Impact of Crossover Bias in Genetic Programming

Nicholas Freitag McPhee, M. Kirbie Dramdahl, David Donatucci
Division of Science and Mathematics
University of Minnesota, Morris
Morris, MN USA-56267
{mcphee, dramd002, donat056}@morris.umn.edu

ABSTRACT

This is probably too long. People often recommend keeping the abstract to somewhere between 150 and 250 words, and this is closer to 500. For the initial submission that's OK, but we may want to move some of this to the introduction and trim down the abstract somewhat.

In tree-based genetic programming with sub-tree crossover, the parent contributing the root portion of the tree (which we refer to as the *root parent*) often contributes more to the semantics of the resulting child than the other parent (the *non-root parent*). In previous research, we found that when the root parent had greater fitness than the non-root parent, the fitness of the child tended to be better than if the reverse were true. Here we explore the significance of that asymmetry by introducing the notion of *crossover bias*, which allows us to bias the system in favor of having the more fit parent be the root parent. To better understand the impact of this bias, we implemented several levels of crossover bias, including 0% bias (root individual chosen randomly, as in traditional sub-tree crossover), 100% bias (the stronger parent is always chosen to be the root parent), 50% bias (bias implemented in half the cases, and the other half chosen randomly), and reverse bias (the weaker parent is always chosen as root parent).

We applied crossover bias to a variety of problems. In most cases we found that using crossover bias either improved performance or had no impact. Our results do, however, indicate the possibility that crossover bias may increase selection pressure and premature convergence – undesirable behavior, as it encourages a genetic programming run to arrive at a solution too quickly, in the process potentially excluding more accurate solutions for a more generalized one.

Our results also demonstrate that the effectiveness of crossover bias is somewhat dependent on the problem, and significantly dependent on other parameter choices. In particular it appears that crossover bias has the largest impact when selection pressure is weaker, and the differences in the fitness of the parents is thus likely to be larger. We also found that the use of elitism reduced the influence of crossover bias. It's possible that crossover bias acts to some degree as an "elitism" operator, making it more likely that the semantics of more fit individuals are copied into the next generation; thus if traditional elitism is being employed this effect is less visible. Another possible explanation for this is that if the most fit individuals are automatically being carried over, there is perhaps less need to produce new, fitter individuals via crossover, reducing or even eliminating the usefulness of crossover bias. Other factors which we found to have potential impact on the effectiveness of crossover bias were tournament size, population size, and possibly the difference in parental fitness.

Categories and Subject Descriptors

[]

General Terms

Keywords

genetic programming, crossover bias, root parent

1. INTRODUCTION

2. GENETIC PROGRAMMING

3. GRAPH DATABASES

4. EXPERIMENTAL SETUP

4.1 Genetic Programming Setup

4.2 Neo4j Setup

5. RESULTS

5.1 Structural problems

5.1.1 K-Landscapes problems

Do we want to to include reverse bias (indicated with -1 in the plots) in all our analysis? If memory servers, nothing much interesting ever comes of it, and we've arguably got more data than we can analyze/discuss here, so I'm inclined to either remove it altogether, or just mention it once somewhere, maybe with one example in a plot. Thoughts?

We did a full sweep of parameters for the K-Landscapes problem with both $K = 2$ and $K = 6$. **All I have added so far is the $K = 6$ data, which is more interesting than the $K = 2$ data because the problem is harder. Do we want to include both, or just forgot the $K = 2$ results?**

- XO bias of reverse, 0, 0.25, 0.5, 0.75, and 1

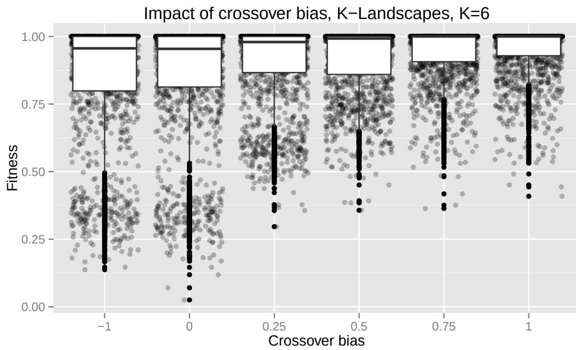

Figure 1: Impact of crossover bias on fitness for K-Landscapes problem, $K = 6$.

