

# Parent Selection and Diversification in Genetic Programming

Thomas Helmuth  
Computer Science  
Washington and Lee Univ  
Lexington, Virginia  
helmuth@wlu.edu

Nicholas Freitag McPhee  
Div of Sci and Math  
Univ of MN, Morris  
Morris, MN 56267  
mcphee@morris.umn.edu

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

## ABSTRACT

More things!

## Keywords

lexicase selection, hyperselection, PushGP, other stuff

## 1. INTRODUCTION

I bet we start here!

Lexicase selection [3] is nifty, eh?

## 2. EXPERIMENTAL DESIGN

Previous work has shown that using lexicase selection results in higher population error diversity than tournament selection across a variety of problems [1, 2]. These examples examined the diversity of entire GP runs, each starting with a different initial population and random number seed.

Here we examine the effects of these parent selection methods on population diversity starting from specific population conditions besides a random initial population. In particular, we want to see how each method changes diversity in populations that occur naturally during an evolutionary run.

In order to produce the populations on which to experiment, we started GP runs and let them continue until they met certain conditions; we then stored those populations and later conducted multiple trials with different random number seeds starting with those stored populations. We used three different stopping conditions in order to generate naturally occurring populations with interesting properties:

1. In a run using lexicase selection, we stopped if the population error diversity was greater than 0.9. This results in very diverse populations, allowing us to observe whether evolution is able to maintain such high diversity in the following generations.
2. In a run using tournament selection, we stopped if the population error diversity was less than 0.15. These

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

GECCO'16, July 20-24, 2016, Denver, Colorado, USA.

© 2016 ACM. ISBN TBA.

DOI: 10.1145/1235

populations allow us to see if methods promote diversification starting from such undiverse populations. They also allow us to see if methods perform differently on a population produced by tournament selection versus one produced by lexicase selection.

3. As described above, we were initially motivated here by observations of runs using lexicase selection undergoing major drops in diversity following hyperselection events, where one or a few individuals in the population received the majority of the parent selections in a generation. We had anecdotally noticed rapid diversity recovery following these events, but not examined them systematically.

In this condition, we stopped a run using lexicase selection when the error diversity reached a level at least 0.25 less than it had been at some point in the previous 10 generations. This allowed us to detect populations that had recently undergone large drops in diversity. We do not know for sure whether those drops are related to hyperselection events, but we expect that they are.

In all three conditions, we only considered populations occurring after generation 10 in order to give evolution a chance to settle down after the extreme shifts that can happen at the beginning of a run.

- We ran each for 20 generations more - 2 problems

## 3. RESULTS

SOOOO MANY GRAPHS!

### 3.1 Starting with high diversity

### 3.2 Starting with low diversity

### 3.3 Starting after a diversity crash

## 4. CONCLUSIONS

I'm hoping we have conclusions.

## Acknowledgments

Lots of cool people helped us.

## 5. REFERENCES

- [1] T. Helmuth, N. F. McPhee, and L. Spector. Lexicase selection for program synthesis: a diversity analysis. In

