Chapter 15

THE EFFECTS OF SIZE AND DEPTH LIMITS ON TREE BASED GENETIC PROGRAMMING

Ellery Fussell Crane[1] and Nicholas Freitag McPhee[1]
[1] *University of Minnesota, Morris, Morris MN 56267, USA*

Abstract Bloat is a common and well studied problem in genetic programming. Size and depth limits are often used to combat bloat, but to date there has been little detailed exploration of the effects and biases of such limits. In this paper we present empirical analysis of the effects of size and depth limits on binary tree genetic programs. We find that size limits control population average size in much the same way as depth limits do. Our data suggests, however that size limits provide finer and more reliable control than depth limits, which has less of an impact upon tree shapes.

Keywords: size limits, depth limits, genetic programming, population distributions, tree shape

1. Introduction

The causes and effects of code growth in Genetic Programming (GP) have been extensively researched (Langdon and Poli, 2002). In order to avoid the negative repercussions of bloat, a variety of corrective measures are employed to keep program sizes in check (Poli, 2003; Silva and Almeida, 2003; Luke and Panait, 2002; Koza, 1992). One frequently used method is to employ a fixed limit on program size by restricting either the depth or the size of syntax trees.

While these limits have the desired effect of keeping the sizes down, little is known about what other impacts such limits might have on the dynamics of GP. Previous research has shown that decisions such as these can have significant effects on the behavior of runs (Gathercole and Ross, 1996) and on important structural features such as the size and shape distributions of populations (Poli and McPhee, 2003; McPhee and Poli, 2002). It would therefore be useful to better understand what structural effects size and depth limits might have, especially given their widespread use.

In (McPhee et al., 2004), we examined these issues using variable length linear structures. Here we extend that work to binary tree GPs. Several important differences exist between these two structures. In variable length linear structures, which are essentially unary trees, a size limit is exactly the same as a depth limit. This is not the case in binary trees, where it is possible to have a large depth and small size.

To evaluate the effects of depth and size limits, we performed a large number of empirical runs using various limits on a problem that induces bloat. In this chapter, we present and analyze data taken from these runs. The focus of this analysis is learning how depth and size limits affect the average size of individuals in a population and how they affect tree shape. From this analysis, we also draw conclusions about the differences between size and depth limits, and provide a tentative recommendation for the use of size limits.

Of special significance to this result is the fact that depth limits have been widely used to combat bloat in genetic programming. This is in part a result of the use of depth limits in John Koza's first two highly influential books (Koza, 1992; Koza, 1994). In explaining his use of depth limits, Koza noted

> ... that for the default value of 17 for the maximum permissible depth ... for a program created by crossover is not a significant or relevant constraint on program size. In fact, this choice permits potentially enormous programs. For example, the largest permissible ... program consisting of entirely two-argument functions would contain $2^{17} = 131{,}072$ functions and terminals. (Koza, 1994, p. 659)

This reasoning regarding depth limits certainly seems plausible, and depth limits have served the needed goal of reducing program size for over a decade. The results in this paper make it clear, though, that depth limits can *severely* constrain the space of trees that GP is likely to explore (supporting, *e.g.*, (Daida, 2003)). Using our definition of depth, a depth limit of 17 theoretically allows for a tree with 262,143 nodes. In doing this study we generated nearly 100 million individuals with depth limit 17 using a problem with a strong tendency to bloat. The largest individual we generated had a size of 341.

While many researchers (including Koza) have moved to using size limits, many continue to use depth limits. Such researchers may be under the mistaken belief that these limits aren't significantly affecting the dynamics of their systems. It is valuable, then, to better understand the impact of both of these widely used types of limits.

Surprisingly, our results show that, with appropriate values, both size and depth limits have nearly the same effects upon the average size of a population. The key difference between the two limits appears to be in how they affect the relationship between population average size and population average depth. Size limits do not seem to affect this relationship at all, while depth limits appear to bias the population towards slightly smaller average depths.

When comparing data from runs using a depth limit with that from runs using comparable size limits, we find that the distributions of sizes are extremely similar. The distribution of depths are also quite similar, but depth limits clearly restrict the depths much more than size limits restrict sizes. In both cases, the distributions are also very similar to the gamma-like distributions seen in earlier work on variable length linear structures.

In Section 2 we present background material necessary for understanding the rest of the chapter, including problem set up and definition of terms. In Section 3 we present and analyze data generated from runs using depth limits, and in Section 4 we do the same for runs using size limits. Based on questions arising from those two sections, we present an analysis of the impact of depth and size limits on tree shape in Section 5. After discussing future avenues of research on this topic in Section 6, we summarize our conclusions in Section 5.

2. Background

In this section, we define several terms and concepts used in this chapter. We also define the test problem and parameters we use.

Convergent average size and the strength of limits

In (McPhee et al., 2004), we defined the notion of a population's *convergent average size* in populations where a strong size limit is in place. We now extend this definition to account for tree depth.

In the presence of bloat, the average size and average depth of individuals in any population increase rapidly during the early generations of a run. After this initial period of unchecked tree growth, the population "hits" the size or depth limit, and the population average size remains at a relatively constant value over time. We refer to this value as the run's *convergent average size*, and more precisely define it as the mean of the population's average size over all of the generations after a run has converged. Figure 15-1 in Section 3 provides several examples of the population average size "converging" after reaching a limit.

