

An analysis of ϵ -lexicase selection for large-scale many-objective optimization

Anonymous

ABSTRACT

In this paper we adapt ϵ -lexicase selection, a parent selection strategy designed for genetic programming, to solve many-objective optimization problems. ϵ -lexicase selection has been shown to perform well in regression due to its use of full program semantics for conducting selection. A recent theoretical analysis showed that this selection strategy preserves individuals located near the boundaries of the Pareto front in semantic space. We hypothesize that this strategy of biasing search to extreme positions in objective space may be beneficial for many-objective optimization as the number of objectives increases. Here, we replace program semantics with objective fitness to define ϵ -lexicase selection for many-objective optimization. We provide a detailed example ϵ -lexicase selection on a 3 objective problem to illustrate its behavior. We then compare this method to multi-objective optimization methods from literature on problems ranging from 3 to 100 objectives. We find that ϵ -lexicase selection outperforms state-of-the-art optimization algorithms in terms of convergence to the Pareto front, spread of solutions, and CPU time for problems with more than 3 objectives.

KEYWORDS

many-objective optimization, selection

ACM Reference Format:

Anonymous. 2018. An analysis of ϵ -lexicase selection for large-scale many-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference 2018 (GECCO '18)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

The multi-objective optimization (MO) community is increasingly interested in algorithms that can scale to large numbers of objectives. Typically the name “many”-objective optimization (MaO) is used to describe cases in which there are four or more competing objectives. Dealing with large numbers of objectives changes the nature of the search space [7, 11] and as a result, different types of algorithms perform well [3]. As research has progressed, studies have analyzed the ability of evolutionary multi-objective optimization (EMO) algorithms to find or approximate Pareto-optimal solution sets for

problems with up to 6 [27], 50 [2], and more recently, 100 objectives [20].

At the same time that EMO research has moved to larger sets of objectives, the genetic algorithm (GA) and genetic programming (GP) communities have shown strong interest in the so-called “multi-objectivization” of single-objective problems [13]. In GP, the idea of restructuring search drivers around program semantics, defined as the outputs or behavior of a GP program, has gained traction [23]. Rather than aggregating performance across training instances, semantic methods can re-define the problem as a set of smaller objectives for driving search. Methods to discover search objectives by clustering [14] and by matrix factorization [22] have been proposed that analyze population semantics, extract sub-problems in the search space, and use multi-objective selection methods according to these sub-problems. Finally, lexicase selection [26], a version of which is studied here, uses program semantics to filter the population via randomized orders of training cases at each selection event. By doing so it is able to adapt selection pressure to subsets of training cases that are harder to solve.

A variant of lexicase selection known as ϵ -lexicase selection was introduced to apply lexicase selection to continuous error spaces for symbolic regression [16]. A recent theoretical analysis considered ϵ -lexicase selection through a multi-objective lens [15]. It showed that ϵ -lexicase selects individuals located near boundaries of the Pareto set defined by the population’s error vectors. In this sense, ϵ -lexicase selection demonstrated an instance of a multi-objective treatment of regression with promising results.

In this paper, we seek to evaluate the performance of ϵ -lexicase selection as a many-objective optimization algorithm. In §2, we describe ϵ -lexicase selection and summarize recent theoretical results related to EMO. We then illustrate the probabilities of selection as a function of Pareto front location on a three objective problem. In §3, we describe our experimental study, which consists of a comparison of three EMO methods on the scalable DTLZ problems with up to 100 objectives. The results are presented in §4 and are discussed in §5 with suggestions for future research directions.

2 METHODS

We first define ϵ -lexicase selection for EMO problem. In EMO, we wish to optimize a set of solutions that approximate the Pareto-optimal set. We have the d -dimensional decision space $X \subset \mathbb{R}^d$, the objective space $Z \in \mathbb{R}^m$ with m objectives, and a vector of functions $\mathcal{F} = (f_1, f_2, \dots, f_m)$ that each map the decision space to real values, i.e. $f_i : X \rightarrow \mathbb{R}$. The optimization problem can be formulated as

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
GECCO '18, July 15–19, 2018, Kyoto, Japan
© 2018 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM. . . \$15.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

$$\min \mathcal{F}(x) = (f_1(x), f_2(x), \dots, f_m(x)), x \in X \quad (1)$$

In EMO, we seek to approximate the Pareto-optimal set with a population of solutions of size N , denoted as $\mathcal{N} = \{x^{(i)}\}_{i=1}^N$, where $x^{(i)} = (x_1, \dots, x_d)$ is a vector of d decision variables. Optimal solutions are those that are non-dominated. Solution x *dominates* solution x' if $f(x) \leq f(x')$ for all $f \in \mathcal{F}$ and there exists at least one $f \in \mathcal{F}$ for which $f(x) < f(x')$. In addition to dominance, the set of solutions in \mathcal{N} should characterize the Pareto-optimal set well, meaning they should 1) cover as much of the Pareto-optimal set as possible and 2) cover it as evenly as possible. In our experiments, we use metrics that consider uniformity as well as convergence.

2.1 ϵ -lexicase selection

We define the ϵ -Lexicase selection algorithm for EMO in Algorithm 1. Each parent selection begins with the entire population in the selection pool. A random “case”, i.e. objective, is drawn from the objectives. For this case, each individual is checked: if it is not within ϵ of the best objective value in the current selection pool, it is removed from selection. Selection continues to consider subsequent cases until the objectives are exhausted or the selection pool is reduced to one individual. If the former happens first, a random individual from the selection pool is chosen.

