NEO X 2022

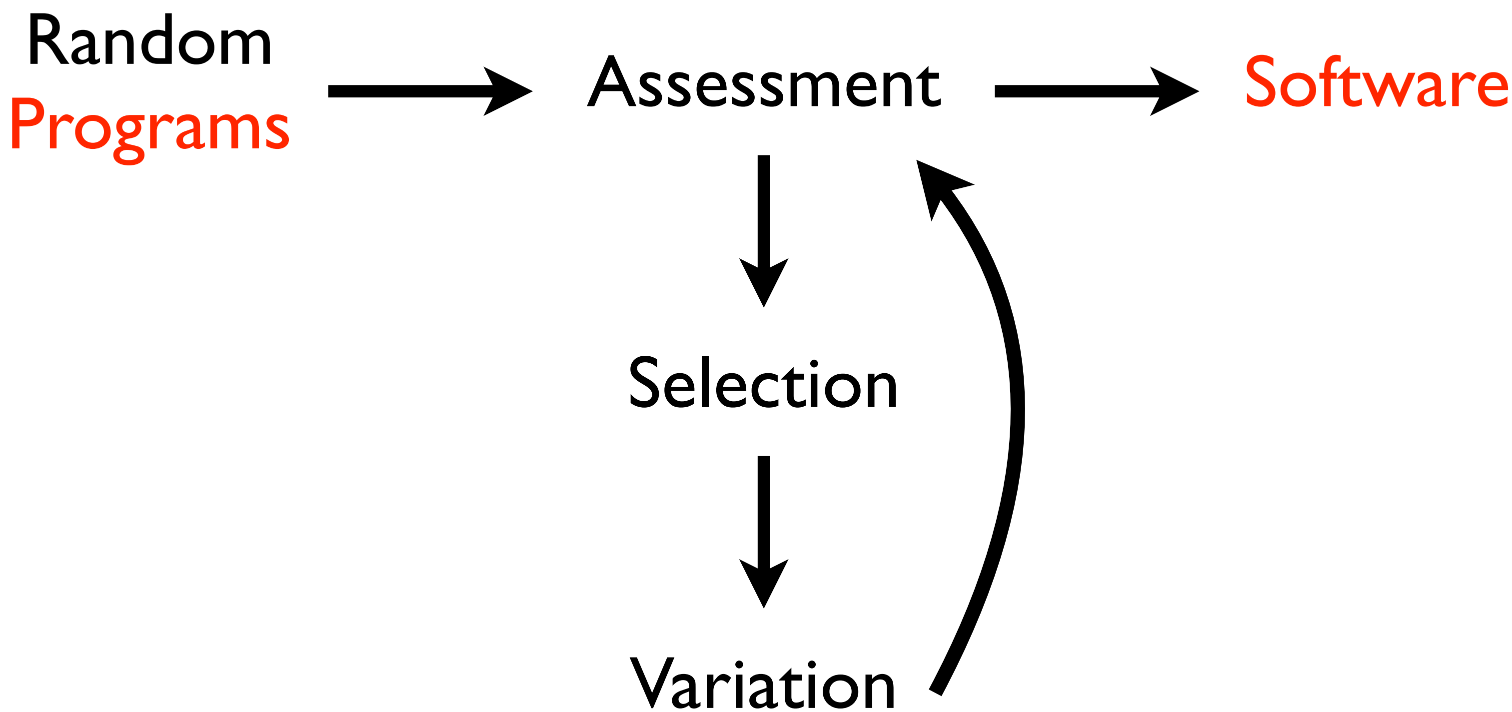
Honor All the Things

A Lexicase Selection Manifesto

Lee Spector
Amherst College
University of Massachusetts, Amherst

Σ

- A narrow problem
- A hack that solves it
- Honor all the things!
- Where, when, how, and why it works
- Much much much broader implications?


Random
Programs
Assessment
Software
Selection
Variation

STAPLES®
SPL-110
12345678
%
OFF
ON/C
MRC
M–
M+
CE
7
8
9
÷
4
5
6
×
1
2
3
–
0
•
=
+

- Don't add up everything you care about
- Instead, consider each thing in its own right
- One at a time, in random order
- **Honor all the things!**
- And all the combinations of all the things

Lexicase Selection Algorithm:

To Pick One Parent

```
1. pool ← population
2. cases ← list of training cases, shuffled
3. while |pool| > 1 and |cases| > 0:
   a. t ← first case in cases
   b. best ← the best error value of any individual in pool on case t
   c. pool ← 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
```

(Slide by Thomas Helmuth and William La Cava)

ϵ-Lexicase Selection for Regression

GECCO '16, July 20 - 24, 2016, Denver, CO, USA

William La Cava*
Department of Mechanical and Industrial Engineering
University of Massachusetts
Amherst, MA 01003
wlacava@umass.edu

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

Kourosh Danai
Department of Mechanical and Industrial Engineering
University of Massachusetts
Amherst, MA 01003
danai@engin.umass.edu

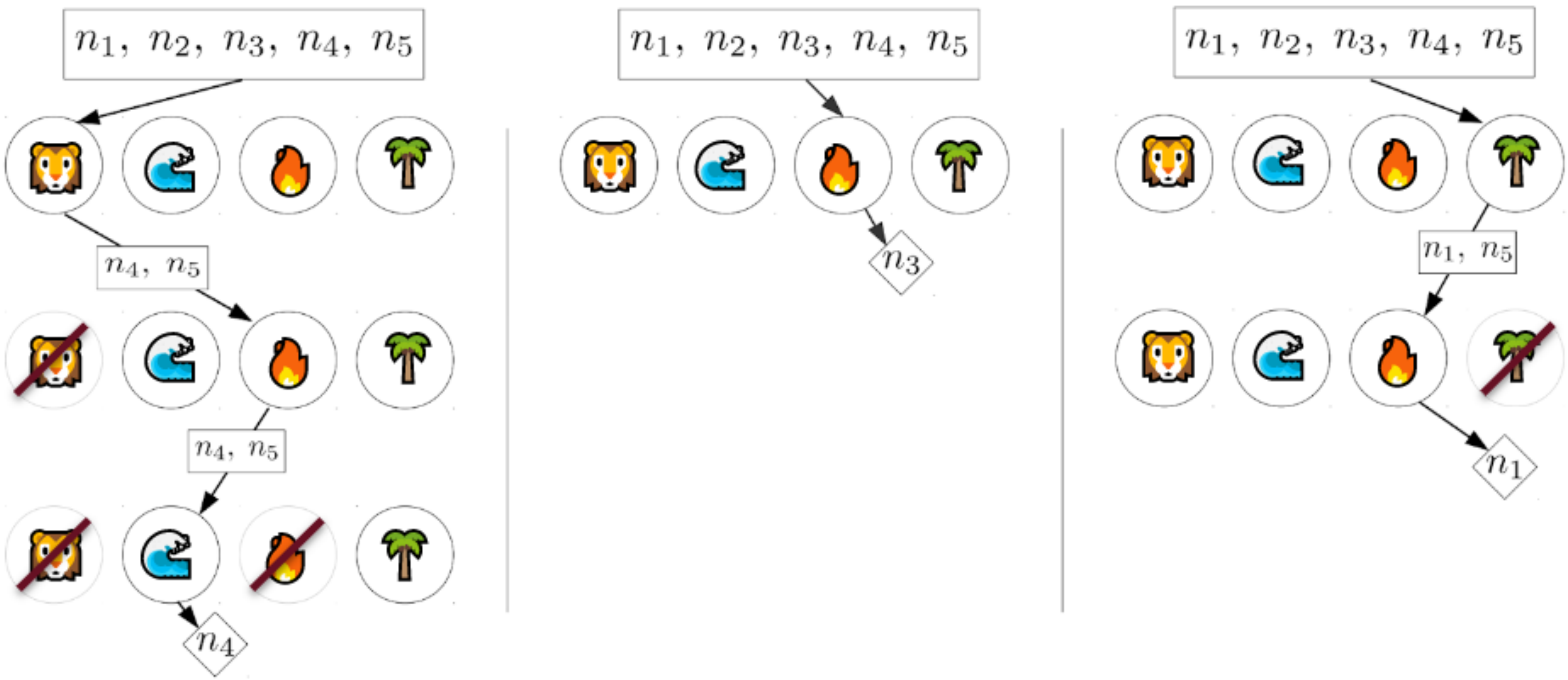
Lexicase Selection Algorithm:

