Tiebreaks and Diversity: Isolating Effects in Lexicase Selection

Jared M. Moore[1] and Adam Stanton[2]

[1]School of Computing and Information Systems,
Grand Valley State University, Allendale, MI, USA
[2]School of Computing and Mathematics,
Keele University, Keele, ST5 5BG, UK
a.stanton@keele.ac.uk

Abstract

A primary goal of evolutionary robotics (ER) is generalized control. That is, a robot controller should be capable of solving a variety of tasks in a domain, rather than only addressing specific instances of a task. Prior work has shown that Lexicase selection is more effective than other evolutionary algorithms for a wall crossing task domain where quadrupedal animats are evaluated on walls of varying height. In this work we expand baseline treatments in this task domain and examine specific aspects of the Lexicase selection algorithm across a variety of different parameter configurations. We identify the most effective Lexicase parameters for this task. Results indicate that Lexicase's success is potentially due to maintaining population diversity at a higher level than other algorithms explored for this domain.

Introduction

Generalized control remains a challenging problem in evolutionary robotics. An effective control strategy should be able to address many instances of a task where conditions are not precisely the same. However, evolved controllers often specialize to certain configurations, preventing them from being successful in other permutations of the same task. Algorithms must therefore address the challenge of balancing success across many configurations.

Evolutionary algorithms (EAs) offer a potential solution to these issues. By emulating the processes that drive biological evolution, solutions are gradually refined according to their performance in a given problem. In this paper, we evaluate simulated quadrupedal animats in a wall crossing task, see Figure 1. Individuals encounter wall heights ranging from very short (floor height) to tall (one half the height of the animat) during evolution. Individuals evolved in these environments are assessed on how well they generalise to the problem of **all** wall heights. Different approaches to exposing evolving populations to varied environments have been tried, in order to improve this generalisation. Previous work has shown varying degrees of success with a variety of evolutionary strategies. Stanton and Channon (2013) showed that oscillating strategies, wherein wall heights varied periodically over the course of evolution, evolved individuals capable of crossing many of the wall heights seen during evolution. However, catastrophic forgetting (McCloskey and Cohen, 1989) remains an issue, where new information about the task displaces older but still useful knowledge. This prevents individuals from addressing the general wall crossing task across all its possible configurations. The order of introduction of the wall heights impacts performance, but evolved individuals do not fully generalize to the problem.

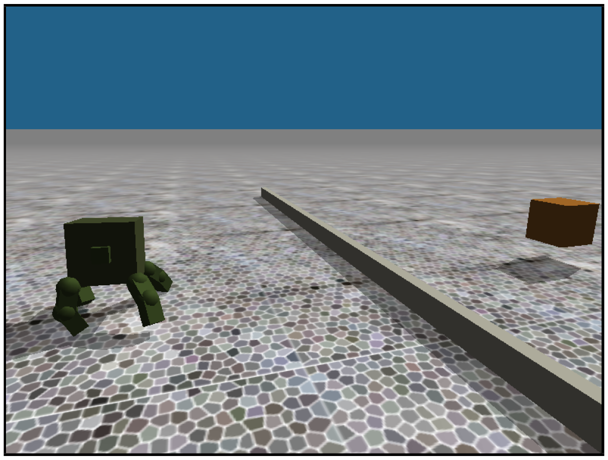

Figure 1: Neurocontrollers evolve to produce gaits that guide the quadrupedal animat (left) across a wall (center) and towards a target, represented by the box (right).

More recently, we evolved individuals for wall crossing behavior using Lexicase selection (Moore and Stanton, 2017). Lexicase selection (Helmuth et al., 2014) offers a potential solution to the task generalization problem through a multi-objective evolutionary approach. During each generation, individuals are assessed on a subset of wall heights, theoretically allowing them to specialize within a generation, but generalize across generations.

The contributions of this work are as follows. We (1) show that Lexicase selection outperforms an expanded set of baseline treatments for the wall crossing task, (2) investigate Lexicase parameter configurations isolating the most effective values for this problem, and (3) attempt to elucidate underlying mechanisms of Lexicase selection that drive the performance differences that we observe in this and prior work. We find that Lexicase's strong performance might be due to diversity maintenance, as all Lexicase treatments maintain higher population diversity over time than other algorithms in this task.

Related Work

Evolutionary robotics (ER) (Nolfi and Floreano, 2000; Doncieux et al., 2015) is the practice of applying concepts from natural evolution to the optimization of robotic systems. Individuals are compared on their fitness (performance) for a given problem. ER based approaches have been successful in many challenging problems, including optimization of morphology (Auerbach and Bongard, 2010; Cheney et al., 2013), legged gaits (Clune et al., 2009), object manipulation (Bongard, 2008), and the study of intelligence (Corucci et al., 2017). Tasks such as locomotion can be distilled into a single performance measure to evaluate an individual's fitness, but as problems become increasingly complex, multiple objectives are needed to assess performance.

