Human Guidance Approaches for the Genetic Improvement of Software

Benjamin J. Craine
School of Computing and Communications
Lancaster University
UK
b.craine@lancaster.ac.uk

Penn Faulkner Rainford
Department of Computer Science
University of York
UK
penn.rainford@york.ac.uk

Barry Porter
School of Computing and Communications
Lancaster University
UK
b.f.porter@lancaster.ac.uk

ABSTRACT

Existing research on Genetic Improvement (GI) of source code to improve performance [10] has examined the mixed application of code synthesis and traditional GI mutation/crossover to gain higher-performing individuals that are tailored to particular deployment contexts, for examples such as hash tables or scheduling algorithms. While demonstrating successful improvements, this research presents a host of challenges [9], from search space size to fitness landscape shape, which raise questions on whether GI alone is able to present a complete solution. In this position paper we propose to *augment* GI processes with Human Guidance (HG) to offer a co-pilot paradigm which may overcome these challenges.

1 INTRODUCTION

HG is a technique to bias algorithms to produce more desirable outputs, and is being explored in multiple generative and synthesis fields [4, 11–13] to bridge shortcomings in various machine learning (ML) applications [5, 8]. GI encounters some of the same issues (over-fitting, dataset quality/availability, emergent behaviour[2, 3, 9]), but as far as we are aware augmenting GI with human guidance has yet to be deeply explored; applying it to the specific domain of mixed code synthesis and traditional GI presents a particularly challenging space with the potential for significant improvements.

HG of ML naturally sits on fuzzy boundaries. At each stage of the development of generative systems, human choices in design, implementation and execution affect what the system has the potential to (and does) generate. When designing systems used to study HG, experimentation should clearly show that HG was a net positive to the system. This requires a system to operate under a HG and non-HG paradigm whilst remaining comparable. An example of such an approach is GPT-3 and its derivative instructGPT [1]. A user of these two systems would view them as operationally equivalent, yet would observe clear differences in their outputs. With the only difference between the two models being that instructGPT's internal parameters were tuned using HG, one could then deduce the difference to be caused by HG. When studying HG's effect on GI we would similarly seek to be able to isolate its precise outcomes.

We propose three research avenues to incorporate HG into GI: HG based on visualisations of the state and search space, where guidance operates on the hyper-parameter level of the Genetic Algorithm (GA), allowing quicker convergence; HG based on curated source code examples to feed to a GI process which, in turn, steer the GI to search around the properties found in those examples; and HG based on direct gene editing where a human co-pilot suggests complex strings of genetic operators to be applied which may allow larger leaps in search space whilst retaining good fitness.

2 HG BASED ON VISUALISATIONS

The goal of HG via novel visualisations is to construct a human-computer interface that provides the human with real-time information on the in-progress genetic search, along with controls to allow human-derived guidance. The visualisations should also show the user the impact that their input has on the search as it progresses.

The challenge of building such a visualisation tool lies in consolidating two main requirements. First, they must be expressive enough to make the most promising search branches apparent to the human. Secondly, the human must be able to interpret the visualised data to make a good decision. We must also find an intuitive way to map the user's intent for the search onto a set of implemented controls we make available, e.g. altering a fitness function.

A visualisation tool may present general GI data, such as fitness landscapes, or domain-specific data, such as a representation of source code, or a combination of these two elements.

An example of visualising general GI data might be a graphical view of the topology of the fitness landscape. By contrast, a tool to present domain-specific data may show clusters of individuals according to a measure of distance that is orthogonal to the fitness landscape (such as code distance, input/output distance, or phylogenetic or phenotype analysis [7, 10]). By presenting selected source-code individuals from such clusters a human may gain unique insights into the search direction which would never be apparent to the GI. These insights may then lead to interventions which alter the search direction using high-level controls.

At least one basis for visualising high dimensional spaces already exists using Euclidean embedding [6]. The open question is whether these types of visualisations lend themselves to our stated goal of allowing humans to glean insights into the GI process, and from this affect positive biases on the process.

3 HG BASED ON CURATED SOURCE CODE

All approaches to program synthesis for general-purpose programming languages face the problem that the search space between some program A and a genetically improved program B is usually massive. This space, even if optimally traversed, would require a long string of specific genetic operations to achieve the transformation from A to B. With the inherent randomness of the GA, getting near this optimal path through fitness-neutral space is unlikely. Furthermore, should such a fitness function exist that guides the

GA close to optimally through the space, that function may promote over-fitting. Therefore, a desirable GA is one that strikes the right balance between exploration of the space serendipitously and exploiting paths the GA is confident will yield good solutions.