The version of ϵ -lexicase selection we consider here is known as “semi-dynamic” ϵ -lexicase selection [15]. In this version, ϵ is defined automatically using a population distribution statistic. Let \mathbf{f}_i be the values for objective i across the population; i.e.,

$$\mathbf{f}_i = \{f_i(x) \mid x \in \mathcal{N}\} \quad (2)$$

We want ϵ to relate to the difficulty of objective i , often estimated using variance. To improve robustness to outliers, ϵ -lexicase selection uses the median absolute deviation as follows:

$$\lambda(\mathbf{f}) = \text{median}(|\mathbf{f} - \text{median}(\mathbf{f})|) \quad (3)$$

$$\epsilon_i = \lambda(\mathbf{f}_i) \quad (4)$$

As shown in Algorithm 1, Eqns. 2-4 are used to determine the cut-off for filtering out candidate solutions during selection. ϵ is calculated only once for each objective each generation.

2.2 Parent properties

Parents selected by ϵ -lexicase selection share certain properties regarding their location in objective space. We define those properties here, transforming them from their definitions for GP program semantics in [15] to objective values. We make use of two definitions: strict ϵ -dominance and the ϵ -Pareto set boundary, which are specialized definitions of dominance and the Pareto set boundary that factor in ϵ . For all of the following definitions, ϵ_j is defined according to Eqn 4.

Algorithm 1 : ϵ -Lexicase Selection applied to individuals $x \in \mathcal{N}$ with objective values $f_i(x)$, $f_i \in \mathcal{F}$.

Selection(\mathcal{N}, \mathcal{F}) :

$\mathcal{P} \leftarrow \emptyset$ ◇ parents
for $f_i \in \mathcal{F}$: ◇ get ϵ for each f_i
 $\epsilon_i \leftarrow \lambda(\mathbf{f}_i)$
do N **times**:
 $\mathcal{P} \leftarrow \mathcal{P} \cup \text{GetParent}(\mathcal{N}, \mathcal{F}, \epsilon)$ ◇ add selection to \mathcal{P}

GetParent($\mathcal{N}, \mathcal{F}, \epsilon$) :

$\mathcal{F}' \leftarrow \mathcal{F}$ ◇ objectives
 $\mathcal{S} \leftarrow \mathcal{N}$ ◇ selection pool
while $|\mathcal{F}'| > 0$ **and** $|\mathcal{S}| > 1$:
 $f_i \leftarrow \text{random choice from } \mathcal{F}'$ ◇ pick random f_i
 $f_i^* \leftarrow \min f_i(x) \text{ for } x \in \mathcal{S}$ ◇ best score on f_i in pool
for $x \in \mathcal{S}$: ◇ filter pool
if $f_i(x) > f_i^* + \epsilon_m$ **then**
 $\mathcal{S} \leftarrow \mathcal{S} \setminus \{x\}$
 $\mathcal{F}' \leftarrow \mathcal{F}' \setminus \{f_i\}$ ◇ remove f_i
return random choice from \mathcal{S}

DEFINITION 2.1 (STRICT ϵ -DOMINANCE). x *strictly ϵ -dominates* x' , i.e., $x \prec_\epsilon x'$, if $f_j(x) + \epsilon_j \leq f_j(x') \forall j \in \{1, \dots, m\}$ and $\exists j \in \{1, \dots, m\}$ for which $f_j(x) + \epsilon_j < f_j(x')$.

Note that a strict definition of ϵ -dominance distinguishes it from the ϵ -dominance concept introduced in [17], which does not require x to improve upon the objectives of x' in order to satisfy $x \prec_\epsilon x'$.

A close look at the filtering step of Algorithm 1 makes it clear that any parent x selected by ϵ -lexicase selection satisfies $x \not\prec_\epsilon x'$ for all $x' \in \mathcal{N}$. Since every selection event begins with the entire population, if there is an individual $x' \prec_\epsilon x$, then x' will remain in the selection pool \mathcal{S} whenever x does, and eventually the objective(s) for which x' outperforms x by at least ϵ will be considered and will result in the filtering of x . Therefore x can never be selected if it is strictly ϵ dominated. (A formal proof is given in [15]).

In addition to being non-dominated according to Definition 2.1, parents selected by ϵ -lexicase selection belong to the ϵ -Pareto set boundary, defined below.

DEFINITION 2.2 (ϵ -PARETO SET BOUNDARY). $x \in \mathcal{N}$ is an ϵ -Pareto set boundary if $x \not\prec_\epsilon x' \forall x' \in \mathcal{N}$ and $\exists j \in \{1, \dots, m\}$ for which $f_j(x) \leq f_j(x') + \epsilon_j \forall x' \in \mathcal{N}$.

THEOREM 2.1 ([15]). *If individuals are selected from a population \mathcal{N} by ϵ -lexicase selection, then those individuals are ϵ -Pareto set boundaries of \mathcal{N} .*

See [15] for a straightforward proof of Theorem 2.1. In summary, ϵ -lexicase selection selects individuals that are not strictly ϵ -dominated and are also within ϵ of the boundary of the Pareto front. Because subsequent objectives continue to winnow the selection pool \mathcal{S} , the parents of ϵ -lexicase are actually ϵ -Pareto set boundaries of \mathcal{S} for each objective that is traversed. This means that despite belonging to the ϵ -Pareto set of \mathcal{N} , ϵ -lexicase selection will filter out the most extreme points if those points are not within ϵ of the best objective scores in \mathcal{S} for subsequent objectives. The loss of extreme

solutions is also a property of other ϵ -based selection methods [18], and methods to address it have been proposed [10]. §2.3 provides an illustration of the probabilities of selection that make the locations from which parents are selected more clear.