Spector (2012) introduced Lexicase selection for modal problems in genetic programming (GP). Modal problems typically are considered many-objective problems, with more than five objectives needed to assess individuals. Lexicase replaces selection in a typical generational GA and considers performance in each objective individually. However, during a selection event, the objectives used to compare individuals varies. Only a subset of objectives are used every generation, but over the course of an evolutionary run, it is highly likely that all objectives will be used to assess performance. Details of the Lexicase selection algorithm and modifications in this paper are presented in the next section. Helmuth et al. (2014) demonstrated that Lexicase solves benchmark challenging problems in GP. Although originally proposed for problems in GP, Lexicase has been effective in ER, where problems can fall in the many-objective domain. Moore and McKinley (2016) found that Lexicase outperformed a traditional generational GA. Furthermore, Helmuth et al. (2016) found that Lexicase selection appears to select individuals differently than other traditional selection methods like tournament selection, potentially impacting diversity. Moore and Stanton (2017) applied Lexicase selection to a previously investigated wall crossing task (Stanton, 2017) finding that Lexicase selection evolved more effective individuals than evolutionary strategies tailored to the task. However, the reason why Lexicase selection outperformed previous strategies, and an in-depth parameter search remain unexplored. In this study, we extend the previous investigation by performing additional baseline treatments, exploring a more complete range of Lexicase algorithm parameters, and investigate specific aspects of algorithm performance elucidating why Lexicase outperforms other approaches in this task.

Methods

The simulation environment, wall crossing task, and quadruped animat in this paper are the same as used in previous work (Moore and Stanton, 2017). We describe them here for completeness, including the simulation parameters and animat configuration.

Quadruped Animat Figure 1 shows the quadrupedal animat. The animat has a cuboid torso with four two-segment legs, each on a corner. The hip is a 2 degree of freedom (DOF) joint with the ability to raise and lower while also sweeping horizontally. The knee is a 1 DOF joint. Table 1 specifies the individual parameters for the animat.

Head Dimension	$0.2 \times 0.2 \times 0.2$
Leg Segment Dimension	$0.075 \times 0.05 \times 0.05$
Head Mass	2.0
Leg Segment Mass	0.5
Hip Vertical Axis	range $[-\frac{\pi}{4}, \frac{\pi}{4}]$
Hip Horizontal Axis	range $[0, \frac{\pi}{2}]$
Knee Horizontal Axis	range $[0, \frac{\pi}{2}]$
Maximum Torque	0.125

Table 1: Physical parameters of robot. Adapted from Moore and Stanton (2017).

Joints commands are specified with a Proportional-Derivative (PD) control mechanism (Reil and Husbands, 2002) taking a target angle as input and applying torque according to Equation 1,

$$T = k_s \times (\theta_d - \theta) - k_d\dot{\theta} \tag{1}$$

where T is applied torque, k_s and k_d are spring and damper constants, θ_d is desired angle, θ the current angle and $\dot{\theta}$ the angle change from the last timestep. As in previous work, $k_s = k_d = 0.5$.

Simulation Environment The Open Dynamics Engine (ODE) (Smith, 2013) version 0.15.2 was used. ODE is a real-time rigid body physics engine handling the interaction between components of the animat, as well as interactions with the ground and obstacles. Table 2 lists the simulator parameter values.

Wall Crossing Task We employ a wall crossing task where animats must evolve both legged locomotion and the ability to cross a wall of varying heights to reach a specific target on the other side of the wall. Wall height is fixed for a simulation, ranging from 0.01 up to a maximum value of 1.0 in 0.01 increments. Fitnesses represent the distance from the target position and are negative values. A maximum fitness of 0.0 indicates that an animat reached the target. Fitness values represent behaviors according to the following: (1) reached objective ($== 0.0$), (2) crossed wall ($>= -0.2$), (3) stuck on wall ($>= -0.6$), (4) reached wall ($>= -1.0$), and (5) did not reach wall (< -1.0).

Timestep	0.02 seconds
Gravity	-1.2
Friction Model	Pyramid approximation, $\mu = 2.0$
Global ERP	0.2
Global CFM	5.0×10^{-5}
Wall Dimension	$0.05 \times 5.0 \times h$
Wall Position	$x = 1; y = 0$
Target Position	$x = 2; y = 0$
Start Location	$x = 0; y = 0$
Simulation Time	20 seconds

Table 2: ODE Simulation Environment Configuration

Controller A controller comprises a fixed, fully-connected feed-forward ANN. Inputs include environmental sensors, see Table 3, as well as oscillating signals from sinusoidal functions driving regular locomotion. The ANN has 12 hidden nodes and 12 outputs specifying desired angles for each joint at the next timestep. Hidden nodes use the *tanh* function, and outputs use the logistic function. ANN updates are tied to the simulation, with inputs propagated completely through the network at each timestep.

