

Nomenclature

- (training) **Cases**:
 - Samples of training data
 - Sometimes referred to as "test cases"
- **Semantics**:
 - The behavior of a GP program on the training cases
 - The genome of a GA
- **Errors**:
 - The (absolute, squared etc.) difference between an individual's semantics and the desired semantics on the training cases

Training Data					
Cases	x1	x2	x3	x4	Target
1	1	0	86	7.5	6
2	0	1	3	6.9	3
3	1	3	45	12.3	8
4	1	6	-6	0.78	9
5	0	5	29	1.2	2

Individual A	
Case	Semantics
1	16
2	8
3	13
4	-6
5	12

Individual Errors						
Case	A	B	C	D	E	
1	10	8	73	15	15	
2	5	7	60	12	12	
3	5	8	0	14	0	
4	15	8	0	15	106	
5	10	7	1	1	1	
Total Error:	45	38	134	57	134	

Nomenclature

- (training) **Cases**:
 - Samples of training data
 - Sometimes referred to as "test cases"
- **Semantics**:
 - The behavior of a GP program on the training cases
 - The genome of a GA
- **Errors**:
 - The (absolute, squared etc.) difference between an individual's semantics and the desired semantics on the training cases

Training Data					
Cases	x1	x2	x3	x4	Target
1	1	0	86	7.5	6
2	0	1	3	6.9	3
3	1	3	45	12.3	8
4	1	6	-6	0.78	9
5	0	5	29	1.2	2

Individual A	
Case	Semantics
1	16
2	8
3	13
4	-6
5	12

Individual Errors						
Case	A	B	C	D	E	
1	10	8	73	15	15	
2	5	7	60	12	12	
3	5	8	0	14	0	
4	15	8	0	15	106	
5	10	7	1	1	1	
Total Error:	45	38	134	57	134	

Nomenclature

- (training) **Cases**:
 - Samples of training data
 - Sometimes referred to as "test cases"
- **Semantics**:
 - The behavior of a GP program on the training cases
 - The genome of a GA
- **Errors**:
 - The (absolute, squared etc.) difference between an individual's semantics and the desired semantics on the training cases

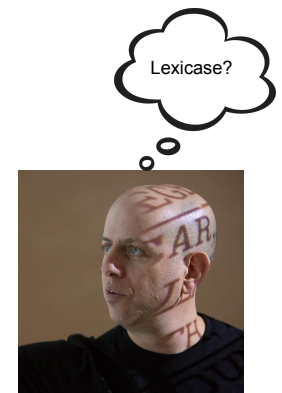
Training Data					
Cases	x1	x2	x3	x4	Target
1	1	0	86	7.5	6
2	0	1	3	6.9	3
3	1	3	45	12.3	8
4	1	6	-6	0.78	9
5	0	5	29	1.2	2

Individual A	
Case	Semantics
1	16
2	8
3	13
4	-6
5	12

Individual Errors						
Case	A	B	C	D	E	
1	10	8	73	15	15	
2	5	7	60	12	12	
3	5	8	0	14	0	
4	15	8	0	15	106	
5	10	7	1	1	1	
Total Error:	45	38	134	57	134	

Origin Story

- Late one night...
- How can Calculator Behavior be evolved?
 - Multiple unrelated functions
 - Different test cases
- How to maintain behaviors that are good on problem subsets?

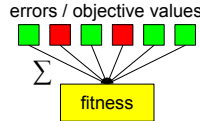


Lee Spector

Spector, Lee. (2012). Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. *GECCO*.

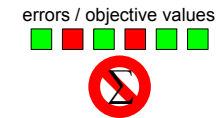
Motivations

- ❖ Most parent selection methods use a single *aggregated* fitness value
 - Ex: total error across set of training cases
 - Even multi-objective methods (e.g. NSGA-II) and quality diversity methods aggregate errors
- ❖ Obscures useful info
 - Ex: Individual Q performs well on some cases and poorly on others
 - perhaps Q has genetic material worth propagating!
 - Q has poor total error
 - Q not likely selected by tournament selection
 - The skill Q is good at may be lost in the population
- ❖ Generalists vs. Specialists



Motivation: Semantic-Aware Selection

- ❖ De-aggregating fitness
- ❖ Aggregating creates an "Information Bottleneck" - taking a rich amount of information in errors and reducing it to a single value
 - Krawiec
- ❖ Semantic-aware selection methods make use of all semantics/errors



- Krawiec, K., et al. (2015). Behavioral Program Synthesis: Insights and Prospects. *GPTP*
- Krawiec, K., & O'Reilly, U.-M. (2014). Behavioral Programming: A Broader and More Detailed Take on Semantic GP. *GECCO*.