Closely related to convergent average size is the notion of size or depth limit strength. Though all of the runs using limits that we examined experienced the convergence described above, it is clear from both (McPhee et al., 2004) and the work presented later in this chapter that not all limits cause the same amount of deviation from the convergent average size. Some limits cause very small amounts of variation from the convergent average size, and we refer to them as *stronger* limits than those which cause larger amounts of variation. Once again, this is easy to observe in Figure 15-1, where the larger limits clearly have more variation around the convergent average size than do the smaller limits.

We therefore define a size or depth limit's strength as the standard deviation of population average size over all of the generations after a run has converged.

Binary Syntax Trees

In our previous work (McPhee et al., 2004) we studied the impact of size limits on variable length linear structures. Those structures were essentially trees with two different unary functions (labeled 0 and 1) and a single type of leaf (labeled 0).

In this chapter we extend our previous work to binary trees, which are more frequently used than linear structures. We will have two functions or internal nodes, again labeled 0 and 1, and a single type of leaf or terminal node, again labeled 0. Thus individuals will consist of binary trees where every internal node is labeled with a 0 or a 1, and every leaf is labeled with a 0.

We also define the *size* of a tree to be the number of nodes (both internal nodes and leaves) in the tree. The *depth* of a tree is the number of edges along the longest path from the root node to a leaf. Thus, for example, a tree consisting of just a single leaf node has depth 0 and size one, while a full tree of depth 2 has size 7. More generally, the size of a full tree of depth d is $2^{d+1} - 1$.

Crossover Operator

Because our primary interest is the effect of size limits on code growth due to crossover, we focus exclusively on the standard subtree-swapping GP crossover operator. Thus there will be no use of mutation or any other genetic operators in this study.

The crossover operator acts by removing a non-empty subtree of an individual and replacing it with a subtree taken from another individual. In the work reported here, the subtrees are chosen uniformly from the set of all a tree's (non-empty) subtrees, including the entire tree itself. Note that we are *not* using any sort of bias. This includes, for example, the common bias of choosing 90% of the crossover nodes as internal nodes.

The One-Then-Zeros Problem

We have used the *one-then-zeros* problem in a number of previous studies of the effect of bloat and genetic operators on variable length linear structures (McPhee et al., 2004; Rowe and McPhee, 2001). This problem has the advantage of being simple to explain and amenable to schema theory analysis. It also has a natural tendency to bloat, *i.e.*, the average size of individuals tends to increase over time in a manner that is not directly dependent on their fitness.

One limitation of this previous work has been the restriction to variable length linear structures, while a large proportion of the GP community uses (non-unary) tree structures to represent expressions and programs. In this study we extend our earlier work to examine binary trees, and as a result need to generalize the one-then-zeros problem to the case of binary trees.

We thus introduce the *degree-N one-then-zeros problem.* In this problem the trees will consist of N-ary internal nodes, all labeled either 0 or 1, and leaf nodes, all labeled 0. Regardless of the degree, the fitness function is the same. The fitness of a tree (or string in the unary case) is 1 if the root node is labeled 1, and *all* other nodes (internal and leaf) are labeled 0; the fitness is 0 otherwise. Thus the only fit trees are those that follow the pattern, and those trees are all equally fit. Given this, our earlier work used the degree-1 one-then-zeros problem, and the work presented here uses the degree-2 one-then-zeros problem.[1]

Another important property of this problem is that it has no direct structural bias in the sense that (with two exceptions discussed below) the fitness function doesn't favor any particular sizes or shapes. Thus most of the data on sizes, depths, and tree shapes presented in this paper are being driven by the underlying dynamics of GP and standard subtree crossover, and not by particular properties of this problem. The two exceptions are (a) trees with a single (leaf) node are guaranteed to be unfit (since the only leaf label is 0), so there is a bias away from that particular tree shape and (b) this problem induces bloat, so there is a general pressure towards larger sizes and depths. If, as seems likely, the bloat is being driven in large part by the benefits of accurate replication (McPhee and Miller, 1995), then this can be obtained using any large tree, regardless of its shape and depth.

Experimental Setup

All the runs presented in this paper use the same parameters with the exception of the size or limit.

Number of generations All runs were for 3,000 generations.

Control strategy We use a non-elitist generational control strategy.

Initialization The populations were initialized entirely with fit individuals consisting of full trees of depth 2.

Size and depth limits These were implemented such that an otherwise fit individual received a fitness of 0 if its size was *strictly greater* than the size limit, or if its depth was *strictly greater* than the depth limit.

Selection mechanism We used fitness proportionate selection in these experiments. Since all individuals have either fitness 0 or 1, this reduces to uniform selection from the set of individuals with fitness 1.

[1] This could obviously be generalized further to account for trees with a mixture of node arities, but that would add complexity that would only complicate the current presentation.

Operators We used crossover exclusively in these experiments, so every individual was constructed by choosing two fit parents and performing subtree crossover as described above. There was no mutation or copying of individuals from one generation to the next.

In each run the convergent average size of the population was calculated by taking the mean value of the population averages in the final 1000 generations of the run. This region was selected because in all cases studied here the population had always converged by generation 2000.