2.3 Illustration with three objectives

The performance of ϵ -lexicase selection is known to suffer when small numbers of training cases are available for GP tasks. In the EMO setting, this corresponds to small numbers of objectives. Since ϵ -lexicase selection depends on randomized orderings of the cases to select different individuals, and there are $m!$ possible orderings of cases, when m is small, the number of parent selection outcomes shrinks. For example, when $m=2$, there are 2 possible orderings of cases; for $m=10$, there are more than 3.6 million.

We illustrate this shortcoming of ϵ -lexicase selection on the DTLZ1 problem considering 3 objectives. The Pareto set of the DTLZ1 problem consists of a triangular plane with boundaries at $\mathcal{F} = (0,0,0.5)$, $(0,0.5,0)$, and $(0.5,0,0)$. The top plot of Figure 1 shows a sampling of the Pareto set, colored by the probabilities of selection under ϵ -Lexicase selection for this population (cf. Eqn.4 [15]). As the plot makes clear, individuals on the Pareto front are not treated equally by ϵ -lexicase selection. The only individuals that have any chance of parent selection are within ϵ of the boundaries of the Pareto set. In addition, among solutions near the boundaries, the most extremely positioned solutions are dropped from selection. Since the extreme boundary solutions are not within ϵ of the best solutions for the subsequent objectives, they are filtered during selection.

As a result of this behavior, we expect ϵ -lexicase to perform poorly for small objectives, and the bottom two plots of Figure 1 exemplify this for the DTLZ1 problem. The non-dominated sorting genetic algorithm (NSGA-II) and ϵ -lexicase selection are each run for 100 generations with population sizes of 300, and their final populations are shown. NSGA-II does a good of finding solutions spread among the front, whereas ϵ -lexicase selection instead finds Pareto front solutions located near the boundaries, as we would expect.

Given this example, why would ϵ -lexicase selection be good for many-objective optimization problems? It is important to consider how the objective space changes as the number of objectives grow. Pareto dominance relations become less useful; at $m = 10$, for example, approximately 90% of random solutions drawn from an m -dimensional cube will be non-dominated [11]. The number of boundaries grows as m , and for a fixed population size, the proportion of random solutions near boundaries will also increase. As mentioned above, the set of possible objective orderings increases factorially as well, which improves the potential selection outcomes under ϵ -lexicase selection.

In this high-dimensional setting, ϵ -lexicase selection provides a natural preference relation for solutions near boundaries of the front. ϵ -lexicase selection provides higher selection probabilities to individuals that perform well on objectives

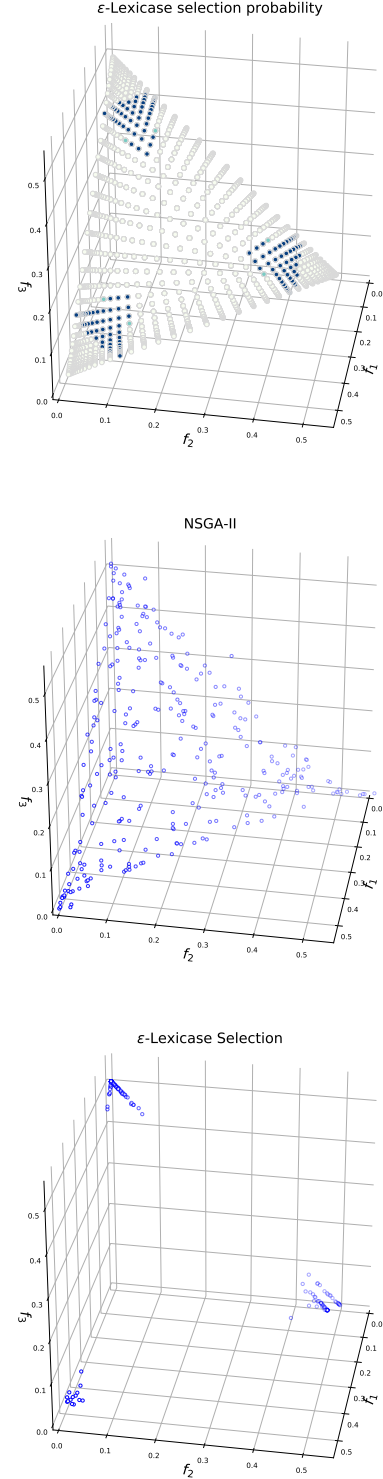


Figure 1: Objective space characterization on the DTLZ1 problem with 3 objectives. (Top) Probabilities of selection among individuals on the Pareto front under ϵ -Lexicase selection. (Middle) ϵ -Lexicase selection solutions. (Bottom) NSGA-II solutions.

not solved by most of the population. Objectives that are more difficult to perform well on will filter more individuals from the population when they are chosen in selection, resulting in higher selection probabilities for the individuals that solve them. Conversely, objectives that are easy to perform well on, or have approximately similar performance across the population, filter few individuals and have a small impact on selection. These properties result in high behavioral diversity in GP, as demonstrated in empirical studies [8, 15]. We expect behavioral diversity in a GP settings to translate to diversity in objective scores in the MaO setting. For an in-depth analysis of probabilities of selection with lexicase selection, the reader is referred to [15]; code to calculate probabilities of selection and generate the example is available here¹.