When it's applicable

- When *fitness* can be decomposed into component parts.
 - Summations / averages over cases (mean squared error, etc)
- Places it doesn't apply:
 - Single output, black-box function optimization
- There are *enough* component parts of fitness
 - There are factorial(n_{cases}) ways to be selected with lexica selection

Who is rewarded

- Individuals that are good at cases that others aren't good at
 - More cases ✓
 - More difficult cases ✓

Example Uses of Lexicase Selection

GP Program Synthesis

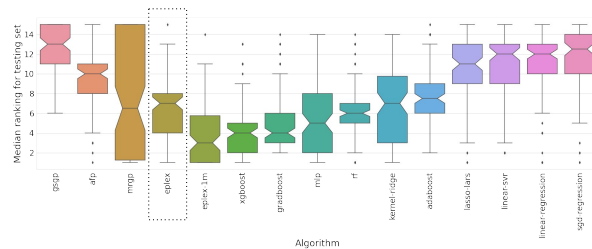
- ❖ Program synthesis: generating programs with multiple data types and control flow structures
- ❖ Lexicase selection has outperformed tournament selection and other selection methods across many benchmark problems

Problem	Tourn	IFS	Lex
Number IO	68	72	98
Small Or Large	3	3	5
For Loop Index	0	0	1
Compare String Lengths	3	6	7
Double Letters	0	0	6
Collatz Numbers	0	0	0
Replace Space with Newline	8	16	51
String Differences	0	0	0
Even Squares	0	0	2
Wallis Pi	0	0	0
String Lengths Backwards	7	10	66
Last Index of Zero	8	4	21
Vector Average	14	13	16
Count Odds	0	0	8
Mirror Image	46	64	78
Super Anagrams	0	0	0
Sum of Squares	2	0	6
Vectors Summed	0	0	1
X-Word Lines	0	0	8
Pig Latin	0	0	0
Negative To Zero	10	8	45
Scrabble Score	0	0	2
Word Stats	0	0	0
Checksum	0	0	0
Digits	0	1	7
Grade	0	0	4
Median	7	43	45
Smallest	75	98	81
Syllables	1	7	18
Problems Solved	13	13	22

- Thomas Helmuth and Lee Spector. (2015) General program synthesis benchmark suite. *GECCO*
- Forstnerlechner, S. et al. (2017). A Grammar Design Pattern for Arbitrary Program Synthesis Problems in Genetic Programming. *EuroGP*.

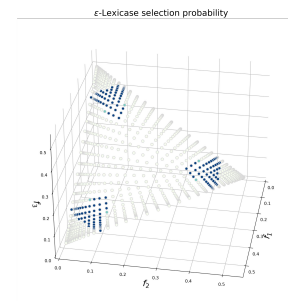
Regression

- Epsilon-lexicase selection has been shown to outperform many state-of-the-art GP and ML methods for regression

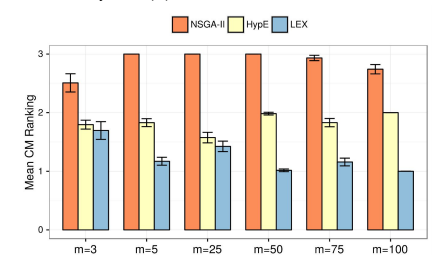


- La Cava, W. et al (2016). Epsilon-Lexicase Selection for Regression. *GECCO*
- Orzechowski, P. et al. (2018) Where Are We Now? A Large Benchmark Study of Recent Symbolic Regression Methods. *GECCO*

Many objective optimization



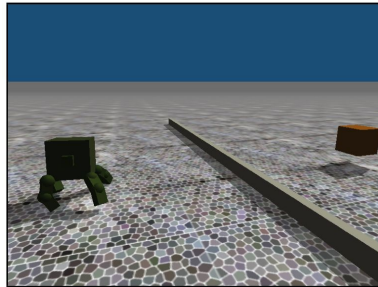
Convergence Measure Rankings, DTLZ problems, for increasing numbers of objectives (m)



La Cava, W. & Moore, J. H. (2018) An Analysis of ϵ -Lexicase Selection for Large-Scale Many-Objective Optimization. *GECCO*

Evolutionary Robotics

- ❖ In a simulated quadrupedal animat application, lexicase selection outperformed other selection methods



Moore, J. M., & Stanton, A. (2018). Tiebreaks and Diversity: Isolating Effects in Lexicase Selection. *ALIFE*.