We did a series of about 30 runs each for a variety of size and depth limits in order to better understand the larger trends. In particular, we looked at a series of depth limits ranging from 5 to 50. We chose a set of 10 values following a geometric (exponential) series, yielding the set of values {5, 6, 8, 10, 13, 17, 23, 29, 38, 50}. We chose the geometric series in an effort to broadly sample this range while still focusing more on the smaller values where (as was seen in (McPhee et al., 2004)) small differences were likely to be more significant. We then used a similar set of size limits ranging from 50 to 5,000, yielding the values {50, 83, 139, 232, 387, 645, 1077, 1796, 2997, 5000}. To better see the impact of some even larger size limits, we also did runs with size limits of 10,000, 12,000, and 15,000. Due to space limitations, only a representative sample of these runs are discussed in this paper, but the trends we present here hold for the entire data set.

3. Depth Limit Analysis

Figure 15-1 presents data about population average size over time for runs using a number of different depth limits. Each point in this graph represents the average size of the individuals in the population at a specific generation for one run. This provides excellent visual evidence that depth limits have an impact upon population size that is extremely similar to that of the size limits examined in (McPhee et al., 2004). In each case, we see the average size of the population increase rapidly in the early generations due to bloat and then quickly reach a convergent average size.

Similar to (McPhee et al., 2004), the strength of the limit being used seems to control how much variation there is once the convergent average size has been reached. In the case of the depth limit 8 data, for instance, this variance is very small- no more than about 3. The depth limit 50 data, however, varies by as much as two hundred. Clearly, the stronger depth limit of 8 provides much tighter bounds on the convergent average size than does the weaker limit of 50. This observation has led us to the more precise definition of size limit strength given in Section 2.

A key feature of Figure 15-1 is that the population average sizes of runs using depth limits are very small relative to the maximum size allowable by the

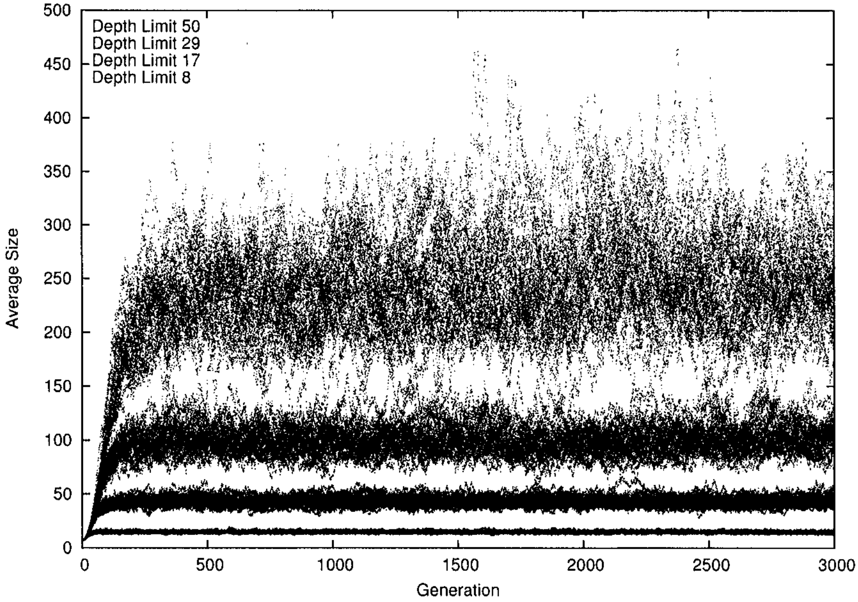

Figure 15-1. Population average size over time for a large number of runs using various depth limits. The "bands" of data correspond, from top to bottom, to runs using depth limits of 50, 29, 17, and 8.

depth limit. Depth limit 17, for instance, would allow for a maximum tree size of $2^{18} - 1$, or 262,143. The convergent average size of the runs using depth 17, however, is approximately 42. This is clearly very much smaller than the possible program sizes allowable by the limit, and it is not *a priori* obvious that this would be the case. As mentioned in Section 1, literature suggests that using a depth limit like 17 allows for the exploration of the space of very large trees. As we shall examine in Section 5, program sizes within a population appear to have a left skewed gamma distribution. This indicates that very little exploration of large sizes is in fact occurring. This is an important result, and suggests that existing assumptions about the behavior of depth limits are incorrect.

Interestingly, the average depth of the population appears very correlated with the population's average size. In other words, there seems to be very little variation in average size for a given average depth. Figure 15-2 illustrates this phenomenon by presenting the average sizes that were contained in Figure 15-1 and their corresponding average depths without accounting for time. Though there seems to be a general relationship between average size and average depth, it is also clear that each depth limit behaves slightly differently. There seems to be a "natural" relationship between average size and average depth