1	$\sin(2\pi t)$
2	$\cos(2\pi t)$
3	balance: $\arccos(\mathbf{H}[10])$
4	$(\lvert\overrightarrow{H_l}\rvert - \lvert\overrightarrow{H_r}\rvert) \div H_{width}$
5–12	hip joint angles
13–16	knee joint angles

Table 3: ANN controller inputs. $\mathbf{H}$ is the rotation matrix of the animat's head, $|\overrightarrow{H_x}|$ is the distance one side of the head to the target and H_{width} is head width.

Evolutionary Algorithm We use a generational GA with a genome specifying floating-point weights for the ANN controllers. A population consists of 50 individuals, with each randomly initialized. 20 replicates are conducted per treatment, with replicate number used to seed the random number generator. Fitness-based tournament selection is used for non-Lexicase treatments using single-point crossover and a tournament size of 5 individuals. Mutation rate is $\frac{2}{N}$, where N is the length of the genome. For Lexicase strategies, a new population is created each generation using Lexicase selection (described next) to choose parents, and creating children using the same recombination parameters. Morphological parameters are fixed. Evolution is conducted for 5,000 generations in each treatment. Fitness is defined as Euclidean distance from a target at the end of a simulation.

Lexicase Selection Algorithm 1 presents Lexicase selection with modifications originally introduced in Moore and Stanton (2017), and used for the Lexicase treatments in this study. Lexicase selection compares individuals against a number of objectives per selection event. By generating a subset of the entire objective space, individuals are compared based on their performance in the first objective (lines 3-19). When two or more individuals are tied on the current objective, tied individuals are then compared on the next objective in the random ordering. If the subset of objectives is exhausted and ties are still present, the algorithm selects an individual at random from the sample of tied individuals (line 21).

Algorithm 1 Lexicase Selection Pseudocode. Adapted from Spector (2012), Moore and Stanton (2017)

```
 1: $subset \leftarrow GetSubsetOfPopulation(population, 5)$
 2: $obj\_order \leftarrow Shuffle(fitness\_objectives)$
 3: for $obj$ in $obj\_order$ do
 4:     $r\_sub \leftarrow RankIndivInSubset(subset, obj)$
 5:     $tie\_index \leftarrow 0$
 6:     for $i$ in $1$ $to$ $length(r\_sub)$ do
 7:         if $r\_sub[i][obj] \geq thresh * r\_sub[0][obj]$ then
 8:             $tie \leftarrow True$
 9:             $tie\_index \leftarrow i$
10:         end if
11:     end for
12:     if $tie$ is $True$ then
13:         $subset \leftarrow r\_sub[0 : tie\_index]$
14:     else
15:         $tie \leftarrow False$
16:         $subset \leftarrow r\_sub[0]$
17:         $break$
18:     end if
19: end for
20: if $tie$ is $True$ then
21:     return $RandomChoiceFromPopSubset(subset)$
22: else
23:     return $subset[0]$
24: end if
```

Initially, Lexicase selection was proposed for problems in GP, but in ER, fitness values are typically real values with two individuals having identical performance only in the case of clones. We therefore adopt a modification to the Lexicase algorithm proposed in Moore and McKinley (2016). Two or more individuals are "tied" if they are within a threshold of performance on an objective compared to the best individual in that objective (lines 7-10). This also relaxes performance requirements, selecting individuals that might be completing the objective, but not in the most "optimal" way. We term the threshold, *fuzz factor*, varying the range of equivalent performance as described in Experiments and Results. This approach is likely functionally equivalent to Epsilon Lexicase selection (La Cava et al., 2016) but we have not yet compared the two side-by-side to verify identical performance.

Treatments Our objective is first to demonstrate the performance of four baselines, and then to compare our Lexicase parameter sweep to these baselines.

- *Direct* This treatment presents only the highest wall to agents at every evaluation.
- *Comprehensive* This treatment evaluates each agent in each of the 100 different wall heights at each evaluation and records the mean value for the fitness for that evaluation. While this is computationally intensive, it exposes individuals to every wall height equally.
- *Random* Agents are presented with walls of random height drawn from a uniform distribution between 0 and h_{max} at each generation.
- *Oscillating Max 100* The best treatment from Stanton and Channon (2013) with tournament selection. Wall height is changed cyclically (100 generation period) up to the max wall height.
- *Lexicase xF yE* Agents compete in Lexicase tournaments with *y* environments in each competition, using a fuzz factor of *x* to blur differences between the top individuals. *x* and *y* are varied to investigate how the fuzz factor and number of environments affects the progression of the search, including the maintenance of diversity and the need for tie-breaking.

