A Comparison of Structural Complexity Metrics for Explainable Genetic Programming

Karina Brotto Rebuli
Department of Veterinary Sciences,
University of Torino, Italy
karina.brottorebuli@unito.it

Mario Giacobini
Department of Veterinary Sciences,
University of Torino, Italy
mario.giacobini@unito.it

Sara Silva
LASIGE, Department of Informatics,
Faculty of Sciences,
University of Lisbon, Portugal
sara@fc.ul.pt

Leonardo Vanneschi
NOVA Information Management School,
Universidade Nova de Lisboa,
Lisbon, Portugal
lvanneschi@novaims.unl.pt

ABSTRACT

Genetic Programming (GP) has the potential to generate intrinsically explainable models. Despite that, in practice, this potential is not fully achieved because the solutions usually grow too much during the evolution. The excessive growth together with the functional and structural complexity of the solutions increase the computational cost and the risk of overfitting. Thus, many approaches have been developed to prevent the solutions to grow excessively in GP. However, it is still an open question how these approaches can be used for improving the interpretability of the models. This article presents an empirical study of eight structural complexity metrics that have been used as evaluation criteria in multi-objective optimisation. Tree depth, size, visitation length, number of unique features, a proxy for human interpretability, number of operators, number of non-linear operators and number of consecutive non-linear operators were tested. The results show that potentially the best approach for generating good interpretable GP models is to use the combination of more than one structural complexity metric.

CCS CONCEPTS

• **General and reference → Metrics**; Empirical studies; • **Theory of computation → Genetic programming**.

KEYWORDS

explainable AI, interpretable models, complexity metrics

ACM Reference Format:
Karina Brotto Rebuli, Mario Giacobini, Sara Silva, and Leonardo Vanneschi. 2023. A Comparison of Structural Complexity Metrics for Explainable Genetic Programming. In *Genetic and Evolutionary Computation Conference Companion (GECCO '23 Companion), July 15–19, 2023, Lisbon, Portugal.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3583133.3590595

1 INTRODUCTION

Machine Learning (ML) models are known to have a high ability to capture non-linear patterns in data. To do that, they can easily get overly complex. When this happens, they become prone to overfitting, require extra computational power, and cannot be understood by humans. When there is a lack of knowledge on how the models transform the input data to generate the output prediction, they are called black-box, or opaque, models. This is the case for several ML techniques, for instance, deep learning neural networks [1]. On the contrary, models that can be understood by humans are called white-box, or transparent, models. Transparent models are not only more reliable [18], they also have the potential of generating knowledge by discovering unknown rules among the features [10] and of revealing errors in the input data [15]. Explainable models can be *intrinsically explainable*, when the model itself can be understood by humans, or *post-hoc explainable*, when the predictions of the opaque model are themselves modelled by a surrogate transparent model. Following the nomenclature in [11], we refer to intrinsically explainable models as interpretable models.

Tree-based Genetic Programming (GP) [5] evolves a direct representation model that usually does not require any coding-decoding phase. Thus, it has the potential of being an interpretable technique. However, in practice, this potential is not fully achieved because the trees typically tend to grow during the evolution of the algorithm. Many works have been proposed to control the increasing of the size and complexity of GP solutions (see Sec. 2). However, interpretability is not achieved only by controlling the size or the mathematical complexity of the function.

The present empirical study aimed at understanding how the existing techniques to control the structural complexity of the GP solutions can improve their interpretability. Eight structural complexity metrics were studied: depth, size, number of operators, number of non-arithmetic operators, number of consecutive non-arithmetic operators, proxy for human interpretability [17], visitation length and number of unique features. They were used, together with the train error, as optimisation criteria in multi-objective problems.

The manuscript is organised as follows: Section 2 presents how structural complexity has been addressed in GP models; Section 3 presents the experimental study conducted with these methods; Section 4 presents and discusses the results of the experiments;

GECCO '23 Companion, July 15–19, 2023, Lisbon, Portugal
© 2023 Copyright is held by the owner/author(s).
ACM ISBN 978-8-4007-0120-7/23/07.
https://doi.org/10.1145/3583133.3590595

finally, Section 5 concludes the manuscript with the main remarks and ideas for future work.

