Fitness Agnostic Adaptive Sampling Lexicase Selection

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

[1]School of Computing,
Grand Valley State University, Allendale, MI, USA
[2]School of Informatics and Digital Engineering,
Aston University, Birmingham, UK
jared.moore@gvsu.edu, adam@machinemachines.org

Abstract

Lexicase selection is an effective many-objective evolutionary algorithm across many problem domains. Lexicase can be computationally expensive, especially in areas like evolutionary robotics where individual objectives might require their own physics simulation. Improving the efficiency of Lexicase selection can reduce the total number of evaluations thereby lowering computational overhead. Here, we introduce a fitness agnostic adaptive objective sampling algorithm using the filtering efficacy of objectives to adjust their frequency of occurrence as a selector. In a set of binary genome maximization tasks modeled to emulate evolutionary robotics situations, we show that performance can be maintained while computational efficiency increases as compared to ϵ-Lexicase.

Introduction

Evolutionary many-objective optimization approaches apply selection operators across a number of objectives to evolve robust solutions. Over many generations, individuals evolve within a population excelling on one or more objectives. As individuals gain proficiency within the population, selection, recombination, and mutation further refine individuals to solve challenging tasks. Lexicase selection (Spector, 2012) has evolved solutions in many-objective optimization problems across a variety of problem domains (Helmuth et al., 2014; He et al., 2022). Applied to evolutionary robotics (ER) (Floreano et al., 2008; Doncieux et al., 2015), Lexicase can effectively leverage modal objectives (e.g. assess a quadrupedal animat on distance traveled, efficiency, leg behaviors) (Moore and Clark, 2021) or to variants of a similar objective (e.g. wall-crossing across different wall heights) (Moore and Stanton, 2017). Lexicase can be compute intensive, especially in the ER domain, where multiple physics simulations might be required per individual to compute fitnesses across objectives.

Despite computational overhead, Lexicase selection has been shown to be effective in evolving generalized behavior. First, effective locomotion for variants of a single task, wall-crossing (Moore and Stanton, 2019, 2020) with 100 unique wall heights, and then to evolving two separate behaviors, wall-crossing and turn-and-seek (Moore and Stanton, 2021). For the wall crossing and turn-and-seek animats,

each task comprises 100 unique environments resulting in a 200 objective optimization problem. Evolved individuals exhibit generalization across the unique configurations of a single task, and multi-modal behavior solving wall crossing and turn-and-seek with a single artificial neural network controller. Although the two behaviors appear distinct, locomotion and navigation to a target are required for both tasks. Within a task, each unique environmental configuration follows the general goals of the larger meta task (e.g. wall crossing or turn-and-seek). For wall-crossing, wall heights increase incrementally, while in turn-and-seek the objectives progressively move the navigation target from an individual's left, to front, and finally to the right. The relationships between individual objectives likely mean that there are general behaviors across objectives. Finding and exploiting the relationships between objectives might lead to a more efficient search of the problem space, freeing compute resources to add in additional tasks or lengthening the evolutionary time allotted to a run. For ER tasks, algorithmic efficiency gains can significantly reduce the computational run time as physics simulations are often the most time intensive portion of the process. This is especially the case where each objective requires its own physics simulation to generate a fitness value as-in (Moore and Stanton, 2018, 2019, 2020; Stanton and Moore, 2022) where the 100 or 200 objective problems make it computationally infeasible to evaluate every individual in the population across all objectives every generation.

In this study, we propose a fitness agnostic mechanism to leverage hidden relationships between objectives augmenting the Lexicase selection operator. Our adaptive sampling strategy focuses on gathering information about objectives by recording data from the selection step complementing weighted shuffling originally introduced in Troise and Helmuth (2018) where the entire population's performance is used to inform objective weighting. Specifically, adaptive sampling maintains a record of how effective individual objectives are at separating individuals during comparison in selection. Objectives that filter out more individuals have their chance of being used as a selector increased, while

those that are poor filters are decreased. As information is gathered about selection events, the algorithm changes the probability of how frequently an objective is used in the selection process. We hypothesize that this adaptive approach will find objectives that are good selectors while using poor selectors infrequently. This would lead to reductions in the number of evaluations used during a run which subsequently reduces the computational overhead of simulating objectives in domains like ER.

Three adaptive sampling strategies are compared to ϵ-Lexicase on a binary genome problem emulating the relationships that arise between objectives in the ER experiment in (Moore and Stanton, 2018). Results show that adaptive sampling based on filtering efficacy during selection can increase performance while also decreasing the number of objectives needed for comparison. Adaptive strategies do not always exceed fitness or efficiency of ϵ-Lexicase selection but performance *only* degrades to be comparable to ϵ-Lexicase. Finally, we show that the most effective adaptive sampling strategy can be tuned to improve computational efficiency while maintaining overall fitness performance.