2.4 Relation to many-objective optimization

Many-objective problems are typically solved with other optimization algorithms, such as those proposed in [3]. Two popular classes of many-objective optimization algorithms are reference point methods and indicator-based methods. Reference point based methods drive search by decomposing the search space into weighted vectors that provide a sub-problem trajectory for solutions. Reference point based methods include MOAE/D [21], NSGA-III [24], and others that are benchmarked in [20]. Indicator-based algorithms attempt to quickly calculate a single performance metric that captures dominance and spread of solutions. A popular choice for such a metric is the hypervolume, which measure of spread and convergence of a set of solutions [1]. Indicator-based methods include IBEA [29] and HypE [2]. For surveys of methods for many-objective optimization, the reader is referred to [3, 19].

Most EMO methods for MaO do not explicitly focus on boundary solutions. Current views on the role and usefulness of boundary solutions in driving search are mixed. Deb et. al. [5] provided empirical evidence supporting the idea that boundary solutions contributed an outsized portion of hypervolume measures for some problems. However, a more in depth analysis by Auger et. al. [1] reached the opposite conclusion, suggesting that, in the general case, solutions near the ‘knee’ of the Pareto front contributed more to hypervolume measures due to the angle they formed with neighboring solutions. Wagner et. al suggested favoring boundary solutions hinders progress in MaO [27]. Nevertheless, EMO algorithms that exploit boundary solutions have been proposed, both for dimensionality reduction [25] and for speeding up non-dominated sorting routines [28].

2.5 Complexity

The worst-case time complexity of ϵ -lexicase selection is $O(N^2m)$, which occurs if every solution performs the same on every objective. This scenario is unlikely for two reasons: 1), the space of floating-point objective values is large, and 2) ϵ -lexicase selection biases selection towards individuals with

¹<https://www.github.com/anon-gecco/emo-lex>

Table 1: Population size (number of generations) for each algorithm in the experimental study.

| m | Problem | |
|-----|------------|------------|
| | DTLZ1,3,4 | DTLZ2 |
| 3 | 100 (500) | 100 (500) |
| 5 | 100 (1000) | 100 (800) |
| 25 | 125 (1000) | 125 (1000) |
| 50 | 250 (1000) | 250 (1000) |
| 75 | 375 (1000) | 375 (1000) |
| 100 | 500 (1000) | 500 (1000) |

unique objective performance. In practice, the wall-clock run-times of ϵ -lexicase selection have been demonstrated to scale better than $O(N^2m)$ [9, 15]. We empirically assess computation times for ϵ -lexicase selection in our experiments in §3.

3 EXPERIMENT

For our experiment we focus on four scalable test problems proposed in [6] known as DTLZ1, DTLZ2, DTLZ3, and DTLZ4. Each problem scales to any number of objectives and dimensions in decision space. For each problem, we consider six values of m : 3, 5, 25, 50, 75, and 100. In accordance with [6] and [20], we use $d = m + 4$ decision variables for DTLZ1 and $d = m + 9$ decision variables for DTLZ2 through DTLZ4. The population sizes and generation counts for each experiment is detailed in Table 1. Each experiment was run for 30 trials. The complete code to reproduce the experiment and is provided on github².

In an effort to conduct a reliable benchmark comparison, we implemented ϵ -lexicase selection as a selector using the PISA framework, described in [12] and available online³. We used PISA’s implementations of the DTLZ problems and the comparison methods described below.

For each problem, we used simulated binary crossover for recombination and variable mutation. Variable mutation probability was set to 0.01. The distribution indices for mutation and recombination were set to 20 and 15, respectively.

3.1 Comparison methods

LEX. The name LEX is used to refer to the ϵ -lexicase selection implementation used in this experiment. Each generation, Algorithm 1 is used to choose parents for variation. In existing GP implementations of this algorithm, the offspring completely replace the parent population. In contrast to most EMO algorithms, ϵ -lexicase selection has not been used with any explicit survival step in previous work, nor has it made use of an archive. In keeping with previous implementations, LEX behaves the same way. Because of this it is possible to lose Pareto-optimal individuals, and indeed any individuals, during search.

²<https://www.github.com/anon-gecco/emo-lex>

³<http://www.tik.ee.ethz.ch/pisa/>

NSGA-II. NSGA-II [4] sorts individuals based on their Pareto front ranking and their distances to other solutions in objective space. Since it is based on Pareto dominance, we expect it to perform well for small numbers of objectives as shown in the example in §2.3 and to scale poorly to large numbers of objectives, as the literature suggests [3].

HypE. HypE is an EMO that estimates the hypervolume measure to guide the search process. Its main innovation is to use a Monte Carlo algorithm to quickly estimate the hypervolume for large numbers of objectives. In previous studies, it was shown to perform well on problems with up to 50 objectives [2].

3.2 Metrics

We use two measures to compare performance of the algorithms: the convergence measure, CM, and the inverted generational distance, IGD. Neither measure is used explicitly as a loss function in any of the algorithms, making them fair choices. Both metrics are recommended and used in previous work [3, 20].

CM works by measuring the distance of the solution population to the Pareto front in objective space. CM is defined as:

$$CM = \begin{cases} \frac{1}{N} \sum_{i=1}^N |\sum_{j=1}^m f_j(x_i) - 0.5|, & \text{DTLZ1} \\ \frac{1}{N} \sum_{i=1}^N |\sum_{j=1}^m f_j^2(x_i) - 1|, & \text{DTLZ2-4} \end{cases} \quad (5)$$

