Empirical Investigation of Size-Based Tournaments for Node Selection in Genetic Programming

Thomas Helmuth
Computer Science
University of Massachusetts
Amherst, MA 01003
thelmuth@cs.umass.edu

Lee Spector
Cognitive Science
Hampshire College
Amherst, MA 01002
lspector@hampshire.edu

ABSTRACT

In genetic programming systems, genetic operators must select nodes upon which to act; the method by which they select nodes influences problem solving performance and possibly also code growth. A recently proposed node selection method using "size-based tournaments" has been shown to have potential, but variations of the method have not been studied systematically. Here we extend the ideas of size-based tournaments and test how they can improve problem-solving performance. We consider allowing tournament size to depend on whether we are selecting nodes within "donors" for crossover, "recipients" for crossover, or targets of mutation. We also consider tournaments that bias selection toward smaller trees rather than larger trees. We find that differentiating between donors and recipients is probably not worthwhile and that size 2 tournaments perform near-optimally.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.2.2 [**Artificial Intelligence**]: Automatic Programming—*Program modification*

General Terms

Algorithms

Keywords

genetic programming, node selection, tournaments, crossover, mutation

1. INTRODUCTION

In tree-based genetic programming (GP), genetic operators such as crossover and mutation change programs to create new individuals. Genetic operators act on subtrees rooted at nodes within program trees; for example, crossover replaces a chosen node[1] with a node selected from another program. The node selection mechanism determines how nodes are selected for use by the genetic operators. Node selection mechanisms usually use a non-deterministic approach, with many biasing selection toward subtrees with certain properties. Node selection bias can affect the size, shape, and location of crossover and mutation points, which can in turn affect the progression of evolution toward fitter programs.

Most GP systems use one of two node selection mechanisms: "unbiased" selection or "Koza 90/10" selection [2]. Unbiased selection gives all nodes in the program tree equal chance of being selected. In order to bias node selection toward larger nodes, Koza 90/10 selection sets specific probabilities for selecting internal and terminal nodes [2]. In particular, this mechanism selects an internal node with 90% probability and a terminal node with 10% probability. Both of these methods have drawbacks, as discussed in [1].

The recently proposed method of *size-based tournaments* biases node selection toward larger nodes by conducting tournaments that are similar to the tournaments commonly used for parent selection [1]. Each tournament randomly selects a number of nodes from the program tree with uniform probability; it then choses the largest of these nodes for use by mutation or crossover. In some experiments, size-based tournaments have been shown to increase problem-solving performance without causing additional code bloat [1].

The size-based tournaments selection mechanism normally selects the largest node in the tournament, where size is based on number of nodes in the subtree. By decreasing the size of a tournament, one removes bias towards selecting larger nodes, since fewer nodes participate in each tournament. In the extreme case of tournaments of size 1, there is no bias towards larger nodes, which results in unbiased selection. Taking this idea further, we devise tournaments that, instead of preferring larger nodes, prefer smaller nodes. In order to continue the trend of smaller tournaments indicating preference of smaller nodes, we call these *negative tournaments*. For example, a size-based tournament of size -3 selects three nodes from the individual and then chooses the smallest of the three nodes. Negative size-based tournaments allow us to put more pressure towards smaller nodes than is possible with normal size-based tournaments.

While crossover is often described as the exchange of subtrees between individuals, one can just as easily designate one individual as the donor and the other as the recipient, where a subtree from the donor is selected to replace the recipient's selected subtree. The original work uses the same tournament size whether the node is being selected for mutation, crossover donor, or crossover recipient [1]. A natural

[1]Although nodes in a program tree and the subtrees rooted at those nodes are different things, for the sake of brevity we will use the term "nodes" to refer to the subtrees that they root.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

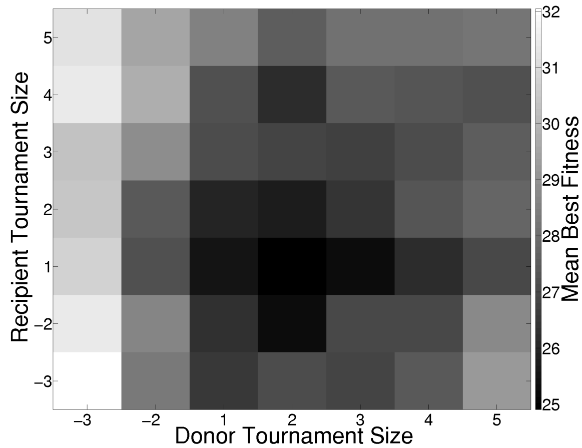

Figure 1: Mean best fitness results for the artificial ant problem. The x- and y-axes represent the tournament size used for donor and recipient respectively. Lower (darker) is better.

alternative is to use different tournament sizes for selecting nodes for mutation, donor, and recipient. Separate tournament sizes for donor, recipient, and mutation can possibly affect the sizes of individuals in the population, as well as overall problem solving ability.