To Pick One Parent

```
1.  pool ← population
2.  cases ← list of training cases, shuffled
3.  while |pool| > 1 and |cases| > 0:
    a.  t ← first case in cases
    b.  best ← the best error value of any individual in pool on case t
    c.  pool ← 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
```

within ϵ

Lexicase Selection



(Slide by Ryan Bahlous-Boldi)

Biological Selection

- Survive challenges that you happen to face
- Until you can reproduce
- Each challenge may be competitive

Multi/Many Objectives?

- Often hundreds or more objectives; "multi" usually means ≤ 3; "many" often still means ≤ 20
- Objectives may be parts of traditional objectives
- Often uncompromising

vs.

- Lexicase selection may also help for traditional many-objective optimization tasks

Multi-objective optimization

From Wikipedia, the free encyclopedia

See also: Multiple-criteria decision analysis and Vector optimization

Multi-objective optimization (also known as **multi-objective programming**, **vector optimization**, **multicriteria optimization**, **multiattribute optimization** or **Pareto optimization**) is an area of multiple criteria decision making that is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously. Multi-objective optimization has been applied in many fields of science, including engineering, economics and logistics where optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Minimizing cost while maximizing comfort while buying a car, and maximizing performance whilst minimizing fuel consumption and emission of pollutants of a vehicle are examples of multi-objective optimization problems involving two and three objectives, respectively. In practical problems, there can be more than three objectives.

RESEARCH-ARTICLE

Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report

Author: Lee Spector Authors Info & Claims

GECCO '12: Proceedings of the 14th annual conference companion on Genetic and evolutionary computation • July 2012
• Pages 401–408 • https://doi.org/10.1145/2330784.2330846

Table 1: Fitness components (errors by case, with smaller being better) and lexicase selection probabilities for a small population on a hypothetical problem.

	Fitness case				Lexicase selection probability	Total error
Individual	*a*	*b*	*c*	*d*		
#1	2	2	4	2	0.250	10
#2	1	2	4	3	0.000	10
#3	2	2	3	4	0.333	11
#4	0	2	5	5	0.208	12
#5	0	3	5	2	0.208	10

Journals & Magazines > IEEE Transactions on Evolutio... > Volume: 19 Issue: 5

Solving Uncompromising Problems With Lexicase Selection

Publisher: IEEE

Cite This

Thomas Helmuth ; Lee Spector ; James Matheson

Date of Publication: 13 October 2014

TABLE II
ALGEBRAIC OPERATORS DEFINING THE FINITE ALGEBRAS IN THIS PAPER.

A_1 *	0	1	2
0	2	1	2
1	1	0	0
2	0	0	1

A_2 *	0	1	2
0	2	0	2
1	1	0	2
2	1	2	1

prior experiments with the problems; none were particularly optimized.

Previous work in genetic programming for finite algebras has created human-competitive results (and won a "Humies" Gold Prize) [27]. Here, we borrow a problem from that work to use as a benchmark. This problem, which we will simply call the **finite algebras problem**, is to find a discriminator term in a three-element, single-operator algebra. A *discriminator term* [28] is a ternary function $t(x, y, z)$ satisfying

$$t(x,y,z) = \begin{cases} x \text{ if } x \neq y \\ z \text{ if } x = y \end{cases}$$

TABLE V
RESULTS ON THE FINITE ALGEBRAS PROBLEM USING THE ALGEBRA A_2 WITH 100 RUNS IN EACH CONDITION. IFS GIVES RESULTS USING IMPLICIT FITNESS SHARING PARENT SELECTION. THE SUCCESS RATE OF EACH SET OF RUNS USING TOURNAMENT SELECTION OR IMPLICIT FITNESS SHARING IS COMPARED WITH THE SUCCESS RATE USING LEXICASE SELECTION; WE PRESENT THAT DIFFERENCE AND A 95% CONFIDENCE INTERVAL OF THAT DIFFERENCE.

Parent Selection Method	Tournament Size	Success Rate	Difference in Success Rate with Lexicase	95% Confidence Interval of Difference in Success Rate
Lexicase	-	1.0	-	-
Tournament	2	0	1.0	[0.953, 1.0]
Tournament	3	0.06	0.94	[0.869, 0.974]
Tournament	4	0.12	0.88	[0.795, 0.930]
Tournament	5	0.14	0.86	[0.772, 0.914]
Tournament	6	0.16	0.84	[0.749, 0.898]
Tournament	7	0.17	0.83	[0.737, 0.890]
Tournament	8	0.10	0.90	[0.819, 0.946]
Tournament	9	0.26	0.74	[0.638, 0.813]
Tournament	10	0.18	0.82	[0.726, 0.882]
IFS	2	0.28	0.72	[0.616, 0.795]
IFS	3	0.61	0.39	[0.286, 0.479]
IFS	4	0.74	0.26	[0.167, 0.343]
IFS	5	0.83	0.17	[0.090, 0.243]
IFS	6	0.84	0.16	[0.082, 0.232]
IFS	7	0.83	0.17	[0.090, 0.243]
IFS	8	0.88	0.12	[0.050, 0.185]
IFS	9	0.79	0.21	[0.124, 0.288]
IFS	10	0.72	0.28	[0.185, 0.364]

