


A Double Lexicase Selection Operator for Bloat Control in Evolutionary Feature Construction for Regression

Hengzhe Zhang

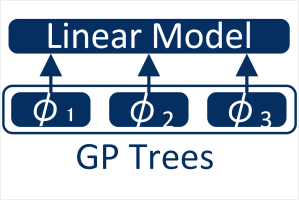
Victoria University of Wellington

22/06/2023

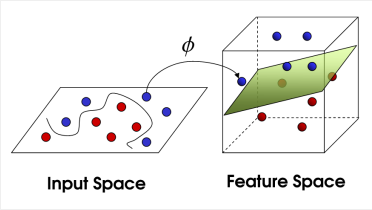
Table of Contents

Background

- The general idea of feature construction is to construct a set of new features $\{\phi_1, \ldots, \phi_m\}$ to improve the learning performance on a given dataset $\{\{x_1, y_1\}, \ldots, \{x_n, y_n\}\}$ compared to learning on the original features $\{x^1, \ldots, x^p\}$.
- Genetic programming (GP) has been widely used to automatically construct features because of its flexible representation and gradient-free search mechanism.



(a) Feature Construction on Linear Regression



(b) New Feature Space

Bloat refers to the tendency of GP solutions to become more complex over time without improving the fitness value.

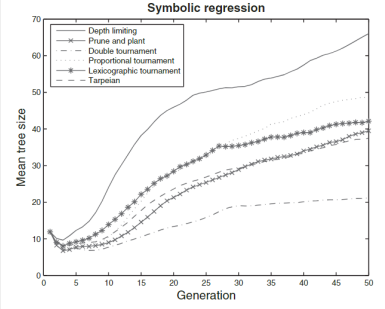


Figure: Growth of program size

The explanations for bloat include:

- Hitchhiking
- Defense against crossover
- Removal bias
- The nature of a program search space

Regardless of the reason for bloat, it is widely acknowledged that solving bloat can increase search efficiency and improves the interpretability of the final model.

- Depth Limit: Set a strict depth limit for each GP tree.
- Variation Operator
  - Prune-and-Plant (PAP)
  - Semantic Approximation (SA)
- **Selection Operator**
  - **Double Tournament Selection (DTS)**
  - **Semantic Tournament Selection (TS-S)**
- Fitness Function
  - Tarpeian
  - Alpha-Dominance MOGP

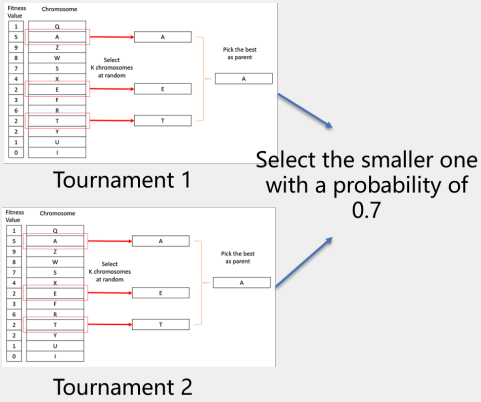


Figure: Double Tournament Selection

Double Stages of DTS:

- Stage 1: Tournament selection, get individuals A, B
- Stage 2: Select the smaller one in A,B with a probability of 0.7

Advantage:

- Applicable for every scenario (GPSR, GPHH)

Disadvantage:

- May lead to worse diversity, given that the tournament selection operator is repeatedly used. -> Lexicase Selection!

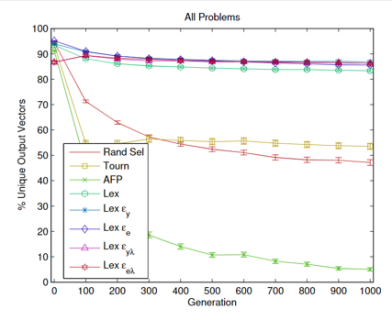


Figure: Tournament vs Lexicase

- Tournament Selection produces a lot of semantically equivalent individuals
- Lexicase Selection preserves a very good population diversity

Why lexicase selection?

- It is not necessary to sum up all errors as a scalar for EA methods.
  $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$

How to perform lexicase selection?