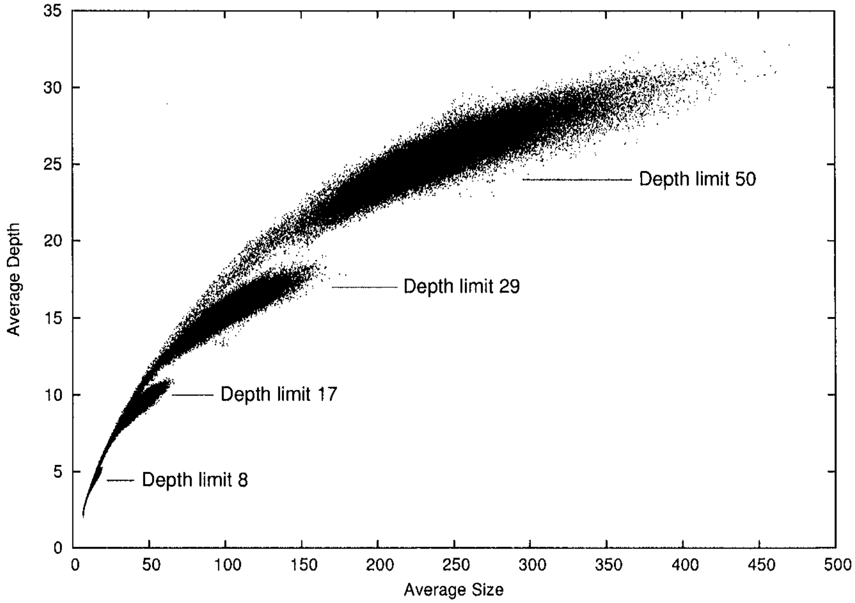

Figure 15-2. Population average size versus population average depth for runs using a variety of depth limits. The labeled clusters represent the space of convergent values for runs using different depth limits.

that populations would follow in the absence of any size or depth limits. (See Figure 15-4 for an additional example, and Section 6 for additional discussion.) Indeed, it appears that for all of the depth limits we examined, runs follow this "natural" relationship until they reach convergence, where they cluster slightly below the "natural" curve. As the corresponding depths for thee average sizes are lower than those in the natural relationship, this suggests that depth limits cause trees to become slightly more bushy once the population has reached convergence. We examine this idea further in Section 5.

An Exceptional Case

We performed hundreds of runs to generate the data presented in this study. As we have shown, the behavior exhibited by runs using certain depth limits is remarkably consistent. In Section 4, we show this to be true for size limits as well. There was one run out of the hundreds, however, which displayed startlingly different behavior.

This run, which used a depth limit of 23, had a convergent average size of about 50,000. Every other run using depth limit 23 had a convergent average size of approximately 70. Further, there were individuals in the exceptional case which reached sizes of upwards of 3.7 million nodes. These observations made us conclude, initially, that some form of programming or software error was responsible for the deviant behavior of the run. Further investigation revealed the truth: the run, though definitely abnormal, was valid.

Examination of the run's early generations suggested that, through a series of stochastic events, the population grew to consist of large, bushy trees, rather than the usual "stringy" trees which seem to be common in the other runs (see Section 5) and which are predicted by (Daida, 2003). This initial behavior likely produced a positive feedback loop which led to a continued increase in tree size. This resulted in the enormous average size observed after the population had reached convergence.

This exceptional run, therefore, provides us with an example of the kind of behavior implied by the quote in Section 1. Though we are in no position to claim just how frequently this actually occurs, the fact that it happened only once in the hundreds of runs we performed suggests it is very rare. It also suggests disturbing implications about the reliability of depth limits. Though this errant run may be the exception (and our data certainly supports that idea), the fact that it is possible to unpredictably have program sizes balloon vastly beyond normal ranges makes the choice of using depth limits questionable. Size limits, for instance, would not have allowed the behavior described above, as they explicitly limit program size.

There are at least two specific concerns about the possibility of this sort of aberrant run. The first is the obvious implications for computing resources. Using our hardware, for instance, a typical run using a depth limit of 23 took approximately five minutes to complete. The exceptional run took about 8 hours to complete. Though the times are, of course, specific to both our problem and hardware, it seems reasonable to assume a proportionate amount of resources would be required for a similar run using other problems and hardware. Second, and perhaps more important, is the problems of doing statistical analysis on a set of runs containing such outlier results.

4. Size Limit Analysis

Figure 15-3 presents data in much the same fashion as Figure 15-1, though for runs using a variety of size limits rather than depth limits. Like the runs using depth limits, discussed in Section 3, we see a distinct convergence in both size and depth after a very small number of generations, again mirroring the the findings of (McPhee et al., 2004). Figure 15-3 and 15-1 are in fact extremely similar. The scales of the two graphs differ, but this is simply due to the disparate strengths of the limits being shown. From a comparison of the