Other Evolutionary Computation Results

- ❖ Boolean logic and finite algebras problems using GP
 - Liskowski, P. et al. (2015) Comparison of semantic-aware selection methods in genetic programming. *GECCO*.
- ❖ Learning Classifier Systems
 - Aenugu, S., & Spector, L. (2019). Lexicase Selection in Learning Classifier Systems. *GECCO*.
- ❖ Boolean constraint satisfaction using GA
 - Metevier, B. et al. (2019) Lexicase selection beyond genetic programming. *GTP*.

The Lexicase Selection Algorithm

Lexicase Selection Algorithm:

To Pick One Parent

1. `pool` $\leftarrow$ `population`
2. `cases` $\leftarrow$ list of training cases, shuffled
3. **while** `|pool|` > 1 and `|cases|` > 0:
 - a. `t` $\leftarrow$ first case in `cases`
 - b. `best` $\leftarrow$ the best error value of any individual in `pool` on case `t`
 - c. `pool` $\leftarrow$ filter `pool` to include only individuals with error of `best` on `t`
 - d. pop `t` from `cases`
4. **if** `|pool|` = 1:
 - a. return the one individual in `pool`
5. **else:**
 - a. return random individual from `pool`

Thomas Helmuth, et al. (2015) Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*.

Lexicase Selection Examples

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 1

Case order: 5, 2, 3, 1, 4

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 1

Case order: 5, 2, 3, 1, 4

- ❖ 5: best is 1, pool = {C, D, E}

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 1

Case order: 5, 2, 3, 1, 4

- ❖ 5: best is 1, pool = {C, D, E}
- ❖ 2: best is 12, pool = {D, E}
 - Note: best is always relative to pool, not full population

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 1

Case order: 5, 2, 3, 1, 4

- ❖ 5: best is 1, pool = {C, D, E}
- ❖ 2: best is 12, pool = {D, E}
 - Note: best is always relative to pool, not full population
- ❖ 1: best is 15, pool = {D, E}

	Individual				
Case	X	X	X	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 1

Case order: 5, 2, 3, 1, 4

- ❖ 5: best is 1, pool = {C, D, E}
- ❖ 2: best is 12, pool = {D, E}
 - Note: best is always relative to pool, not full population
- ❖ 1: best is 15, pool = {D, E}
- ❖ 3: best is 0, pool = {E}
- ❖ return E

	Individual				
Case	X	X	X	X	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 2

Case order: 1, 2, 5, 4, 3

- ❖ 1: best is 8, pool = {B}
- ❖ return B

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Lexicase Selection: Example 3

Case order: 3, 5, 4, 1, 2

- ❖ 3: best is 0, pool = {C, E}
- ❖ 5: best is 1, pool = {C, E}
- ❖ 4: best is 0, pool = {C}
- ❖ return C

	Individual				
Case	A	B	C	D	E
1	10	8	73	15	15
2	5	7	60	12	12
3	5	8	0	14	0
4	15	8	0	15	106
5	10	7	1	1	1
Total Error:	45	38	134	57	134

Interactive Demonstration

Interactive Demonstration: Select Talk Participants

- ❖ Visit linked website to get a list of 6 random digits
 - These are your "errors" on 6 cases
- ❖ Everyone click the "raise hand" button
 - Under reactions
 - Keep your hand up as long as you're in the selection pool
- ❖ We will run through the lexicase algorithm:
 - shuffle the 6 error indices (0 through 5)
 - shrink pool, one case at a time, until one individual remains



<https://www.random.org/integers/?num=6&min=0&max=9&col=100&base=10&format=html&rnd=new>

Epsilon Lexicase

Working with floating point semantics

- ❖ When program semantics/errors are floating point, it is much less likely to have ties.
 - This leads to very shallow selection events using lexicase selection
- ❖ Epsilon-lexicase selection
 - Relaxes the lexicase filtering step
 - Only individuals who fall outside of some epsilon of best are filtered each step

- La Cava, W. et al (2016). Epsilon-Lexicase Selection for Regression. *GECCO*
- La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexicase Selection and Epsilon-Lexicase Selection. *Evolutionary Computation*.

epsilon-Lexicase Selection Algorithm:

