



HAL
open science

Veni, Vidi, Evolvi commentary on W. B. Langdon's “Jaws 30”

Giovanni Squillero, Alberto Tonda

► **To cite this version:**

Giovanni Squillero, Alberto Tonda. Veni, Vidi, Evolvi commentary on W. B. Langdon's “Jaws 30”. Genetic Programming and Evolvable Machines, 2023, 24 (2), pp.24. 10.1007/s10710-023-09472-0 . hal-04307703

HAL Id: hal-04307703

<https://hal.science/hal-04307703>

Submitted on 26 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Veni, Vidi, Evolvi commentary on W. B. Langdon's "Jaws 30"

Giovanni Squillero¹ · Alberto Tonda^{2,3}

Accepted: 13 October 2023
© The Author(s) 2023

Genetic programming (GP) has been pushing the boundaries of what a computer may achieve in an autonomous way since its introduction [1]. Over the years, John Koza himself tracked some one hundred results that are competitive with human-produced ones in a wide variety of fields,¹ and success stories have been steadily published by both scholars and practitioners in the specialized literature. However, we cannot help noticing that today GP is largely underutilized in the real-world domains where it was originally supposed to excel. Artificial intelligence is considered a core technology of the fourth industrial revolution (4IR, or Industry 4.0), but, while machine learning (ML) is explicitly mentioned, there is little doubt that the term refers to statistical models and neural networks, not GP nor other evolutionary algorithms.

Regression is a paradigmatic example of this trend. In the early 1990s, researchers excitedly demonstrated the GP's ability to evolve mathematical functions that could fit to a set of data, but after 20 years, deep neural networks are showing competitive performances [2]—the two winners of the GECCO22 SRBench competition on inter-pretable symbolic regression for data science² do not exploit GP, nor do

¹ <https://www.human-competitive.org/awards>.

² <https://cavalab.org/srbench/competition-2022/>.

Giovanni Squillero and Alberto Tonda have contributed equally to this work.

This comment refers to the article available at <https://doi.org/10.1007/s10710-023-09467-x>.

Special Issue: Thirtieth Anniversary of Genetic Programming: On the Programming of Computers by Means of Natural Selection.

✉ Giovanni Squillero
giovanni.squillero@polito.it

Alberto Tonda
alberto.tonda@inrae.fr

¹ Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Turin, Italy

² UMR 518 MIA-PS, INRAE, AgroParisTech, University of Paris-Saclay, 22, Place de L'Agronomie, 91120 Palaiseau, France

³ UAR 3611 CNRS, Institut des Systèmes Complexes de Paris Île-de-France (ISC-PIF), 113, Rue Nationale, 75013 Paris, France

they mention “evolutionary computation” in their descriptions. Nowadays, the most popular algorithms for classification are either ensembles of boosted trees, like XGBoost [3], or, again, deep neural networks [4]. A cursory analysis of the number of *likes* on GitHub repositories provides a rather clear overview of the situation: as of July 22, 2023, one may observe 176k stars for Tensorflow,³ 68k for PyTorch,⁴ and 55k for scikit-learn.⁵ TPOT,⁶ a GP-based optimizer for ML pipelines, scores nearly 10k stars, presumably thanks to the “ML” connection, DEAP,⁷ a library for evolutionary optimization that also includes GP has 5.2k stars, but *jenetics*,⁸ *gplearn*,⁹ *tiny-gp*¹⁰ and the other 7 projects listed when searching for “genetic programming” cumulatively got less than 2k stars.

Here, we would like to draw the readers’ attention to an archetypal topic, although “less explored” as pointed out by prof. Langdon: the creation of *computer programs*. When the goal is to generate fragments frequently coded by humans, such as API calls or common algorithms, the task can be delegated almost safely to neural networks. Non-evolutionary, “AI-powered” tools like ChatGPT, Github Copilot, or Tabnine are practical because of their speed and the reduced amount of meta-parameters that need to be tweaked. Extremely complex models are often available out-of-the-box, already trained, and may be tweaked with reduced effort exploiting transfer-learning techniques; moreover, practitioners are finding clever workarounds, such as the down-casting of the weights for inference, to allow even large models to work on end-user PCs.

However, while deep learning (DL) methodologies have been shown able to efficiently learn from huge amounts of data and interpolate among existing results, GP displayed a unique ability to slowly unfold brand new solutions; and in the generation of never-before-written programs it could be thriving with little to no competition. For example, in the creation of assembly-language programs to test modern microprocessors [5] there are no libraries of already-written solutions and the goal is to create a unique program from scratch, targeting a new hardware design. More broadly, whenever the goal is to devise a test, by definition, one cannot exploit already-existing material, and therefore neural-network models trained on available data are of little use. GP- and other evolutionary-based techniques have been and still are perfectly suited as fuzzer and feedback-based test generators [6, 7]. In another emblematic case study, GP was able to design a novel antenna [8], proving its effectiveness in creating a structure considerably different from human blueprints. A hypothetical generative DL system applied to the same task would be unlikely to uncover such a solution, as the final result fell well outside the distribution of

³ <https://github.com/tensorflow/tensorflow>.