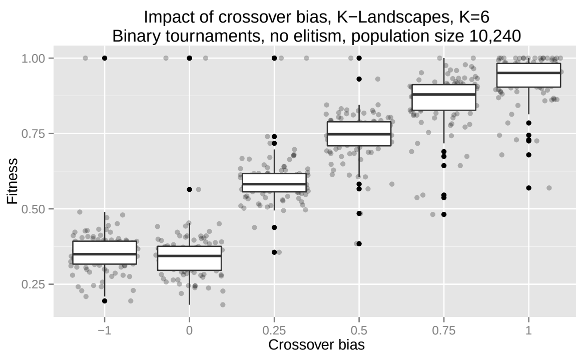

Figure 2: Impact of crossover bias on fitness for K-Landscapes problem, $K = 6$, restricted to binary tournament selection, no elitism, and population size 10,240.

- Population sizes of 1,024 and 10,240
- Elitism of 0% or 1%
- Tournament sizes of 2, 3, 5, and 7

Figure 1 shows the impact of crossover bias on this problem across all the combinations of parameter values. Increasing the amount of crossover bias appears to consistently improve the fitness of the results. All the differences are statistically significant ($p < 0.012$) except for the difference between reverse bias (-1) and no bias (0), and the difference between bias probability 0.75 and 1.0.

Figure 2 shows the subset set of this information with just binary tournament selection, no elitism, and population size of 10,240. It's clear that the impact of crossover bias is much stronger in this case than in the general case shown in Figure 1. Here all the differences are strongly statistically significant ($p < 10^{-11}$) except for the difference between reverse bias (-1) and no bias (0). Increasing the crossover bias increases the number of "perfect" solutions discovered as well as just increasing the fitness. 15 solutions were discovered in the 100 runs with crossover bias at 1.0, where only 1 or 2 solutions were discovered for each of the other crossover bias probabilities; this difference is statistically significant with $p \leq 0.03$ using a pairwise test of proportions.

Figures 3 and 5 are variants of Figures 1 and 2 using violin plots instead of box plots. I think I definitely prefer the violin plots in Fig 2 (the nice stair step plot), but I'm less certain about the other one. Thoughts?

Figure 4 is the same as Figure 3, but with the scale transformed by squaring the values. It turns out that log didn't work; it just increased the compression up at the top because the values are between 0 and 1.

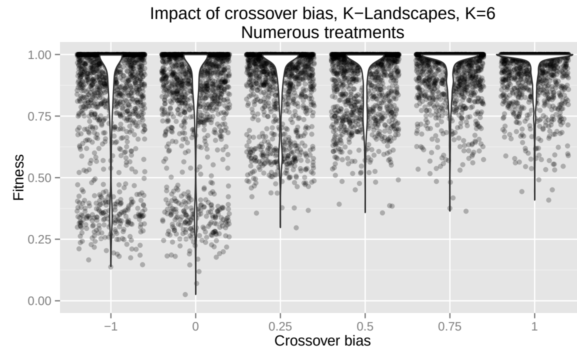

Figure 3: Impact of crossover bias on fitness for K-Landscapes problem, $K = 6$.

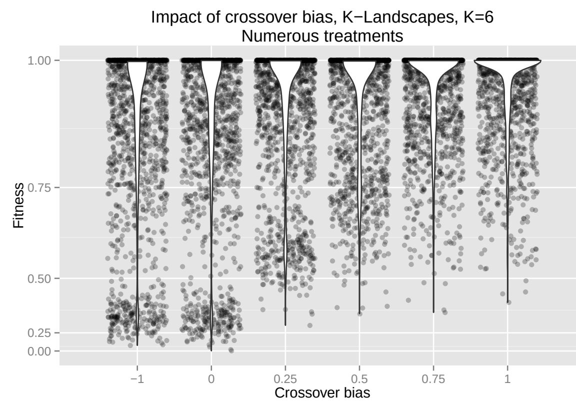

Figure 4: Impact of crossover bias on fitness for K-Landscapes problem, $K = 6$. Note the y-axis has been transformed to provide more detail in the upper end of the fitness range.

5.1.2 Order tree problem

5.1.3 Lid problem

We need to add plots for this problem.

5.2 U.S. Change problem

5.3 Symbolic regression problems

5.3.1 Sine problem

Figure 12 shows the hits results for the sine regression problem (without reverse bias – I think I figured we were going to drop it when I ran these.) with the "strong treatment", i.e., binary tournaments, no elitism, and the large population (10,240). Figure 13 is the same data, but with violin plots instead of box plots. All these differences are statistically significant ($p < 10^{-5}$ using a pairwise Wilcoxon rank sum test) except for the difference between the bias of 0.75 and 1.0. Figure 14 shows the same results, but for fitness (probably sum of error?) instead of hits. Again, all of these are significant except for bias of 0.75 and 1.0.