Background and Related Work

Lexicase selection is a many-objective selection operator that has found success in a variety of domains from genetic programming (Spector, 2012), geosciences (He et al., 2022) and ER (Moore and Stanton, 2017). Lexicase is effective at maintaining population diversity and specialists within a population which contribute to its success in many-objective search problems (Helmuth et al., 2016; Dolson and Ofria, 2018). ϵ-Lexicase (La Cava et al., 2016) is especially important in the ER domain as small performance differences on metrics such as distance traveled, or locomotive efficiency, don't produce meaningful differences in the observed behaviors of animats. Adding a "close enough" factor allows two individuals within $\epsilon * max\_fitness$ in the pool of candidates to move on to be compared on other objectives.

For domains such as ER, Lexicase selection can be compute intensive as physics simulations are needed for evaluating objectives. If every objective needs to be evaluated at each generation, and objectives require individual simulations, computational overhead grows rapidly with any added objectives. A number of techniques have been explored addressing efficiency in Lexicase including downsampling (Moore and Stanton, 2017, 2018, 2019; Hernandez et al., 2019), where the number of objectives considered during selection is limited with ties resolved through a random selection, or cohort Lexicase selection (Hernandez et al., 2019), which is similar to an island model genetic algorithm. Both approaches have yielded computational efficiencies allowing for longer evolutionary searches given the same compute budget (Ferguson et al., 2020). These techniques rely on random sampling of the objectives and evolutionary timelines to evolve effective individuals.

It is possible that information about each objective's effectiveness as a selector could be leveraged to further enhance the computational efficiency of Lexicase selection. Troise and Helmuth (2018) proposed using prior knowledge about the difficulty of objectives to influence the ordering of objectives during Lexicase selection. Although performance of the weighted shuffling did not exceed that of traditional Lexicase with uniform shuffling, it has the potential to reduce the number of individual evaluations needed during an evolutionary run as shown in Ding et al. (2022). There, weighted shuffling was applied with the weights of each objective modified based on the success of subsets of the population on an objective. In this work, we further examine weighted shuffling by exploring a fitness agnostic measure adjusting the weights of objectives during selection without requiring additional evaluations of the population.

Fitness Agnostic Adaptive Sampling Lexicase Selection

Fitness Agnostic Adaptive Sampling (FAAS) ϵ-Lexicase Selection develops a list of weights to apply to objectives while the evolutionary process is ongoing. Under downsampled ϵ-Lexicase selection, a set of objectives are drawn randomly and used to evaluate a set of individuals during an individual selection event. While individuals undergo selection, the set of individuals are compared on their performance in the first objective. Only those that are within $\epsilon * max\_fitness$ of the top performing individual advance to the next objective under consideration. If at any point there only remains one individual, that individual is returned as the selected individual and the evaluation process stops. Together with downsampling, FAAS ϵ-Lexicase selection can be effective and computationally tractable in the ER domain. The information to facilitate adaptive sampling can be generated alongside the normal selection process by evaluating an objective's ability to separate individuals from the selection pool. An objective's "filtering effectiveness" can be determined by calculating the ratio of how many individuals were filtered out divided by how many individuals were under consideration by the objective. A list of filtering ratios, each associated with an objective, can be maintained and updated over the course of a run and used to influence which objectives appear in selection. The following subsections further describe FAAS and variations considered in this study.

Objective Selection

A key aspect of making Lexicase selection computationally efficient is to simulate only what is needed to evaluate individuals during selection. Downsampling helps reduce that cost by limiting the number of objectives considered during an individual selection event but it remains a stochastic sampling from the entire objective space. Typically, objectives are selected randomly with a uniform distribution from the pool of objectives. Sampling objectives for the evaluation

phase with FAAS takes into account the filtering effectiveness of an objective. Initially, all objectives have an equal probability of being used for selection. The probabilities of objectives change as information is gained about the population and objectives. Each objective is assigned a chance of selection relative to its filtering effectiveness ratio as a percentage of the total ratios across all objectives. Better filters have a higher proportional representation in a roulette wheel scheme while those that have been poor filters are allotted a smaller portion. By not fully eliminating poor filtering objectives, they have a chance to increase their probability of being used for selection over time. This could occur as an objective may transition from a poor filter to an effective filter as populations evolve and higher fitness individuals integrate new behaviors. Initially challenging environments might have a low filtering power early in evolution as the population may not yet have evolved the behaviors required to be successful. As a population gains proficiency, objectives may become better filters while initially good filters might be less effective due to the population broadly having competency for that specific objective.

Objective Ratio Assignment Strategies

The weight of an objective is calculated as an offset from a baseline. Weights are initialized to a fixed value for equal initial consideration of all objectives. They are adjusted based on feedback gathered from the selection of individuals during Lexicase. The number of individuals filtered by and the number of individuals considered by a given objective provide a filtering effectiveness value. How this value is used to adjust ratios dictates the behavior of adaptive sampling. We next describe three strategies to integrate new information about filtering effectiveness.

