Chapter 12

CONTENT DIVERSITY IN GENETIC PROGRAMMING AND ITS CORRELATION WITH FITNESS

A. Almal, W. P. Worzel[1], E. A. Wollesen[1] and C. D. MacLean[1]
[1]*Genetics Squared Inc., 210 S. Fifth Ave, Suite A, Ann Arbor, MI 48104*

Abstract A technique used to visualize DNA sequences is adapted to visualize large numbers of individuals in a genetic programming population. This is used to examine how the content diversity of a population changes during evolution and how this correlates with changes in fitness.

Keywords: genetic programming, diversity, chaos game, fitness correlation.

1. Introduction

Genetic Programming (GP) has borrowed theory extensively from Genetic Algorithms (GAs). It is widely accepted that the building-block hypothesis (Holland, 1975) holds true for GP and Poli has proven a Schema Theorem (Holland, 1975) for GP (Poli and McPhee, 2001).

At the same time, there have been voices of dissent. Angeline (Angeline, 1997) has described crossover as "macro mutation" that is as likely to be destructive of existing building blocks as it is to create new building blocks. Daida *et al.* (Daida et al., 2003) has suggested that GP is dominated by structural considerations that significantly constrain the possible search space, thus limiting the importance of the Schema Theorem. McPhee and Hopper (McPhee and Hopper, 1999) and Daida *et al.* (Daida, 2004) both showed that the genetic material in the final generation of evolution could be traced to a very limited subset of the initial generation. Daida *et al.* (Daida, 2004) also suggests that tournament selection is better than fitness proportional selection at reaching a solution precisely because diversity is reduced quickly to a limited set of building blocks that are then shuffled to find their best combination. This is contrary to accepted wisdom that it is desirable to maintain diversity as long as possible

in order to search for the best building blocks available. Instead Daida *et al.* (Daida, 2004) argues that for reasons of computational efficiency, it is better to allow fast convergence on a small number of building blocks that are selected from the initial populations. Without early convergence, a GP system will be forced to spend an inordinate amount of time evaluating inferior individuals.

This paper introduces a means for visualizing Genetic Programming content and structure so that aspects such as diversity and structure within a population may be examined during evolution and related to the progression of fitness. This may be used to test some of the theories described above as well as giving GP users some insight into the appropriateness of GP parameter settings for the problem being solved.

2. Content Mapping

Chaos Game

Genetic programming systems, as with other evolutionary systems, are generally not in equilibrium. The dynamics of the system are usually non- linear in behavior and genetic programming systems tend to be very sensitive to initial conditions. Due to these properties, a genetic programming system may be described as a chaotic dynamical system. By applying chaos theory to the dynamics of evolution in GP, it may be possible to better understand the emergence of non-random patterns during the evolutionary process.

The Chaos Game is an interactive approach to teaching students about fractals and, indirectly, about chaotic dynamical systems. From a starting point within a simple geometric figure such as a triangle or a square, a point is plotted some fraction of a distance toward one of the figure's vertices. This is repeated, varying the targeted vertex until a figure emerges. For example, if a triangle is used and a point is plotted half way from the current position to the targeted vertex and the vertex is randomly selected, a Sierpinski triangle is created. This may be turned into a game by providing a target for the line to reach and requiring the student to pick the vertex toward which he or she moves (Voolich and Devaney, 2005).

If a square is used instead of a triangle and each corner is labeled with one of the bases in DNA (*i.e.*, A, T, C and G), then each sequence of DNA will create a different graph. By plotting multiple sequences in this way, the Chaos Game can be used for a variety of things such as identifying recurring sequences, and identifying functional regions of DNA (Jeffrey, 1990) (V. Solovyev, 1993). This method is now widely used for sequence analysis and in particular for the discovery of particular sequences of interest for further analysis.

The Circle Game

By moving from a polygon to a circle, a more flexible system is created with the values being mapped distributed evenly around the circle. This is equivalent to a polygon inset within a circle with the vertices touching the edge of the circle. By using this to plot individuals in genetic programming populations, the emergence of structure and content "motifs" during evolution may be tracked.