2 COMPLEXITY OF GP MODELS

The exact definition of complexity depends on its context of application. In ML field, it usually refers to the size and non-linearity degree of the model [12, 19]. In GP, two kinds of complexity have been studied, the structural and the functional [7, 8]. This work focus on the strategies that have been proposed to control the structural complexity with multi-objective optimisation, also referred as parsimony pressure. In this case, a second fitness measure is used as an additional criterion for selection. The main approaches found in literature use (i) structural fitness measures, and (ii) complexity scores. They are described hereafter. *Structural fitness measures.* In addition to the size and depth, other metrics for the structure of the trees were proposed in literature. Smits and Kotanchek [14] used as complexity measure the tree depth, the total number of nodes, the total number of features, the number of unique features and the ratio of operator nodes of a tree. Keijzer and Foster [3] proposed a measure called visitation length (λ), defined by:

$$\lambda \ = \ \pi + \zeta + S \tag{1}$$

where π is the internal path length of the tree, the ζ is its external path length and S is its size. It measures the total number of nodes that need to be visited starting at the root node. De Vega et al. [2] proposed to use the computational time of the model as a fitness measure to control bloat [9]. To use it, however, the system and the hardware of the computer must be highly controlled. *Complexity scores.* LaCava et al. [6] proposed a score system in which all operators of the tree receive a value between 1 and 4 according to the mathematical complexity of the operators. The final score is given by the sum of the scores of all internal nodes of the solution. Kommenda et al. [4] proposed a score system that assess also the complexity of the terminal nodes of the trees. It first assigns to terminal nodes 1 if they are a constant or 2 if they are a feature of the dataset. Then, the internal nodes are assigned with the value resulting from the combination of the complexity scores of all of its branches. In their experiments, tree size and visitation length generated smaller models. Virgolin et al. [17] proposed a non-arbitrary score system, built with a linear model that defines the relationship between the tree structure and the complexity of its mathematical expression. The proposed model, called Proxy for Human Interpretabily (ϕ), was defined by:

$$\phi(s, no, nao, naoc) \ = \ 79.2 - 0.2S - 0.5no - 3.4nao - 4.5naoc \tag{2}$$

In this score system, the only distinction among the operators of a solution is with regard to its arithmetic or non-arithmetic nature. Besides that, it does not consider the number of features.

Most of the mentioned publications focus on limiting the growth of the solutions and were engineered to control bloat. Potentially, all these strategies increase the interpretability of the models. However, this is not guaranteed because although the interpretability, size (or depth) and complexity of the models are related, they do not necessarily imply one another. It is still an open question the advantages and drawbacks of these approaches for improving the interpretability of GP. The present empirical study is a first investigation on how some of the most important the existing techniques to control the structural complexity of GP solutions can improve the interpretability of the models.

3 EXPERIMENTAL STUDY

The following structural complexity metrics were considered in the experimental study: tree depth (d), size (S), visitation length (λ), number of unique features (θ), the proxy for human interpretability [17] (ϕ), number of operators (no), number of non-arithmetic operators (nao), and number of consecutive non-arithmetic operators ($naoc$). The experiments used a multi-objective GP with Nested Tournament (NT) selection method [16]. NT has one tournament for each objective to be optimised. All tournaments had size 2. The first tournament used as selection criterion the Root Mean Squared Error (RMSE), $RMSE = \frac{1}{M}\sqrt{\sum_{i=1}^{M}(y_i - \bar{y_i})^2}$, where M is the number of observations, y_i is the i^{th} target and $\bar{y_i}$ is the tree output for the i^{th} observation. The extra tournament(s) used one of the above mentioned complexity metrics. Two experiments were performed with more than two optimisation criteria: one using RMSE, size and unique features (denoted by $S\theta$) and another using the RMSE and ϕ components (S, no, nao and $naoc$), denoted by ϕC. The elitism was applied combining an archive of the best individuals for each objective and the Euclidean distance to the ideal candidate solution. The ideal candidate individual is a point in the fitness space that corresponds to the best value found so far for each objective. The elite individual is the closest individual from this ideal one, and the worst individual is the farthest individual from it.