FAAS ϵ-Lexicase Base FAAS ϵ-Lexicase, herein referred to as *Adaptive*, maintains a running tally of how many individuals enter a selection event and how many individuals are filtered. When an objective is used for selection, its ratio is updated based on the following formula:

$$ratio_i = ratio_i + \lambda((num\_fil_i/num\_in_i) - ratio_i) \quad (1)$$

where $ratio_i$ is the ratio for objective i, num_fil_i is the number of individuals filtered, num_in_i is the total number of individuals evaluated by the objective, and λ is a constant update rate (1.0 in this study). One possible side effect of this method is that a given objective's ratio could fall to zero, or near zero, eliminating an objective from being considered. Accordingly, *Adaptive* enforces a minimum ratio of 0.05.

Sliding Window FAAS ϵ-Lexicase Ratios based on the entire historical record for an objective might suppress an objective's chance of being selected in later generations where it might actually be a more effective filter. To address this possibility, we introduce a sliding window wherein the last 800 objective evaluations are used to determine the ratios of the objectives. In *Slide*, an objective's ratio is calculated during a run from the data points in the window tied to that objective. If an objective has not been used within the sliding window, its ratio is set to the mean filtering ratio of all objectives. As evolution progresses, older data that could influence objective ratios is discarded, while newer information informs the current state of the evolutionary search. Adjusting the size of the sliding window influences how much new information steers the selection of objectives for evaluation.

Restorative FAAS ϵ-Lexicase An alternative approach to a sliding window is to implement a restorative adjustment for low performing objectives. In *Restore*, objective ratios are adjusted as in *Adaptive* FAAS ϵ-Lexicase but no minimum ratio is enforced. Instead, when objectives are sampled for a new Lexicase selection, objectives that are less than the mean ratio across objectives are adjusted upwards at 10% of their difference from the current ratio to the mean of all objectives. Over time, low performing objective ratios will increase, enhancing their chance of getting selected again.

Binary Genome Tests

The three FAAS ϵ-Lexicase variants are compared to ϵ-Lexicase selection on a series of four binary genome maximization tests. Genomes comprise 500 bits, with objectives being composed of subsets of the total genome space. Each objective attempts to maximize the sum of its constituent bits. For reporting purposes, we report a metric, All_Fit, which is the sum of the bits across all loci although no objective is evolved to maximize the entire genome specifically in three of the four tests. Figure 1 shows the configuration of the four tests conducted in this study. We modified the one-max problem to mimic the relationships between objectives that we hypothesize exist in ER tasks like wall-cross or turn-and-seek. Our goal is to evaluate the performance of FAAS ϵ-Lexicase on a domain where the relationships between objectives are explicitly defined and intuitive. If FAAS ϵ-Lexicase is effective, we hypothesize that the computational cost will be lower than ϵ-Lexicase. The following paragraphs describe the 4 tests that we conduct.

Test 1: Non-Overlapping Objectives *T1-No Overlap* is a 500 bit binary problem with 50 objectives. Each objective spans 10 genes. No genes are shared between objectives. This test establishes a baseline of performance between ϵ-Lexicase and FAAS ϵ-Lexicase. In this test, we hypothesize that FAAS ϵ-Lexicase will not significantly favor any individual objective as there are no shared genetic regions between objectives.
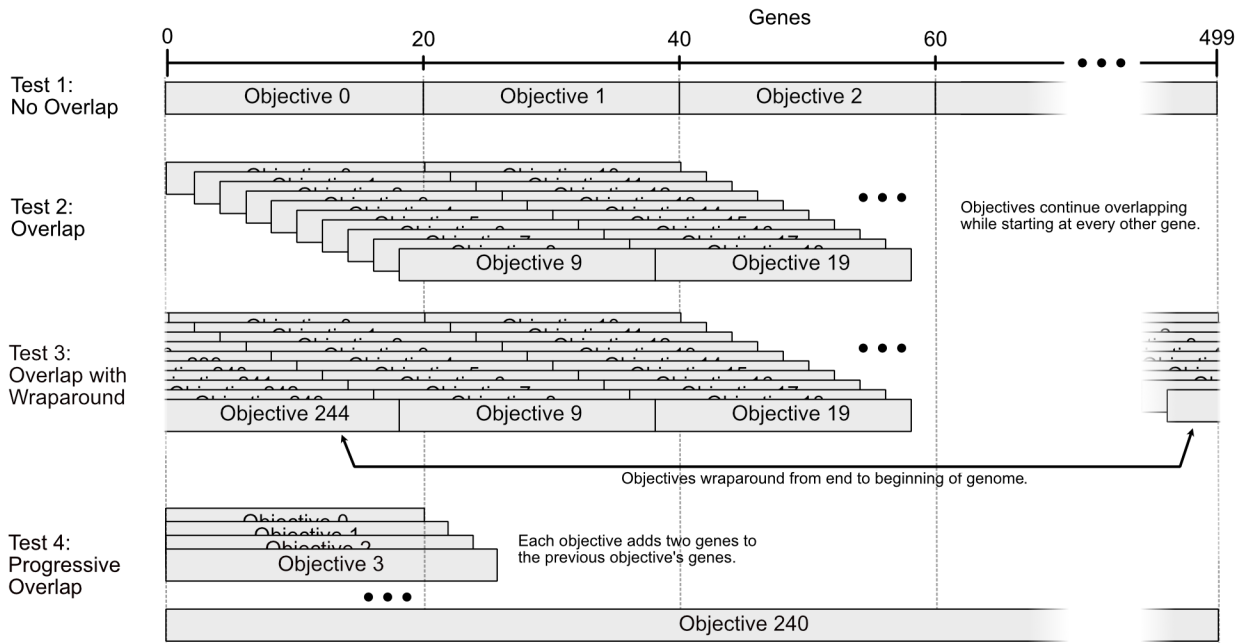
Figure 1: The four tests used in this study. *T1-No Overlap* has no overlapping objectives. *T2-Overlap* has overlapping objectives but ends are not overlapped. *T3-Wraparound* has all objectives containing equal overlap. *T4-Progressive* progressively increases the number of genes in an objective with higher objectives fully encompassing the genes from lower objectives.