GP Program Synthesis

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

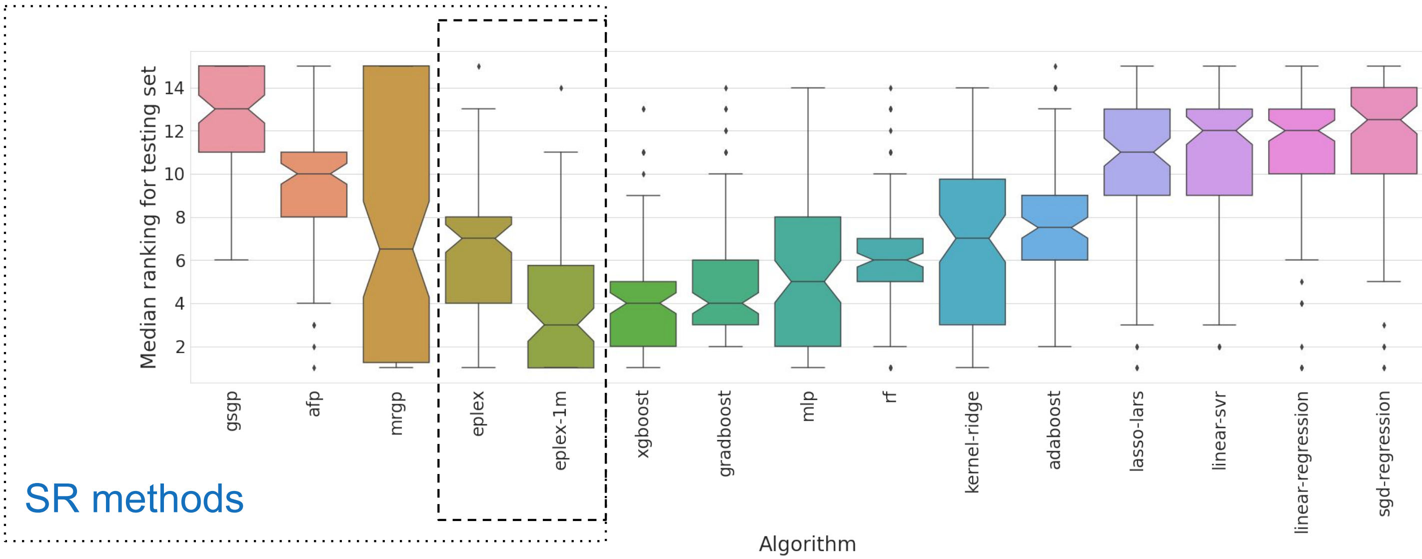
Problem	Tourn	IFS	Lex
Number IO	68	72	98
Small Or Large	3	3	5
For Loop Index	0	0	1
Compare String Lengths	3	6	7
Double Letters	0	0	6
Collatz Numbers	0	0	0
Replace Space with Newline	8	16	51
String Differences	0	0	0
Even Squares	0	0	2
Wallis Pi	0	0	0
String Lengths Backwards	7	10	66
Last Index of Zero	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*
- Forstenlechner, S. et al. (2017). A Grammar Design Pattern for Arbitrary Program Synthesis Problems in Genetic Programming. *EuroGP*.

(Slide by Thomas Helmuth and Bill La Cava)

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*

(Slide by Thomas Helmuth and Bill La Cava)

Lexicase Selection Beyond Genetic Programming

Genetic Programming Theory and Practice XVI

Blossom Metevier, Anil Kumar Saini, and Lee Spector

Table 3: Success rate for the genetic algorithm with fitness proportionate, tournament (size 2), and lexicase parent selection for each studied combination of v (number of variables) and c (number of constraints). Underlines indicate statistically significant improvements, determined using a pairwise chi-square test with Holm correction and $p < 0.05$.

Number of Variables (v)	Number of Constraints (c)	Fitness Proportionate	Tournament (size 2)	Lexicase
20	8	0.835	0.867	<u>0.992</u>
20	12	0.940	0.954	<u>1.000</u>
20	16	0.980	0.987	<u>1.000</u>
20	32	0.999	1.000	1.000
30	8	0.415	0.475	<u>0.889</u>
30	12	0.614	0.697	<u>0.995</u>
30	16	0.815	0.869	<u>1.000</u>
30	32	0.983	0.995	1.000
40	8	0.205	0.257	<u>0.689</u>
40	12	0.224	0.310	<u>0.927</u>
40	16	0.433	0.576	<u>0.993</u>
40	32	0.861	0.944	<u>1.000</u>

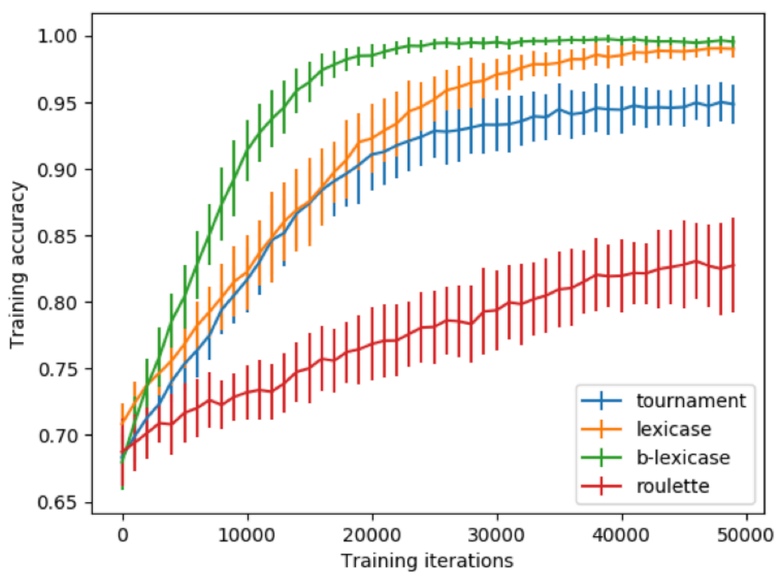
- Boolean constraint satisfaction problems
- Genetic algorithm with fixed-length linear genomes

GECCO '19, July 13–17, 2019, Prague, Czech Republic

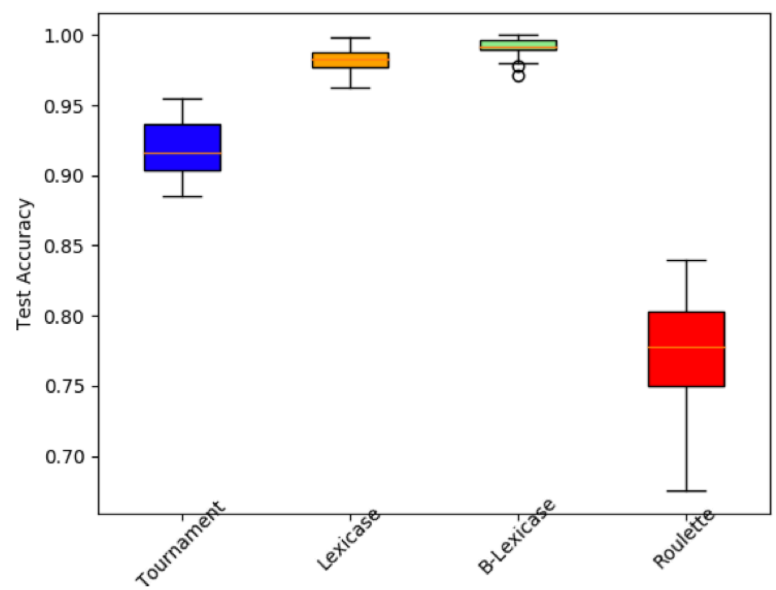
Lexicase Selection in Learning Classifier Systems

Sneha Aenugu
University of Massachusets Amherst
Amherst, Massachusetts
saenugu@umass.edu

Lee Spector
Hampshire College
University of Massachusetts Amherst
Amherst, Massachusets
lspector@hampshire.edu