Unlike CM, the IGD score takes into account the spread of solutions in the final population. To compute the IGD, one must have a reference set of Pareto-optimal objective values, denoted P^* . Then IGD is calculated as

$$IGD = \frac{1}{|P^*|} \sum_{f^* \in P^*} \text{dist}(f^*, \mathcal{N}) \quad (6)$$

where dist is the distance of f^* to its nearest neighbor in \mathcal{N} . Since the Pareto-optimal set cannot be enumerated, we generated reference sets by uniformly sampling the Pareto-optimal front for each problem and level of m 10,000 times. The method for generating P^* is introduced in [21] and described here. The points were generated by randomly sampling m -dimensional weight vectors (w) and computing the intersection of these vectors with the Pareto front. The intersection point is on the Pareto front and satisfies

$$f_i^*(x) = \begin{cases} 0.5 * w_i / \sum_{i=1}^m w_i, & \text{DTLZ1} \\ w_i / \sqrt{(\sum_{i=1}^m w_i^2)}, & \text{DTLZ2-4} \end{cases} \quad (7)$$

4 RESULTS

The CM and IGD scores of the experiments are shown in Figures 2 and 3. Each quadrant of the figures contains the results from a specific DTLZ problem, with columns corresponding to the numbers of objectives. The log-normalized median values of the metrics are plotted along with their distributions. For each problem and objective size, we compute relative rankings of the methods according to CM and IGD; these rankings are summarized in the bar plots in Figures 4 and 5, respectively.

The CM results show interesting trends as a function of m . For $m = 3$, the three methods perform the most similarly. For 5 objectives and higher, the diminishing performance of NSGA-II is evident. For most problem instances of $m > 5$, LEX achieves a lower CM than HypE, with some exceptions. HypE tends to perform more poorly as the number of objectives increase, most notably on the DTLZ4 problem. At $m = 100$, LEX achieves the best convergence for each problem.

The IGD results are less strongly in favor of LEX, especially for $m < 25$. At $m = 3$, NSGA-II outperforms both LEX and HypE as expected, but NSGA-II's performance rapidly deteriorates. For $m > 25$, the median LEX scores are the lowest on average, as Figure 5 makes clear.

We conduct pairwise Wilcoxon rank-sum tests on these results to test for significant differences. The pairwise tests are conducted between LEX and the other two methods. The p -values are detailed in Table 4. The p -values are significant for nearly every comparison, which is unsurprising given the small variability noted in Figures 2 and 3.

Finally, we report the CPU times for each algorithm run in Figure 6. Given the heterogenous nature of the cluster used for these runs, these results should be analysed with a grain of salt. For the 3 and 5 objective cases, NSGA-II has the lowest times, followed by LEX. For 25 objectives and higher, the LEX runs complete in the lowest median CPU time.

Table 2: Significance test p -values comparing LEX to NSGA-II and HypE using the pair-wise Wilcoxon rank-sum test with Bonferroni correction. Bold indicates $p < 0.01$.

| Problem | m | CM | | IGD | |
|---------|-----|-----------------|-----------------|-----------------|-----------------|
| | | NSGA-II | HypE | NSGA-II | HypE |
| DTLZ1 | 3 | 3.60e-13 | 2.32e-06 | 1.63e-07 | 2.32e-06 |
| | 5 | 5.07e-17 | 1.29e-07 | 6.09e-16 | 1.63e-04 |
| | 25 | 4.80e-13 | 1.90e-07 | 4.80e-13 | 1.90e-07 |
| | 50 | 4.80e-13 | 4.80e-13 | 4.80e-13 | 4.80e-13 |
| | 75 | 1.03e-11 | 1.96e-09 | 1.03e-11 | 2.57e-04 |
| | 100 | 2.50e-12 | 6.22e-13 | 2.50e-12 | 2.74e-04 |
| DTLZ2 | 3 | 3.60e-13 | 3.60e-13 | 3.60e-13 | 3.60e-13 |
| | 5 | 3.60e-13 | 3.60e-13 | 3.60e-13 | 3.60e-13 |
| | 25 | 2.82e-11 | 2.82e-11 | 5.16e-12 | 5.16e-12 |
| | 50 | 6.79e-11 | 6.79e-11 | 3.64e-12 | 3.64e-12 |
| | 75 | 6.22e-13 | 6.22e-13 | 1.25e-13 | 1.25e-13 |
| | 100 | 2.03e-12 | 1.81e-09 | 1.97e-12 | 2.66e-04 |
| DTLZ3 | 3 | 1.72e-11 | 1.72e-11 | 1.72e-11 | 1.72e-11 |
| | 5 | 5.07e-17 | 1.90e-01 | 5.07e-17 | 1.00e+00 |
| | 25 | 4.80e-13 | 2.65e-06 | 4.80e-13 | 1.72e-11 |
| | 50 | 4.80e-13 | 4.86e-10 | 4.80e-13 | 4.86e-10 |
| | 75 | 1.03e-11 | 9.75e-02 | 1.03e-11 | 8.03e-03 |
| | 100 | 9.28e-13 | 1.49e-06 | 9.27e-13 | 1.49e-06 |
| DTLZ4 | 3 | 1.25e-13 | 5.30e-13 | 1.25e-13 | 5.30e-13 |
| | 5 | 9.04e-11 | 7.04e-05 | 1.64e-10 | 2.09e-04 |
| | 25 | 4.80e-13 | 4.80e-13 | 4.80e-13 | 1.72e-11 |
| | 50 | 4.80e-13 | 4.80e-13 | 4.80e-13 | 4.80e-13 |
| | 75 | 8.76e-12 | 3.18e-11 | 8.76e-12 | 3.18e-11 |
| | 100 | 4.80e-13 | 6.14e-12 | 4.80e-13 | 6.14e-12 |