Test 2: Overlapping Objectives without Wraparound
T2-Overlap comprises 241 20 gene objectives. Objective 0 starts at gene 0 ending at gene 19. Objective 1 starts at gene 2 ending at gene 21. Subsequent objectives follow this pattern starting at every even numbered gene up to gene 480 creating significant gene overlap among adjacent objectives. All of an objective's genes are covered by other objectives except for portions of the first and last objective. We hypothesize that the overlap will lead to FAAS ϵ-Lexicase exhibiting periodic spikes in objective usage relative to adjacent objectives which will be less frequently sampled given the shared evolutionary pressures.

Test 3: Overlapping Objectives with Wraparound *T3-Wraparound* follows the structure of *T2-Overlap*, but with wraparound objectives creating toroidal coverage of the genome space. This wraparound means that the first and last objectives do not have less shared genetic material than the other objectives. Subsequently, we still expect to see the periodic spikes, but without higher usage of the first and last objectives as might be expected in test two.

Test 4: Progressive Objectives *T4-Progressive* focuses on progressive overlap of objectives. The first objective covers genes 0 to 19 with each subsequent objective adding two more genes (e.g. Objective 0: 0-19, Objective 1: 0-21, Objective 2: 0-23 ...). A total of 241 objectives arise

with this scheme. As the objectives increase, they cover a more substantial portion of the genome space. Lower numbered genes will also be covered by every objective while the highest genes will only be covered by a few objectives. This situation is intended to mimic tasks where there may be a common "simple" behavior while more advanced behaviors encompass and expand on those behaviors.

Evolutionary Algorithm and Parameters

For each test, we conduct 20 replicate runs initialized with a unique random seed. Populations of 100 individuals are evolved for 1,000 generations. Mutation rate is $1/genome\_length$ and crossover probability is 50%. A selection subset size of 4 is used during Lexicase selection with an $\epsilon = 0.9$. That is, individuals within 10% of the best individual in a selection subset on the objective currently being evaluated will be considered tied and the selection will proceed to the next objective. To maintain computational efficiency, we employ downsampling with a maximum of 10 objectives per selection evaluated. If a subset of the individuals are considered tied on all 10 objectives, a random individual is selected and a tiebreak is recorded. Each objective is a maximization task summing the number of 1's across the gene range. Population size and number of generations along with the evolutionary parameters have been chosen to follow those in (Moore and Stanton, 2017) as the FAAS ϵ-Lexicase technique is ultimately intended to be applied in

ER. As such, using parameters common in that domain more closely mimics conditions the algorithm will experience in computationally costly tasks.

Results

Fitness Performance Across Tests Figure 2 plots the count of 1's in the 500 gene genome (*All_Fit*) of the best individual per replicate across treatments for every test in this study. Since there is varying gene overlap depending on the test, *All_Fit* provides a standardized score to compare results across tests. High fitnesses evolve in the first three tests while *T4-Progressive* fails to evolve high fitness. *T1-No Overlap* provides a baseline for the FAAS algorithms as each objective is independent of the others. There is no significant difference in *All_Fit* in *T1-No Overlap* between treatments using a Wilcoxon test with Bonferroni Correction. This test is used to conduct all following statistical tests. In *T2-Overlap*, *Slide* significantly outperforms the other three treatments ($p < 0.001$). Fitnesses increase for all *FAAS* treatments with the addition of wraparound objectives in *T3-Wraparound*, but the only significant difference in performance is between *Slide* and ϵ-Lexicase. Unlike prior experiments, fitnesses in *T4-Progressive* are lower in general with a mean around 340 out of a possible 500. This suggests that the task itself might be considerably harder, or the dynamics of the task do not support an adaptive approach.