Figure 15 shows the results for the sine regression problem (without reverse bias – I think I figured we were going to drop it

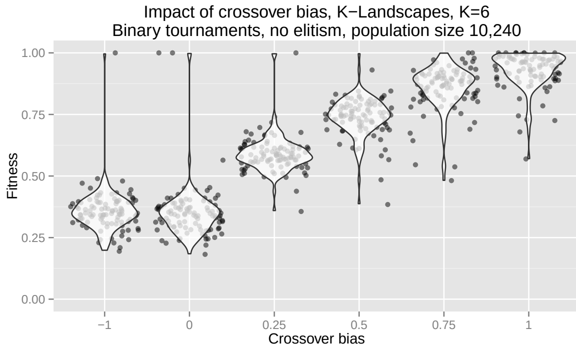

Figure 5: Impact of crossover bias on fitness for K-Landscapes problem, $K = 6$, restricted to binary tournament selection, no elitism, and population size 10,240.

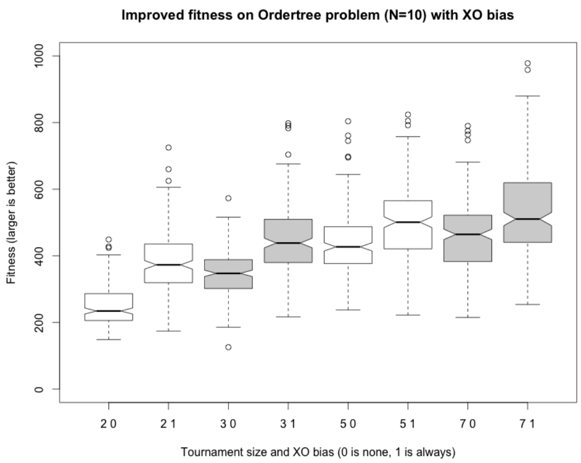

Figure 6: Impact of crossover bias on fitness for Ordertree problem for various tournament sizes.

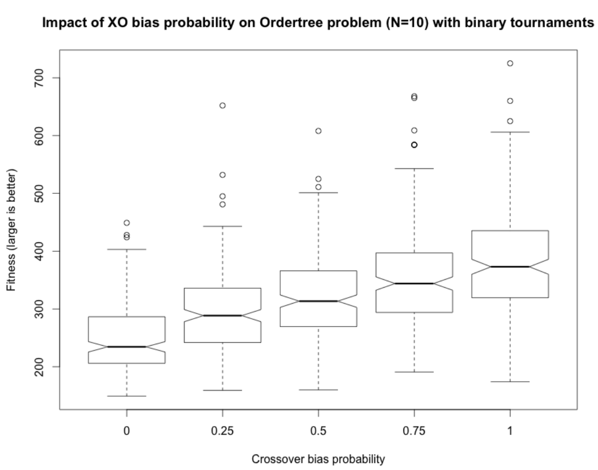

Figure 7: Impact of crossover bias on fitness for Ordertree problem for binary tournaments.

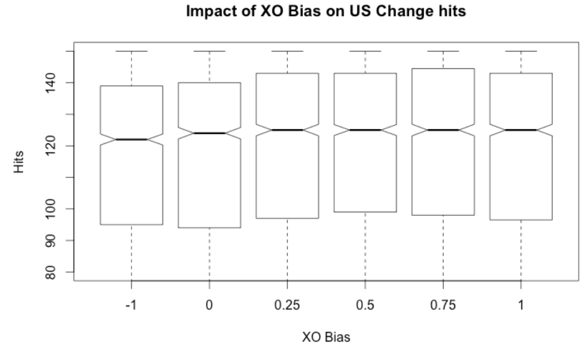

Figure 8: Impact of crossover bias on the number of hits for the US Change problem.

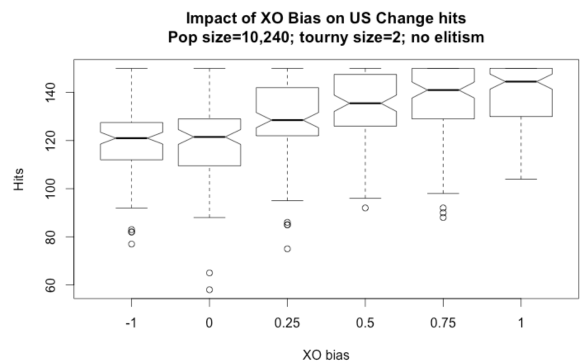

Figure 9: Impact of crossover bias on the number of hits for the US Change problem, limited to binary tournaments and no elitism.