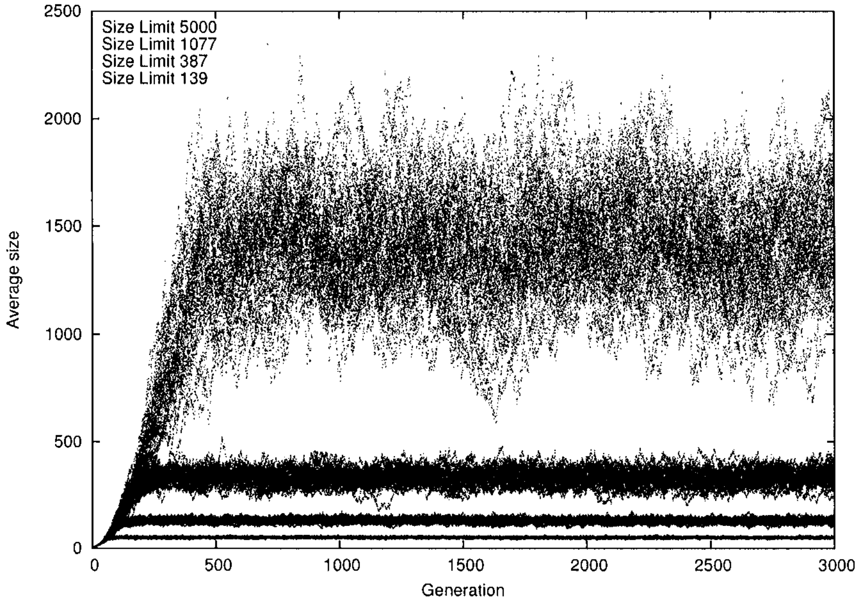

Figure 15-3. Population average size over time for a large number of runs using various size limits. The "bands" of data correspond, from top to bottom, to runs using size limits of 5000, 1077, 387, and 139.

two figures, it appears that size and depth limits have almost the same, if not identical, effects upon population average size.

This is an important observation, as it is not conceptually obvious that size and depth limits would restrict population sizes in a similar way. Indeed, the fact that depth limits, which could conceivably allow an enormous range of sizes, behave in the same way as size limits, which explicitly limit tree size, is quite remarkable.

Figure 15-4 shows the relationship between population average size and population average depth, as we did in in Figure 15-2 in Section 3. Unlike the depth limits analyzed in Section 3, the size limits used here do not display any marked deviance from the "natural" relationship between average size and average depth discussed earlier. This is so much the case, in fact, that it becomes hard to discern which data corresponds to which size limit.

By comparing Figures 15-2 and 15-4, several inferences can be made. The "natural" relationship between average size and average depth for this problem appears the same whether depth or size limits are used. Size limits seem to have no impact upon this relationship. Depth limits, however, evidently bias this relationship to some extent by lowering the depth slightly. Whether this bias

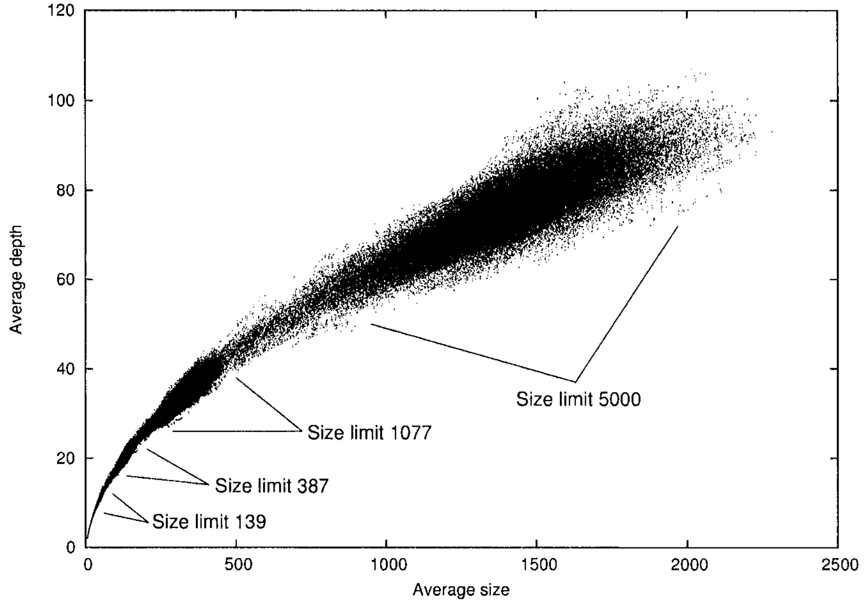

Figure 15-4. Population average size versus population average depth for a number of size limits. The labeled clusters represent the space of convergent values for runs using different depth limits.

has a positive or negative impact upon a given run is almost certainly problem dependent, and there is no evidence to suggest what the extent of the bias might be for problems with fitness functions that alter tree shape.

Sub-Quadratic Relationship Between Size and Depth Limits

Given the close relationship between size and depth limits, an obvious question is, for a given depth limit, what size limit is roughly equivalent in the sense that it yields a similar convergent average size? An initial analysis of our data suggests that, at least for this problem, the relationship can be roughly approximated by $S \approx 0.410063 * D^{1.92}$, where S is the size limit and D is the depth limit. The details of the constants aren't important except to note that the exponent is slightly less than two. Thus the "equivalent" size limit grows roughly with the square (or less) of the depth limit instead of the exponential relationship one might expect.