Analysis Metrics: Tie-breaks In Lexicase, whenever a selection event results in a random choice between equally fit individuals (individuals that performed equally well on all Lexicase objectives), a tie-break event is recorded. Since the population size is 50 and each individual has two parents, a maximum of 100 tie-breaks can occur at each new generation. Here, Lexicase uses a tournament of size 5 to find each parent, so each tie-break can involve up to 5 individuals. Thus, for each generation, we present a metric for tie-breaks which is simply the total number of individuals that participated in a tie-break. In the extreme example, if every selection event in a generation resulted in a tie-break between the maximum number of individuals, this metric would record $50 \times 2 \times 5 = 500$ tie-breaks.

Analysis Metrics: Diversity To measure diversity, we calculate the locus-wise mean genotype for a population at each generation:

$$G^{locus} = \frac{1}{P}\sum_{i=1}^{P} I_i^{locus} \tag{2}$$

, where G is the mean genotype, P is the population size, and I_i^l is the lth locus of the ith individual). Then, the (locus-wise) mean squared difference of each individual's genotype from the mean individual is found, and averaged across all individuals in the population

$$D = \frac{1}{P}\sum_{i=1}^{P}\left[\frac{1}{L}\sum_{l=1}^{L}(I_i^l - G^l)^2\right] \tag{3}$$

where D is the diversity metric and L is the genotype length). This single scalar value, the mean per-locus variance in the population, is used as a proxy for population diversity.

Experiments and Results

Baseline Treatment Performance Our baseline treatments comprise the Direct, Comprehensive, Random, and Oscillating Max 100 strategies described in Methods. Figure 2 plots the performance of the best individual per replicate across the baseline treatments against the best Lexicase treatment in this study. Lexicase significantly outperforms all four treatments.

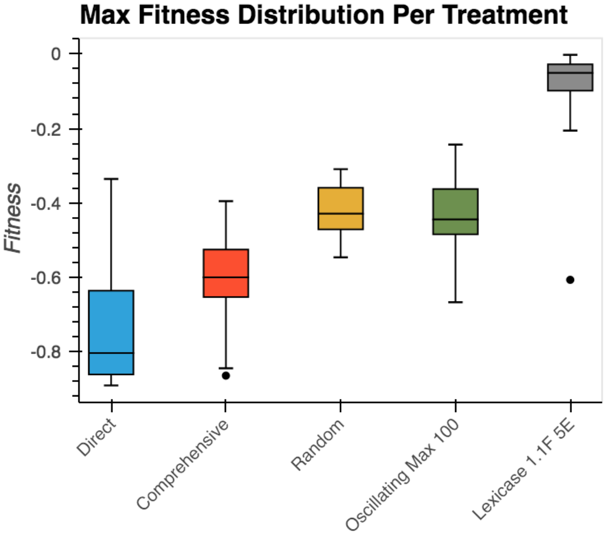

Figure 2: Fitness of the best individual per replicate across 20 replicate runs for the non-Lexicase baseline treatments.

A goal of our study is to understand how the evolved controllers generalize across the 100 different wall heights. Accordingly, Figure 3 plots the performance of each replicate's most effective individual across the 100 wall heights. Yellow shades represent high performing individuals reaching the target while purple shades indicate a failure to cross the wall. As shown in the figure, the Direct treatment results in the poorest individuals in terms of generalization. The Oscillating Max and Random treatments are roughly similar, but it appears that individuals have difficulty crossing walls in the upper half of the ranges introduced during evolution. Whereas the Lexicase 1.1 5E treatment exhibits effective wall crossing behavior for all heights in the majority of replicate runs. We next examine a variety of Lexicase parameterizations to see how different values influence performance of the algorithm.

Lexicase Parameter Search In previous work (Moore and Stanton, 2017), we examined a set of Lexicase pa-