The primitive functions used to build the GP-trees were +, −, ×, sin, cos, power of 2, $\sqrt{}$ and protected ÷ (denominator equals to zero replaced by the constant 1.0×10^{-6}). The terminal set was composed by ephemeral constants from $]-1, 1[$, in addition to the dataset features. Trees were initialised with the ramped-half-and-half method [5], with trees maximum initial depth equals 2 and maximum depth equals 7. The population size was 200 and the number of generations, 500. The probability of crossover and mutation were respectively equals to 0.7 and 0.2. Each experiment was run 30 times with random data partitioning, using 70% of the data for training and 30% for test. The statistical significance of the results were tested with Wilcoxon signed-rank test for two-samples and with Wilcoxon paired test for multiple samples comparisons [13] with Bonferroni correction and level of significance $\alpha = 0.05$. Tables with all p-values are in the Supplementary Material[1] (SM).

Six public[2] real-data datasets were used: *Boston*, with 506 instances and 13 attributes; *ENB Cooling,* and *ENB Heating* with 768 instances and 8 attributes; *Airfoil*, with 1503 instances and 5 attributes; *RB Cost* and *RB Sales*, with 372 instances and 107 attributes. The ENB Cooling and ENB Heating have the same features, but different targets. The same for the RB Cost and RB Sales.

4 RESULTS AND DISCUSSION

4.1 Effect of the metrics on the predictive ability of the models

The first concern on generating less complex models relies on if it keeps its predictive ability. The effect of the parsimony selection

[1] Supplementary material available at https://bit.ly/3XdKDEY.
[2] OpenML https://www.openml.org/.

on the RMSE of the test partition was dataset-dependent. Thus, it was not possible to observe a pattern in the effect of using the complexity metrics on the predictive ability of GP. Tables SM 1 to 6 present the p-values for the tests of the differences between the mean test error using only RMSE versus using RMSE and the other complexity metrics on each dataset.

For the Boston dataset, only the d worsened the RMSE of the test partition. For the ENB Cooling and ENB Heating, all complexity metrics worsened the RMSE of the test partition, except the *nao* and *naoc*. For the Airfoil, the *no*, S and λ worsened the RMSE of the test partition. For RB Cost and RB Sales, none of the complexity metrics worsened the RMSE of the test partition.

4.2 Effect of the metrics on the generalisation ability of the models

In addition to maintaining the predictive ability, it is important to verify whether the algorithms maintain their generalisation ability. This can be inferred by the difference between train and test errors. A much higher test error than a train error is an indication that the algorithm is overfitting. Table 1 shows the mean RMSE in train and test partitions for each experiment and dataset. The p-values are in Table SM 7. In general, the $S\theta$ had the best effect on controlling the overfitting of the algorithms. The ϕ and λ were also good.

For the Boston dataset, the mean test error was larger than the mean train error in all experiments. On the other hand, for the RB Sales and RB Cost, the mean test error was smaller than the mean train error in all experiments. This indicates that for these datasets the use the parsimony selection did not have a significant effect in the generalisation ability of the model. For the ENB Cooling dataset, the p-value for the difference between the train and test errors was not significant. For the ENB Heateing dataset, the mean test error was larger than the mean train error when using only the RMSE. For both datasets, all solutions using more than one optmisation criterium had the mean test error larger than the mean train error, except the ϕ, $S\theta$ and λ solutions. For the Airfoil dataset, the test error was larger than the train error using the RMSE and *naoc*. The use of the other metrics improved the generalisation ability of the algorithm, as with them the test error was equals or smaller than the train error.