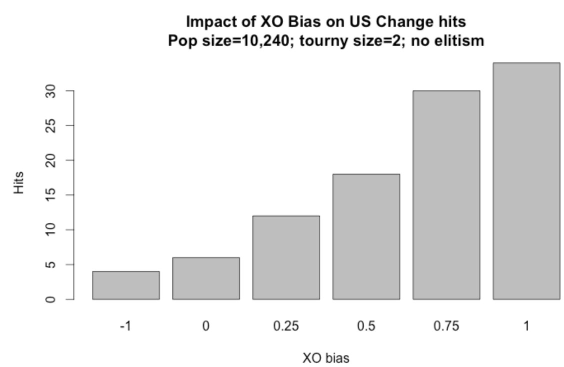

Figure 10: Impact of crossover bias on the number of successes runs for the US Change problem, limited to binary tournaments and no elitism.

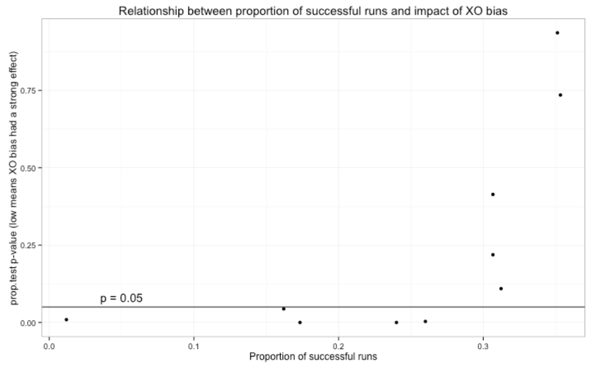

Figure 11: Relationship between proportion of successful runs and the impact of crossover bias.

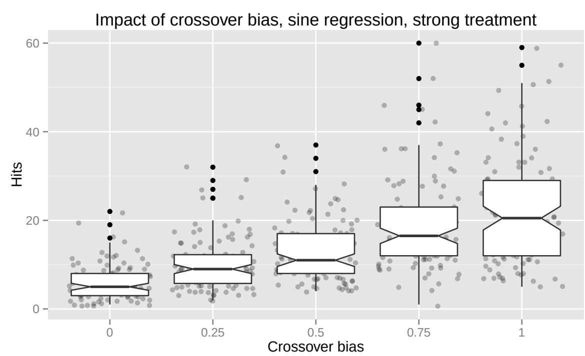

Figure 12: Impact of crossover bias on the sine symbolic regression problem with binary tournaments, no elitism, and population size 10,240

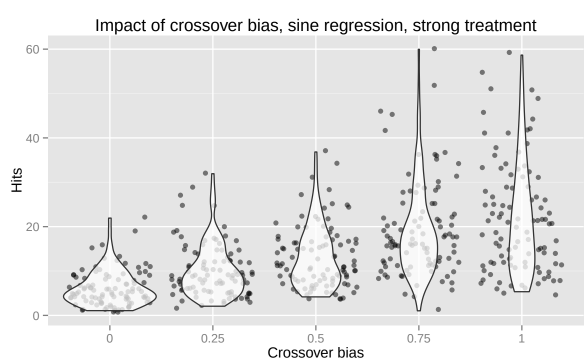

Figure 13: Impact of crossover bias on the sine symbolic regression problem with binary tournaments, no elitism, and population size 10,240

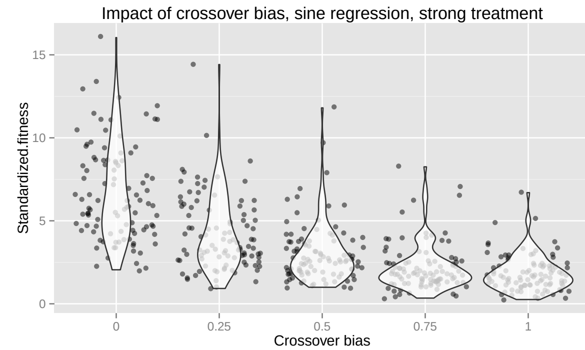

Figure 14: Impact of crossover bias on the sine symbolic regression problem with binary tournaments, no elitism, and population size 10,240