Computational Cost Across Tests Figure 3 plots how many objectives cumulatively were used in selection events across replicates. *Slide* has significantly fewer evaluations for *T1-No Overlap* and *T2-Overlap* compared to the other three treatments. *Slide* only has significantly fewer evaluations than ϵ-Lexicase in *T3-Wraparound* with no significant difference from the other two FAAS treatments. No significant differences arise between treatments in *T4-Progressive*.

Figure 4 plots the cumulative count of objectives evaluated during selection for *T2-Overlap* across treatments. The behavior shown in the figure is similar for *T1-No Overlap* and *T3-Wraparound*, where significant differences arise between *Slide* and ϵ-Lexicase. *Slide* requires slightly fewer objective evaluations at every generation resulting in a computational savings by the end of a run of 14% in *T1-No Overlap* and 4% in *T2-Overlap* and *T3-Wraparound*. For a typical ER run simulating 10 seconds of locomotion as in (Moore and Stanton, 2018), this can save 1 hour of compute time per replicate with up to 80 hours of savings across a 4 treatment, 20 replicate experiment.

Surprisingly, *Slide* has significantly fewer evaluations in *T1-No Overlap* where no shared genetic regions exist between objectives. Figure 5 plots the number of objectives used during evaluation per generation across the four treatments. Throughout the course of the run, *Slide* consistently maintains a lower number of objective evaluations. We hypothesize that this could be an artifact of *Slide* maintaining the filtering effectiveness measure for the last 800 objective evaluations only. Assuming a population size of 100 individuals, the window could effectively look back from 1 to 10 generations depending on selection depth and frequency of crossover. In *T1-No Overlap*, it may be that *Slide* is downweighting objectives that aren't good filters perhaps due to specialization by the population. This could be enough to push selection towards objectives where there is more variation in the population. By exploring these spaces, *Slide* doesn't evaluate as many objectives per selection resulting in the decreased cumulative number of evaluations.

Objective Usage Characteristics FAAS ϵ-Lexicase is intended to exploit shared relationships between objectives to more efficiently explore search spaces in computationally intensive problems. Accordingly, Figures 6 and 7 plot the genes covered by an objective on the x-axis with the frequency an objective was used on the y-axis. ϵ-Lexicase has relatively equal usage of objectives considering that objective selection is a stochastic process. *Slide* shows a larger disparity in objective usage. Even with the noise in the adaptive process, adjacent objectives in *Slide* often exhibit differences in usage. Since there is significant shared genetic information between adjacent objectives, one objective with high usage would tend toward suppressing usage of its neighbors.

Slide Further Investigation

Slide is the most effective FAAS ϵ-Lexicase approach in the initial tests. In the first part of results, a window size of 800 was used. Next, we conduct a second set of experiments with variations of the window size determining how long objective information is maintained. Window sizes are 200, 400, 800, 1600, 3200, 6400, and 25600.

Figure 8 shows the number of objectives used during selection for each replicate in *T1-No Overlap*. Fitness plots are not shown as the window size does not significantly impact fitness with scores similar to those of *Slide* from Figure 2 *T1-No Overlap*. For this configuration, there is a slow increase in the number of objectives used for selection as the sliding window size increases. There is a split between the window sizes with significant differences in the number of objectives between window sizes of 3,200 and under compared to 6,400 and 25,600. As the window size increases, more objectives are evaluated but still less than those of ϵ-Lexicase for *T1-No Overlap*.