4.3 Effect of the metrics on the interpretability of the solutions

Interpretability is not directly measurable and it depends on the use of the model. In this Section, the structure of the best solutions in terms of train error are compared. As we were interested in *intrinsically* understandable models, no simplification was considered. Besides the structure of the solutions, two other criteria were used for their comparisons: the train error and the use of frequent features among the best solutions for the respective dataset. The analysed solutions are presented in Table SM 8.

For the Boston dataset, all best solutions used exclusively the dataset feature f_5, except the λ solution, which was composed just by the f_{10} feature. This indicates that f_5 is the most important feature of this dataset. The solutions generated with *no* and $S\theta$ were considered the best in terms of interpretability. Their train errors are at the same scale of the train errors of the other analysed solutions, they are small and their mathematical expression is simple, as they do not contain any trigonometric operators. For the ENB Cooling dataset, all best solutions used the dataset feature f_4, except the $S\theta$ solution, which did not use any dataset feature. The *no* solution was considered the best because it is very small, simple and, in addition to the feature f_4, it uses the f_1 and f_3. Both were also used by the RMSE solution, the best solution in terms of train error. Thus, the *no* solution likely incorporated relevant information available in the dataset. For the ENB Heating dataset, all best solutions used the feature f_4, except the ϕ and $S\theta$, which did not use any feature. The ϕC was selected as the best by the same reasons of the *no* solution for the Boston dataset. For the Airfoil dataset, the solutions generated by S, $S\theta$ and λ did not contain any dataset feature. The ϕC was chosen as the best because it is small, mathematically simple, and it uses two dataset features. The use of two features is an advantage in this dataset with only 5 features and for which many solutions did not use any of them. For the RB Cost dataset, all best solutions used the f_{11}, despite the large number of features of this dataset (107). The solutions with the simpler structure were the ϕC, d, *no*, S and λ. The ϕC was chosen as the best because its structure is small and simple, and it has the smallest train error after the RMSE, *nao* and *naoc* solutions. For the RB Sales dataset, all best solutions used the f_{11}, as observed in the RB Cost dataset. Among them, the ϕC and the *no* were the best in terms of interpretability. The ϕC is bigger, but it only uses the sum and subtraction operators.

All best solutions generated with *nao* and *naoc* were too big to be analysed. However, the differences between their train errors and the train errors of the RMSE solution were small. Indeed, their train error were smaller than the train error of the RMSE solution for the RB Cost ans RB Sales datasets. Only *no* and λ criteria generated solutions with size < 20 for all datasets, but only the *no* solution was selected as the best (Boston, ENB Cooling and RB Sales datasets). Most of the solutions selected as the best in term of interpretability were generated by a composition of more than one structural complexity metric. The $S\theta$ was selected for the Boston dataset and the ϕC was selected for the ENB Heating, Airfoil, RB Cost and RB Sales datasets. Interestingly, the ϕ and ϕC used the same criteria, but the former used by them combined in the ϕ formula and the latter used them separated. It is not possible to know it the best performance of the ϕC in comparison to the ϕ is due to the preservation of the multi-objective nature of the problem or if a formula of ϕ can work, but it needs to be improved.

5 CONCLUSIONS

The need for Machine Learning models whose predictions and structure can be understood by humans has increased with the use of these models by society. Tree-based Genetic Programming (GP) has the potential of generating intrinsically explainable, called interpretable, models as its solutions usually do not require any coding-decoding phase. However, this potential is frequently not fulfilled because the solutions tend to grow too much during the evolution. Many structural complexity metrics have been proposed in the literature to be used in parsimony selection for generating smaller and less complex GP models. However, although interpretability, size and complexity of the models are related, they do not necessarily imply one another. Thus, it is not guaranteed that these approaches will increase the interpretability of the models.

Table 1: Train and test mean RMSE for each experiment and dataset. Bold pairs are significantly different ($\alpha = 0.05$, with Bonferroni correction for multiple comparisons). Values with (*) indicate the cases in which the error in the test partition is smaller than the error in train partition.