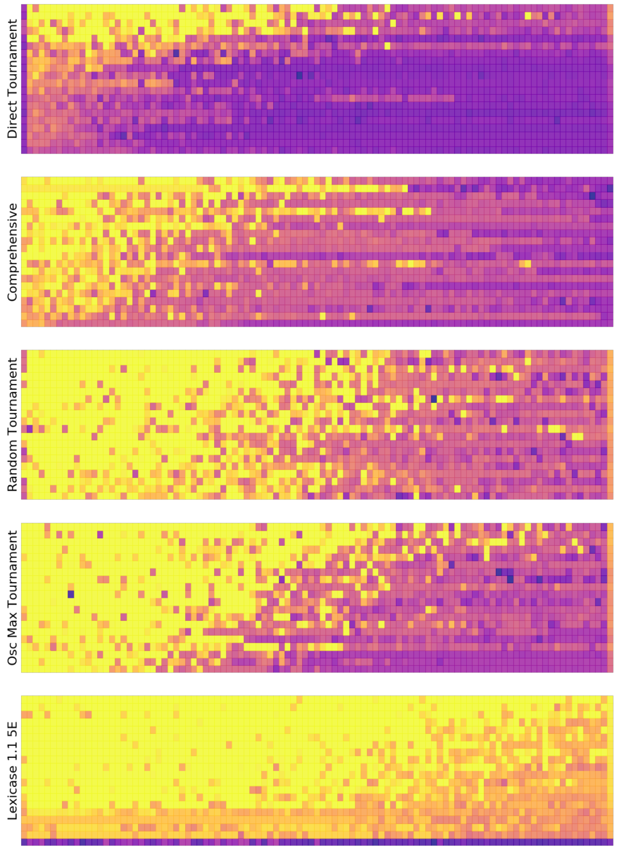

Figure 3: Performance of the best individual per replicate across all 100 environments post evolution. Darker shades indicate poor performance, lighter shades indicate success. Wall heights increase from left to right.

rameters encompassing (1) fuzz factors of 1.05, 1.10, and 1.15, (2) environments of 1, 2, 5, 10, and 20, and (3) generations both set to balance the number of evaluations against the number of environments per generation and strictly 5,000 generation runs for fuzz factors of 1.10. In that work, we initially sought to maintain the number of evaluations in each Lexicase configuration by ensuring the number of environments evaluated in a generation and the total number of generations per replicate run multiplied to 250,000. Here, we relax that requirement and instead evolve for 5,000 generations as some of the higher environment replicates appeared to suffer from substantially reduced evolutionary time, only 250 generations in the 20 environment configuration versus 2,500 for 2 environments. Furthermore, we add two new fuzz factors, 1.00 and 1.50. The 1.00 fuzz factor demonstrates what happens when two individuals are only considered equal when they have the same performance, a situation highly unlikely to arise in this real valued fitness task. 1.50 fuzz factor effectively reduces the selective pressure for fit individuals as those that are only half as good as the best individual in a given selection event are considered "equal" in performance for an environment being evaluated.

Figure 4 plots the performance of the best individual per replicate across the different fuzz factors sampling the number of environments at 1, 2, 5, 10, and 20. Binning the treatments by fuzz factor shows a trend that emerges as the number of environments considered during selection increases. Maximum fitness typically increases from 1 environment to 5 environments. Performance then generally tapers off, but not significantly, when comparing treatments pairwise using a t-test, as the number of environments increases up to 20. However, performance in the 20 environment case remains higher than the 1 environment treatments. The 1.10 fuzz factor results in the most effective evolved individuals. Although fitness is similar compared to other fuzz factors, the distribution of fitnesses among 5, 10, and 20 environments has less variance, with replicates in general having high fitnesses. Perhaps somewhat surprisingly, the 1.50 fuzz factor has higher performance than the 1.0 fuzz factor. This occurs even though individuals that are only half as effective compared to the best individual during a selection event in a given environment are considered "tied".

After sweeping the fuzz factors, we next evaluate the number of environments for the 1.10 fuzz factor. We have previously used 1, 2, 5, 10 and 20 environments to assess the influence of the number of environments on Lexicase performance. Here, we evaluate all possible number of environments from 1 to 20. Figure 5 plots the fitness of the best individual across replicates for treatments ranging from 1 to 20 environments. Fitnesses increase as the number of environments used per selection event increases from 1 to 5. Performance then slightly decreases, but not significantly, as the number of environments increases up to 20. Only the 1 environment treatment is significantly lower than all others using a t-test. Still, it appears that 5 environments is the optimal number of environments to use with Lexicase selection in this task.

Tiebreaks and Diversity Given Lexicase's performance compared to the other algorithms, we next examine population diversity and the number of tiebreaks during Lexicase selection. Figure 6 plots the average population diversity across replicates. The sharp drop in diversity across treatments is due to the initially random populations rapidly shedding some genes during the initial increase in fitness. Oscillating Max 100 has the lowest overall population diversity along with the second lowest average maximum performance. Oscillating Max 100, Direct, and Random all have lower diversity than the best Lexicase treatment from each of the fuzz factor groups. Effective Lexicase treatments all fall into the same range of diversity between 6.5 and 8.0.