- Step 1: Construct a filter
  $MAD\left(\mathbf{e}_t\right) = \lambda\left(\mathbf{e}_t\right) = \mathrm{median}_j\left(\left|\mathbf{e}_{t_j} - \mathrm{median}_k\left(\mathbf{e}_{t_k}\right)\right|\right)$
- Step 2: Remove bad individuals based on the filter
- Step 3: Construct more filters until only one individual remains

Algorithm

Double Lexicase Selection

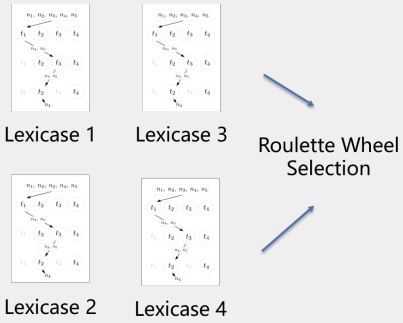


Figure: Double Lexicase Selection

Double Stages of DLS:

- Stage 1: Lexicase selection, get *k* individuals A, B, C, D, …to form a candidate pool
- Stage 2: Roulette Wheel Selection on *k* individuals negatively proportionate to tree size

Advantage:

- Applicable for every scenario (GPSR, GPHH)
- Fully exploit semantics because of using the lexicase selection operator

Experimental Settings

- 98 Regression datasets are used in the experiments, which are all datasets in PMLB with less than 2000 data items.
- The size of datasets range between 47 and 1059, and the dimension of datasets are between 2 and 124.

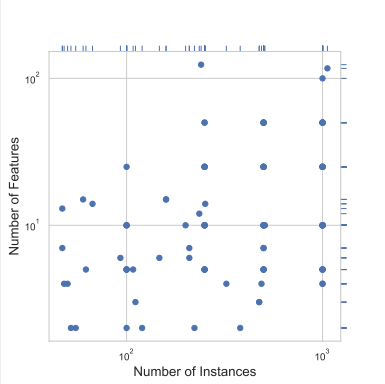


Figure: Properties of experimental datasets

Seven baseline bloat control methods:

- Depth Limit
- Double Tournament Selection (DTS) (ECJ 2006)
- Tarpeian (EuroGP 2003)
- Prune and plant (PAP) (ECJ 2010)
- TS-S (Inf. Sci. 2018)
- DSA (Swarm Evol. Comput. 2020)
- α-MOGP (TEVC) (TEVC 2022)

Parameter settings are common settings in GP.

Parameter	Value
Population Size	1000
Maximal Number of Generations	100
Crossover and Mutation Rates	0.9 and 0.1
Maximum Tree Depth	10
Maximum Initial Tree Depth	6
Number of Trees in An Individual	10
Functions	+, -, *, AQ, Sin, Cos, Abs, Max, Min, Negative

Figure: Parameter settings for all experiments.

Experimental Results

- Only DLS, DSA and Tarpeian methods have similar or better performance on most datasets compared to the depth limit method.
- For the top three algorithms, the DLS method is better than the Tarpeian and DSA methods.

	αMOGP	Tarpeian	DTS	PAP	TS-S	DSA	DepthLimit
DLS	13(+)/83(∼)/2(-)	8(+)/90(∼)/0(-)	37(+)/58(∼)/3(-)	45(+)/49(∼)/4(-)	46(+)/41(∼)/11(-)	7(+)/91(∼)/0(-)	14(+)/79(∼)/5(-)
αMOGP	–	4(+)/87(∼)/7(-)	35(+)/55(∼)/8(-)	44(+)/45(∼)/9(-)	38(+)/45(∼)/15(-)	7(+)/88(∼)/3(-)	4(+)/85(∼)/9(-)
Tarpeian	–	–	35(+)/60(∼)/3(-)	42(+)/50(∼)/6(-)	39(+)/48(∼)/11(-)	4(+)/94(∼)/0(-)	7(+)/90(∼)/1(-)
DTS	–	–	–	18(+)/76(∼)/4(-)	7(+)/75(∼)/16(-)	1(+)/71(∼)/26(-)	13(+)/50(∼)/35(-)
PAP	–	–	–	–	4(+)/71(∼)/23(-)	1(+)/62(∼)/35(-)	10(+)/48(∼)/40(-)
TS-S	–	–	–	–	–	8(+)/65(∼)/25(-)	19(+)/40(∼)/39(-)
DSA	–	–	–	–	–	–	7(+)/82(∼)/9(-)

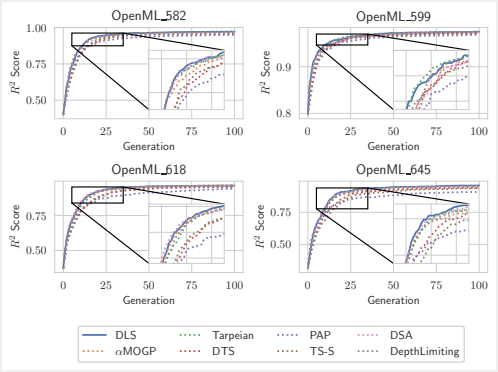
Figure: Statistical comparison of **test** R^2 **score** for different bloat control methods. ("+","∼", and "-" indicate using the method in a row is better than, similar to or worse than using the method in a column.)