From a practitioner's standpoint, this reinforces the idea that one can use size limits to achieve a qualitatively similar results to those obtained with depth limits. It also suggests that "equivalent" size limits are polynomial (quadratic

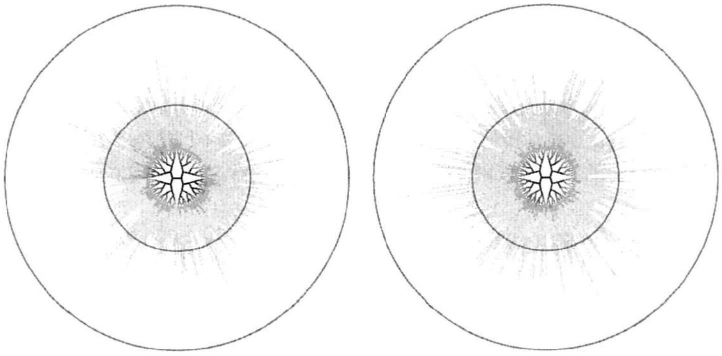

Figure 15-5. Visualization of all of the 10,000 individuals taken from the last 10 generations (generations 2991-3000) of a representative run using depth limit 17 (on the left) and size limit 118 (on the right). The inner circle is at depth 17, and the outer circle is at depth 40.

or slightly sub-quadratic in our case) in the depth limit and not exponential as one might expect.

5. Impact of Limits on Tree Shapes

In the previous section we found that there are depth and size limits that lead to similar convergent average sizes. We saw earlier, however, that depth limits tend to push the tree shapes off the "natural" shape and size limits don't (see, *e.g.*, Figs 15-2 and 15-4). This then raises the question of whether the shapes of the trees using "equivalent" size and depth limits are in fact different. To see this we used the visualization techniques of (Daida et al., 2005) to visualize the entire population of a single run for two pairs of limits (depth limit 17 and size limit 118, and depth limit 50 and size limit 600) that are roughly equivalent. By equivalent, we mean that in each pair the size and depth limits produced similar convergent average sizes.

Fig 15-5 shows a visualization of every individual present in each of the last 10 generations (*i.e.*, generations 2991 to 3000)[2] of a representative run using depth limit 17 (on the left) and size limit 118 (on the right). The inner circle is at depth 17, so the size limit case has more trees that exceed that depth, and they exceed it by considerably more. Thus while the *average* sizes and depths of these two runs are extremely close, their distributions seem to be somewhat different.

[2] Note, then, that each graph is displaying an aggregate view of 10,000 individuals.

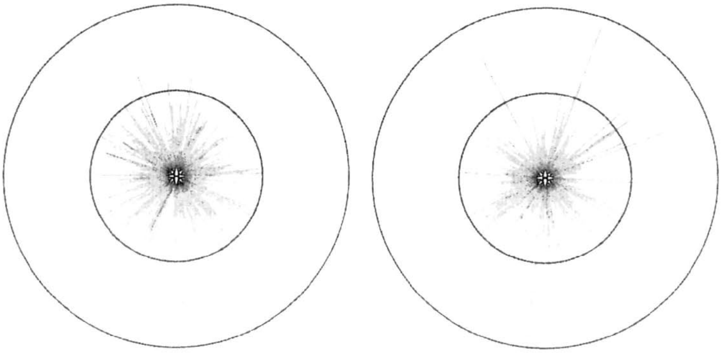

Figure 15-6. Visualization of the entire population of 1000 individuals in the final generation (generation 3000) of a representative run using depth limit 50 (on the left) and size limit 600 (on the right). The inner circle is at depth 50, and the outer circle is at depth 100.

Figure 15-6 shows a visualization of all the individuals present in the last generation (*i.e.*, generation 3000) of a representative run using depth limit 50 (on the left) and size limit 600 (on the right). The inner circle is at depth 50, and again the size limit case has more trees that exceed that depth, and they exceed it by considerably more.

One of the key features of the visualizations in (Daida et al., 2005; Daida, 2003) was the lack of variety of tree shapes, with the majority of the trees sharing a significant amount of structure. In our visualizations, however, there is a *much* wider variety of sizes and shapes. In Figure 15-5, for example, there are at least a few trees containing branches in almost every part of the space up to depth 17, whereas the population visualizations in (Daida et al., 2005; Daida, 2003) cover only a tiny fraction of the space.

It seems likely that this is a result of structural differences between the degree-2 one-then-zeros problem used here, and the regression problems used in (Daida et al., 2005; Daida, 2003). In the one-then-zeros problem, all that matters is the simple pattern of having a one at the root and zeros elsewhere (which is largely independent of tree size and shape) and avoiding size or depth limits as appropriate. This implies that the "meaning" of subtrees is largely independent of context in the one-then-zeros problem, so a subtree can be moved, via crossover, to an entirely different location in the tree without (in many cases) changing the fitness. This is in strong contrast to most GP problems (like regression), where context is crucial to the "meaning" of a subtree, and moving a subtree to a different location often has a large, and typically detrimental, effect on the fitness. This context dependence presumably plays a large role in the uniformity of shapes seen in (Daida et al., 2005; Daida, 2003), just as the lack

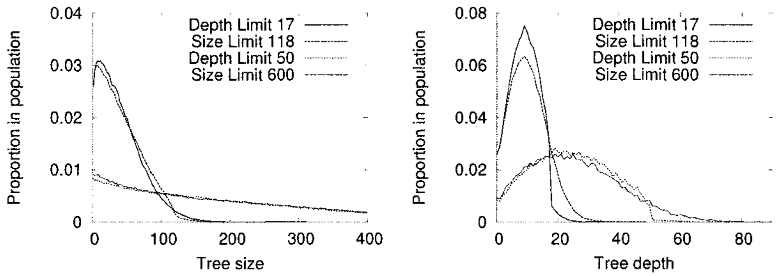

Figure 15-7. Distribution of sizes (left) and depths (right) for depth limits 17 and 50 and size limits 118 and 600. Note the different scales for proportions.

of this sort of dependence presumably plays a large role in the dispersion of shapes in our examples.