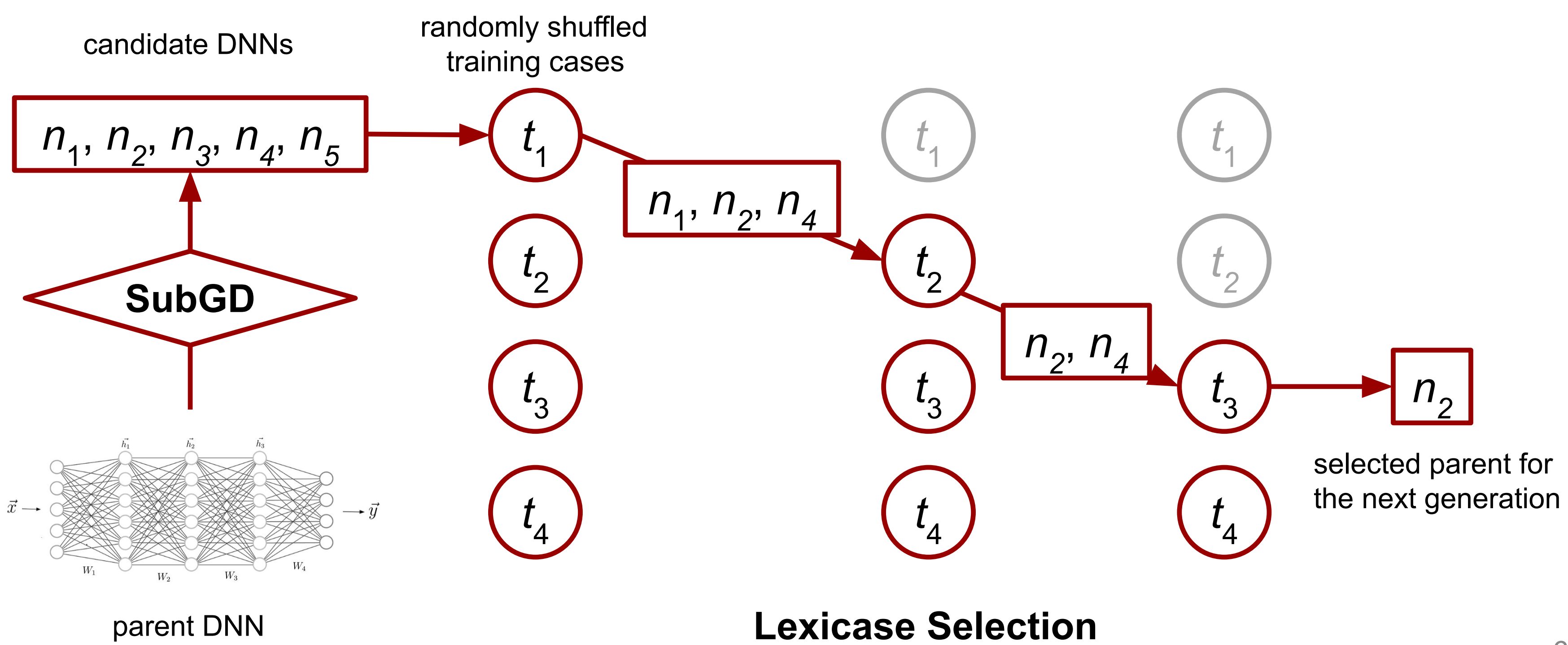
(a) Training Accuracy



(b) Test Accuracy

- 20-bit multiplexer problem
- Batch lexicase selection

Gradient Lexicase Selection



Optimizing Neural Networks with Gradient Lexicase Selection, Li Ding and Lee Spector, *ICLR 2022*

(Slide by Li Ding)

www.nature.com/scientificreports

scientific reports

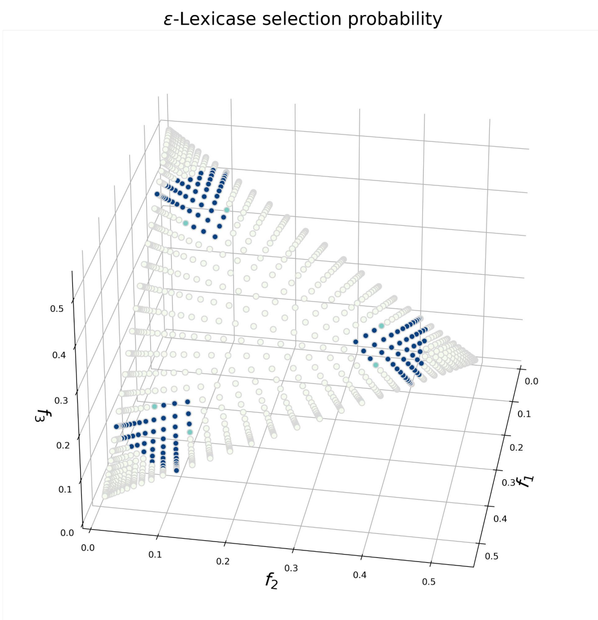
OPEN

Conservation machine learning: a case study of random forests

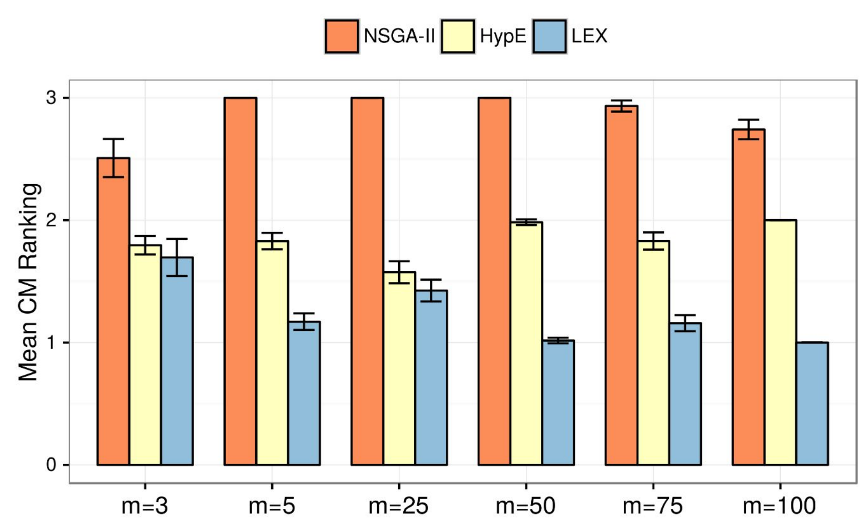
Moshe Sipper[1,2]✉ & Jason H. Moore[1]

Conservation machine learning conserves models across runs, users, and experiments—and puts them to good use. We have previously shown the merit of this idea through a small-scale preliminary experiment, involving a single dataset source, 10 datasets, and a single so-called cultivation method—used to produce the final ensemble. In this paper, focusing on classification tasks, we perform extensive experimentation with conservation random forests, involving 5 cultivation methods (including a novel one introduced herein—*lexigarden*), 6 dataset sources, and 31 datasets. We show that significant improvement can be attained by making use of models we are already in possession of anyway, and envisage the possibility of repositories of *models* (not merely datasets, solutions, or code), which could be made available to everyone, thus having conservation live up to its name, furthering the cause of data and computational science.