To Pick One Parent

1. `pool` $\leftarrow$ population
2. `cases` $\leftarrow$ list of training cases, shuffled
3. while `|pool|` > 1 and `|cases|` > 0:
 - a. `t` $\leftarrow$ first case in `cases`
 - b. `best` $\leftarrow$ the best error value of any individual in `pool` on case `t`
 - c. `epsilon` $\leftarrow$ median absolute deviation of population on case `t`
 - d. `pool` $\leftarrow$ filter `pool` to include only individuals within `epsilon` of `best`
 - e. pop `t` from `cases`
4. if `|pool|` = 1:
 - a. return the one individual in `pool`
5. else:
 - a. return random individual from `pool`

(static) epsilon-Lexicase Selection Algorithm:

To Pick One Parent

1. `pool` $\leftarrow$ population
2. `cases` $\leftarrow$ list of training cases, shuffled
3. while `|pool|` > 1 and `|cases|` > 0:
 - a. `t` $\leftarrow$ first case in `cases`
 - b. `best` $\leftarrow$ the best error value of any individual in `pool` on case `t`
 - c. `epsilon` $\leftarrow$ median absolute deviation of population on case `t`
 - d. `pool` $\leftarrow$ filter `pool` to include only individuals within `epsilon` of `best`
 - e. pop `t` from `cases`
4. if `|pool|` = 1:
 - a. return the one individual in `pool`
5. else:
 - a. return random individual from `pool`

Calculated once per generation

(dynamic) epsilon-Lexicase Selection Algorithm:

To Pick One Parent

1. `pool` $\leftarrow$ population
2. `cases` $\leftarrow$ list of training cases, shuffled
3. while `|pool|` > 1 and `|cases|` > 0:
 - a. `t` $\leftarrow$ first case in `cases`
 - b. `best` $\leftarrow$ the best error value of any individual in `pool` on case `t`
 - c. `epsilon` $\leftarrow$ median absolute deviation of `pool` on case `t`
 - d. `pool` $\leftarrow$ filter `pool` to include only individuals within `epsilon` of `best`
 - e. pop `t` from `cases`
4. if `|pool|` = 1:
 - a. return the one individual in `pool`
5. else:
 - a. return random individual from `pool`

Calculated dynamically

Optimizations and Tricks

Pre-Selection Filtering: Motivation

- ❖ In GP, programs often produce the same error vector as other programs
 - Call these *equivalent*
- ❖ If 2 or more equivalent programs would make it to the end of lexicae, we would need to look at every case to find this out
 - This is inefficient
 - If only one such individual existed, we could stop lexicae earlier

Case	Individual	
	A	B
1	17	17
2	0	0
3	4	4
4	12	12
5	1	1

Pre-Selection Filtering: Algorithm

- ❖ Group individuals into *equivalence classes* based on their error vectors
 - once per generation
- ❖ Run lexicae selection on *error vectors*, *one from each equivalence class*
 - instead of individuals
- ❖ After picking an error vector, select a random individual from its equivalence class as a parent
- ❖ This has no *functional* effect on the results of lexicae - same probability of selection for every individual
- ❖ Can provide substantial speedup of running times

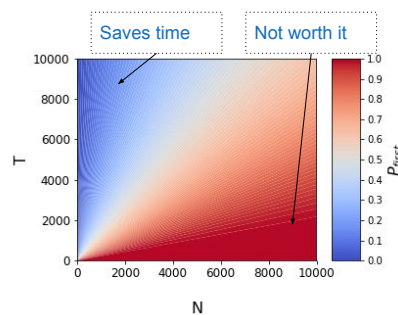
Thomas Helmuth, et al. (2020) On the importance of specialists for lexicae selection. *GPEM*

Lazy Evaluation

- Some training cases may not get used for selection
- Computational savings depend on the ratio of training cases (T) to population size (N).
- Every case probably comes first in selection when

$$T \leq \frac{1}{1 - (0.5)^{1/N}}$$

- Otherwise, lazy evaluation may see significant gains in performance.



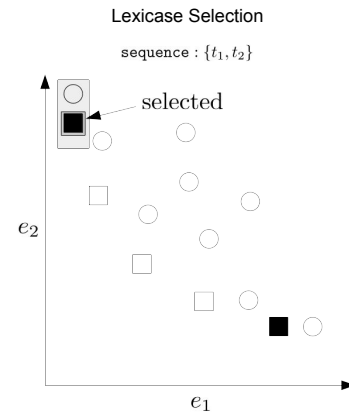
The probability of a case appearing first.