In this approach, to represent the content of a GP expression the tokens being tracked (*i.e.*, terminals and operators) are evenly spaced around a circle. By rendering a GP derived function as a linear string, the sequence of tokens may be plotted. As in the Chaos Game, beginning at the center of the circle, a point is plotted from the current location to a point halfway to the location of the point on the circle where the next token in the function lies. This is repeated until the function has been fully graphed in the circle and then repeated for all members of the population. (Koelle,) An alternative version plots a line from the current location to a point half way to the appropriate vertex rather than a single point. This has the virtue of showing ordered patterns that repeat within the population but at the cost of creating a more tangled plot.

It can be seen that the chaos game can capture the content diversity and show the emergence of patterns, however if we want to identify the 'motifs,' it requires us to represent the structure of the expression as well since $a \times b + c$ is quite different from $a \times (b + c)$ but their content plots would be identical. In order to do this we propose a modified approach that represents both the structure and the content.

The equation shown in Equation 12.1 can be easily mapped into a binary tree structure as shown in Figure 12-1.

$$[h!](Op3\ (Op2\ T1\ T2)\ T4) \tag{12.1}$$

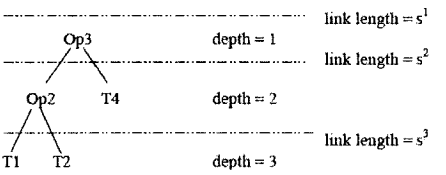

Figure 12-1. Binary Tree Representation of Equation 12.1

In the modified algorithm, the nodes are plotted using the rules for the circle game. However, the length of the links for these nodes are given by s^d, where s is a scaling factor arbitrarily chosen between 0 and 1, and d is the depth of the node the link is leading to in the binary tree. Also the link for a node in the plots should originate from the location of its parent. For example, the

sequence of plotting for Equation 1, will be: plot a line from origin half the distance ($s = 0.5$) towards $Op3$, move a quarter distance towards $Op2$, move one-eighth of the distance towards $T1$, come back to the starting point for $Op2$, move one-eighth of the distance towards $T2$, come back to $Op3$ and move a quarter distance towards $T4$. The scaling parameter s can be chosen to be any arbitrary value, keeping in mind that it controls the visual divergence in the plot. Figure 12-2 shows an example of this for the expression shown in Equation 12.1.

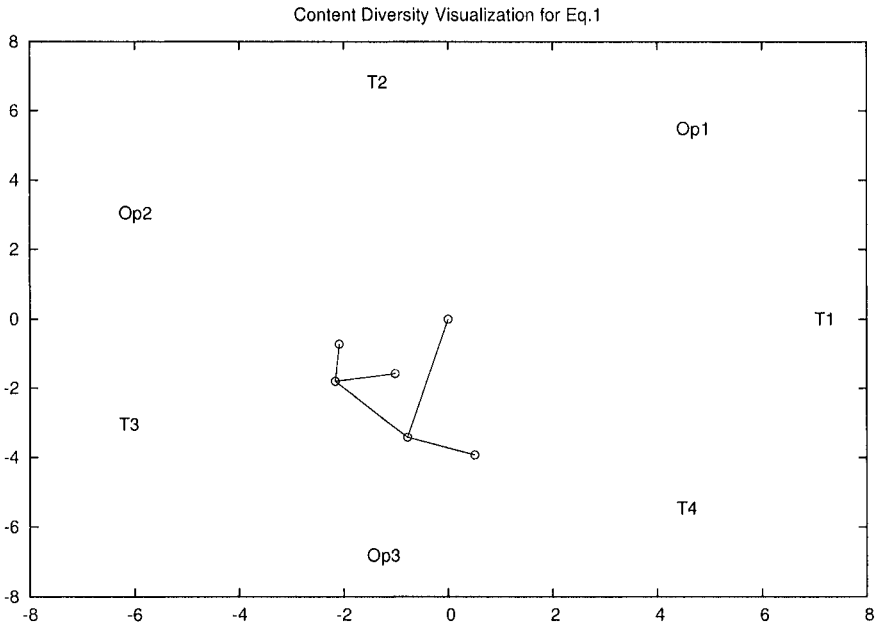

Figure 12-2. Modified Circle Plot for Equation 12.1

If we add Equations 12.2 and 12.3 and plot all three equations together using using different pens, we get the plot shown in Figure 12-3. This shows that similar expressions can be distinguished but at the same time their structural and content similarities can be spotted.