Many objective optimization



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



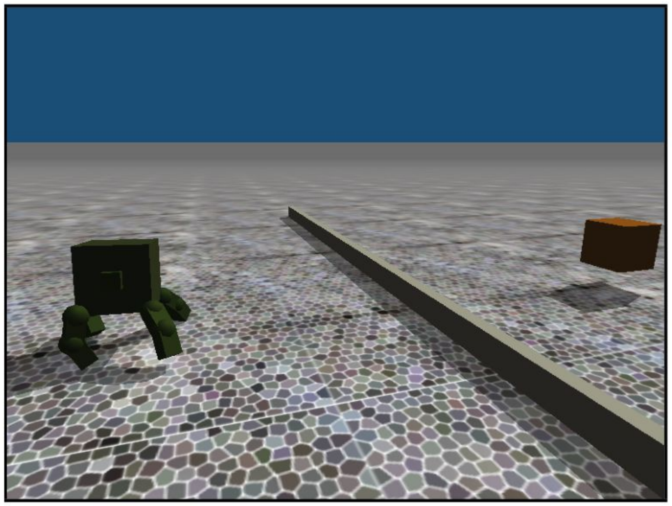
La Cava, W. & Moore, J. H. (2018) An Analysis of ϵ-Lexicase Selection for Large-Scale Many-Objective Optimization. *GECCO*

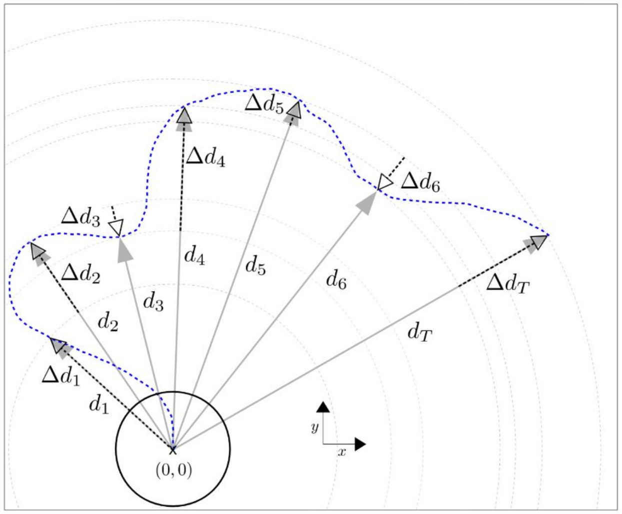
(Slide by Thomas Helmuth and Bill La Cava)

Problems from Deb, K., Thiele, L., Laumanns, M., Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds) Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing. Springer, London. https://doi.org/10.1007/1-84628-137-7_6

Evolutionary Robotics

- Quadruped animat application, lexicase selection outperformed other selection methods
- Works well for soft robotics evolution of locomotion





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

La Cava, W., & Moore, J. H. (2018). Behavioral search drivers and the role of elitism in soft robotics. *Artificial Life*, 206–213.

(Slide by Thomas Helmuth and Bill La Cava)

Operations Research Perspectives 9 (2022) 100237

Contents lists available at ScienceDirect

Operations Research Perspectives

journal homepage: www.elsevier.com/locate/orp

Optimization of subsurface models with multiple criteria using Lexicase Selection

Yifan He [a], Claus Aranha [b,*], Antony Hallam [c], Romain Chassagne [c]

[a] School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Japan
[b] Faculty of Engineering, Information and Systems, University of Tsukuba, Tsukuba, Japan
[c] Energy, Geoscience, Infrastructure and Society, Heriot-Watt University, United Kingdom

ARTICLE INFO

Keywords:
Seismic History Matching
Lexicase Selection
Multi-objective optimization
Multi-Objective Evolutionary Algorithm

ABSTRACT

Seismic History Matching (SHM) is a key problem in the geosciences community, requiring optimal parameters of a subsurface model that match the observed data from multiple in-situ measurements. Therefore, the SHM problems are usually solved with Multi-Objective Evolutionary Algorithms (MOEAs). This group of algorithms optimize multiple objectives simultaneously, considering the trade-off between objectives. However, SHM requires the solutions that are good on all objectives rather than a trade-off. In this study, we propose a Differential Evolution algorithm using Lexicase Selection to solve the SHM problems. Unlike the MOEAs, this selection method pushes the solutions to perform well on all objectives. We compared this method with two MOEAs, namely Non-dominated Sorting Genetic Algorithm II and Reference Vector-guided Evolutionary Algorithm, on two SHM problems. The results show that this method generates more solutions near the ground truth.

Volume 27, Issue 3
Fall 2019

September 01 2019

A Probabilistic and Multi-Objective Analysis of Lexicase Selection and ε-Lexicase Selection

In Special Collection: CogNet

William La Cava, Thomas Helmuth, Lee Spector, Jason H. Moore

Author and Article Information

Evolutionary Computation (2019) 27 (3): 377–402.

https://doi.org/10.1162/evco_a_00224 Article history

Cite Permissions Share Views

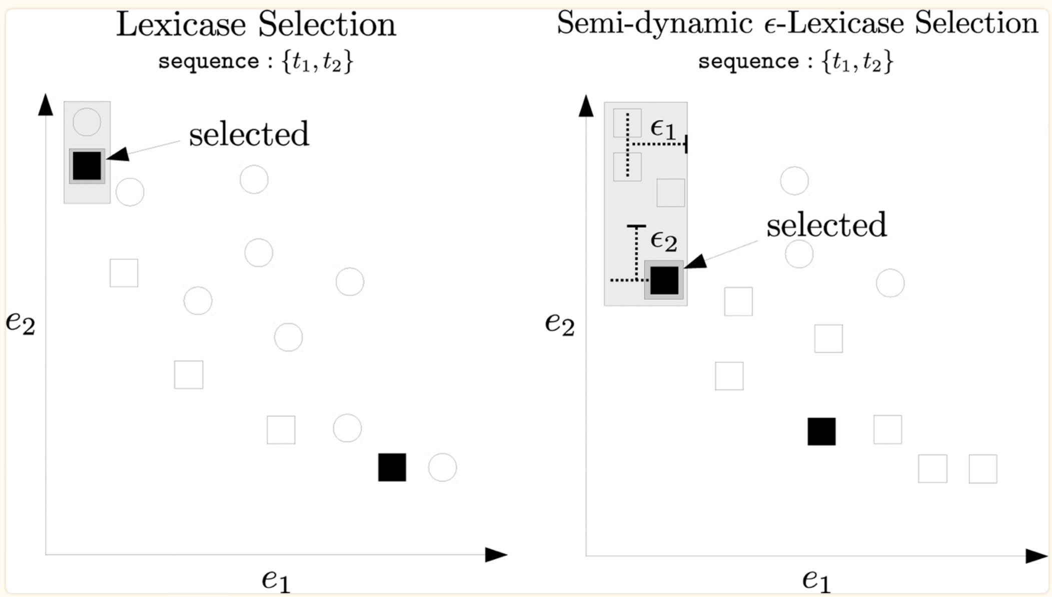


Figure 3:

An illustration of the performance of lexicase selection (left) and semi-dynamic ϵ-lexicase selection (right) in a scenario involving two cases. Each point represents and individual in the population. The squares in the left figure are individuals in the Pareto set, and the squares on the right are individuals in the ϵ-Pareto set. A selection event for case sequence $\{t_1, t_2\}$ is shown by the gray rectangles. The black points are individuals that could be selected by any case ordering.