Figures 15-5 and 15-6 speak volumes about the distribution of tree *shapes*, but leave open the question of how the sizes and depths are distributed. Previous work on variable length linear structures (Poli and McPhee, 2003; McPhee et al., 2004) has shown a strong tendency for the size distribution of populations to be similar to a gamma distribution, with a very large proportion of short strings balancing out a small number of much longer strings. An open question has been whether these results would generalize to N-ary trees, and the distributions in Figure 15-7 suggest that they do.

The graphs in Figure 15-7 show the distribution of sizes depths for the same two pairs of depth and size limits used in Figures 15-5 and 15-6. In all cases the distributions are again very similar within each pair, lending weight to the idea that corresponding size and depth limits can have very similar impacts on population structure. Note, for example, the size distributions for depth limit 50 and size limit 600, which are nearly indistinguishable over the bulk of their range.

We also find in all cases that the distributions are similar to the gamma-like distributions found in earlier work on variable length linear structures. Thus we find here that the distributions of both sizes and depths are skewed significantly to the left, with a large number of small sizes/depths being balanced by a much smaller number of large sizes/depths.

These graphs also point out the specific impacts of size and depth limits on particular distributions. In the size distribution graph we see a sharp dip in the size limit 118 distribution right around size 118. There is a similar, but smaller, dip in the size limit 600 distribution that is off the right hand side of the graph. There are also similar, but more pronounced, dips in the depth distributions for the runs using depth limits, which again suggests that depth limits are having a stronger (perhaps undesirable) impact on our population distributions.

It's worth noting that in each case where a limit creates a dip in the corresponding distribution, there is perforce an increase in some other part of the distribution to compensate. In the depth limit 17 depth distribution, for example, this is seen quite clearly as a significant increase in the proportions of depths around 10, indicating that a size limit of 118 allows for a slightly broader exploration of a range of depths than does the otherwise similar depth limit of 17. Similarly, in the size limit 600 size distribution the small dip (not visible in this graph) leads to a small rise in the proportions of very small trees when compared to the depth limit 50 distribution. These dips and compensations are consistent with predictions from the "theory of holes" (Poli and McPhee, 2003; McPhee et al., 2004), where schema theory analysis shows that limits like these (in the case of variable length linear structures) lead to the sort of shifts in distributions seen in this work.

6. Future Work

This study directly addresses one of the major questions from (McPhee et al., 2004), namely how well the distribution results from variable length linear structures generalize to N-ary trees. Two other questions from that earlier paper remain open, however. First, prior results on different mutation operators (Rowe and McPhee, 2001) and combinations of genetic operators (McPhee and Poli, 2002) suggest that these can themselves act to limit size and depth, so studying their interaction with explicit limits might be fruitful. Second, preliminary data suggests that population size plays a significant role in determining the strength of limits and the convergent average sizes and depths. The specifics of this relationship are unclear at the moment and warrant further investigation.

Additionally, this work on binary trees raises questions about the "natural" relationship between size and depth (see Figures 15-2 and 15-4). This seems likely to be related to both the the Flajolet line (Langdon and Poli, 2002, Chapter 11) and Region I of (Daida and Hilss, 2003). Exploring the details of these relationships is beyond the scope of this paper, but such an exploration would likely be fruitful.

The work presented here is all for a single "toy" problem, and a key question is obviously how well the results generalize to other problems. Since our results on the relationship between size and shape look quite similar to results obtained by other researchers with a broader range of problems (Langdon and Poli, 2002; Daida and Hilss, 2003, Chapter 11), we can hope that other results will generalize (at least qualitatively) as well. As seen in Section 5, however, there is at least one important structural difference between the degree-2 one-then-zeros problem and the regression problems studied in (Daida et al., 2005). Thus some additional work is clearly necessary to better understand which results will generalize to other problems, and to what degree.

The exceptional case discussed in Section 3 appears to be reasonably rare (we only saw such a thing once in over 300 runs), but we currently lack sufficient data to estimate how often it is likely to occur. Given how profoundly different the performance and results of such a run are going to be, knowing more about their frequency would be helpful.

We've seen (*e.g.*, Section 5) that there are size and depths limits that lead to similar outcomes. It would be useful to know more about the nature of that relationship, with the ultimate goal being the development of a model with predictive power that would allow us to map from size limits to roughly equivalent depth limits and vice versa.

7. Conclusion

Throughout this chapter, we have examined the behavior of depth limits and size limits on binary tree genetic programs. The results of this investigation have yielded several major findings.