$$(Op3\ (Op2\ T1\ T3)\ T4) \tag{12.2}$$

$$(Op2\ (Op1\ T1\ T2)\ T2) \tag{12.3}$$

Showing Content Diversity During Evolution. By looking at the structural content plots for an entire population during evolution we can gain a glimpse of the dynamic changes in structure and content. There are two different types

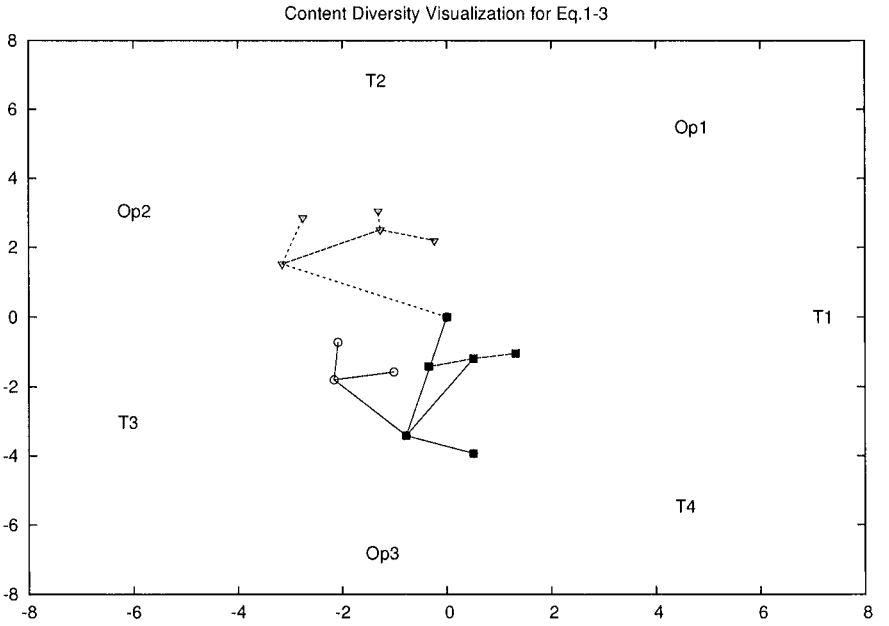

Figure 12-3. Circle Plot or Equations 12.1-12.3

of plots we use to study evolution. In one we plot the entire graph and in the other we plot the nodes and the links are omitted. Both of these methods have unique qualities, the former tells us about the connectivity of the nodes(an essential feature for finding the motifs) and the latter approach gives a nice visual representation of the diversity during evolution. Especially interesting are the emergence of the circular fractals in these plots. These suggest that the GP system is searching for the appropriate combination of elements in a structure.

Figure 12-4 shows a population of individuals at generation 0 of a run while Figure 12-5 shows the population at generation 10. Figure 12-6 shows it at generation 20 and 12-7 at the final generation, generation 40. By comparing these images we can see the appearance of shared content and structure within the population emerging from the random "ball of string" in generation 0. By the final generation shown in Figure 12-7, we can see how the content diversity has been reduced to a comparatively small number of variables and the structure is fairly similar across the individuals in the population.

The plots of only the nodes for the same problem follow in Figures 12-8 through 12-11.