2. EXPERIMENTAL RESULTS

To test the effectiveness of size-based tournaments of differing tournament sizes, we conducted runs on three standard test problems: symbolic regression, artificial ant, and 11-multiplexer. We conducted sets of runs for each possible pair of donor and recipient tournament sizes from the set $\{-3, -2, 1, 2, 3, 4, 5\}$. To simplify things, we used a constant node selection tournament size of 2 for mutation across all runs. As a comparison, we conducted runs using unbiased selection and Koza 90/10 selection. We performed all runs using ECJ[2] with standard parameters as given in [1]. All problems used the standard formulations from [2].

We are primarily interested in the problem solving abilities of size-based tournaments. In order to test how well each node selection condition solves each problem, we measured the success rate, computational effort, and the mean best fitness for each set of runs. We only report the mean best fitnesses here, as we have limited space. The mean best fitness of a GP run is the mean of the best individual fitnesses attained in each run. For all runs described here, fitness is defined as a measure of error with lower numbers being better and solutions having fitness values of zero.

We tested each node selection mechanism on a symbolic regression problem with the function $f(x) = x^4 + x^3 + x^2 + x$. When we examined the mean best fitnesses for all donor and recipient tournament size combinations, we found that the best results for mean best fitness are concentrated around donor size 3 and recipient size 2. We tested the same node selection methods on the artificial ant problem using the Santa Fe trail [2]. Figure 1 gives the mean best fitness results for the artificial ant problem runs. This figure is representative of those we found for the other two tested problems. Mean best fitness results are clustered around donor size 2 and recipient size 1. Finally, we tested the node selection techniques on the 11-multiplexer problem [2]. Donor size 5 and recipient size 4 gave the lowest mean best fitness, although donor size 2 and recipient size 2 performed almost as well. Negative tournament sizes rarely gave above-average performance on any problem, and were never optimal.

For these three problems, we also find that size-based tournaments using donor and recipient size 2 outperforms the unbiased and Koza 90/10 selection mechanisms on mean best fitness, success rate, and computational effort. Additionally, we find that differentiating between donors and recipients is probably not worthwhile; runs using size 2 tournaments for both donors and recipients perform near-optimally in all of our tests. While our focus is primarily on problem-solving performance, our preliminary finding shows that runs using size 2 tournaments have comparable program sizes to those using unbiased and Koza 90/10 selection methods.

[2]http://cs.gmu.edu/~eclab/projects/ecj/

3. CONCLUSIONS AND FUTURE WORK

We have explored size-based tournaments to see what parameters give the best performance. We have shown that negative tournaments, which favor smaller nodes instead of larger ones, do not perform better than tournaments that favor larger nodes. More importantly, we have shown that differentiating tournament sizes between donor and recipient individuals is not likely to be worthwhile.

In particular, using size 2 tournaments resulted in near-optimal mean best fitnesses, success rates, and computational efforts on the tested problems. Size 2 tournaments also performed favorably compared to unbiased and Koza 90/10 selection mechanisms. Size-based tournaments are simple to implement and produce well-reasoned biases in node selection, attributes the proposed alternatives lack. We therefore recommend using size-based tournaments of size 2 as a general and effective node selection method.

4. ACKNOWLEDGMENTS

Thanks to Kyle Harrington, Brian Martin, Emma Tosch, Kwaku Yeboah Antwi, and Nathan Whitehouse for feedback that improved this work. This material is based upon work supported by the National Science Foundation under Grant No. 1017817. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

5. REFERENCES

[1] T. Helmuth, L. Spector, and B. Martin. Size-based tournaments for node selection. In M. Nicolau, editor, *GECCO 2011 Graduate students workshop*, pages 799–802, Dublin, Ireland, 12-16 July 2011. ACM.

[2] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.