Figure 9 plots the number of objectives used during selection per replicate for *T2-Overlap*. A similar pattern is observed in *T3-Wraparound* (not shown). It appears that there is a range of optimal window sizes in terms of the compute time required for a replicate. Too small a window size
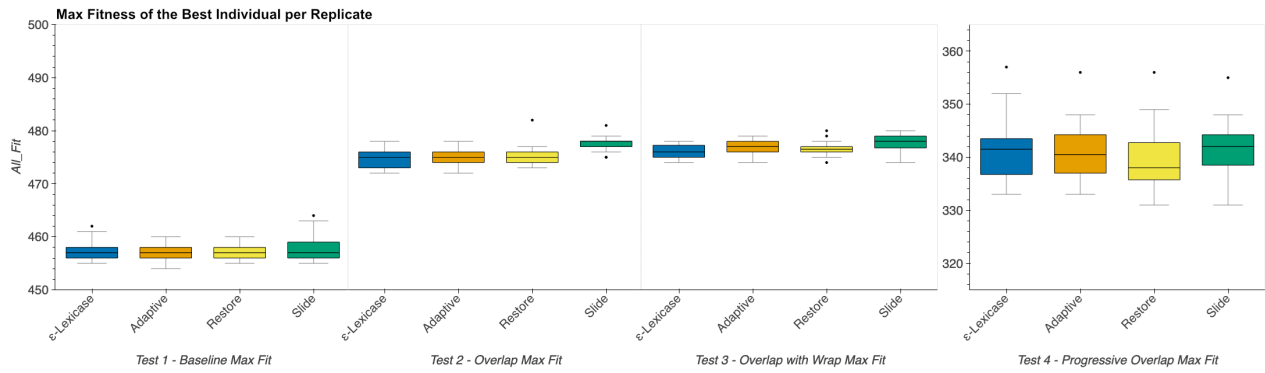
Figure 2: Cumulative count of 1's in the best individual per replicate per treatment. Note the different y-axis for *T4-Progressive* where all four treatments failed to evolve high performing individuals.
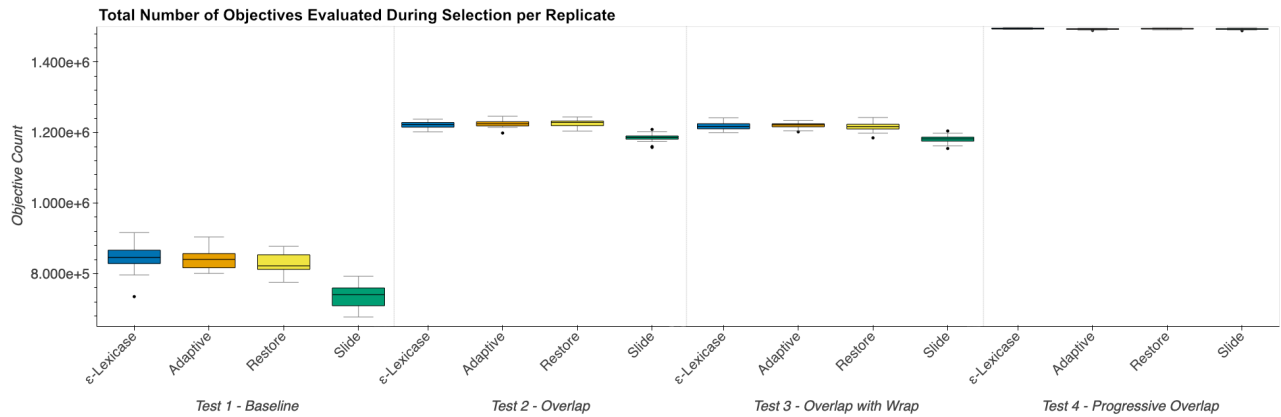

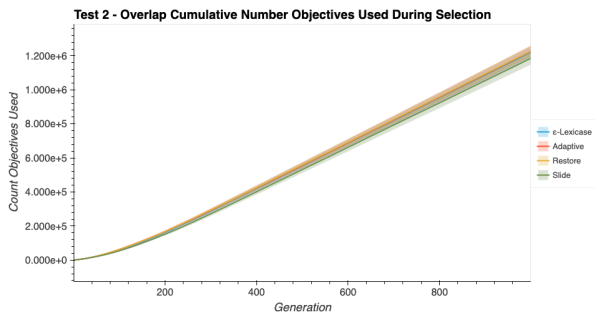Figure 3: Number of objectives evaluated per replicate per treatment.

Figure 4: Cumulative count of how many objectives were considered during selection over the course of a replicate run for *T2-Overlap*. Slide evaluates fewer objectives over the course of the evolutionary run resulting in a significant difference in total objectives evaluated after 1,000 generations. Shaded regions represent the 95% confidence intervals.
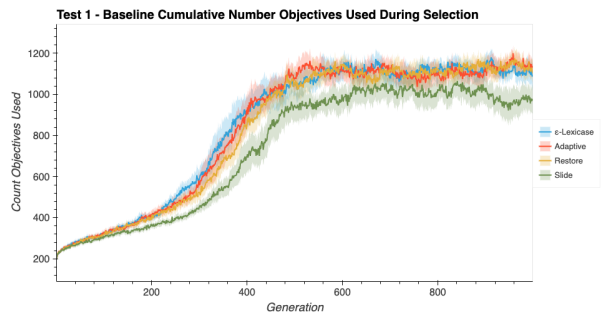

Figure 5: Count of how many objectives were considered during selection over the course of a replicate run for *T1-No Overlap*. Slide requires fewer evaluations over time, even when the objectives do not have overlapping gene loci. Shaded regions represent the 95% confidence intervals.