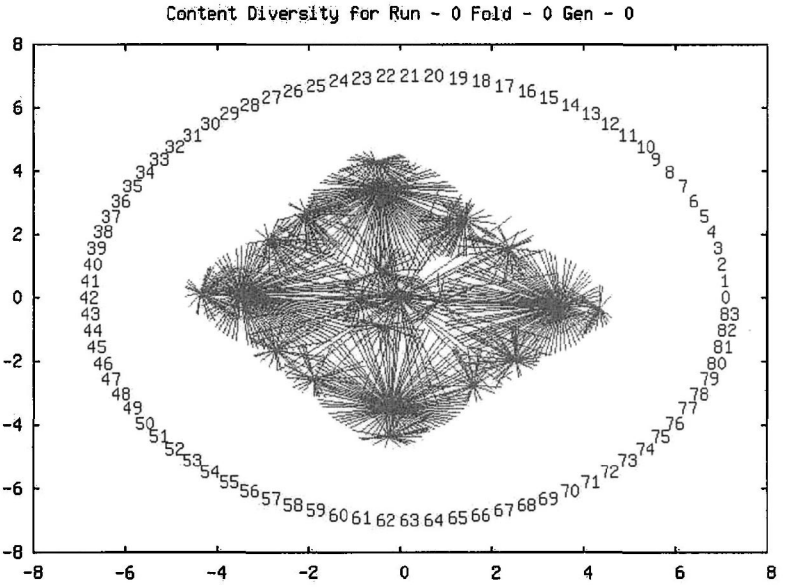

Figure 12-4. Generation 0 Content Plot

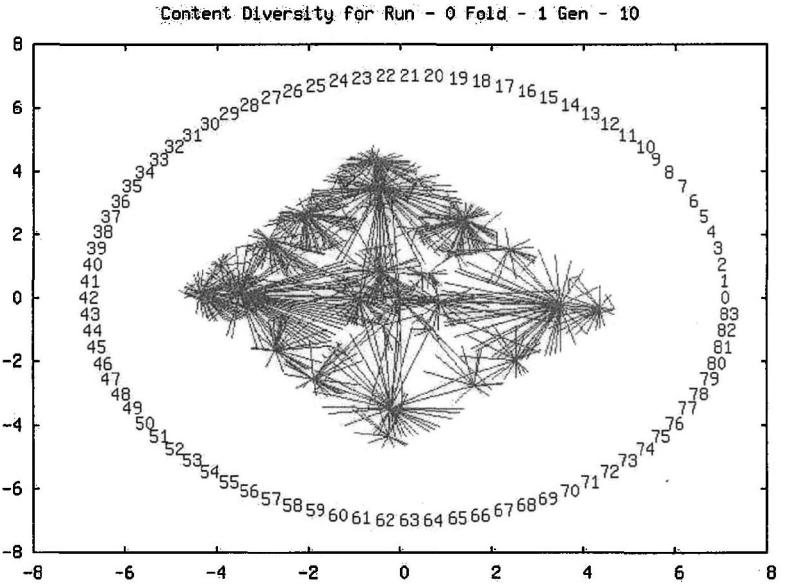

Figure 12-5. Generation 10 Content Plot

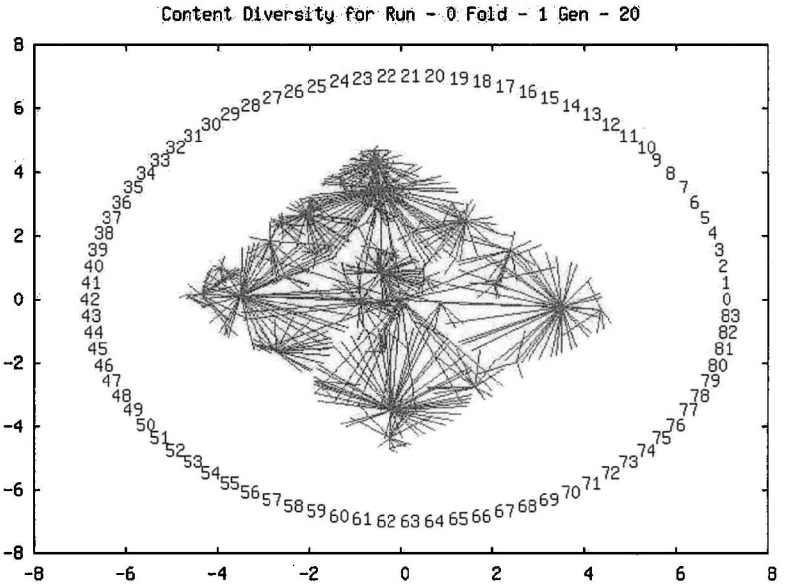

Figure 12-6. Generation 20 Content Plot

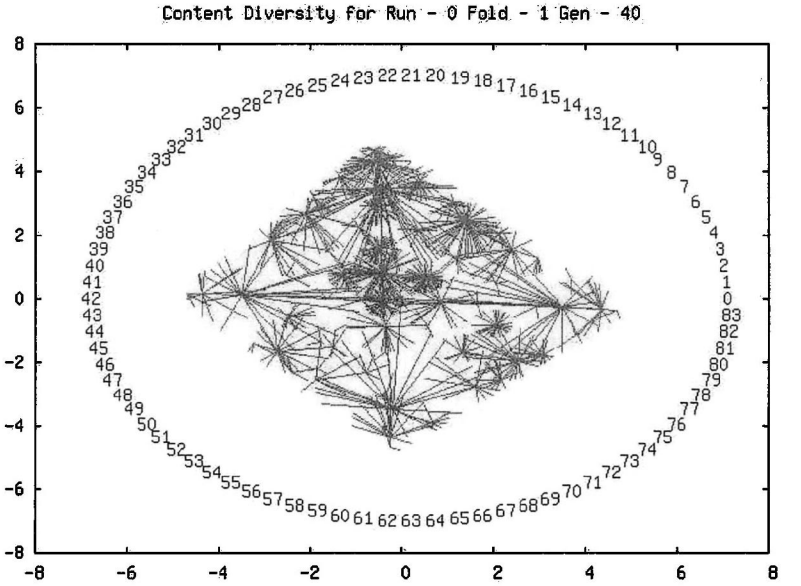

Figure 12-7. Generation 40 Content Plot

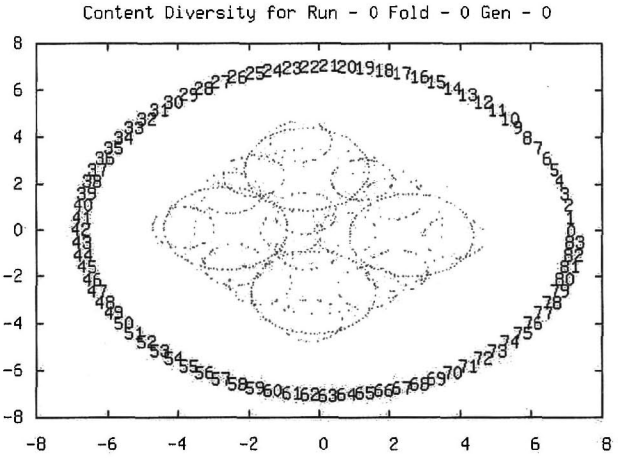

Figure 12-8. Generation 0 Content Plot - Endpoints

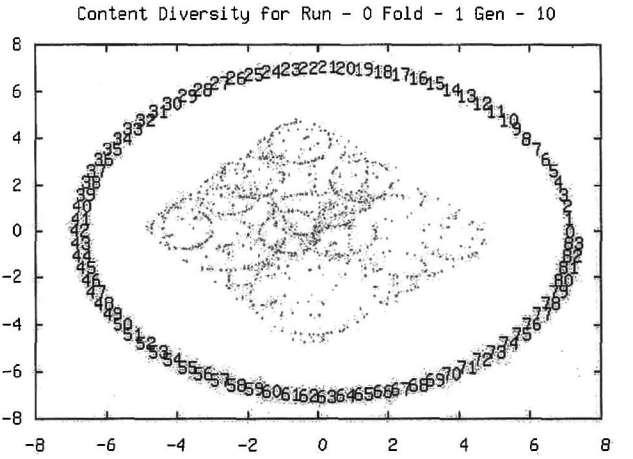

Figure 12-9. Generation 10 Content Plot - Endpoints

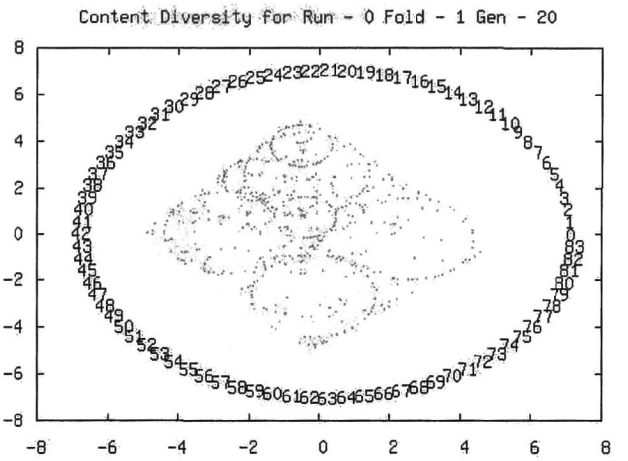

Figure 12-10. Generation 20 Content Plot - Endpoints

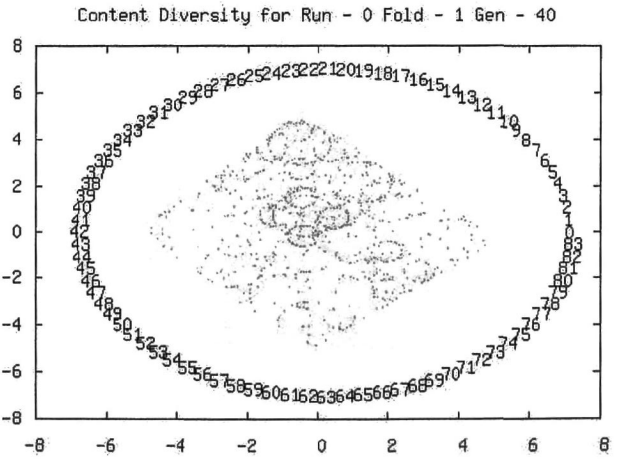