- DLS is a successful bloat control method, as it reduces model sizes on all datasets.
- When comparing the PAP, DSA, DTS and TS-S operators, the DLS operator is worse at reducing model size.
- However, DLS is better than the PAP, DSA, DTS and TS-S operators in terms of test R^2 scores.

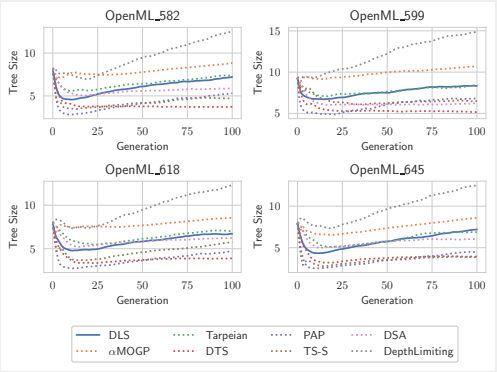
	αMOGP	Tarpeian	DTS	PAP	TS-S	DSA	DepthLimit
DLS	86(+)/12(∼)/0(-)	19(+)/78(∼)/1(-)	2(+)/9(∼)/87(-)	11(+)/45(∼)/42(-)	0(+)/10(∼)/88(-)	11(+)/38(∼)/49(-)	98(+)/0(∼)/0(-)
αMOGP	–	0(+)/11(∼)/87(-)	0(+)/0(∼)/98(-)	0(+)/27(∼)/71(-)	0(+)/0(∼)/98(-)	0(+)/1(∼)/97(-)	74(+)/19(∼)/5(-)
Tarpeian	–	–	0(+)/6(∼)/92(-)	13(+)/41(∼)/44(-)	0(+)/10(∼)/88(-)	2(+)/43(∼)/53(-)	98(+)/0(∼)/0(-)
DTS	–	–	–	54(+)/28(∼)/16(-)	31(+)/19(∼)/48(-)	87(+)/11(∼)/0(-)	98(+)/0(∼)/0(-)
PAP	–	–	–	–	8(+)/22(∼)/68(-)	43(+)/17(∼)/38(-)	98(+)/0(∼)/0(-)
TS-S	–	–	–	–	–	81(+)/14(∼)/3(-)	98(+)/0(∼)/0(-)
DSA	–	–	–	–	–	–	98(+)/0(∼)/0(-)

Figure: Statistical comparison of **model sizes** for different bloat control methods. ("+","∼", and "-" indicate using the method in a row is better than, similar to or worse than using the method in a column.)

- DLS operator has good effectiveness in terms of R^2 scores over the whole evolution process and thus achieves good final accuracy.
- Depth limit cannot effectively control tree sizes. In contrast, the DLS operator can effectively control tree size to a relatively low level.



(a) Evolutionary plots of test R^2 score for different bloat control methods.



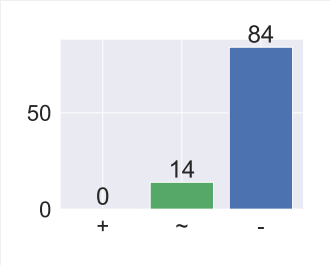
(b) Evolutionary plots of average tree sizes for different bloat control methods.

Algorithm	R^2 Score Rank	Size Rank
DLS	3.43 (1)	4.71 (5)
Tarpeian	3.93 (2)	4.92 (6)
DSA	4.16 (3)	3.9 (4)
αMOGP	4.22 (4)	7.06 (7)
DepthLimit	4.29 (5)	7.88 (8)
DTS	5.03 (6)	2.09 (2)
TS-S	5.15 (7)	1.7 (1)
PAP	5.79 (8)	3.74 (3)

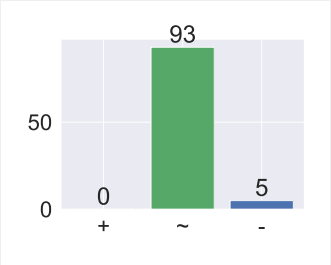
Figure: Friedman's rank of test R^2 scores and tree sizes on all datasets for different bloat control methods.

