Function Choice, Resiliency and Growth in Genetic Programming*

Sireesha Besetti
University of Idaho
Moscow
Idaho-83844
bese9744@uidaho.edu

Dr.Terence Soule
University of Idaho
Moscow
Idaho-83844
tsoule@cs.uidaho.edu

ABSTRACT

In this paper we examine how the choice of functions in a genetic program (GP) affects the rate of code growth and the development of resilient individuals. We find that functions or combination of functions that produce the most resilient individuals often, but not always, produce the most growth. Finally, the results confirm that there is a high correlation between the resiliency of parent individuals and the fitness of their offspring.

Categories and Subject Descriptors

I.2.2 [**Automatic Programming**]: Program Synthesis

General Terms

Algorithms

Keywords

Genetic programming, function choice, growth, resiliency

1. INTRODUCTION

Resiliency is a general measure of how much an individual is affected by genetic variation. More resilient individuals are the ones whose fitness is least likely affected by genotypic changes. There is clear evidence that in evolutionary systems there is a significant evolutionary pressure in favor of solutions that are more resilient [1,2,3,4]. Code growth, or bloat, is a well known strategy that is adopted by evolving individuals to increase resiliency. Evolutionary pressure is important because it may have a significant affect on the evolutionary process and may hinder, or help, the practical goal of evolving more fit individuals. Understanding the causes and affects of the pressure for resilient individuals may help to design evolutionary algorithms that take advantage of this pressure.

In this paper it is demonstrated that in a GP the function set affects both the growth rate and the resiliency of the evolved individuals. Further, the pressure for resiliency has significant affect on the ratio of functions in evolving programs. We find that there is a close correlation between the resiliency of parents and the fitness of offspring. Comparison between a simple problem and a symbolic regression problem is performed.

Recent research has shown that there is no simple connection between introns, growth, resiliency and survival. The underlying evolutionary force driving growth is not pressure for growth directly, but pressure for resilient individuals, which only manifests itself as growth in some cases [2,3,4]. Finally, it has been shown that, with simple genes, pressure for resiliency also effects which genes are preferred [3]. With more complex genes the simple relationship between the terminals and the individual's fitness cannot be expected to exist. This paper begins to address the question of how pressure for resiliency affects gene ratios when there can be complex interactions between the genes. The results show that resiliency is a significant factor in gene choice even for these fairly complex operations in which there is no simple functional relationship between function and fitness.

2. EXPERIMENT

In GP the results are very problem specific. Simple problems facilitate the understanding of the underlying evolutionary pressures, in this case for code growth and resiliency. Thus, our first problem is very simple. The problem is to evolve an expression to reach a fixed target value (T = 50) using combination of functions and a range of terminals. Fitness is the absolute difference between the value of the evaluated tree and the target value. The second problem is a symbolic regression problem with the target function $1+3*x^1+3*x^2+x^3$. Fitness is the absolute error between the target function and the evolved expression. The parameters of the two experiments are listed in Table 1. Offspring that are larger than 10,000 nodes are discarded and their parents are kept instead.

3. CONCLUSION

Very generally, the results show that different function sets produce different rates of growth, different fitness and different levels of resiliency in the evolved solutions, even with the simple problem where all function sets are sufficient

*This publication was made possible by NIH Grant P20 RR16448 from the COBRE Program of the National Center for Research Resources.

GECCO'05, June 25–29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

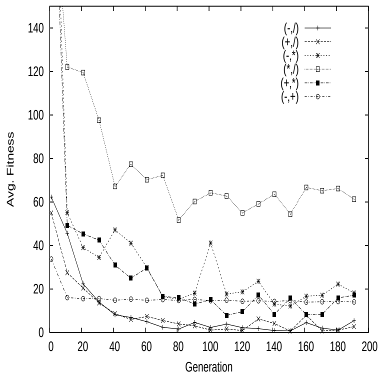

Figure 1: The average fitness for the function sets (+,-), (+,*), (+,/), (-,*), (-,/), (*,/) averaged across 50 trials. The average fitness is best when division is combined with functions other than multiplication.

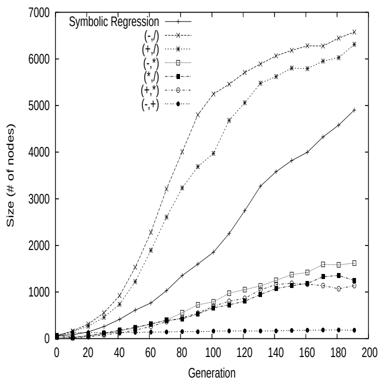

Figure 2: Average size for each of the function pairs and the symbolic regression problem averaged across 50 trials.

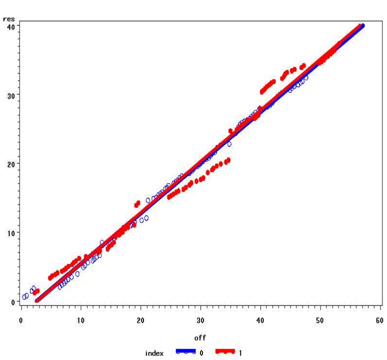

Figure 3: The correlation between offspring fitness and resiliency values in case of addition and division, averaged across 50 independent multiple crossover and then again. Index 0 indicates first offspring and index 1 indicates second offspring.

Table 1: Problem parameters. For the simple problem all sets of two function were tested: (+,-), (+,/), etc.

Population Size	100
Initialization	Ramped, half and half
Number of Trials	50
Number of Generations	200
Function	(+,-,*,/)
Terminal Set	(-10,10)
Selection	Tournament selection(3)
Crossover	90/10 rule
Mutation	-
Elitism	Best individuals are preserved
Division function	Protected

Table 2: Simple problem, average values for average, and best fitness in the last generation.

Functions	Avg. best fitness	Avg. average fitness
Add, Sub	0.00114792	14.814
Add, Mul	0.0978987	17.0739
Add, Div	0.000329666	1.15273
Sub, Mul	0.0929832	19.0967
Sub, Div	0.000868683	0.543712
Mul, Div	0.921442	55.264
All	0.0209537	305.417

to solve the problem. The results also show selective pressure in favor of certain functions over others, independent of the function's apparent contribution to fitness. The results strongly suggest that functions that produce the most resilient individuals have the highest growth rates (e.g. division). However, it is also clear that the interaction between functions is very significant. Finally there is a high correlation between resiliency, as measured by the average fitness change due to crossover, and both parent fitness and offspring fitness.

(1)Peter Smith and Kim Harries. Code growth, explicitly defined introns, and alternative selection schemes.*Evolutionary Computation,*6(4):339-360,1998.

(2)Terence soule.*Operator choice and the Evolution of Robust solutions.* Genetic Programming Theory and Practice,Kluwer,2003.

(3)Terence Soule, Robert Heckendorn, and Jian Shen. Solution stability in evolutionary computation. In Ilyas Cicekli, Nihan Kesim Cicekli and Erol Gelenbe, editors,*Proceedings of the 17th International Symposium on Computer and Information Systems,* pages237-241,2002.

(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.

Table 3: Simple problem, ratio of the numbers of functions in the final generation, averaged over 50 trails.

Ratio of functions	Ratio value
Div/Mul	1.922
Div/Sub	1.9082
Div/Add	1.9804
Mul/Sub	1.7673
Mul/Add	1.7998
Sub/Add	2.0839