). It is neither an efficient nor uniform method of tree generation (cf. Rémy's algorithm in [15]). It does suggest, however, the use of set and number theory, and invites comparison with work in DLAs and stochastic OL-systems.
- *The proposed model accounts for the deviations away from his estimates of mean performance in the depth 0 – 26 range.* General estimates of size and shape in [2, 3, 14] were significantly lower in the depth range 0 – 26 than those papers' indicated empirical results. Furthermore, the model estimates in those works also underestimates growth in this depth range in comparison to their own empirical work (e.g., Figure 10 in [14]).
- *The proposed model predicts for identifiable partitions in the search space of size and shape.* Predictions of tree size and shape in [14] are based on a combinatorial estimate of program likelihood—structure is neither explicit nor fundamental to the generation of this estimate. Consequently, the distribution of programs in the space of size and depth is relatively smooth within the boundaries of full and minimal trees. That paper's author has subsequently stated, "The use of a depth limit rather than size limit on the evolution of the program trees may encourage the formation of nearly full trees of the maximum permitted depth (p. 425, [3])." The proposed model predicts for the existence of identifiable partitions in the search space of size and shape, namely Regions I, II, III, and IV. Although the existence of Region IV has been well-established and is known to be taboo, the proposed model indicates that Regions III are also taboo. These regions are a direct consequence of structure and their presence does imply that there might not exist any fitness function in standard GP that will allow for search in Regions III. We leave to future work for empirical evidence of such regions.

VII. Conclusions

This paper described a general theoretical model of size and shape evolution in genetic programming. The proposed model incorporated a mechanism that was analogous to ballistic accretion in physics. The model has indicated a four-region partition of GP search space. It further indicated that two of these regions are not searchable by GP.

Acknowledgments

I thank the following individuals and organizations for their help: I. Kristo, S. Daida, L.M. Sander, CSCS U-M / Santa Fe Institute Fall Workshops, S. Stanhope, P. Litvak, S. Yalcin. D. Maclean, W. Worzel, and UM-ACERS teams Meta-Edge, Royal, Borges, and Binomial-3.

References

[1] T. Soule, J. A. Foster, and J. Dickinson, "Code Growth in Genetic Programming," in *Genetic Programming 1996: Proceedings of the First Annual Conference: July 28–31, 1996, Stanford University* J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds. Cambridge: The MIT Press, 1996, pp. 215 – 223.

[2] W. B. Langdon, T. Soule, R. Poli, and J. A. Foster, "The Evolution of Size and Shape," in *Advances in Genetic Programming 3*, L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. J. Angeline, Eds. Cambridge: The MIT Press, 1999, pp. 163–190.

[3] W. B. Langdon, "Scaling of Program Fitness Spaces," *Evolutionary Computation*, vol. 7, pp. 399 – 428, 1999.

[4] D. E. Knuth, *The Art of Computer Programming: Volume 1: Fundamental Algorithms*, vol. 1, Third ed. Reading: Addison–Wesley, 1997.

[5] T. A. Witten and L. M. Sander, "Diffusion-Limited Aggregation: A Kinetic Critical Phenomenon," *Physics Review Letters*, vol. 47, pp. 1400 – 1403, 1981.

[6] A. Lindenmayer, "Mathematical Models for Cellular Interaction in Development, Parts I and II," *Journal of Theoretical Biology*, vol. 18, pp. 280 – 315, 1968.

[7] P. Eichhorst and W. J. Savitch, "Growth Functions of Stochastic Lindenmayer Systems," *Information and Control*, vol. 45, pp. 217 – 228, 1980.

[8] J. M. Daida, R. B. Bertram, J. A. Polito 2, and S. A. Stanhope, "Analysis of Single-Node (Building) Blocks in Genetic Programming," in *Advances in Genetic Programming 3*, L. Spector, W. B. Langdon, U.-M. O'Reilly, and P. J. Angeline, Eds. Cambridge: The MIT Press, 1999, pp. 217–241.

[9] J. M. Daida, J. A. P. 2, S. A. Stanhope, R. R. Bertram, J. C. Khoo, S. A. Chaudhary, and O. Chaudhri, "What Makes a Problem GP-Hard? Analysis of a Tunably Difficult Problem in Genetic Programming," *Journal of Genetic Programming and Evolvable Hardware* , 2001.

[10] J. R. Koza, F. H. Bennett II, D. Andre, and M. A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*. San Francisco: Morgan Kaufmann Publishers, 1999.

[11] F. Gruau, "Cellular Encoding of Genetic Neural Networks," Laboratoire de l'Informatique du Parallélisme, Ecole Normale Supérieure de Lyon, Lyon, Technical Report 92–21, 1992.

[12] P. Senge, *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York: Doubleday/Currency, 1990.

[13] P. Flajolet and A. Odlyzko, "The Average Height of Binary Trees and Other Simple Trees," *Journal of Computer and System Sciences*, vol. 25, pp. 171 – 213, 1982.

[14] W. B. Langdon, "Size Fair and Homologous Tree Crossovers for Tree Genetic Programming," *Genetic Programming and Evolvable Machines*, vol. 1, pp. 95 – 119, 2000.

[15] L. Alonso and R. Schott, *Random Generation of Trees*. Boston: Kluwer Academic Publishers, 1995.