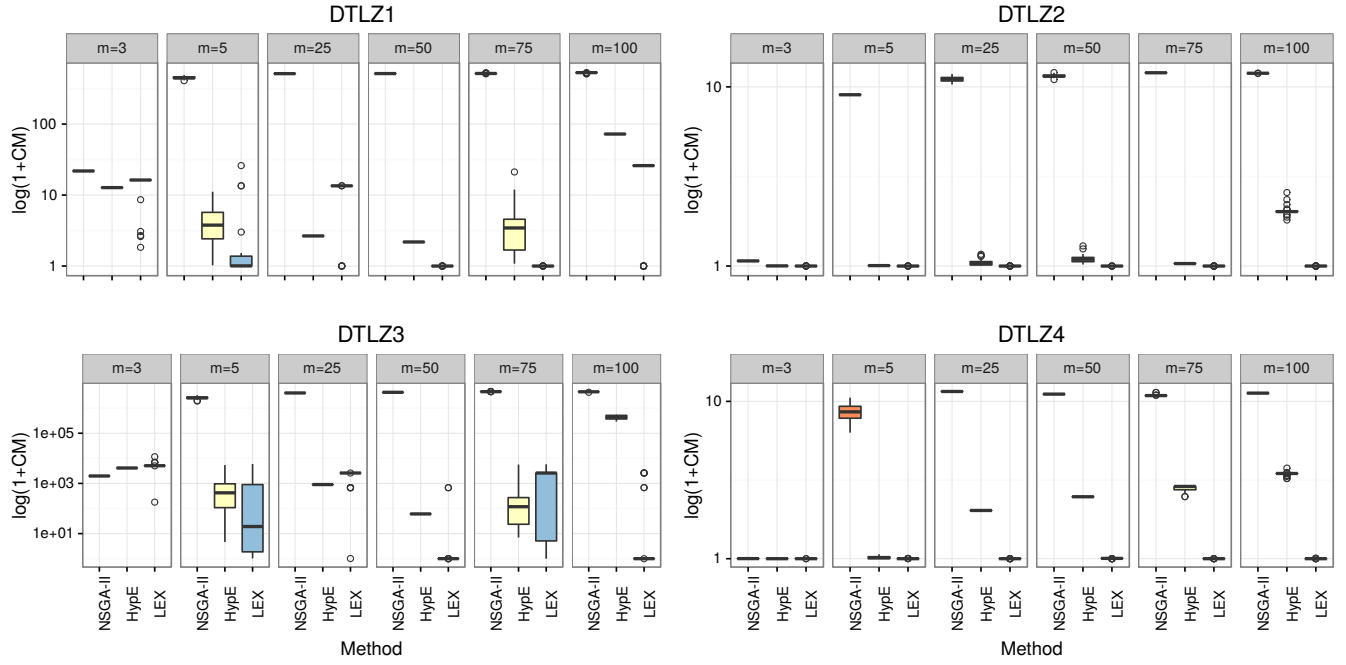


Figure 2: Convergence Measure (CM) of the final populations of each selection method. Each quadrant is a DTLZ benchmark problem, and the columns correspond to the number of objectives.

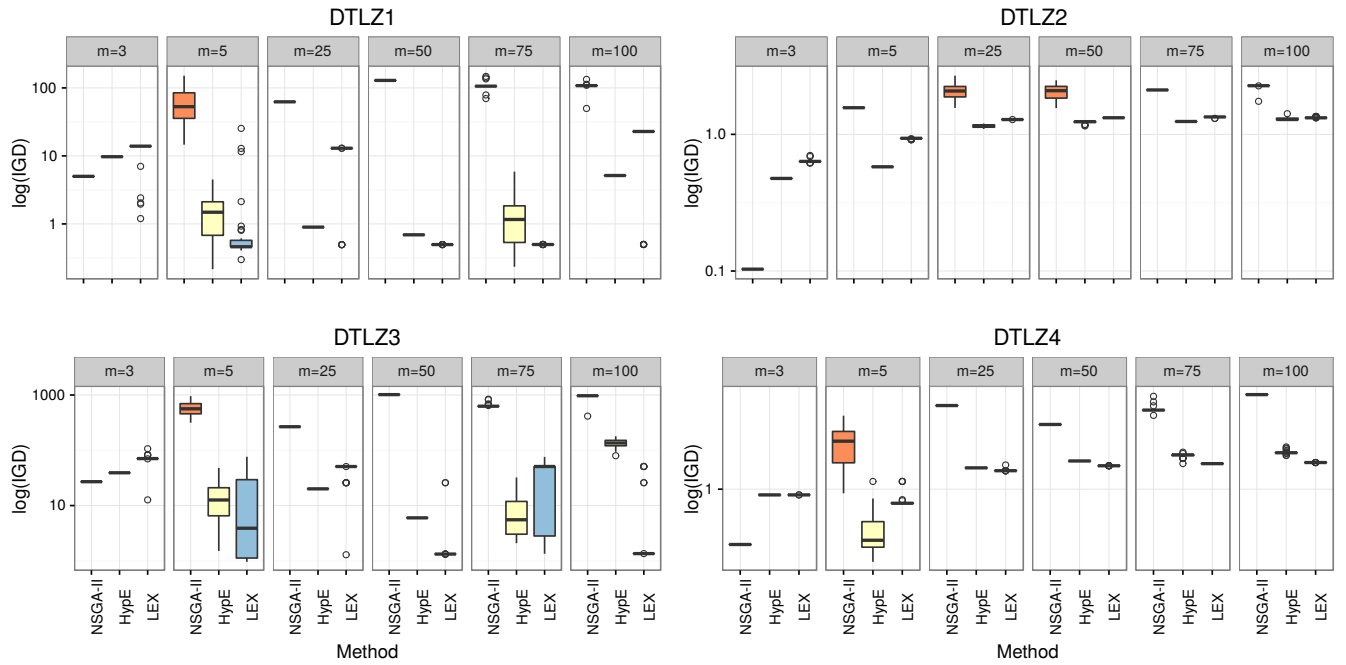


Figure 3: Inverted Generational Distance (IGD) of the final populations of each selection method. Each quadrant is a DTLZ benchmark problem, and the columns correspond to the number of objectives.