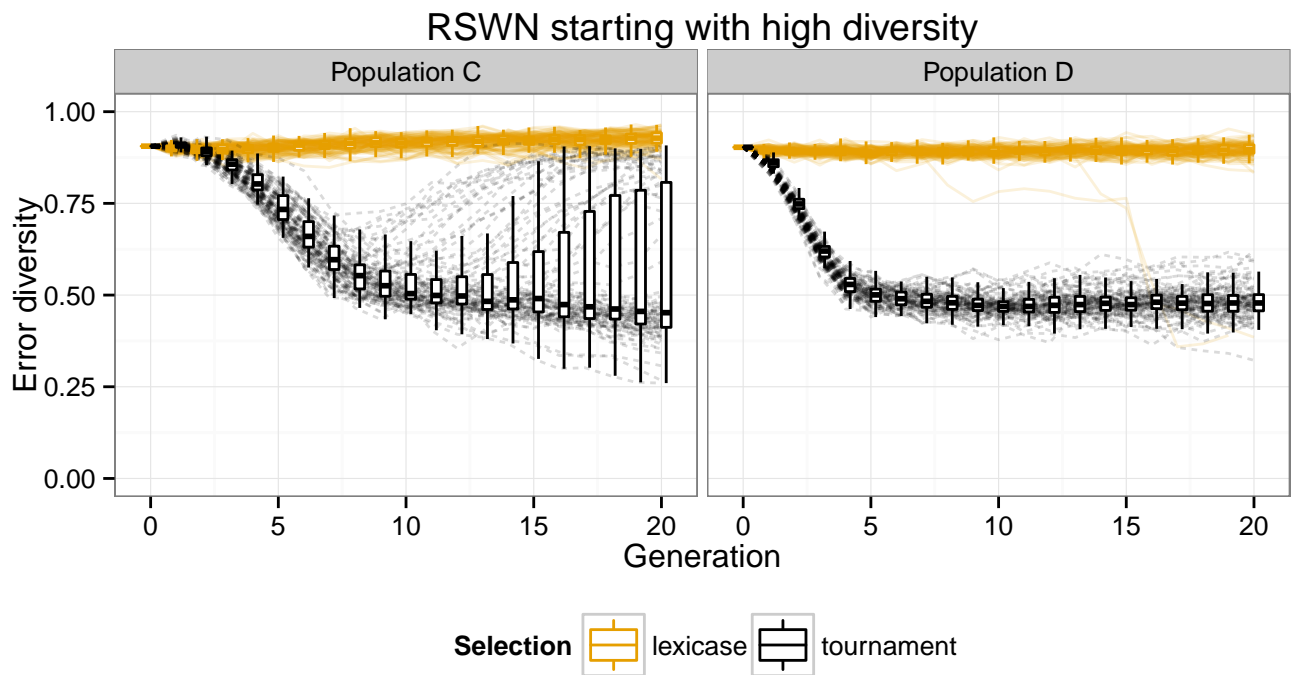


Figure 1: Changes in diversity over 100 “re-runs” of the replace-space-with-newline problem with both lexicase and tournament selections, starting from a population with high diversity coming from a lexicase selection run.

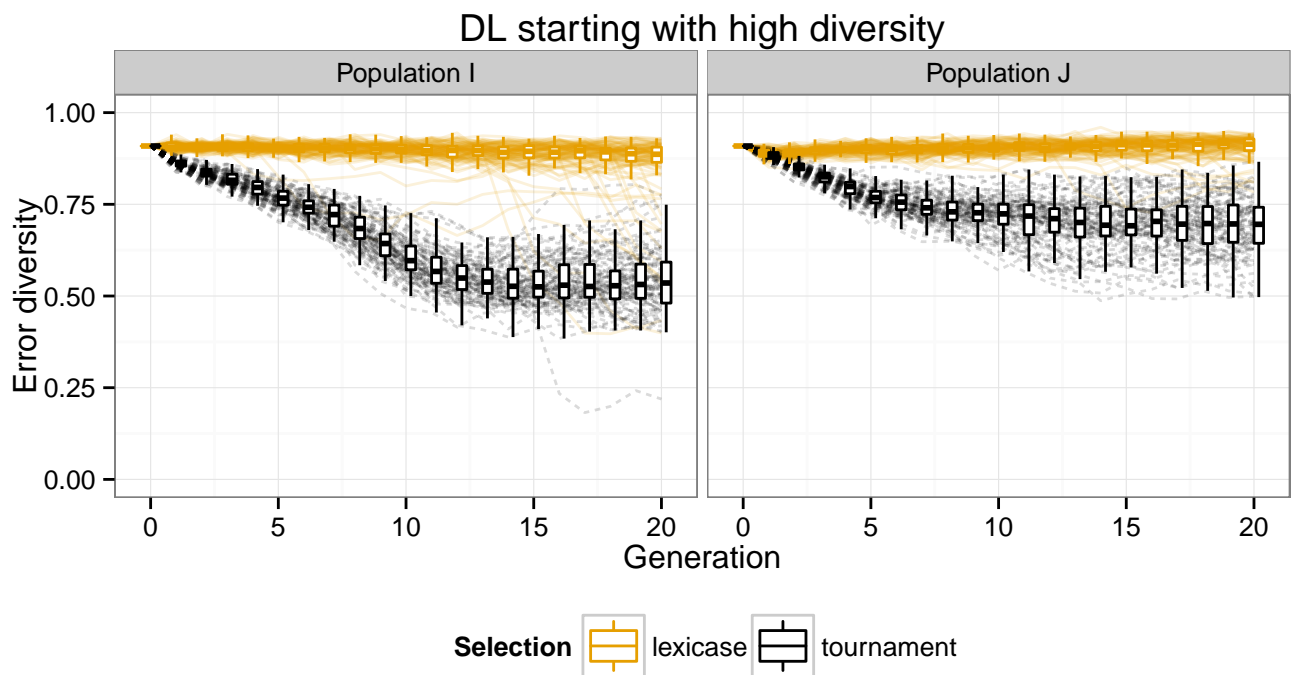


Figure 2: Changes in diversity over 100 “re-runs” of the double-letters problem with both lexicase and tournament selections, starting from a population with high diversity coming from a lexicase selection run.