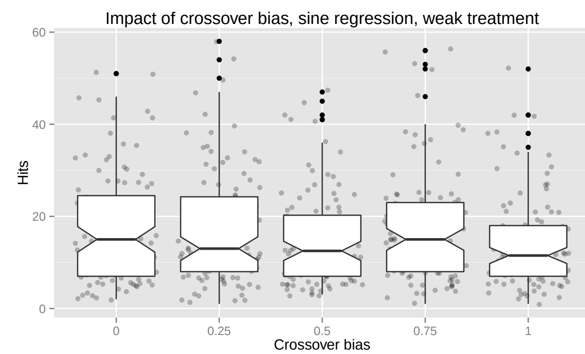

Figure 15: Impact of crossover bias on the sine symbolic regression problem with tournament size 7, 0.1% elitism, and population size 1,024.

when I ran these.) with the "weak treatment", i.e., tournament size 7, 0.1% elitism, and the small population (1,024). Figure 16 is the same data, but with violin plots instead of box plots. None of these differences are statistically significant using a pairwise Wilcoxon rank sum test. Figure 17 shows the same results, but for fitness (probably sum of error?) instead of hits. None of these differences are significant.

Figures 18 and 19 show the change in fitness over time for the strong and weak configurations. Figure 20 shows fitness over time for the "weak" configuration with population size 10,240 (instead of 1,024 for the "normal" weak configuration). Increasing the pop size definitely improves the performance by quite a lot (no surprise). The different bias levels are all essentially the same at the end of the 100 generations, but interestingly there is definitely a spread around 15-20 generations that is a really clean "more bias is better" demonstration except for the fact that 0.75 and 1.0 are essentially the same.

5.3.2 Pagie-1 problem

5.4 Santa Fe Trail problem

I did this small-ish set of runs early on, so the setup for this isn't really the same as for the later problems. The only "parameter sweep" I did was XO bias on (i.e., 1) or off (i.e., 0). I only ran 50 generations, and the setting to stop early if a solution was found was turned on, so not all the runs went out to 50 generations. There

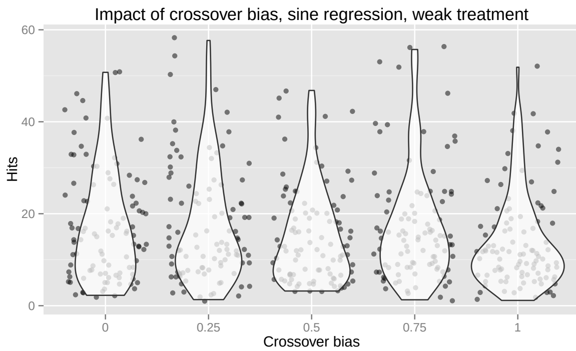

Figure 16: Impact of crossover bias on the sine symbolic regression problem with tournament size 7, 0.1% elitism, and population size 1,024.

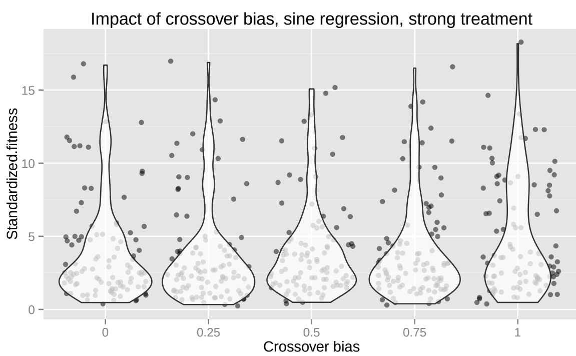

Figure 17: Impact of crossover bias on the sine symbolic regression problem with tournament size 7, 0.1% elitism, and population size 1,024.

Figure 18: Impact of crossover bias on the sine symbolic regression problem with binary tournaments, no elitism, and population size 10,240.

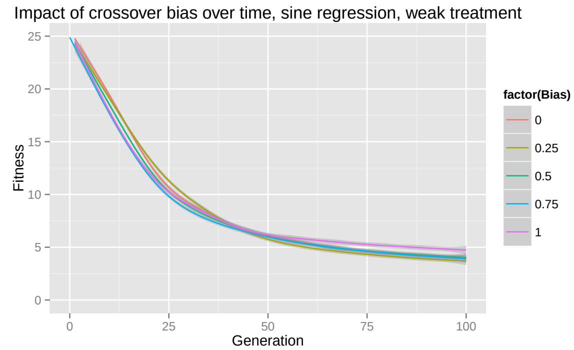

Figure 19: Impact of crossover bias on the sine symbolic regression problem with tournament size 7, 0.1% elitism, and population size 1,024.