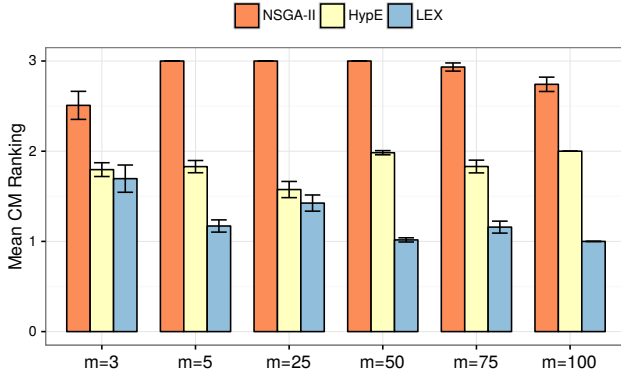


Figure 4: Average CM rankings of each algorithm as a function of m .

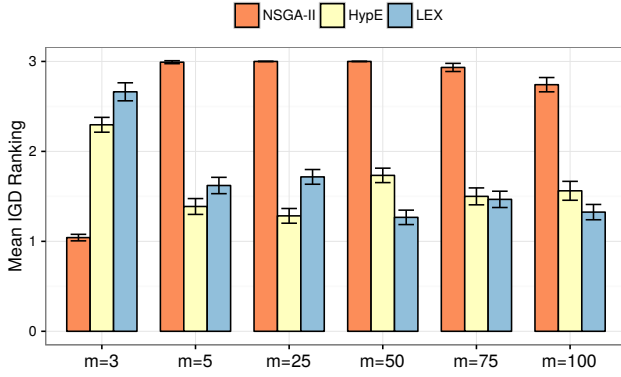


Figure 5: Average IGD rankings of each algorithm as a function of m .

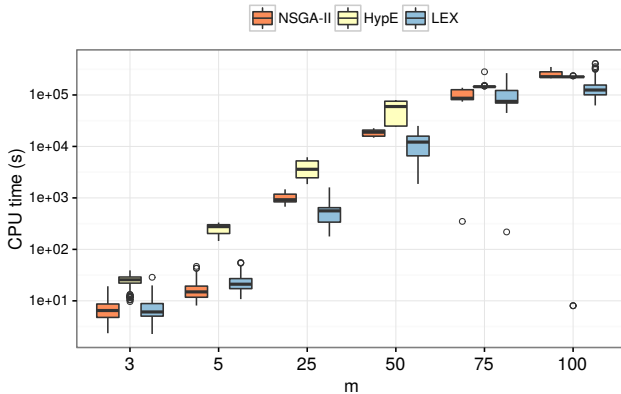


Figure 6: CPU time for each algorithm as a function of m .

5 DISCUSSION & CONCLUSIONS

In this paper we adapt a selection method developed in the GP field to many-objective optimization. We were motivated to do so by recent theoretical results showing a relation between ϵ -lexicase selection and Pareto-set boundary exploitation. Our analysis in §2 illustrates how ϵ -lexicase performs selection in objective space and why such an approach may be useful as the number of objectives grow.

The results of our experiment suggest that ϵ -lexicase selection can perform very well on many-objective optimization problems. In comparison to NSGA-II and HypE, ϵ -lexicase selection exhibits better convergence to the Pareto front, increasingly so for more objectives. Furthermore, for $m > 3$, ϵ -lexicase selection is able to preserve diversity among its solutions better than NSGA-II and in a similar manner to HypE, as measured by IGD.

Future work should consider more adaptations of ϵ -lexicase selection to incorporate concepts from other EMOs. For example, ϵ -lexicase selection could be integrated with an archiving strategy as is common with other methods. We also intend to compare ϵ -lexicase selection to other EMO algorithms, especially reference-point based methods, which are not considered in this study. In a broader sense, the success of ϵ -lexicase selection suggests that it is useful to promote solutions near Pareto-set boundaries that perform well on randomized subsets of other objectives.

6 ACKNOWLEDGMENTS

Omitted for review

REFERENCES

- [1] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. 2009. Theory of the hypervolume indicator: optimal -distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*. ACM, 87–102.
- [2] Johannes Bader and Eckart Zitzler. 2011. HypE: An Algorithm for Fast Hypervolume-based Many-objective Optimization. *Evol. Comput.* 19, 1 (March 2011), 45–76. https://doi.org/10.1162/EVCO_a.00009
- [3] Shelvin Chand and Markus Wagner. 2015. Evolutionary many-objective optimization: A quick-start guide. *Surveys in Operations Research and Management Science* 20, 2 (Dec. 2015), 35–42. <https://doi.org/10.1016/j.sorms.2015.08.001>
- [4] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *Parallel Problem Solving from Nature PPSN VI*, Marc Schoenauer, Kalyanmoy Deb, Gunther Rudolph, Xin Yao, Evelynne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel (Eds.). Vol. 1917. Springer Berlin Heidelberg, Berlin, Heidelberg, 849–858. <http://repository.ias.ac.in/83498/>
- [5] Kalyanmoy Deb, Manikant Mohan, and Shikhar Mishra. 2005. Evaluating the -Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. *Evolutionary Computation* 13, 4 (Dec. 2005), 501–525. <https://doi.org/10.1162/106365605774666895>
- [6] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. 2005. *Scalable test problems for evolutionary multiobjective optimization*. Springer. <http://link.springer.com/10.1007/1-84628-137-7.6>
- [7] Marco Farina and Paolo Amato. 2002. On the optimal solution definition for many-criteria optimization problems. In *Fuzzy Information Processing Society, 2002. Proceedings. NAFIPS. 2002 Annual Meeting of the North American*. IEEE, 233–238.

