Long-Term Evolution of Genetic Programming Populations

William B. Langdon
Department of Computer Science, University College London, Gower Street, WC1E 6BT, UK.

ABSTRACT
Evolving binary mux-6 trees for up to 100000 generations, during which some programs grow to more than a hundred million nodes, suggests the landscape which GP explores contains some very smooth regions. Although the GP population evolves under crossover, our unbounded GP appears not to evolve building blocks. We do see periods of tens even hundreds of generations where even although each member of the population occupies a different point in the genotypic search space, they are lie at exactly the same point in the phenotypic landscape. Phenotypic convergence whilst retaining genotypic diversity is typical of GP and, we suggest inherent in highly redundant variable length representations.

2 RESULTS
2.1 Evolution of Size: Gambler’s Ruin
In all experiments we do see enormous increases in size (see Figure 1). If everyone in the population has the same fitness, selection appears to become random and children are as likely to be smaller than their parents as they are likely to be bigger. If selection is switched off for many generations, we see an apparently random walk in average tree size, with falls and rises. (We return to this in Section 3.) However there are substantial falls in size which might be due to the discovery of smaller than average individuals of max fitness but with a higher than average effective fitness. Nonetheless over thousands of generations the removal of smaller trees is sufficient to continue to bloat the population.

2.2 Fitness Convergence
As expected, the number of individuals with maximum fitness in the population grows rapidly towards 100%. However as Figure 1 shows, typically it does not remain at 100% but hovers slightly below 100%. Figure 1 also shows a general downward trend towards fitness convergence.

2.3 Evolution of Size and Shape
Even after evolving for 1000s of gens in small populations, we continue to see the impact of fitness selection on the distribution of tree sizes. And, although the distribution of tree sizes versus their depths is close to that of random trees, the distribution of tree sizes does not approach the limiting distribution Poli predicted assuming no fitness EuroGP-2007

2.4 Evolution of Effective Code
Figure 2 (top right) shows that the size of effective code is fairly stable. If we look at the effective code in every tree in the population at generation 500–4000 (Figure 2) we see they are very similar. Typically the effective part of the code lies in a few hundred nodes around the root node (yellow) which is protected against crossover by evolved constants. The constants head large sacrificial subtrees of ineffective code. Figure 2 shows the effective code is conserved over many generations. We see this in all pop=500 runs (except 102, which gets stuck at fitness=62). However the details of the evolved effective code differ from run to run.
Figure 2: Effective code in a pop of 500 binary trees after 500, 1000, 2000 and 4000 gens. (Each panel showing the 496-500 trees with max fitness.) Note the similarity of the eff. code even though separated by 1000s of gens. By gen 4000 although mean size is 105 880, eff. code is limited to the 721 nodes shown. Almost all the population have 82 eff. nodes in common (yellow). Darker colours indicate eff. code which only occurs in ≤148 (blue) or ≤22 (black) trees. Code which does not effect any program’s output is not plotted.

Figure 3: Ten extended population fifty runs.

3 A LIMIT TO BLOAT

When the bloated trees become so large that number of times the high fitness core near the root node is disrupted falls well below one per generation there is no driving force to grow the trees. Hence there is a balancing point near tree size ≈ popsize × core codesize. Averaged over ten extended runs over 100 000 generations, the median mean tree size in the 10 populations was 42 507 (see Figure 3).

GP code http://www.cs.ucl.ac.uk/.../GPmux6.tar.gz