Figure 12-11. Generation 40 Content Plot - Endpoints

3. Fitness Plots

Correlation between content, structure and fitness can be made by comparing fitness plots with the circle plots above. Scatter plots of the individual fitness values in a test population shown in Figure 12-7 have a surprising diversity of fitness among the population, even late in the evolutionary process. The fitnesses of all individuals have been sorted by the training set fitnesses (not shown here) with the least fit individuals appearing at the left end of the graph and the most fit at the right end. Figure 12-12 shows the fitness distribution in generation 0, 12-13 at generation 10, Figure 12-14 at generation 20, and Figure 12-15 at the end of the GP run, generation 40.

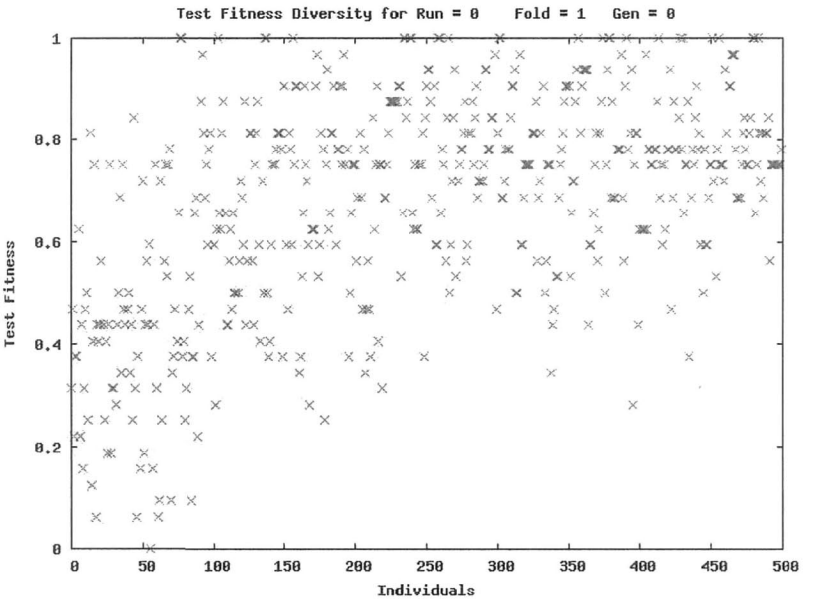

Figure 12-12. Test Fitnesses at Generation 0

By comparing the circle plots and the fitness, we can see that although the content diversity narrows, the fitness variance among individuals remains high but we can also see that there are certain fitness bands that dominate the population as the content goes down.

4. Conclusions and Future Work

The examples shown above were developed in a multi-deme system using generational evolution on a classification problem with a particular fitness measure suited for the type of classification problem we were working on. Any