Multi-objective interpretations

La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexicae Selection and Epsilon-Lexicae Selection. *Evolutionary Computation*

Lexicase Selections are Pareto Optimal

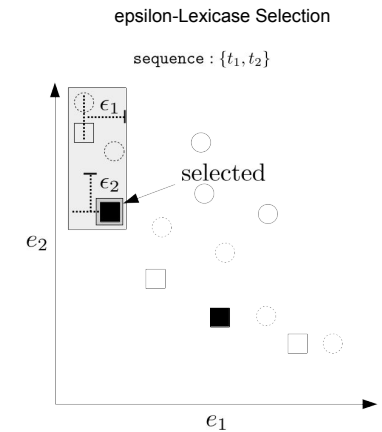
- Individuals who are selected are on the Pareto front defined by the cases



La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexicase Selection and Epsilon-Lexicase Selection. *Evolutionary Computation*

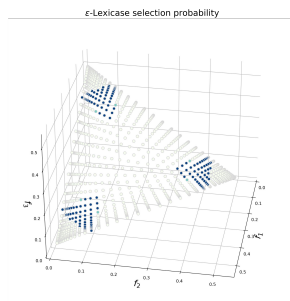
epsilon-Lexicase Selections are epsilon-Pareto Optimal

- Epsilon Lexicase selects individuals that are *epsilon*-Pareto Optimal
- Within epsilon of the Pareto Optimal points
- It does *not* necessarily select the Pareto Optimal points

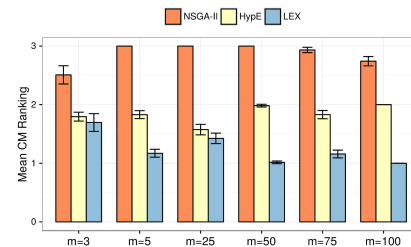


La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexicase Selection and Epsilon-Lexicase Selection. *Evolutionary Computation*

Many objective optimization



Convergence Measure Rankings, DTLZ problems, for increasing numbers of objectives (m)



La Cava, W. & Moore, J. H. (2018) An Analysis of ϵ -Lexicase Selection for Large-Scale Many-Objective Optimization. *GECCO*

Why does Lexicase Selection Work?

Specialists vs. Generalists

❖ Specialists:

- relatively low errors on a subset of training cases
- relatively high errors on other training cases
- poor total error (aggregate fitness) relative to population

Low (good) errors on green cases
Ex: 8 9 1 8 7 0 0 7 9 = 49
High (bad) errors on red cases total

❖ Generalists:

- similar errors on all training cases
- not particularly low errors on any training cases
- good total error relative to population

Mediocre errors on all cases
Ex: 3 2 3 3 3 3 4 2 3 = 26
total

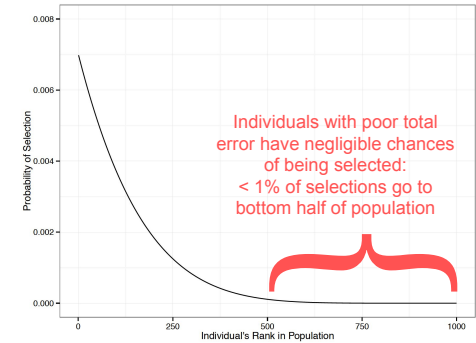
Thomas Helmuth et al. (2019) Lexicase selection of specialists. *GECCO*

Specialists vs. Generalists

❖ Which are better to select?

- Aggregating errors emphasizes generalists
- Lexicase selection emphasizes specialists

❖ Empirical answer is specialists in most cases

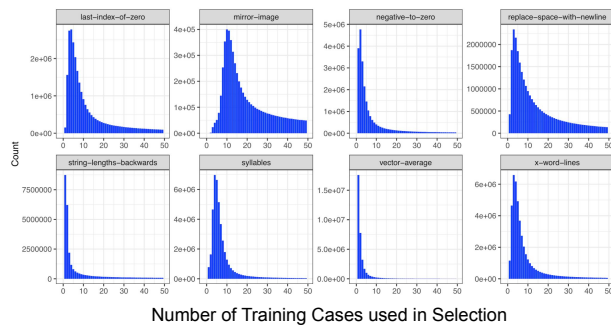


Ex: Tournament size = 7

Thomas Helmuth et al. (2019) Lexicase selection of specialists. *GECCO*

Groups of Cases

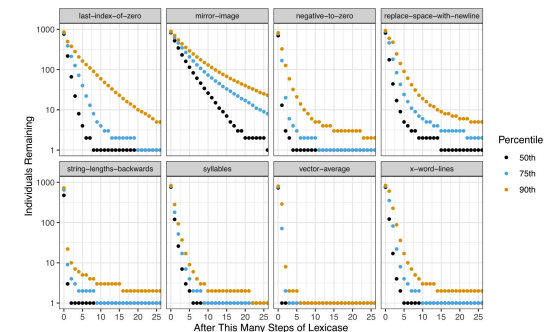
- ❖ Most lexicase selection events use small numbers of cases
- ❖ Cases near the beginning of the shuffled list have largest impact on selection
- ❖ Cases near end of list have little or no impact!