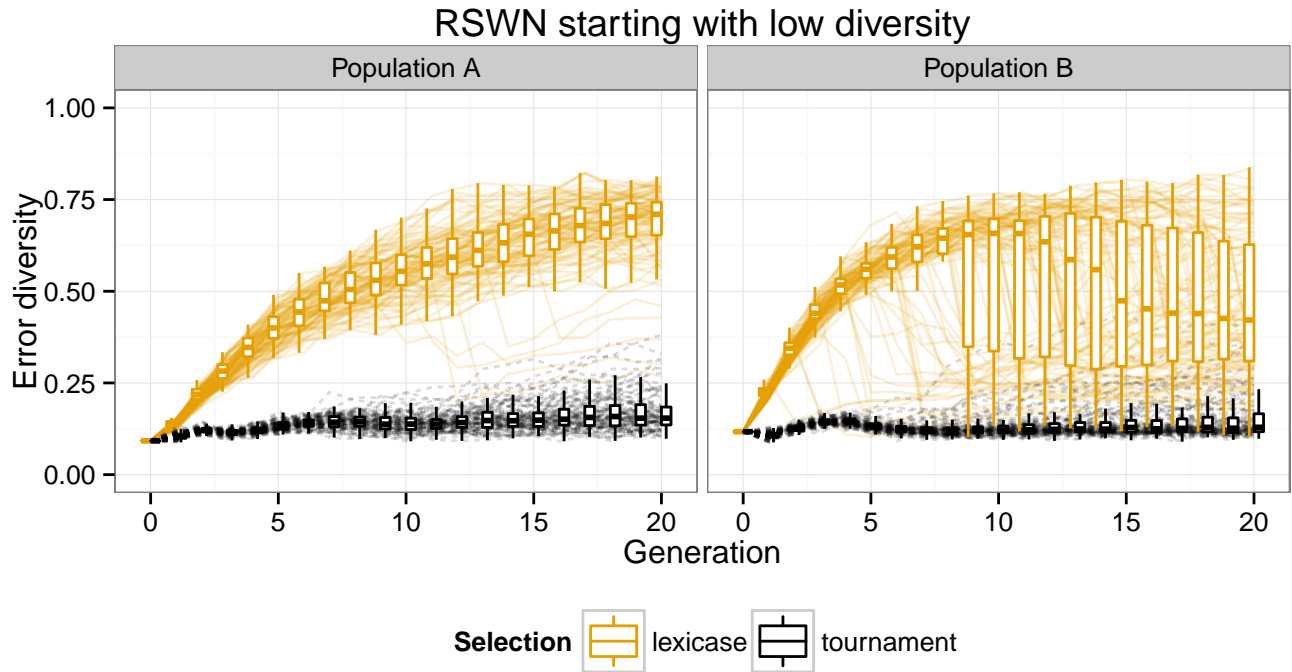


Figure 3: Changes in diversity over 100 “re-runs” of the replace-space-with-newline problem with both lexicase and tournament selections, starting from a population with low diversity coming from a tournament selection run.