RESEARCH-ARTICLE PUBLIC ACCESS

The Impact of Hyperselection on Lexicase Selection

Authors: Thomas Helmuth, Nicholas Freitag McPhee, Lee Spector Authors Info & Claims

GECCO '16: Proceedings of the Genetic and Evolutionary Computation Conference 2016 • July 2016 • Pages 717–724 • https://doi.org/10.1145/2908812.2908851

Published: 20 July 2016 Publication History

Table 5: Number of successful runs out of 100 for each parent selection method on each problem. For each problem, underline indicates significant improvement over tournament selection using a pairwise chi-squared test with Holm correction and 0.05 significance level. SLT and lexicase selection never have significantly different success rates.

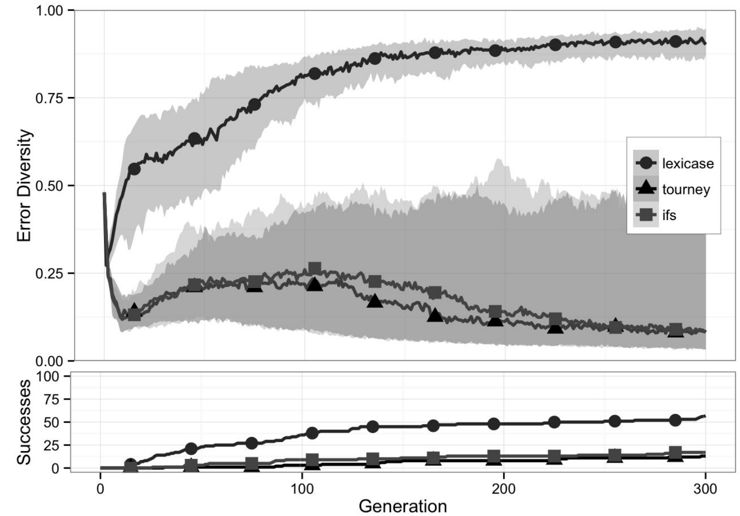
Problem	Lex	Tourn	SLT
Count Odds	8	0	5
Double Letters	6	0	4
Mirror Image	78	46	84
Negative To Zero	45	10	53
Replace Space with Newline	51	8	61
String Lengths Backwards	66	7	79
Syllables	18	1	13
Vector Average	16	14	30
X-Word Lines	8	0	4

Sampled **L**exicase **T**ournament: selects only lexicase-selectable individuals, at tournament selection frequencies

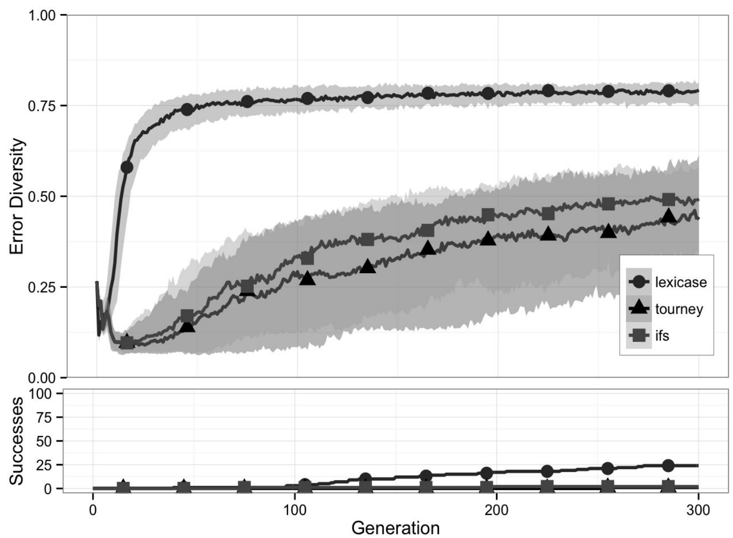
Population Diversity in GP

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

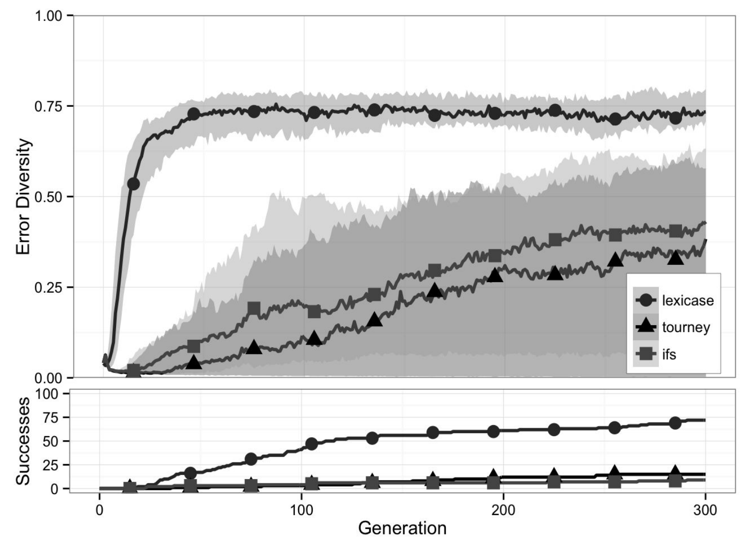
Replace Space with Newline



Syllables



Negative to Zero



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

(Slide by Thomas Helmuth and Bill La Cava)

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 total

High (bad) errors on red cases

- **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

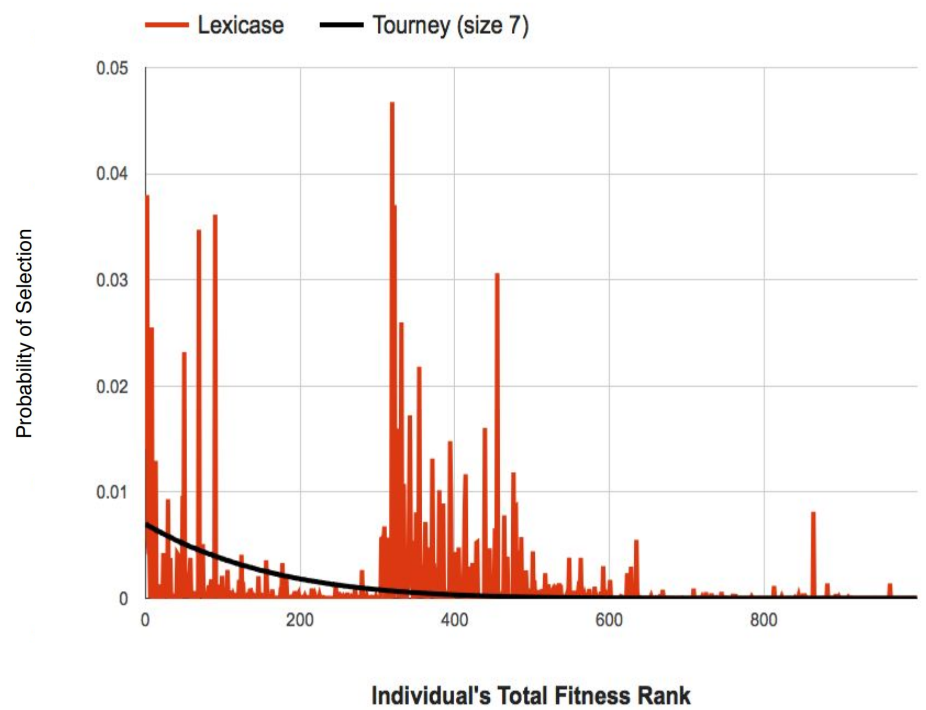
Ex: 3 2 3 3 3 3 4 2 3 = 26 total

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

(Slide by Thomas Helmuth and Bill La Cava)