- <http://ieeexplore.ieee.org/abstract/document/1018061/>
- [8] Thomas Helmuth, Nicholas Freitag McPhee, and Lee Spector. 2016. Effects of Lexicase and Tournament Selection on Diversity Recovery and Maintenance. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. ACM, 983–990. <http://dl.acm.org/citation.cfm?id=2931657>
 - [9] T. Helmuth, L. Spector, and J. Matheson. 2014. Solving Uncompromising Problems with Lexicase Selection. *IEEE Transactions on Evolutionary Computation* PP, 99 (2014), 1–1. <https://doi.org/10.1109/TEVC.2014.2362729>
 - [10] Alfredo G. Hernandez-Daz, Luis V. Santana-Quintero, Carlos A. Coello Coello, and Julin Molina. 2007. Pareto-adaptive -dominance. *Evolutionary computation* 15, 4 (2007), 493–517.
 - [11] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. 2008. Evolutionary many-objective optimization: A short review. In *IEEE congress on evolutionary computation*. Citeseer, 2419–2426. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.139.2046&rep=rep1&type=pdf>
 - [12] J. Knowles, L. Thiele, and E. Zitzler. 2006. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. 214. Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland.
 - [13] Joshua D. Knowles, Richard A. Watson, and David W. Corne. 2001. Reducing local optima in single-objective problems by multi-objectivization. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 269–283.
 - [14] Krzysztof Krawiec and Paweł Liskowski. 2015. Automatic derivation of search objectives for test-based genetic programming. In *Genetic Programming*. Springer, 53–65. http://link.springer.com/10.1007/978-3-319-16501-1_5
 - [15] William La Cava, Thomas Helmuth, Lee Spector, and Jason H. Moore. 2017. epsilon-Lexicase selection: a probabilistic and multi-objective analysis of lexicase selection in continuous domains. *arXiv:1709.05394 [cs]* (Sept. 2017). <http://arxiv.org/abs/1709.05394> arXiv: 1709.05394.
 - [16] William La Cava, Lee Spector, and Kourosh Danai. 2016. Epsilon-Lexicase Selection for Regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. ACM, New York, NY, USA, 741–748. <https://doi.org/10.1145/2908812.2908898>
 - [17] Marco Laumanns, Lothar Thiele, and Eckart Zitzler. 2006. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research* 169, 3 (March 2006), 932–942. <https://doi.org/10.1016/j.ejor.2004.08.029>
 - [18] Marco Laumanns, Lothar Thiele, Eckart Zitzler, and Kalyanmoy Deb. 2002. Archiving with guaranteed convergence and diversity in multi-objective optimization. In *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 439–447. <http://dl.acm.org/citation.cfm?id=2955562>
 - [19] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-Objective Evolutionary Algorithms: A Survey. *Comput. Surveys* 48, 1 (Sept. 2015), 1–35. <https://doi.org/10.1145/2792984>
 - [20] Ke Li, Kalyanmoy Deb, Tolga Altinoz, and Xin Yao. 2017. Empirical investigations of reference point based methods when facing a massively large number of objectives: First results. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 390–405.
 - [21] Ke Li, Kalyan Deb, Qingfu Zhang, and Sam Kwong. 2015. *An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition*. Vol. 19.
 - [22] Paweł Liskowski and Krzysztof Krawiec. 2017. Discovery of Search Objectives in Continuous Domains. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17)*. ACM, New York, NY, USA, 969–976. <https://doi.org/10.1145/3071178.3071344>
 - [23] Paweł Liskowski, Krzysztof Krawiec, Thomas Helmuth, and Lee Spector. 2015. Comparison of Semantic-aware Selection Methods in Genetic Programming. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO Companion '15)*. ACM, New York, NY, USA, 1301–1307. <https://doi.org/10.1145/2739482.2768505>
 - [24] Mohamed Wiem Mkaouer, Marouane Kessentini, Slim Bechikh, Kalyanmoy Deb, and Mel Cinnide. 2014. High dimensional search-based software engineering: finding tradeoffs among 15 objectives for automating software refactoring using NSGA-III. ACM Press, 1263–1270. <https://doi.org/10.1145/2576768.2598366>
 - [25] H. K. Singh, A. Isaacs, and T. Ray. 2011. A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation* 15, 4 (Aug. 2011), 539–556. <https://doi.org/10.1109/TEVC.2010.2093579>
 - [26] Lee Spector. 2012. Assessment of problem modality by differential performance of lexicase selection in genetic programming: a preliminary report. In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference companion*. 401–408. <http://dl.acm.org/citation.cfm?id=2330846>
 - [27] Tobias Wagner, Nicola Beume, and Boris Naujoks. 2007. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In *Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, 742–756. http://link.springer.com/chapter/10.1007/978-3-540-70928-2_56 DOI: 10.1007/978-3-540-70928-2_56.
 - [28] Ing Wang, Student Member, and Xin Yao. [n. d.]. *Corner Sort for Pareto-based Many-Objective Optimization*.
 - [29] Eckart Zitzler and Simon Knzli. 2004. Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature*. Springer, 832–842.