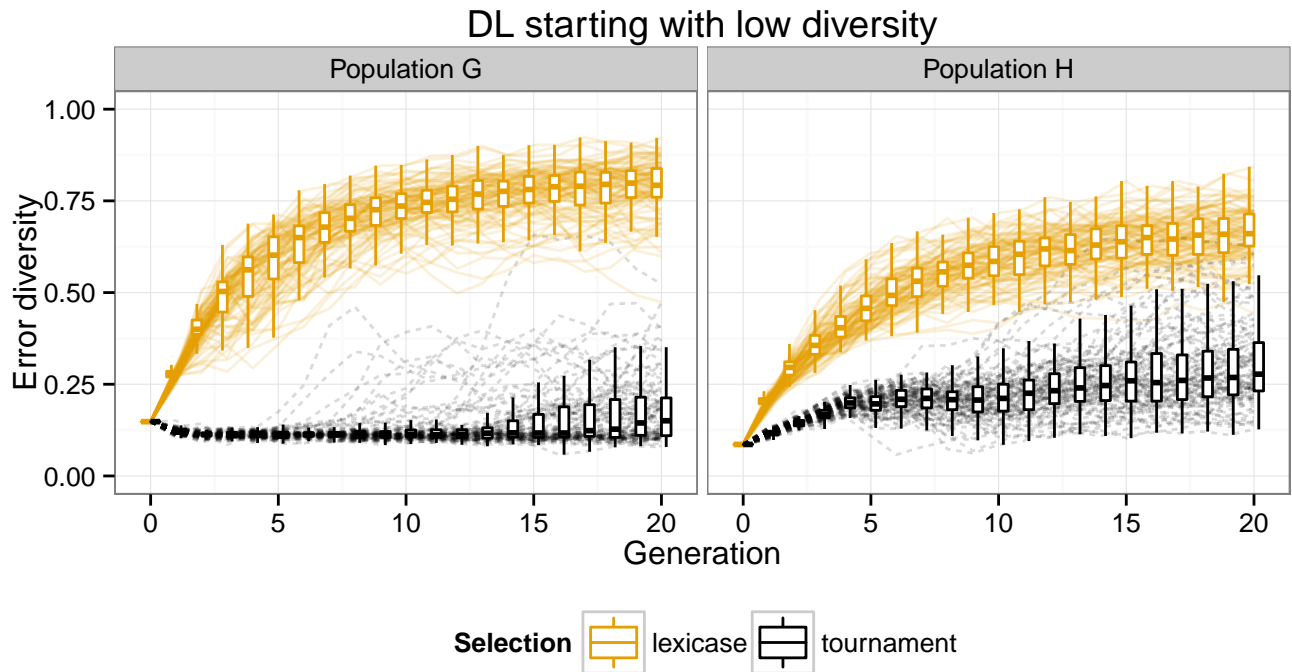


Figure 4: Changes in diversity over 100 “re-runs” of the double-letters problem with both lexicase and tournament selections, starting from a population with low diversity coming from a tournament selection run.