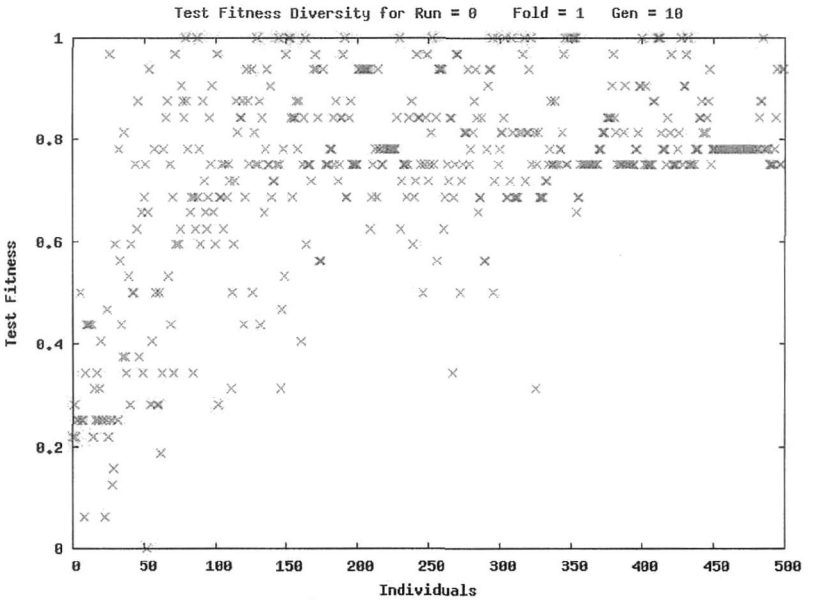

Figure 12-13. Test Fitnesses at Generation 10

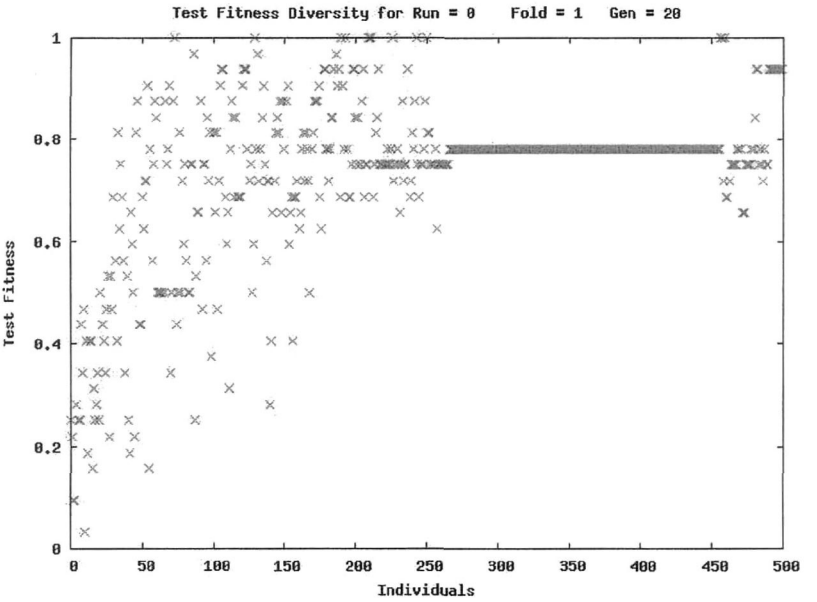

Figure 12-14. Test Fitnesses at Generation 20

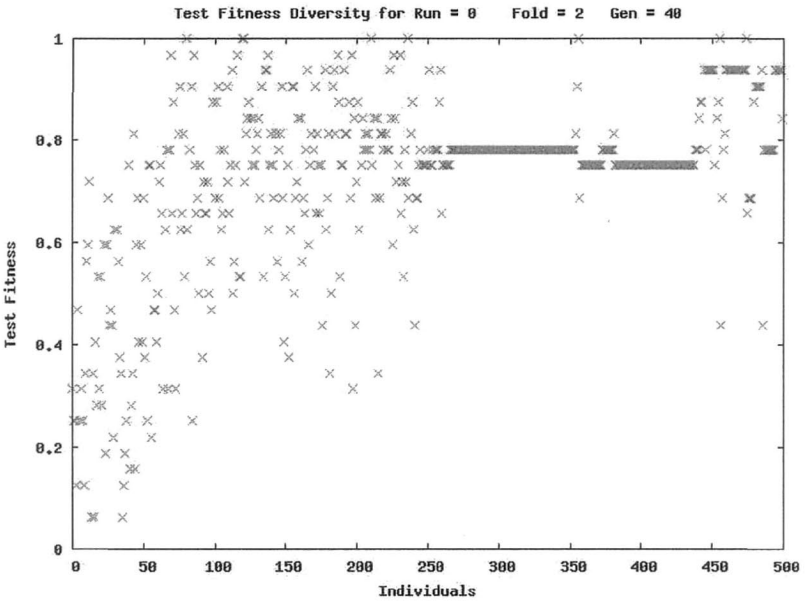

Figure 12-15. Test Fitnesses at Generation 40

general conclusions about GP and the changes in content and its correlation to fitness will have to wait until this approach is applied to more varied problems and environments.