Thomas Helmuth, et al. (2020) On the importance of specialists for lexicase selection. *GPEM*

Individuals Remaining after x Lexicase steps

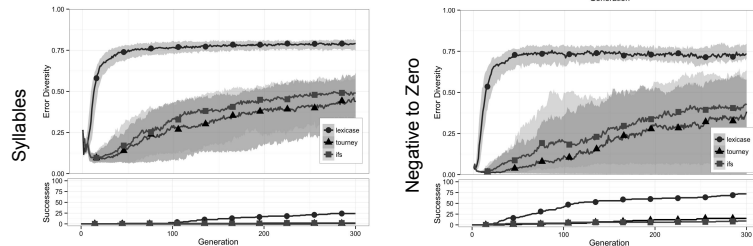
- ❖ Selection pools often reduce to a small number of individuals within 5-10 cases



Thomas Helmuth, et al. (2020) On the importance of specialists for lexicase selection. *GPEM*

Population Diversity in GP

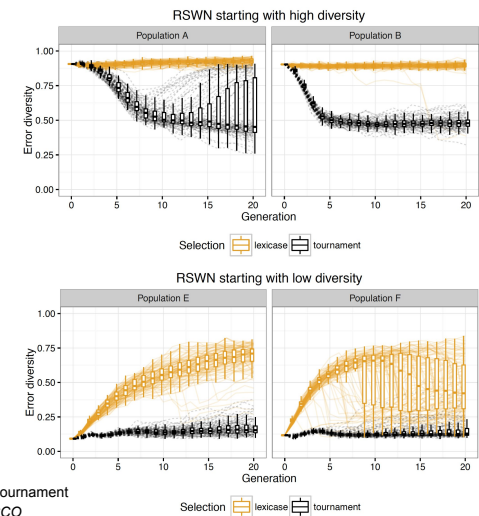
- ❖ Lexicase selection produces and maintains higher levels of population diversity across full GP runs
- ❖ Why?
 - it selects individuals that perform well in different parts of the search space



Thomas Helmuth et al. (2015) Lexicase selection for program synthesis: A diversity analysis. *GPTP*

Experiment: Diversity in GP

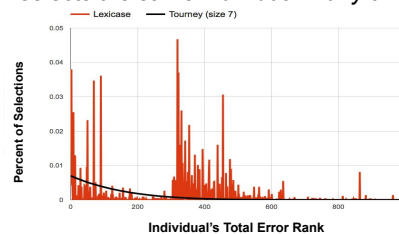
- ❖ Start with the same population
 - tested low diversity and high diversity populations
- ❖ Measure diversity over next 20 generations
- ❖ Tournament selection loses diversity or maintains an initial low diversity
- ❖ Lexicase selection maintains an initial high diversity or increases diversity



Thomas Helmuth et al. (2016) Effects of lexicase and tournament selection on diversity recovery and maintenance. *GECCO*

Hyperselection

- ❖ Lexicase often selects the same individual *many* times in one generation



- ❖ Does this hyperselection help (or hurt) performance?
- ❖ Empirical evidence indicates it neither helps nor hurts!

Thomas Helmuth et al. (2016) The impact of hyperselection on lexicase selection. *GECCO*

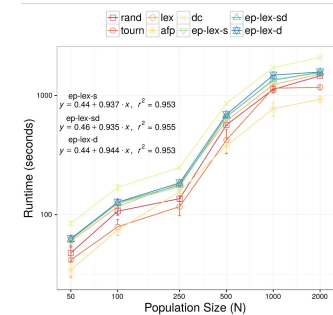
Running Time

Worst case running time

- Population of N individuals, T training cases
- The worst case running time for a single selection event is $O(NT)$
- For a generation, lexibase selection has worst-case complexity $O(N^2T)$
- Occurs when all individuals are identical
 - In other words, doesn't occur with pre-selection filtering
- Rarely observed

Experimental Running Time

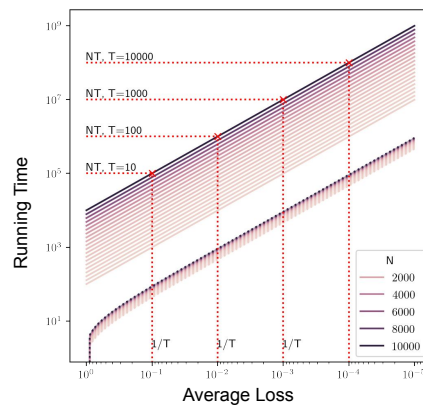
- Observed running time is much better than the worst-case
- Closer to linear in population size



La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexibase Selection and Epsilon-Lexibase Selection. *Evolutionary Computation*

What about expected running time?