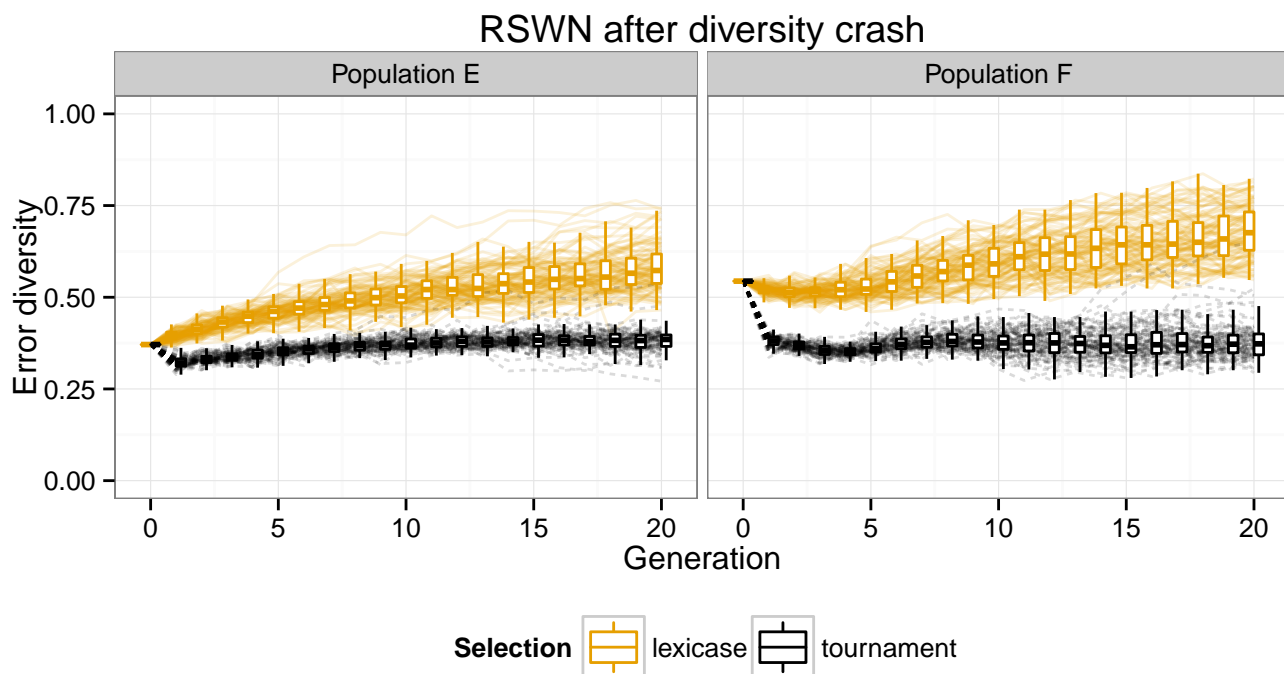


Figure 5: Changes in diversity over 100 “re-runs” of the replace-space-with-newline problem with both lexicase and tournament selections, starting from a population with that had lost diversity in a lexicase selection run.

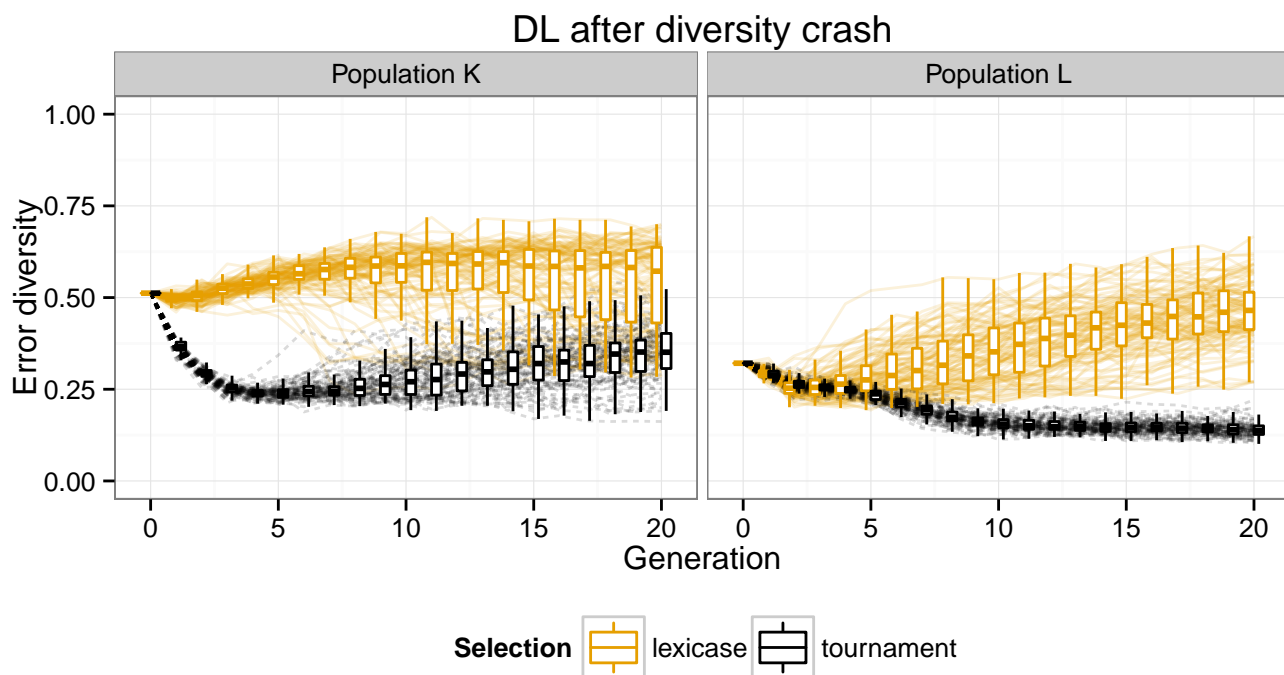


Figure 6: Changes in diversity over 100 “re-runs” of the double-letters problem with both lexicase and tournament selections, starting from a population with that had lost diversity in a lexicase selection run.

*Genetic Programming Theory and Practice XIII*,  
Genetic and Evolutionary Computation. Springer.

- [2] T. Helmuth, L. Spector, and J. Matheson. Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*, 19(5):630–643, Oct. 2015.
- [3] L. Spector. Assessment of problem modality by differential performance of lexicase selection in genetic programming: A preliminary report. In *1st workshop on Understanding Problems (GECCO-UP)*, pages 401–408, Philadelphia, Pennsylvania, USA, 7-11 July 2012. ACM.