One limitation we have encountered is that in problem sets where there are a large number of inputs and a large population, the "ball of string" effect for full plots can make identification of subtle difference difficult as even minor differences begin to run together. We have considered sampling the individuals in a population rather than using the whole population to help deal with this problem. We are also trying 3D plots where the number of repeats of a segment corresponds to plot height. Another interesting experiment might be coloring the individuals according to the fitness and seeing the correspondence in between the fitness, structure and the content diversity.

However, this approach shows potential as a way to model the dynamics of GP by providing insight into both structure and content during evolution. There are a number of questions that could be resolved more completely in terms of GP behavior such as the difference in diversity caused by crossover, a comparison of fitness proportional versus tournament selection, and perhaps most interesting, comparing populations in separate demes and the effect of different rates of transfer between the demes.

Similarly, running with varying probabilities of crossover and mutation and comparing the content distribution and its relationship to fitness will give an indication of how much GP is influenced by the building block hypothesis and the schema theory as opposed to structural limitations.

Also, by comparing the circle plots described here with Daida *et al.*'s structure plots (Daida et al., 2003), we will be able to see how much of the structure is captured in the circle plot compared to their approach. If the structure shown in the circle plots does not correspond to the structure relationships shown by Daida *et al.* (Daida et al., 2003), then adding structure plots to circle plots and correlating with fitness should show the interplay between structure, content and fitness, testing many of the current theories in Genetic Programming.

References

Angeline, Peter J. (1997). Subtree crossover: Building block engine or macromutation? In Koza, John R., Deb, Kalyanmoy, Dorigo, Marco, Fogel, David B., Garzon, Max, Iba, Hitoshi, and Riolo, Rick L., editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 9–17, Stanford University, CA, USA. Morgan Kaufmann.

Daida, Jason (2004). Considering the roles of structure in problem solving by a computer. In O'Reilly, Una-May, Yu, Tina, Riolo, Rick L., and Worzel, Bill, editors, *Genetic Programming Theory and Practice II*, chapter 5. Kluwer, Ann Arbor.

Daida, Jason M., Hilss, Adam M., Ward, David J., and Long, Stephen L. (2003). Visualizing tree structures in genetic programming. In Cantú-Paz, E., Foster, J. A., Deb, K., Davis, D., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Standish, R., Kendall, G., Wilson, S., Harman, M., Wegener, J., Dasgupta, D., Potter, M. A., Schultz, A. C., Dowsland, K., Jonoska, N., and Miller, J., editors, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1652–1664, Chicago. Springer-Verlag.

Holland, John H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press, Ann Arbor, Michigan, USA.

Jeffrey, HJ (1990). Chaos game representation of gene structure. *Nucleic Acids Research*, 18(8):2163–2170.

Koelle, Katia. Private communications with Katia Koelle. University of Michigan, Center for the Study of Complex Systems.

McPhee, Nicholas Freitag and Hopper, Nicholas J. (1999). Analysis of genetic diversity through population history. In Banzhaf, Wolfgang, Daida, Jason, Eiben, Agoston E., Garzon, Max H., Honavar, Vasant, Jakiela, Mark, and Smith, Robert E., editors, *Proceedings of the Genetic and Evolutionary Com-*

putation Conference, volume 2, pages 1112–1120, Orlando, Florida, USA. Morgan Kaufmann.

Poli, Riccardo and McPhee, Nicholas Freitag (2001). Exact schema theory for GP and variable-length GAs with homologous crossover. In Spector, Lee, Goodman, Erik D., Wu, Annie, Langdon, W. B., Voigt, Hans-Michael, Gen, Mitsuo, Sen, Sandip, Dorigo, Marco, Pezeshk, Shahram, Garzon, Max H., and Burke, Edmund, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 104–111, San Francisco, California, USA. Morgan Kaufmann.

V. Solovyev, S. Korolev, H. Lim (1993). A new approach for the classification of functional regions of DNA sequences based on fractal representation. *Int. Journal of Genome Research*, 1(2):109–128.

Voolich, Johanna and Devaney, Robert L. (2005). The chaos game. http://math.bu.edu/DYSYS/applets/chaos-game.htm.