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*

(Slide by Thomas Helmuth and Bill La Cava)

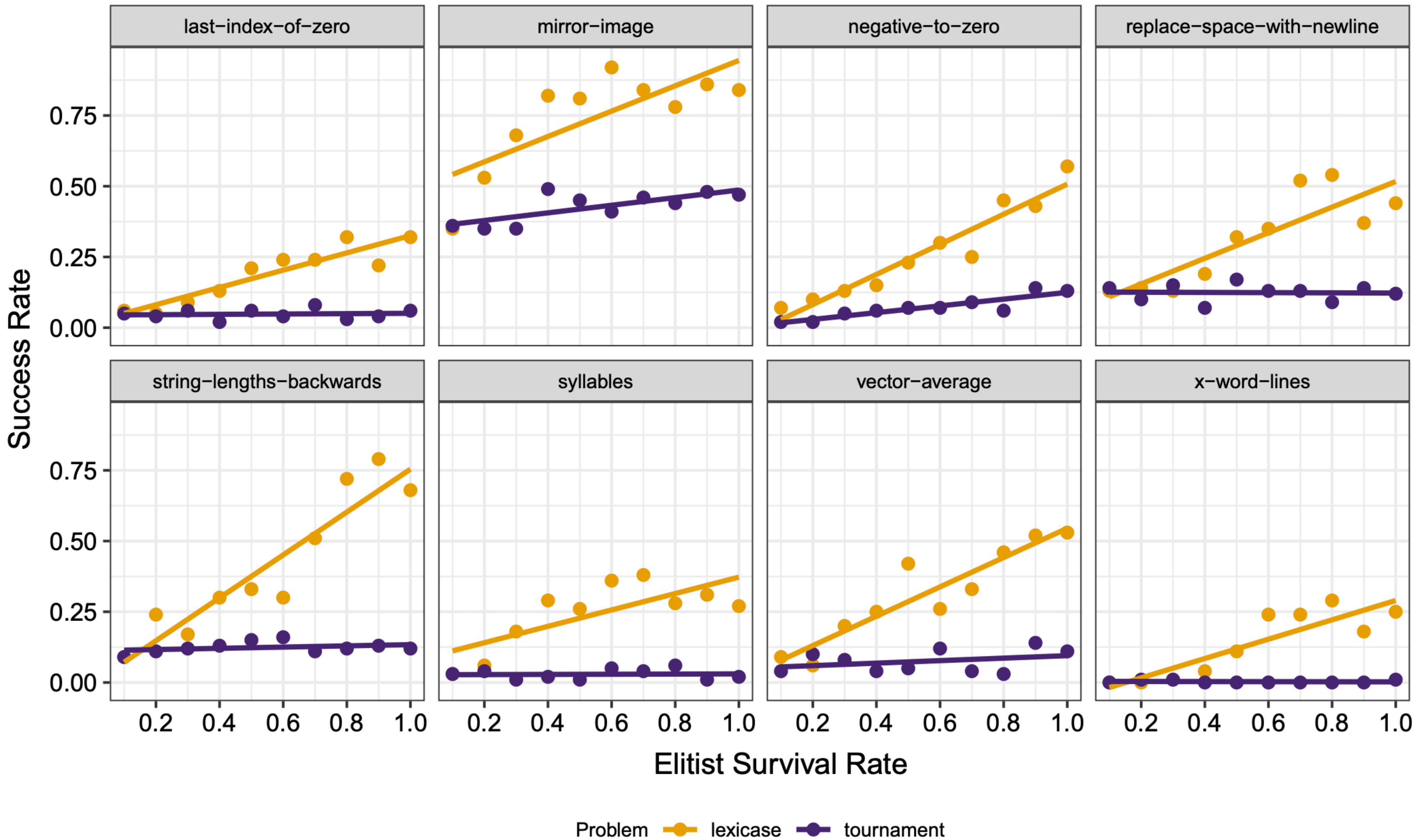


Fig. 5 The impact of elitist survival filtering on the ability of lexicase selection and tournament selection to find generalizing solution programs. As the elitist survival rate increases, lexicase selection tends to find more solutions, while tournament selection does not

Helmuth, T., Pantridge, E. & Spector, L. On the importance of specialists for lexicase selection. *Genet Program Evolvable Mach* **21**, 349–373 (2020). https://doi.org/10.1007/s10710-020-09377-2

We consistently observe lexicase selection running times that are much lower than its worst-case bound of $O(NC)$. Why?

Cava Lab
Advancing interpretable and fair digital health

Join Members Research Publications

William La Cava

Principal Investigator

- Email
- Website
- Google Scholar
- Github
- Twitter
- LinkedIn

Adventures in Lexicase: Running Time

Further Reading Helmuth, T., Lengler, J., & La Cava, W. (2022). Population Diversity Leads to Short Running Times of Lexicase Selection. *Parallel Problem Solving from Nature (PPSN)*. arXiv

https://cavalab.org/2022/08/23/lexicase-running-time.html

https://arxiv.org/abs/2204.06461

Down-sampled Lexicase 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
- ❖ *This has given the best performance on program synthesis problems of any lexicase selection variant*

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

(Slide by Thomas Helmuth and Bill La Cava)