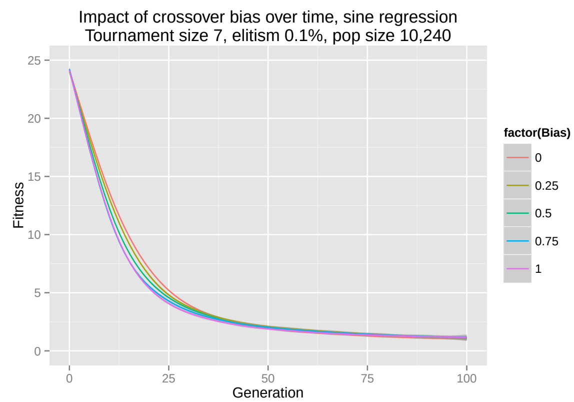

Figure 20: Impact of crossover bias on the sine symbolic regression problem with tournament size 7, 0.1% elitism, and population size 10,240.

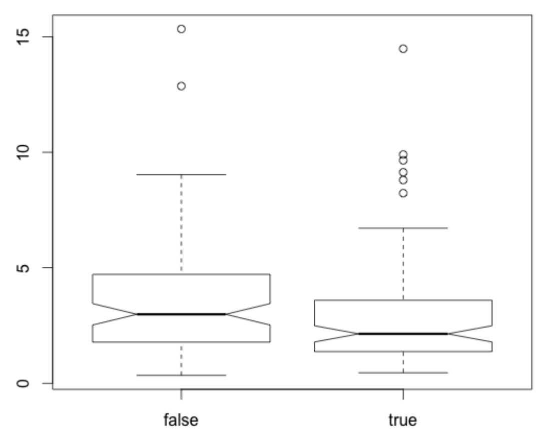

Figure 21: **OLD FIGURE** Impact of crossover bias on the sine symbolic regression problem

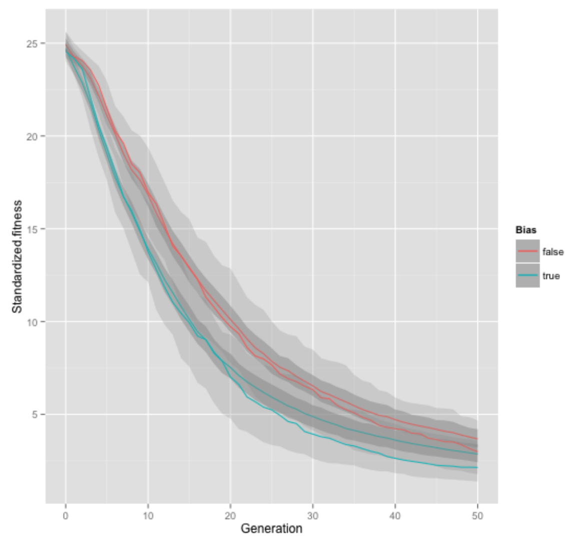

Figure 22: **OLD FIGURE** Impact of crossover bias on the fitness over time for the sine symbolic regression problem

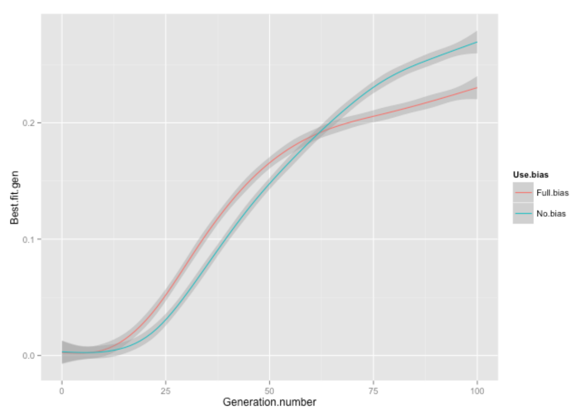

Figure 23: Impact of crossover bias on the fitness over time for the Pagie-1 symbolic regression problem

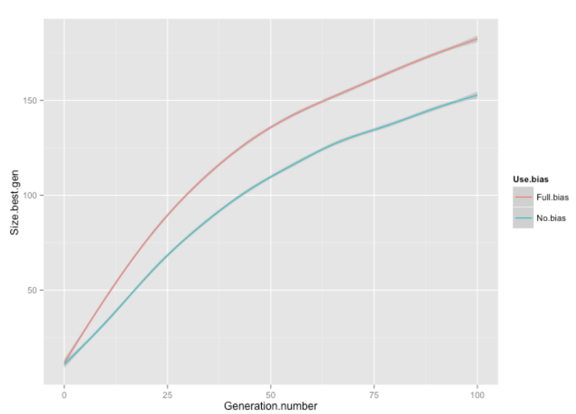

Figure 24: Impact of crossover bias on the tree size over time for the Pagie-1 symbolic regression problem