- Only four methods, DLS, Tarpeian, DSA and αMOGP do not get worse predictive performance on R^2 scores than using depth-limited methods alone.
- DLS operator achieves a good trade-off between test R^2 scores and model size.

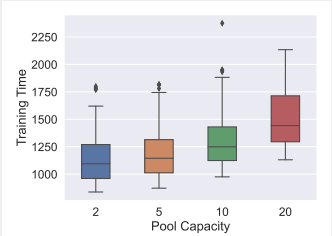
- Model sizes decrease with an increase in capacity.
- Training time will increase significantly when increasing pool capacity from 10 to 20.



(a) Statistical comparison on tree size between using a capacity of 10 and 2.

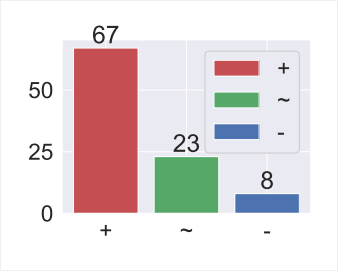


(b) Statistical comparison on tree size between using a capacity of 10 and 5.

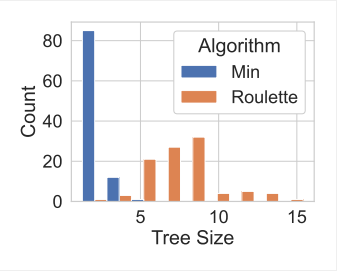


(c) Distribution of training time versus candidate pool capacity.

- Roulette wheel selection operator is significantly better than minimum selection operator in terms of test R^2 scores.
- The minimum selection operator favors very small individuals and thus leads to very poor predictive performance.



(a) Statistical comparison of R^2 scores using roulette instead of minimum as the selection strategy.



(b) Distribution of tree sizes when using roulette or minimum as the selection strategy.

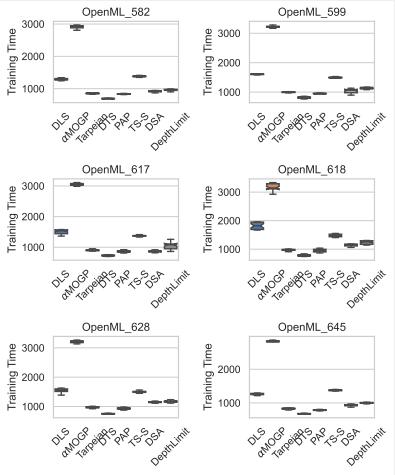


Figure: Box plots on the training time for different bloat control methods.

- The time complexity of DLS is $O(Cap|P|^2n)$, where $|P|$ is the population size, n is the number of data items and Cap is the capacity of the candidate pool.
- DLS is slower than Tarpeian, DTS, PAP, DSA and depth limit methods, but faster than αMOGP.

Compatibility with Multi-objective Methods

- There is no significant difference between integration DLS or not in terms of R^2 scores
- The DLS operator can significantly reduce model sizes on nearly all datasets

	SPEA-II+DLS	NSGA-II	SPEA-II
NSGA-II+DLS	2(+)/89(~)/7(-)	9(+)/81(~)/8(-)	4(+)/80(~)/14(-)
SPEA-II+DLS	–	6(+)/87(~)/5(-)	4(+)/89(~)/5(-)
NSGA-II	–	–	3(+)/91(~)/4(-)

(a) Statistical comparison of test R^2 scores for integration with DLS and MO methods.

	SPEA-II+DLS	NSGA-II	SPEA-II
NSGA-II+DLS	1(+)/86(~)/11(-)	97(+)/1(~)/0(-)	95(+)/3(~)/0(-)
SPEA-II+DLS	–	97(+)/1(~)/0(-)	97(+)/1(~)/0(-)
NSGA-II	–	–	1(+)/95(~)/2(-)

(b) Statistical comparison of model sizes for integration with DLS and MO methods.

Thanks for listening!

Email: Hengzhe.zhang@ecs.vuw.ac.nz

GitHub Project: https://github.com/hengzhe-zhang/DoubleLexicaseSelection