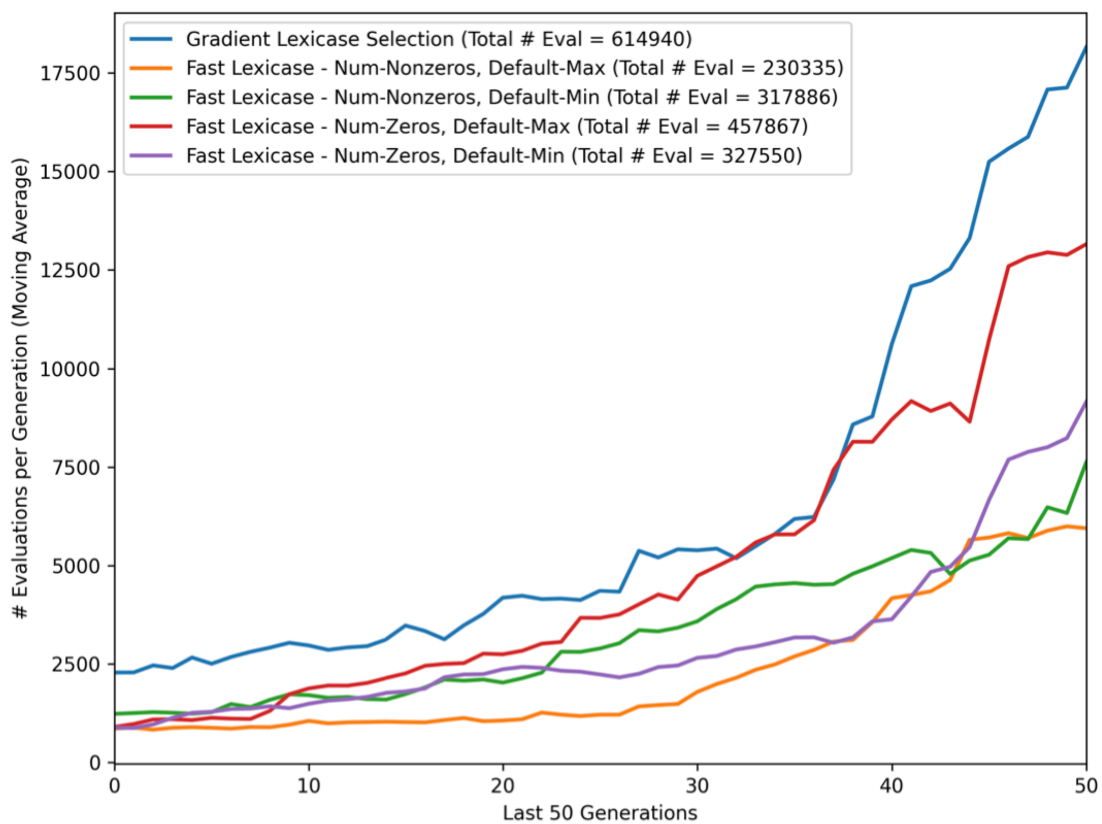
Lexicase Selection at Scale

Li Ding
University of Massachusetts Amherst
liding@umass.edu

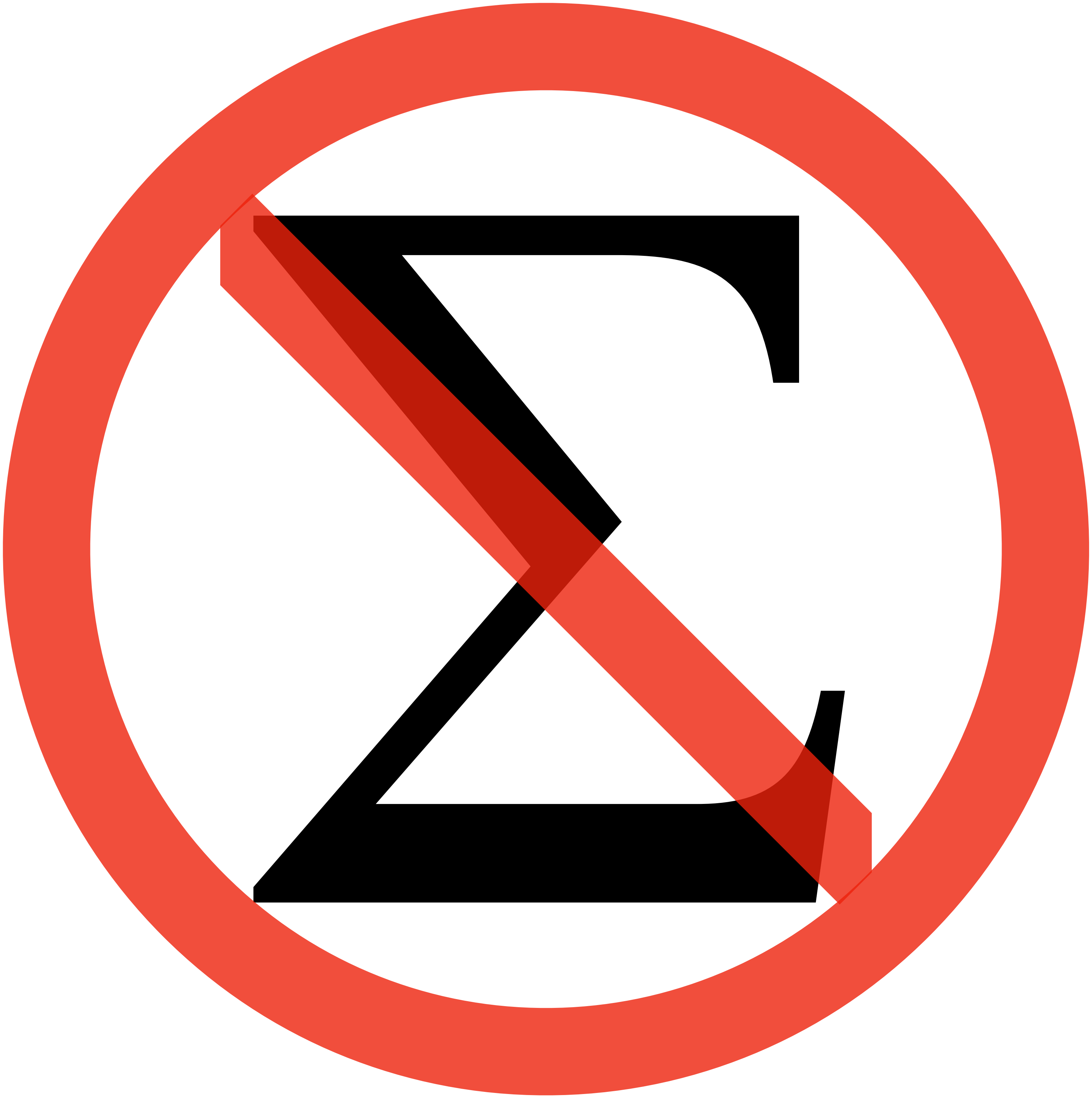
Ryan Boldi
University of Massachusetts Amherst
rbahlousbold@umass.edu

Thomas Helmuth
Hamilton College
thelmuth@hamilton.edu

Lee Spector
Amherst College
University of Massachusetts Amherst
lspector@amherst.edu



- Don't add up everything you care about
- Instead, consider each thing in its own right
- One at a time, in random order
- **Honor all the things!**
- And all the combinations of all the things



If you can't quit Σ

- Well, sometimes you really need it, and/or it works well
- But bear in mind:
  - Even when Σ measures what you ultimately want to achieve, it may not provide the best guidance for getting there
  - When repeated, lexicase selection's effects "add up" over time

Implications

- Other evolutionary algorithms
- Other AI: deep neural networks, reinforcement learning, ...
- Recommendation algorithms?
- College admissions? Hiring? Grant and article reviewing?
- Any situation in which we usually use Σ?

Thanks

Alex Lalejini, Anil Saini, Bill La Cava, Bill Tozier, Blossom Metevier, Charles Ofria, Claudiu Babin, Eddie Pantridge, Emily Dolson, Jason Moore, Li Ding, Michael Garcia, Nic McPhee, Ryan Boldi, Sneha Aenugu, Thomas Helmuth

Members of the ICI/PUSH research group at Amherst College/Hampshire College/UMass Amherst

This material is based upon work supported by the National Science Foundation under Grant No. 1617087. 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.