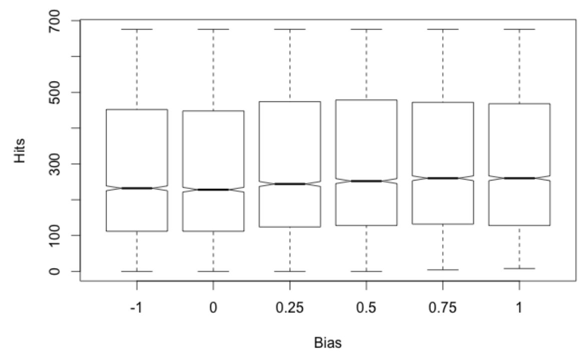

Figure 25: Impact of crossover bias on the number of hits for the Pagie-1 symbolic regression problem. Unfortunately I'm not immediately sure what (sub)set of data this includes.

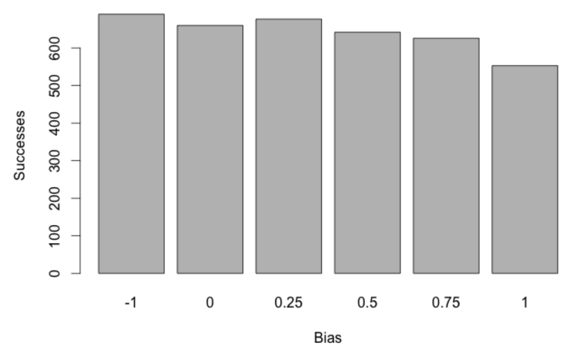

Figure 26: Impact of crossover bias on the number of successes (runs that exactly solve the problem) for the Pagie-1 symbolic regression problem. Unfortunately I'm not immediately sure what (sub)set of data this includes.

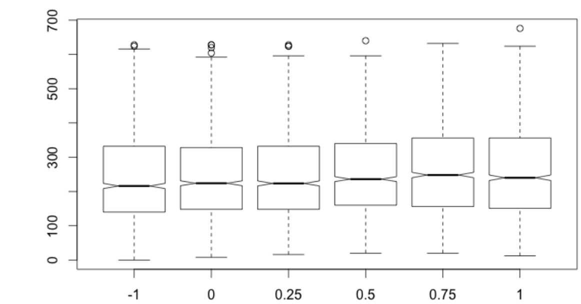

Figure 27: Impact of crossover bias on the number of hits when using the koza2 function set for the Pagie-1 symbolic regression problem.

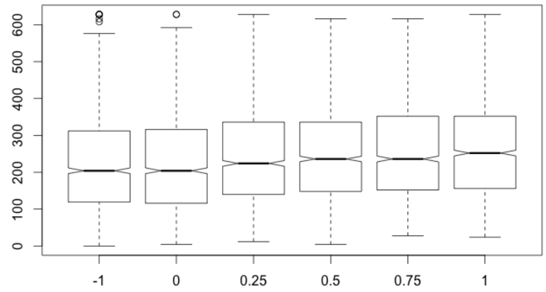

Figure 28: Impact of crossover bias on the number of hits when using the koza2 function set and Tarpeian bloat control for the Pagie-1 symbolic regression problem.

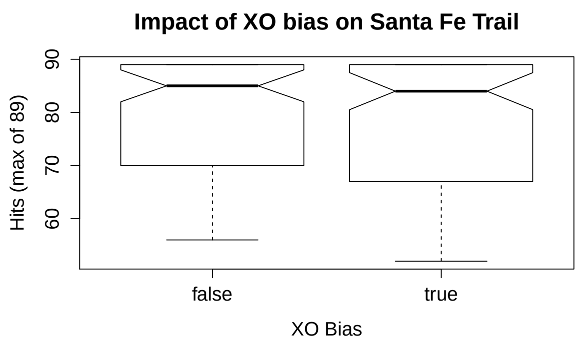

Figure 29: Impact of crossover bias on the number of hits for the Santa Fe Trail problem; the difference isn't statistically significant.

were 200 runs total, 100 with bias and 100 without. The population sizes are 4,096 and the tournament size is the ECJ default of 7; there's no elitism or Tarpeian bloat control.