	Boston		ENB Cooling		ENB Heating		Airfoil		RB Cost		RB Sales	
	train	test	train	test	train	test	train	test	train	test	train	test
RMSE	**3.17**	**24.89**	2.33	3.17	**2.57**	**3.74**	**4.79**	**6.47**	**199.51**	**132.43***	**205.57**	**149.48***
ϕC	**6.53**	**12.45**	**4.92**	**6.71**	**5.01**	**6.83**	7.09	6.67	**286.32**	**184.07***	**283.04**	**197.85***
d	**4.42**	**12.57**	**4.88**	**6.87**	**5.23**	**6.72**	**7.04**	**6.74***	**345.03**	**132.73***	**345.32**	**137.25***
θ	**4.54**	**12.09**	**4.66**	**6.19**	**4.65**	**6.35**	6.48	6.43	**409.03**	**194.12***	**421.45**	**190.91***
nao	**3.19**	**221.09**	**2.81**	**4.17**	**2.91**	**3.58**	5.63	6.60	**175.41**	**119.37***	**172.21**	**115.23***
$naoc$	**3.13**	**13.82**	**2.21**	**3.02**	**2.28**	**3.22**	**5.08**	**6.38**	**175.10**	**145.43***	**180.53**	**126.11***
no	**4.94**	**12.55**	**5.22**	**6.59**	**5.14**	**6.68**	13.12	13.44	**339.35**	**139.30***	**338.43**	**135.63***
ϕ	**8.80**	**13.55**	7.81	9.26	8.07	10.12	12.72	12.56	**333.01**	**140.45***	**339.87**	**172.00***
S	**5.77**	**13.03**	**5.75**	**7.60**	**6.05**	**7.95**	9.64	9.70	**343.11**	**136.80***	**343.14**	**139.65***
$S\theta$	**6.27**	**12.89**	6.58	8.17	5.94	7.52	**6.84**	**6.48***	**409.50**	**151.99***	**404.93**	**157.02***
λ	**7.11**	**11.71**	6.97	8.45	6.25	8.44	15.63	15.49	**345.81**	**153.23***	**343.73**	**131.16***

The present empirical study investigated how some of the most important existing metrics for controlling the structural complexity of GP solutions affect the interpretability of the models. Tree depth, number of unique features, number of operators (*no*), number of non-arithmetic operators (*nao*), number of consecutive non-arithmetic operators (*naoc*), size and a proxy for human interpretability (ϕ) were tested with multi-objective problems. In general, the use of *nao* and *naoc* individually did not have any effect in the generated models. For the other metrics, the effect of their use was dataset-dependent in all results. Importantly, the metrics not always worsened the prediction ability of the models. In general, the most interpretable solutions were generated with the components of ϕ used separately. The *no* and the $S\theta$ also generated good solutions in terms of interpretability. As two of the experiments with the best results apply more than one metric as optimisation criteria, this can indicate that the combination of the structural complexity metrics is a good strategy for generating interpretable GP models with the parsimony selection approach. In fact, this makes sense, as the interpretability of the models is a multi-factorial concept. Thus, further studies on the combination of the structural complexity metrics in parsimony pressure are promising.

ACKNOWLEDGMENTS

This work was partially supported by FCT, Portugal, through funding of research units MagIC/NOVA IMS (UIDB/04152/2020) and LASIGE (UIDB/00408/2020 and UIDP/00408/2020).

REFERENCES

[1] Vanessa Buhrmester, David Münch, and Michael Arens. 2021. Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey. *Machine Learning and Knowledge Extraction* 3, 4 (2021), 966–989. https://doi.org/10.3390/make3040048

[2] Francisco Fernández de Vega, Gustavo Olague, Francisco Chávez, Daniel Lanza, Wolfgang Banzhaf, and Erik Goodman. 2020. *It Is Time for New Perspectives on How to Fight Bloat in GP.* Springer International Publishing, Cham, 25–38. https://doi.org/10.1007/978-3-030-39958-0_2

[3] Maarten Keijzer and James Foster. 2007. Crossover Bias in Genetic Programming. In *Genetic Programming*, Marc Ebner, Michael O'Neill, Aniko Ekárt, Leonardo Vanneschi, and Anna Isabel Esparcia-Alcázar (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 33–44.

