

A Study of the Role of Single Node Mutation in Genetic Programming*

Wei Quan and Terence Soule

Department of Computer Science
University of Idaho
Moscow, ID 83844-1010
tsoule@cs.uidaho.edu

Abstract. In this paper we examine the effects of single node mutations on trees evolved via genetic programming. The results show that neutral mutations are less likely for nodes nearer the root and that as evolution proceeds neutral mutations of nodes near the root are progressively less likely.

Studies of crossover in tree based GP have shown that when smaller and/or deeper branches are selected for crossover the resulting change in fitness is smaller and the probability of fitness neutral crossover is larger [3,1,4,2]. In this paper we continue this research for mutations by studying the relationship between the depth of single node mutations and the probability of fitness neutral mutations.

Our GP is steady-state, population size 100, 0.7 crossover rate, 90/10 crossover, 4 member tournament selection. Results are the average of 100 trials. The initial population is generated with full tree with depth of 4. The test problem is symbolic regression; the target function is $f(x) = x^3 + 2x^2 - 5x + 3$ in the range $[0, 2\pi]$ or $[-\pi, \pi]$. Fitness is the square root of the sum of the squared errors at every test point. The function and terminal set is $\{+, -, *, /(protected)\}$. Mutation changes a single node into another node with the same arity. One mutation is applied per offspring per iteration.

Results are generated by copying the population after 50, 100 and 500 evaluations and exhaustively testing every possible single node mutation on the copied populations. Figure 1 shows the percentage of non-neutral mutations as a function of depth of mutation and the number of evaluations/iterations. There is a strong correlation between the depth of a mutation point and the probability of a fitness neutral mutation. Additionally, after a longer period of evolution neutral mutations are less likely, for all mutation depths studied.

We tested two hypotheses to explain the affect of additional iterations on the percentage of non-neutral mutations:

- 1) As evolution proceeds, the increasing tree size makes mutation near the root less likely to be fitness neutral. This seems reasonable because for larger trees a mutation near the root effects absolutely and proportionally more of the tree.
- 2) As evolution proceeds, the improving fitness makes mutation near the root

* This work supported by NSF EPSCoR EPS-0132626.

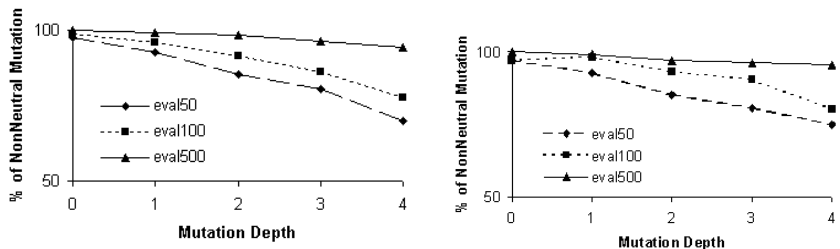


Fig. 1. Average percentage of mutations that are non-neutral as a function of depth after 50, 100 and 500 evaluations for the domains $[-\pi, \pi]$ (left) and $[0, 2\pi]$ (right). Mutations at the root (depth 0) are never neutral. As depth increases neutral mutations become more likely. As evolution proceeds neutral mutations become progressively less likely.

less likely to be fitness neutral. This seems reasonable because as trees become more fit they are presumably less random, thus a random mutation may be more likely to effect their fitness.

To test these hypotheses we compared overall tree depth to the probability of neutral mutations and compared the tree fitness to the probability of neutral mutations. The correlations of both overall depth and of fitness to the probability of neutral mutations were too weak to support the above hypotheses.

Our results show that the affects of single node mutations are similar to the affects of crossover in tree based GP: the probability of a neutral mutation is correlated with the depth of the mutation, at least for mutations near the root. Additionally, the more a program has evolved, the more likely it is that a mutation near the root will change the tree's fitness.

References

1. Christian Igel and Kumar Chellapilla. Investigating the influence of depth and degree of genotypic change on fitness in genetic programming. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference 1999*, pages 1061–1068. Morgan Kaufmann, 1999.
2. Sean Luke. Code growth is not caused by introns. In *Late Breaking Papers, Proceedings of the Genetic and Evolutionary Computation Conference 2000*, pages 228–235, 2000.
3. Terence Soule and Robert Heckendorn. An analysis of the causes of code growth in genetic programming. *Genetic Programming and Evolvable Machines*, 3:283–309, 2002.
4. Matthew J. Streeter. The root causes of code growth in genetic programming. In Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa, editors, *Genetic Programming, 6th European Conference, EuroGP 2003*, pages 443–454, 2002.