Figure 7 plots the average maximum fitness versus average population diversity in the final generation population. The three lowest diversities (Direct, Oscillating Max 100, and Random) also have the lowest performance. The most effective treatments have population diversities in the

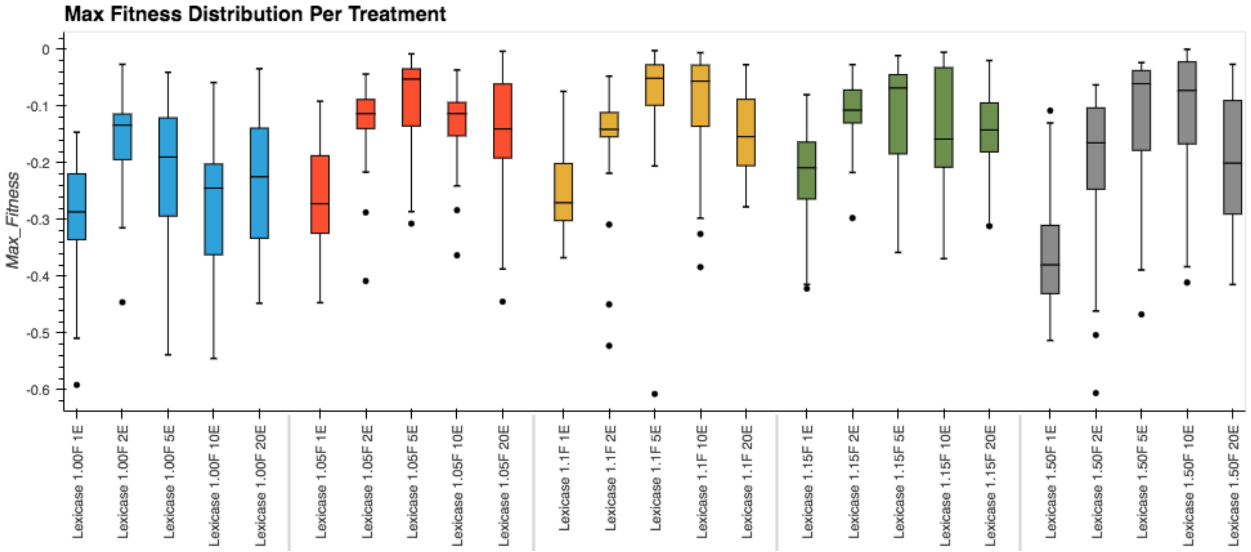

Figure 4: Best individual per replicate across various Lexicase parameter configurations.

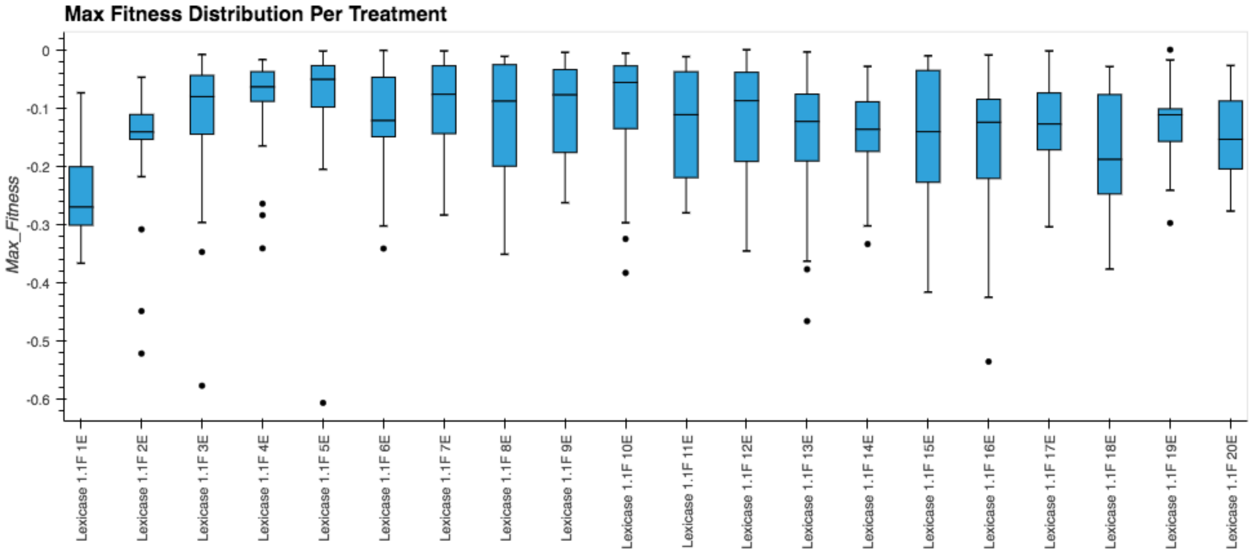

Figure 5: Best individual per replicate across the Lexicase 1.10 fuzz factor treatments.

previously indicated range, with the best treatments falling roughly at 7.0.

For Lexicase treatments within a specific fuzz factor, we observed diversity steadily declining from 1 to 5 environments. For 5 environments and up, diversity stabilizes. Figure 8 plots the population diversity across replicates for Lexicase 1.1 fuzz factor treatments. A low number of environments to compare individuals on likely results in a higher number of tiebreaks which are random selection events on individuals that are tied in the selected environments.