[4] Michael Kommenda, Gabriel Kronberger, Michael Affenzeller, Stephan M. Winkler, and Bogdan Burlacu. 2016. *Evolving Simple Symbolic Regression Models by Multi-Objective Genetic Programming.* Springer International Publishing, Cham, 1–19. https://doi.org/10.1007/978-3-319-34223-8_1

[5] John R. Koza. 1992. *Genetic programming: On the programming of computers by means of natural selection.* MIT Press.

[6] William La Cava, Kourosh Danai, and Lee Spector. 2016. Inference of compact nonlinear dynamic models by epigenetic local search. *Engineering Applications of Artificial Intelligence* 55 (2016), 292–306. https://doi.org/10.1016/j.engappai.2016.07.004

[7] Nam Le, Hoai Nguyen Xuan, Anthony Brabazon, and Thuong Pham Thi. 2016. Complexity measures in Genetic Programming learning: A brief review. In *2016 IEEE Congress on Evolutionary Computation (CEC).* 2409–2416. https://doi.org/10.1109/CEC.2016.7744087

[8] Yi Mei, Qi Chen, Andrew Lensen, Bing Xue, and Mengjie Zhang. 2022. Explainable Artificial Intelligence by Genetic Programming: A Survey. *IEEE Transactions on Evolutionary Computation* (2022), 1–1. https://doi.org/10.1109/TEVC.2022.3225509

[9] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. 2008. *A field guide to genetic programming.* Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`. (With contributions by J. R. Koza).

[10] Piotr Przybyła and Axel J. Soto. 2021. When classification accuracy is not enough: Explaining news credibility assessment. *Information Processing and Management* 58, 5 (2021), 102653. https://doi.org/10.1016/j.ipm.2021.102653

[11] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (2019), 206–215. https://doi.org/10.1038/s42256-019-0048-x

[12] Patrick Schneider and Fatos Xhafa. 2022. *Anomaly Detection and Complex Event Processing Over IoT Data Streams.* Academic Press.

[13] David J. Sheskin. 2011. *Handbook of Parametric and Nonparametric Statistical Procedures.* Chapman and Hall.

[14] Guido F. Smits and Mark Kotanchek. 2005. *Pareto-Front Exploitation in Symbolic Regression.* Springer US, Boston, MA, 283–299. https://doi.org/10.1007/0-387-23254-0_17

[15] Rita T Sousa, Sara Silva, and Catia Pesquita. 2022. Explaining Protein-Protein Interaction Predictions with Genetic Programming, Kalyanmoy Deb (Ed.). *Late-breaking abstracts from EuroGP Conference*, 30–33.

[16] Leonardo Vanneschi, Mauro Castelli, Kristen Scott, and Aleš Popovič. 2018. Accurate High Performance Concrete Prediction with an Alignment-Based Genetic Programming System. *International Journal of Concrete Structures and Materials* 12, 1 (2018), 72. https://doi.org/10.1186/s40069-018-0300-5

[17] Marco Virgolin, Andrea De Lorenzo, Eric Medvet, and Francesca Randone. 2020. Learning a Formula of Interpretability to Learn Interpretable Formulas. In *Parallel Problem Solving from Nature – PPSN XVI*, Thomas Back, Mike Preuss, Andre Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann (Eds.). Springer International Publishing, Cham, 79–93.

[18] Leander Weber, Sebastian Lapuschkin, Alexander Binder, and Wojciech Samek. 2023. Beyond explaining: Opportunities and challenges of XAI-based model improvement. *Information Fusion* 92 (2023), 154–176. https://doi.org/10.1016/j.inffus.2022.11.013

[19] Chenguang Zhu. 2021. *Machine Reading Comprehension: Algorithms and Practice.* Elsevier.