Under some assumptions, we can show that the expected running time of lexibase selection grows *linearly* with the population's average loss, approaching the worst case as the population converges.



Helmuth, T. & La Cava, W. (2021) Expected Running Time of Lexibase Selection. *Under Review*

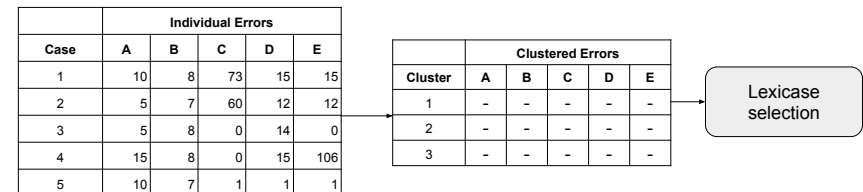
Extensions

Extensions

- ❖ Alternate definitions of epsilon
 - User-defined thresholds
 - Moore & McKinley (2016) A Comparison of Multiobjective Algorithms in Evolving Quadrupedal Gaits. *SAB*
 - La Cava et al (2016) Epsilon lexibase selection for regression. *GECCO*
 - MADCAP epsilon lexibase
 - Spector, L. et al. (2018) Relaxations of Lexibase Parent Selection. *GPTP XV*
- ❖ epsilon-lexibase survival
 - La Cava, W.; Moore, J. (2017) A General Feature Engineering Wrapper for Machine Learning Using epsilon-Lexibase Survival. *EuroGP*
- ❖ Combinations with other methods
 - Novelty search: Knobelty and novelty-lexibase
 - DOCLEX
 - Liskowski, P.; Krawiec, K. (2017) Discovery of Search Objectives in Continuous Domains. *GECCO*
- ❖ Using smaller pools / islands
 - Works when less selection pressure is desirable

Discovery of Objectives + Lexibase Selection

- Apply clustering to population semantics to identify sub-tasks
- Feed these into lexibase selection



Liskowski, P.; Krawiec, K. (2017) Discovery of Search Objectives in Continuous Domains. *GECCO 17*

Down-sampled Lexibase Selection

- ❖ Each generation, use a subsample of the training cases to evaluate individuals
 - Similar to mini-batches used in gradient descent
- ❖ Fewer program evaluations → longer evolution for the same computational cost
- ❖ Works very well, even using small portions (5-10%) of the training set

Weighted Case Shuffling

- ❖ Natural question: is there a better way to shuffle cases than uniformly random?
- ❖ Tested:
 - 3 different weighted shuffle algorithms
 - 9 different bias metrics for weighting cases
- ❖ None of these outperform uniform shuffle!
- ❖ Why? Hypotheses:
 - Lower diversity because of less even emphasis on the search space
 - Fewer selections of specialists that perform well on cases that receive less emphasis

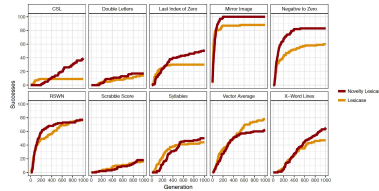
- Hernandez, J. G. et al. (2019). Random subsampling improves performance in lexibase selection. *GECCO*.
- Ferguson, A. J. et al. (2019). Characterizing the Effects of Random Subsampling on Lexibase Selection. *GPTP*.
- Thomas Helmuth and Lee Spector. (2020) Explaining and exploiting the advantages of down-sampled lexibase selection. *ALife*.

Sarah Anne Troise, Thomas Helmuth. (2017) Lexibase selection with weighted shuffle. *GPTP*.

Combining Lexicase and Novelty Search

Novelty Lexicase Selection

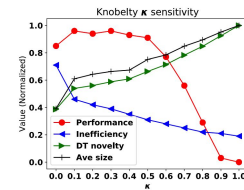
- ❖ Combines novelty scores on each case and errors into one set of cases
- ❖ Produces more diversity and higher successes in long GP runs



Lia Jundt, Thomas Helmuth. (2019). Comparing and combining lexicase selection and novelty search. *GECCO*.