Lexicase selection consistently maintains higher population diversity as compared to the other treatments. One possible explanation for this is the tiebreak operation performed when two or more individuals have been considered "equals" in all environments for that selection event. This random selection might help to maintain population diversity by propagating an individual that is not objectively better. Figure 9 plots the average number of tiebreaks over time for five of the Lexicase 1.1 fuzz factor treatments. Intuitively, tiebreaks would decline as the number of environments increase due to the additional environments being considered for Lexicase. However, that is not the case. 1 environment results in many tiebreaks, with the number declining until 5 environments and then stabilizing beyond that.

Another potential explanation of the reduction in tiebreaks and stabilization in diversity is that, as the number of environments used in the Lexicase selection increase,

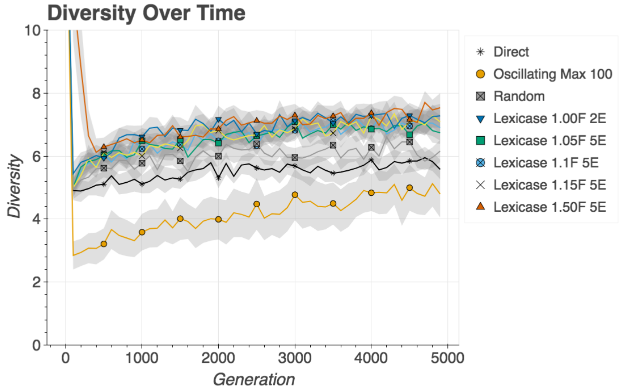

Figure 6: Average population diversity over time across treatments in this study. Shaded areas represent 95% confidence intervals. The most effective treatment per Lexicase fuzz factor value has been chosen as the representative for each set of treatments.

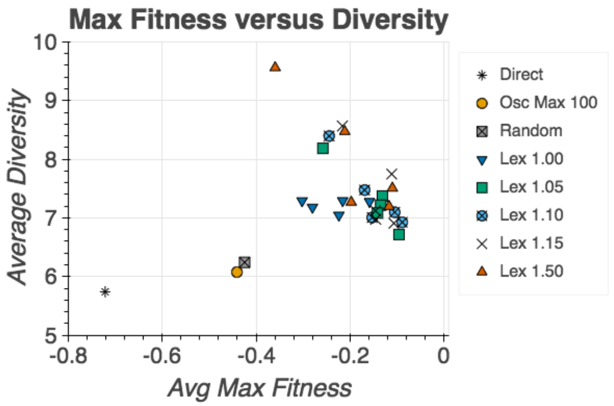

Figure 7: Final generation average maximum fitness versus average population diversity across replicates. Lexicase treatments have been grouped by fuzz factor for clarity.

there is a higher likelihood that similar environments are used during the selection process. That is, in the 20 environment treatments, two of the objectives may be wall heights of 0.25 and 0.26, which are quite similar in terms of wall height. Whereas, in the 2 environment case, there is a higher likelihood of substantially different wall heights being selected for the two objectives. We plan to investigate this hypothesis further in ongoing work.

Conclusions and Future Work

In this paper, we expand our investigation of Lexicase selection for a wall crossing task. As in prior work, results show that Lexicase selection is the most effective evolutionary strategy examined even compared to an expanded set of baseline treatments. There appears to be a sweet spot in parameters centered around 5 environments and a fuzz factor of 1.10. Overall, it appears that the performance of Lexicase is driven by a combination of diversity preservation compared to the baseline treatments as well as the number of objectives considered during evolution. Across the various fuzz factors presented here, we observe that the number of tiebreaks decline as the number of environments considered during selection increases, however, they remain present and increase over evolutionary time. Tiebreaks appear to be beneficial but do decrease as the number of objectives considered increases. Diversity also appears to be a key driver of performance, with Lexicase selection treatments maintaining higher population diversity than comparable treatments investigated.

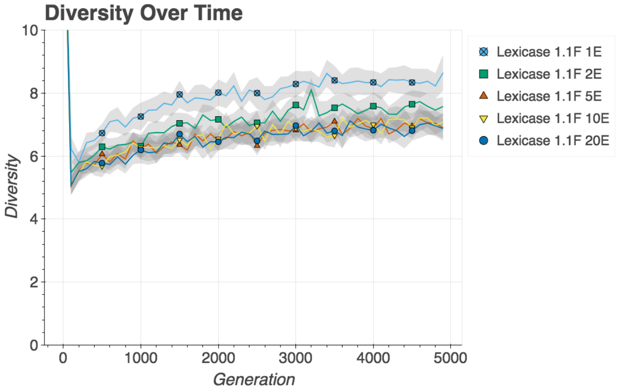

Figure 8: Average population diversity over time across replicates for Lexicase 1.1 fuzz factor treatments. Shaded areas represent 95% confidence intervals.