⁴ <https://github.com/pytorch/pytorch>.

⁵ <https://github.com/scikit-learn/scikit-learn>.

⁶ <https://github.com/EpistasisLab/tpot>.

⁷ <https://github.com/DEAP/deap>.

⁸ <https://github.com/jenetics/jenetics>.

⁹ <https://github.com/trevorstephens/gplearn>.

¹⁰ <https://github.com/moshessipper/tinygp>.

samples it could have had observed. Apart from this niche, in some cases GP seems to have followed the old saying “if you cannot defeat them, join them”: While GP cannot compete with ML and DL directly, its inherent characteristics might be used to support them. GP-based neuro- evolution is again at the forefront, with interesting results, close to or better than the state-of-the-art human-designed networks [9–11]; and even for boosted trees, recent attempts at using ensembles of GP trees evolved with a MAP-Elites [12] scheme were able to outperform classical strategies for boosting [13]—not to mention TPOT [14], seen above, by far the GP-based tool with more *stars* on GitHub. In general, evolutionary ML is a rapidly growing sub-field, with dedicated workshops and tracks.

To conclude, we believe that so far GP failed to be adopted by the mainstream community, especially in industrial contexts. However, this is not just due to the relative effectiveness of the different techniques. The DL/ML scholars managed to advertise their successes and coalesce a large community of practitioners around their algorithms. Just like some researchers kept the fire going while the interest in neural networks waned during the 90s and the 2000s, the GP community should keep working to make the world aware of the potentiality of this approach. After 30 years, the future seems ripe for GP applications; moreover, the ML/DL rising wave is creating vast search spaces that need to be effectively explored (neuro-evolution, diversity of boosted trees); and with DL models being complete black boxes, there is a growing need for explainability, a call for symbolic or neuro-symbolic AI, where GP could provide good solutions, especially in areas where DL fails, like the Abstraction and Reasoning Corpus benchmark [15]. A new GP spring may very well be looming on the horizon.

Funding Open access funding provided by Politecnico di Torino within the CRUI-CARE Agreement.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. J. Koza, Genetic programming: a paradigm for genetically breeding populations of computer programs to solve problems. Technical Report STAN-CS-90-1314 (Dept. of Computer Science, Stanford University, 1990)
2. P.-A. Kamienny, G. Lample, S. Lamprier, M. Virgolin, Deep generative symbolic regression with Monte-Carlo-tree-search (2023). 2302.11223
3. T. Chen, C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, *KDD '16* (ACM, New York, 2016), pp.785–794. <https://doi.org/10.1145/2939672.2939785>
4. I. Goodfellow, Y. Bengio, A. Courville, *Deep Learning* (MIT Press, 2016)

5. G. Squillero, Artificial evolution in computer aided design: from the optimization of parameters to the creation of assembly programs. *Computing* **93**, 103–120 (2011)
6. N. Alshahwan et al., *Deploying Search Based Software Engineering with Sapienz at Facebook* (Springer, 2018), pp.3–45
7. S. Gandini, W. Ruzzarin, E. Sanchez, G. Squillero, A. Tonda, A framework for automated detection of power-related software errors in industrial verification processes. *J. Electron. Test.* **26**, 689–697 (2010). <https://doi.org/10.1007/s10836-010-5184-5>
8. J.D. Lohn, G.S. Hornby, D.S. Linden, An evolved antenna for deployment on NASA's space technology 5 mission. *Genet. Program. Theory Pract.* **2**, 301–315 (2005)
9. R. Miikkulainen et al., *Evaluating Medical Aesthetics Treatments Through Evolved Age-Estimation Models* (ACM, 2021). <https://doi.org/10.1145/3449639.3459378>
10. F. Assuncao, N. Lourenco, P. Machado, B. Ribeiro, DENSER: deep evolutionary network structured representation. *Genet. Program. Evol. Mach.* **20**, 5–35 (2018). <https://doi.org/10.1007/s10710-018-9339-y>
11. E. Real et al., in *Large-Scale Evolution of Image Classifiers*, eds. by D. Precup, Y.W. The. *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 of *Proceedings of Machine Learning Research* (PMLR, 2017), pp. 2902–2911. <https://proceedings.mlr.press/v70/real17a.html>.
12. J.-B. Mouret, J. Clune, Illuminating search spaces by mapping elites (2015). <https://arxiv.org/abs/1504.04909>
13. H. Zhang et al., *MAP-Elites with Cosine-Similarity for Evolutionary Ensemble Learning* (Springer, Switzerland, 2023), pp.84–100. <https://doi.org/10.1007/978-3-031-29573-76>
14. R.S. Olson, N. Bartley, R.J. Urbanowicz, J.H. Moore, *Evaluation of a tree-based pipeline optimization tool for automating data science*, GECCO '16 (ACM, New York, 2016), pp. 485–492. <https://doi.org/10.1145/2908812.2908918>
15. F. Chollet, On the measure of intelligence (2019). <https://arxiv.org/abs/1911.01547>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval, sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com