may prevent the algorithm from gaining enough information about objectives to determine a subset of objectives to focus on. A window size of 200 means that only the last one or two generations would be used to influence objective weighting. Whereas, a large window size could maintain information about objectives for more generations allowing

old information to influence the selection of objectives. If held for too long, old information might not represent the current best set of objectives in terms of filtering efficacy leading to deeper use of objectives during Lexicase selection and increased computation time.
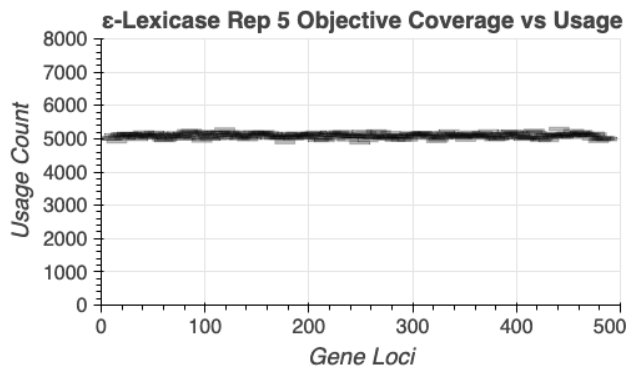
Figure 6: Objective usage count per objective in replicate 5 of ϵ-Lexicase for *T2-Overlap*. Rectangular region indicates the genes covered on the x-axis while the location on the y-axis is how many times the objective was used during selection over the course of the run.
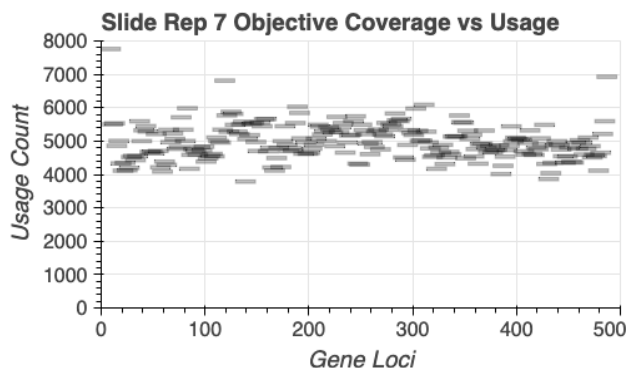

Figure 7: Objective usage count per objective in replicate 7 of *Slide* for *T2-Overlap*. Rectangular region indicates the genes covered on the x-axis while the location on the y-axis is how many times the objective was used during selection over the course of the run.

Conclusion and Future Work

As in related work (Troise and Helmuth, 2018; Ding et al., 2022), the adaptive sampling explored in this study does not always exceed the performance of ϵ-Lexicase but computational overhead can be reduced. The three adaptive sampling strategies yield differing efficiency with a sliding window being the most effective in terms of both performance gains and number of objectives used during selection. Of the three *FASS* approaches proposed, *Slide*'s improvements in fitness and computational efficiency in *T2-Overlap* and *T3-Wraparound*, and a surprising improvement in computational efficiency in *T1-No Overlap* merit further consideration in computationally expensive domains. It also appears that in the worst case (*T4-Progressive*) *Slide* degrades only to have the same performance as ϵ-Lexicase. The objective configuration of *T4-Progressive* may not lend itself to an adaptive approach as lower numbered genes are covered
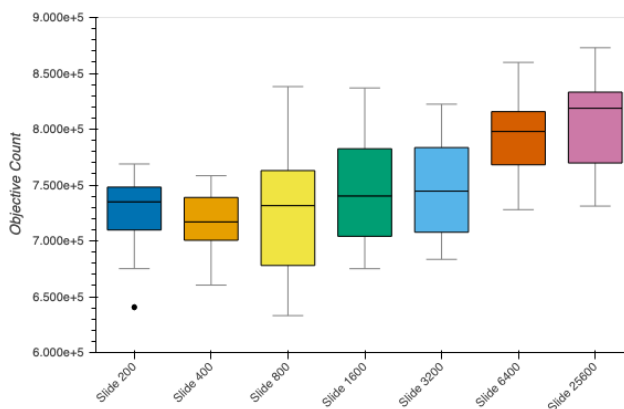

Figure 8: Count of how many objectives were considered during selection over the course of a replicate run for *T1-No Overlap*.
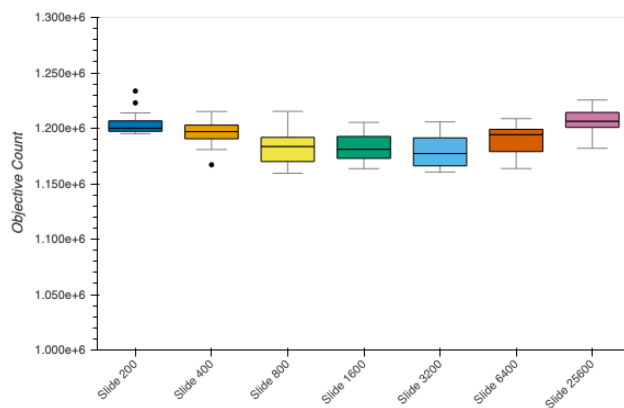

Figure 9: Count of how many objectives were considered during selection over the course of a replicate run for *T2-Overlap* slide exploration.

by every objective. While we hypothesized that the adaptive mechanism would gradually reduce the occurrence of lower objectives over evolutionary time, here it appears that there may be a point at which the adaptive mechanism cannot override the disparity between small and large objectives as determined by number of genes. It could also be that the small population size, in keeping with typical sizes in ER, limit Lexicase's ability to maintain niches in the population specializing on this problem. With so few objectives covering the higher numbered genes the task itself might limit the performance of any variant.