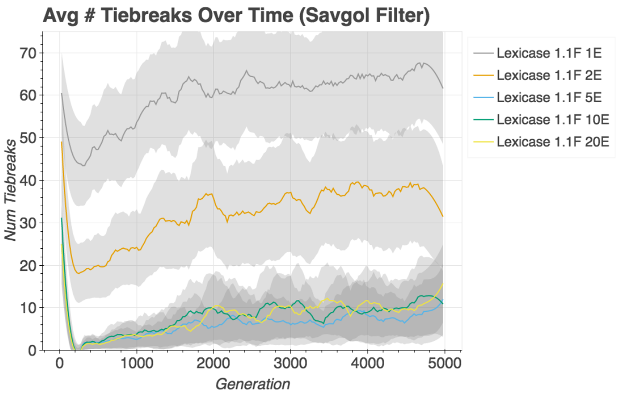

Figure 9: Average number of tiebreaks per replicate over time for Lexicase 1.1 fuzz factor treatments. Savgol smoothing filter applied with window of 15 and polynomial order of 3. Artifacts at beginning and end are due to the filtering. Shaded areas represent 95% confidence intervals.

Future work will investigate Lexicase selection in other ER tasks, alternate tiebreaking strategies, as well as continuing to examine the dynamics underlying the algorithm. The task examined here is a series of related problems, falling under the umbrella of wall crossing. Ongoing experiments apply Lexicase to other ER tasks where objectives are not necessarily sub-problems of a general instance. Furthermore, the tiebreak strategy here was random. Other strategies are possible, weighting performance or diversity, and could impact the performance of the algorithm on a given task. Additional experiments will attempt to isolate other factors such as the selection of objectives and their relatedness to see if they impact tiebreaks, diversity, and performance.

Acknowledgements

The authors would like to thank Keele University and Grand Valley State University for supporting this research.

References

Auerbach, J. E. and Bongard, J. C. (2010). Dynamic resolution in the co-evolution of morphology and control. In *Proceedings of the Twelfth International Conference on Artificial Life*, pages 451–458, Odense, Denmark.

Bongard, J. C. (2008). Behavior chaining: Incremental behavioral integration for evolutionary robotics. In *Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, pages 64–71, Winchester, United Kingdom.

Cheney, N., MacCurdy, R., Clune, J., and Lipson, H. (2013). Unshackling evolution: Evolving soft robots with multiple materials and a powerful generative encoding. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation*, pages 167–174, Amsterdam, The Netherlands. ACM.

Clune, J., Beckmann, B. E., Ofria, C., and Pennock, R. T. (2009). Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 2764–2771, Trondheim, Norway.

Corucci, F., Cheney, N., Kriegman, S., Bongard, J., and Laschi, C. (2017). Evolutionary developmental soft robotics as a framework to study intelligence and adaptive behavior in animals and plants. *Frontiers in Robotics and AI*, 4:34.

Doncieux, S., Bredeche, N., Mouret, J.-B., and Eiben, A. G. (2015). Evolutionary robotics: What, why, and where to. *Frontiers in Robotics and AI*, 2(4).

Helmuth, T., McPhee, N. F., and Spector, L. (2016). The impact of hyperselection on lexicase selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '16, pages 717–724, Denver, Colorado, USA. ACM.

Helmuth, T., Spector, L., and Matheson, J. (2014). Solving uncompromising problems with Lexicase selection. *IEEE Transactions on Evolutionary Computation*, PP(99):1–1.

La Cava, W., Spector, L., and Danai, K. (2016). Epsilon-lexicase selection for regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 741–748, Denver, Colorado, USA. ACM.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165.

Moore, J. M. and McKinley, P. K. (2016). *A Comparison of Multiobjective Algorithms in Evolving Quadrupedal Gaits*, pages 157–169. Springer International Publishing, Aberystwyth, UK.

Moore, J. M. and Stanton, A. (2017). Lexicase selection outperforms previous strategies for incremental evolution of virtual creature controllers. In *Proceedings of the 14th European Conference on Artificial Life*, pages 290–297, Lyon, France. MIT Press.

Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. The MIT Press.

Reil, T. and Husbands, P. (2002). Evolution of central pattern generators for bipedal walking in a real-time physics environment. *IEEE Transactions on Evolutionary Computation*, 6(2):159–168.

Smith, R. (2013). Open Dynamics Engine, http://www.ode.org/.

Spector, L. (2012). Assessment of problem modality by differential performance of Lexicase selection in genetic programming: A preliminary report. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pages 401–408, Philadelphia, Pennsylvania, USA. ACM.

Stanton, A. and Channon, A. (2013). Heterogeneous complexification strategies robustly outperform homogeneous strategies for incremental evolution. In *Proceedings of the 12th European Conference on Artificial Life*, pages 973–980, Taormina, Italy.

Stanton, A. J. (2017). *Simultaneous incremental neuroevolution of motor control, navigation and object manipulation in 3D virtual creatures*. PhD thesis, Keele University.