Knobelly

- ❖ Uses novelty search selection K proportion of the time and lexicase selection (1 - K) proportion of the time



Kelly, J. at al. (2019). Improving Genetic Programming with Novel Exploration-Exploitation Control. *EuroGP*.

Live Demo

Acknowledgments

- ❖ Lee Spector, Nic McPhee, Bill Tozier, Jason Moore
- ❖ Epistasis Lab at UPenn
- ❖ Grants
 - La Cava: NIH K99-LM012926

References (1)

- ❖ Aenugu, S., & Spector, L. (2019). Lexicase Selection in Learning Classifier Systems. *GECCO*.
- ❖ Ferguson, A. J. et al. (2019). Characterizing the Effects of Random Subsampling on Lexicase Selection. *GTPP*.
- ❖ Forstenlechner, S. et al. (2017). A Grammar Design Pattern for Arbitrary Program Synthesis Problems in Genetic Programming. *EuroGP*.
- ❖ Thomas Helmuth and Lee Spector. (2020) Explaining and exploiting the advantages of down-sampled lexicase selection. *ALife*.
- ❖ Thomas Helmuth, et al. (2015) Solving uncompromising problems with lexicase selection. *IEEE Transactions on Evolutionary Computation*.
- ❖ Thomas Helmuth and Lee Spector. (2015) General program synthesis benchmark suite. *GECCO*
- ❖ Thomas Helmuth et al. (2015) Lexicase selection for program synthesis: A diversity analysis. *GTPP*
- ❖ Thomas Helmuth et al. (2016) Effects of lexicase and tournament selection on diversity recovery and maintenance. *GECCO*
- ❖ Thomas Helmuth et al. (2016) The impact of hyperselection on lexicase selection. *GECCO*
- ❖ Thomas Helmuth et al. (2019) Lexicase selection of specialists. *GECCO*
- ❖ Thomas Helmuth, et al. (2020) On the importance of specialists for lexicase selection. *GPEM*
- ❖ Helmuth, T. & La Cava, W. (2021) Expected Running Time of Lexicase Selection. Under Review

References (2)

- ❖ Hernandez, J. G. et al. (2019). Random subsampling improves performance in lexicase selection. *GECCO*.
- ❖ Lia Jundt, Thomas Helmuth. (2019). Comparing and combining lexicase selection and novelty search. *GECCO*.
- ❖ Kelly, J. at al. (2019). Improving Genetic Programming with Novel Exploration-Exploitation Control. *EuroGP*.
- ❖ Krawiec, K., et al. (2015). Behavioral Program Synthesis: Insights and Prospects. *GPTP*
- ❖ Krawiec, K., & O'Reilly, U.-M. (2014). Behavioral Programming: A Broader and More Detailed Take on Semantic GP. *GECCO*.
- ❖ La Cava, W. et al (2016). Epsilon-Lexicase Selection for Regression. *GECCO*
- ❖ La Cava, W.; Moore, J. (2017) A General Feature Engineering Wrapper for Machine Learning Using epsilon-Lexicase Survival. *EuroGP*
- ❖ La Cava, W. & Moore, J. H. (2018) An Analysis of ϵ -Lexicase Selection for Large-Scale Many-Objective Optimization. *GECCO*
- ❖ La Cava, W. et al. (2019). A Probabilistic and Multi-Objective Analysis of Lexicase Selection and Epsilon-Lexicase Selection. *Evolutionary Computation*.
- ❖ Liskowski, P. et al. (2015) Comparison of semantic-aware selection methods in genetic programming. *GECCO*.
- ❖ Liskowski, P.; Krawiec, K. (2017) Discovery of Search Objectives in Continuous Domains. *GECCO*
- ❖ Metevier, B. et al. (2019) Lexicase selection beyond genetic programming. *GPTP*.

References (3)

- ❖ Moore & McKinley (2016) A Comparison of Multiobjective Algorithms in Evolving Quadrupedal Gaits. *SAB*
- ❖ Moore, J. M., & Stanton, A. (2018). Tiebreaks and Diversity: Isolating Effects in Lexicase Selection. *ALIFE*.
- ❖ Orzechowski, P. et al. (2018) Where Are We Now? A Large Benchmark Study of Recent Symbolic Regression Methods. *GECCO*
- ❖ Spector, Lee. (2012). Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. *GECCO*.
- ❖ Spector, L. et al. (2018) Relaxations of Lexicase Parent Selection. *GPTP*
- ❖ Sarah Anne Troise, Thomas Helmuth. (2017) Lexicase selection with weighted shuffle. *GPTP*