Leveraging the filtering efficacy of objectives provides an adaptive weighting approach that doesn't require additional evaluations and should be broadly applicable to any problem implementing Lexicase selection. A final potential benefit of the algorithm is that when it cannot leverage information about objectives, performance returns to be comparable to ϵ-Lexicase. Here, the behavior of *Slide* in *T2-Overlap* and *T3-Wraparound* support our initial hypothe-

sis that an adaptive approach can leverage relationships between objectives. While the test problems posed in this study are not computationally intensive, they mimic the relationships we suspect exist in ER problems like wall-cross and turn-and-seek where each objective requires its own physics simulation. There, the savings in objectives evaluated could allow for more generations, larger population sizes, or reduce the power requirements of a run enhancing the sustainability of the approach.

In this work, we provide an initial exploration of using filtering efficacy as an adaptive mechanism weighting objectives without additional overhead on a binary genome problem. Future work will examine applying *FASS* to more complex problem domains in ER. We plan to examine if the adaptive mechanism can effectively leverage relationships in more complex problems while also revealing unknown relationships between objectives that might inform problem design/dynamics. We are also interested in the impact that FAAS ϵ-Lexicase approaches might have on diversity and exploration, two critical factors in ER.

Acknowledgements

References

Ding, L., Boldi, R., Helmuth, T., and Spector, L. (2022). Lexicase selection at scale. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, pages 2054–2062, New York, NY, USA. Association for Computing Machinery.

Dolson, E. and Ofria, C. (2018). Ecological theory provides insights about evolutionary computation. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '18, pages 105–106, Kyoto, Japan. ACM.

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).

Ferguson, A. J., Hernandez, J. G., Junghans, D., Lalejini, A., Dolson, E., and Ofria, C. (2020). *Characterizing the Effects of Random Subsampling on Lexicase Selection*, pages 1–23. Springer International Publishing, Cham.

Floreano, D., Husbands, P., and Nolfi, S. (2008). Evolutionary Robotics. In *Handbook of Robotics*. Springer Verlag, Berlin.

He, Y., Aranha, C., Hallam, A., and Chassagne, R. (2022). Optimization of subsurface models with multiple criteria using lexicase selection. *Operations Research Perspectives*, 9:100237.

Helmuth, T., McPhee, N. F., and Spector, L. (2016). Effects of lexicase and tournament selection on diversity recovery and maintenance. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, GECCO '16 Companion, pages 983–990, New York, NY, USA. Association for Computing Machinery.

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

Hernandez, J. G., Lalejini, A., Dolson, E., and Ofria, C. (2019). Random subsampling improves performance in lexicase selection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, pages 2028–2031, New York, NY, USA. Association for Computing Machinery.

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.

Moore, J. M. and Clark, A. J. (2021). Supervision and Evolution: Pretraining Neural Networks for Quadrupedal Locomotion. In *Proceedings of The 2021 Conference on Artificial Life*, volume ALIFE 2021: The 2021 Conference on Artificial Life. MIT Press. 57.

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.

Moore, J. M. and Stanton, A. (2018). Tiebreaks and diversity: Isolating effects in lexicase selection. In *Proceedings of the 16th International Conference on the Simulation and Synthesis of Living Systems*, pages 590–597, Tokyo, Japan. ACM.

Moore, J. M. and Stanton, A. (2019). The limits of lexicase selection in an evolutionary robotics task. In *The 2019 Conference on Artificial Life: A Hybrid of the European Conference on Artificial Life (ECAL) and the International Conference on the Synthesis and Simulation of Living Systems (ALIFE)*, pages 551–558, Newcastle-Upon-Tyne, UK. MIT Press.

Moore, J. M. and Stanton, A. (2020). When specialists transition to generalists: Evolutionary pressure in lexicase selection. *Artificial Life Conference Proceedings*, pages 719–726.

Moore, J. M. and Stanton, A. (2021). Objective Sampling Strategies for Generalized Locomotion Behavior with Lexicase Selection. In *Proceedings of The 2021 Conference on Artificial Life*, volume ALIFE 2021: The 2021 Conference on Artificial Life, page 73.

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 Moore, J. M. (2022). Lexicase Selection for Multi-Task Evolutionary Robotics. *Artificial Life*, 28(4):479–498.

Troise, S. A. and Helmuth, T. (2018). Lexicase selection with weighted shuffle. In Banzhaf, W., Olson, R. S., Tozier, W., and Riolo, R., editors, *Genetic Programming Theory and Practice XV*, pages 89–104, Cham. Springer International Publishing.