These were early runs of the ant problem on the cluster. There are only 50 generations, and all we (currently) have is bias off and bias fully on (i.e., bias rate of either 0 or 1). The population sizes are 4096 (instead of either 1,024 or 10,240 as on most of the other problems). The tournament size is the ECJ default (7), and thereÕs no elitism or Tarpeian bloat control.

As we can see in Figure 29, using crossover bias on this problem appears to reduce the number of hits; this difference isn't statistically significant, however. (p-value of 0.2003 using a Wilcoxon rank sum test.) There are 39 successes without crossover bias, i.e., 39 out of 100 runs attained the best fitness of 89 hits, and 31 with crossover bias. This is also not statistically significant (p-value of 0.2994 using a 2-sample test for equality of proportions).

Figure 30 shows the hits over time, although it's skewed because runs that found the solution terminated early and thus "disappear" from this plot. It does, however, suggest that using crossover bias may be causing premature convergence that is limiting the ability to continue exploring and find a solution.

These results really don't tell us anything since none of the differences are statistically significant. Given what we've learned in other places, we could re-run these with small tournaments and larger populations. That would certainly be more consistent with our other runs, and it might turn up something interesting. Or we could just decide we don't care about this problem (it's not a particularly favored problem these days, and not recommended in the benchmarks paper) and drop all this. Thoughts?

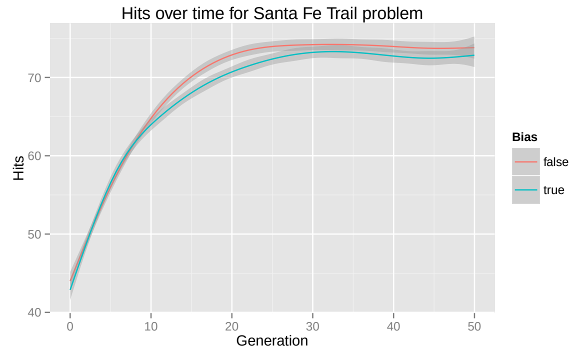

Figure 30: Number of hits over time for the Santa Fe Trail problem, both with and without crossover bias. This plot is skewed because runs that found the solution (89 hits) ended early, and thus "disappear" from the plot. This explains the slight dip in the line for runs with crossover bias turned on (the *true* line) towards the end.

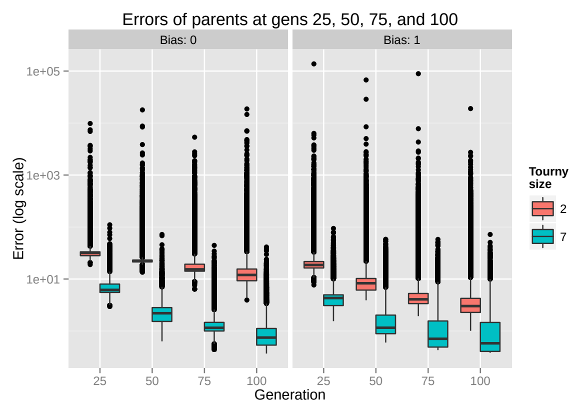

Figure 31: Plot of the errors of all individuals chosen as parents from some sine runs. **Explain this. Do we want/need this plot?**

6. DISCUSSION

Why does all this happen this way? For example, why does crossover bias have a much stronger effect when using binary tournaments than when using larger tournament sizes such as 7. One possible explanation is that with large tournaments, the difference in fitness between the two parents is likely to be closer, because the larger tournaments help ensure that both parents are from the more highly fit part of the population. To better understand this, blah, blah, blah **We need to turn this into actual text**.

7. CONCLUSIONS

Acknowledgements

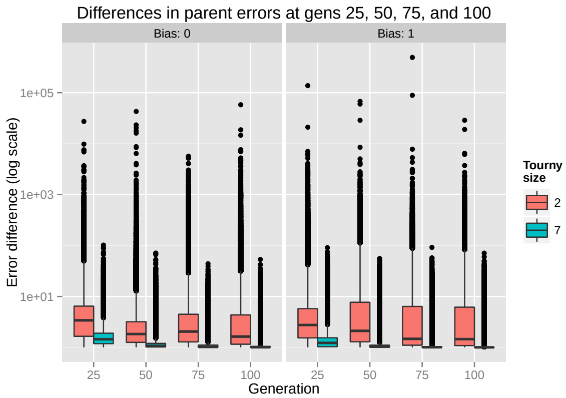

Figure 32: Plot of the differences in parent errors from some sine runs. **Explain this**.

8. REFERENCES