In Section 3 we show that depth limits, contrary to GP folklore, do not typically allow for large ranges of tree size. Instead, we observe that they produce tree sizes that are nearly the same as those produced by size limits with maximum sizes that are orders of magnitude below the maximum size possible using the depth limit. In only one case out of the hundreds of runs generated for this study did we observe tree sizes that were anywhere near the maximum possible using depth limits. This leads us to conclude that although in the vast majority of cases depth limits seem to control code growth very similarly to size limits, their consistency is questionable. Furthermore, since the one case where this inconsistency manifested took vastly more computational resources than the normal cases and led to results that were wildly different from the other cases, the unreliability of depth limits is worrying.

In both Sections 3 and 4 we show that there is a well defined relationship between population average size and population average depth which is visible using either size limits or depth limits. Size limits did not appear to affect this relationship in any meaningful way, though depth limits appeared to add a small yet significant bias towards smaller depths. Though it is unclear how strong this bias actually is, lack of understanding regarding it supports the idea that using depth limits holds a great deal of uncertainty.

Visualization of our populations suggests that runs with size limits are able to explore more of the tree space than those with depth limits. We also showed that both types of limits induce gamma-like distributions of both sizes and depths, similar to those seen in earlier work with variable length linear structures (Poli and McPhee, 2003; McPhee et al., 2004).

Another finding of this study has been that our observations of how size limits affect population average size were almost identical to those made in our earlier work using variable length linear structures (McPhee et al., 2004). This has important implications about the generalizability of research using linear structures. Use of analytical tools such as schema theory on N-ary syntax trees is exceedingly difficult, which makes the use of linear structures to simplify analysis desirable. A question that has always arisen from such analysis is whether the results can be generalized to N-ary trees. We show in this study that, in at least the context we use here, many of them do.

It's important to remember that all these results are in the context laid out in Section 2, including the use of the degree-2 one-then-zeros problem, so care must be taken to not over generalize. We do believe however, that many of these results will generalize, at least qualitatively, to a variety of other problems.

Acknowledgments

We wish to thank Wolfgang Banzhaf and Jason Daida for their helpful reviews of this chapter. We would also like to thank all of the participants in the GPTP workshop for their insightful comments and suggestions.

The population visualizations in Section 5 were generated using Daida, *et al*'s Mathematica notebook; we are very grateful for their willingness to share these resources. The remainder of our graphs were generated with gnuplot, and we greatly appreciate the work of all the authors of gnuplot and the many other open source tools that were used in generating and analyzing our data, and presenting our results.

References

Daida, Jason M. (2003). What makes a problem GP-hard? A look at how structure affects content. In Riolo, Rick L. and Worzel, Bill, editors, *Genetic Programming Theory and Practice*, chapter 7, pages 99–118. Kluwer.

Daida, Jason M. and Hilss, Adam M. (2003). Identifying structural mechanisms in standard genetic programming. In Cantú-Paz, E. et al., editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1639–1651, Chicago. Springer-Verlag.

Daida, Jason M., Hilss, Adam M., Ward, David J., and Long, Stephen L. (2005). Visualizing tree structures in genetic programming. *Genetic Programming and Evolvable Machines*, 6. Prepublication Date: 6 August 2004.

Gathercole, Chris and Ross, Peter (1996). An adverse interaction between crossover and restricted tree depth in genetic programming. In Koza, John R., Goldberg, David E., Fogel, David B., and Riolo, Rick L., editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 291–296, Stanford University, CA, USA. MIT Press.

Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA.

Koza, John R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts.

Langdon, W. B. and Poli, Riccardo (2002). *Foundations of Genetic Programming*. Springer-Verlag.

Luke, Sean and Panait, Liviu (2002). Lexicographic parsimony pressure. In Langdon, W. B. et al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 829–836, New York. Morgan Kaufmann Publishers.

McPhee, Nicholas Freitag, Jarvis, Alex, and Crane, Ellery Fussell (2004). On the strength of size limits in linear genetic programming. In Deb, Kalyanmoy et al., editors, *Genetic and Evolutionary Computation – GECCO-2004, Part II*, volume 3103 of *Lecture Notes in Computer Science*, pages 593–604, Seattle, WA, USA. Springer-Verlag.

McPhee, Nicholas Freitag and Miller, Justin Darwin (1995). Accurate replication in genetic programming. In Eshelman, L., editor, *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)*, pages 303–309, Pittsburgh, PA, USA. Morgan Kaufmann.

McPhee, Nicholas Freitag and Poli, Riccardo (2002). Using schema theory to explore interactions of multiple operators. In Langdon, W. B. et al., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 853–860, New York. Morgan Kaufmann Publishers.

Poli, Riccardo (2003). A simple but theoretically-motivated method to control bloatin genetic programming. In Ryan, C. and *et al*, editors, *Proceedings of the Sixth European Conference on Genetic Programming (EuroGP-2003)*, volume 2610 of *LNCS*, pages 204–217, Essex, UK. Springer Verlag.

Poli, Riccardo and McPhee, Nicholas Freitag (2003). General schema theory for genetic programming with subtree-sw apping crossover: Part II. *Evolutionary Computation*, 11(2).

Rowe, Jon E. and McPhee, Nicholas F. (2001). The effects of crossover and mutation operators on variable length linear structures. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco, California, USA. Morgan Kaufmann.

Silva, Sara and Almeida, Jonas (2003). Dynamic maximum tree depth. In Cantú-Paz, E. et al., editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1776–1787, Chicago. Springer-Verlag.