One possible avenue to achieve such a desirable GA, using HG, would be to have an experienced engineer curate a set of programs that are distant in space but contain known useful properties. These programs should generally function correctly, and should also represent a range of fitness scores for a particular context. A mathematical relationship could be formed between this set of programs which gives the minimum and expected distance travelled by a GA between those same programs, providing trajectories and distances through neutral space. A key question here is whether such a mathematical correlation or relationship exist between the high-level parameterisation of the GA and the expected distance travelled by the GA between some selection of programs? If so, it may be the case that guiding HG at the parameter level can achieve our goal of navigating neutral space better; otherwise the best options for HG may lean toward more direct forms of intervention. Exploring the use of other ML techniques for navigating search spaces appears may help to explore this potential correlation between the GA parameters and the positive navigation of neutral space.

What we would expect to see from this approach is a reduction in time taken between reaching genetically improved individuals, whilst minimising the time spent exploring entirely neutral areas of space. Key observations of this may include new useful individuals containing fewer neutral source code modifications.

4 HG BASED ON DIRECT GENE EDITING

When synthesising programs belonging to a general purpose language, most of the search space is neutral with regards to fitness evaluation. GAs struggle to navigate this space, seen as the plateau of a fitness of the best individual plotted against the number of generations. If this plateau is caused by the GA travelling through neutral space, there are few ways to predict with confidence that the GA will discover an improved individual in an acceptable time frame. Such plateaus can be caused by other factors, but it is neutral space traversal we wish to address through human guidance.

Perhaps the most direct approach for HG to address the neutral space problem is to have an experienced engineer suggest search areas to the GA in domain-specific terms. This may manifest in many ways, all of which equate to temporarily or partially replacing the stochastic GA elements with a set of genetic modifications to apply, relocating it to space(s) deemed to be promising.

This kind of very direct intervention treads the most fine line between the human providing the answer and the human and GI working in a jointly-useful relationship. The ideal scenario for direct gene editing is that the GI locates an high-utility area of program space that the human engineer would not have considered, and the human engineer is then able to offer additional inspiration to enhance the solution even further. The extent to which these co-pilot relationships manifest, rather than either the GI or human becoming dominant, and while maintaining the utility of the GI to autonomously locate novel solutions, is a key open question.

5 DISCUSSION

A pivotal challenge in investigating all aspects of HG for GI lies in delicately balancing the influence of humans on an independent GI process in a co-pilot relationship. Striking the right balance is essential to ensure that human involvement enhances rather than overrides the GI process, preserving its desirable characteristics. While it's evident that GI alone faces limitations in addressing specific challenges, an open question is how we evaluate the efficacy of different HG approaches. The focus is on identifying scenarios where HG-aided GI becomes the most viable and effective solution.

Our upcoming endeavours involve a comprehensive exploration of the potential applications of HG within our specific research problem domain. This entails navigating large, predominantly neutral search spaces to unearth individuals with heightened performance.

ACKNOWLEDGEMENT

This work was supported by the Leverhulme Trust Research Grant 'Genetic Improvement for Emergent Software', RPG-2022-109.

REFERENCES

[1] Long Ouyang et al. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155 [cs.CL]

[2] Ivo Gonçalves and Sara Silva. 2013. Balancing Learning and Overfitting in Genetic Programming with Interleaved Sampling of Training Data. In *Proceedings of the 16th European Conference on Genetic Programming* (Vienna, Austria) *(EuroGP'13)*. Springer-Verlag, Berlin, Heidelberg, 73–84.

[3] Ivo Gonçalves, Sara Silva, Joana B. Melo, and Joao Carreiras. 2012. Random Sampling Technique for Overfitting Control in Genetic Programming. 218–229.

[4] Kori Inkpen, Shreya Chappidi, Keri Mallari, Besmira Nushi, Divya Ramesh, Pietro Michelucci, Vani Mandava, Libuše Hannah Vepřek, and Gabrielle Quinn. 2023. Advancing Human-AI Complementarity: The Impact of User Expertise and Algorithmic Tuning on Joint Decision Making. *ACM Trans. Comput.-Hum. Interact.* 30, 5, Article 71 (sep 2023), 29 pages.

[5] Harsurinder Kaur, Husanbir Singh Pannu, and Avleen Kaur Malhi. 2019. A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Comput. Surv.* 52, 4, Article 79 (aug 2019), 36 pages.

[6] Krzysztof Michalak. 2019. Low-Dimensional Euclidean Embedding for Visualization of Search Spaces in Combinatorial Optimization. *IEEE Transactions on Evolutionary Computation* 23, 2 (2019), 232–246.

[7] Hirotaka Moriguchi and Shinichi Honiden. 2010. Sustaining Behavioral Diversity in NEAT. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (Portland, Oregon, USA) *(GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 611–618.

[8] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D. Lawrence. 2022. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Comput. Surv.* 55, 6, Article 114 (dec 2022), 29 pages. https://doi.org/10.1145/3533378

[9] Penny Faulkner Rainford and Barry Porter. 2021. Open Challenges in Genetic Improvement for Emergent Software Systems. In *2021 IEEE/ACM International Workshop on Genetic Improvement (GI)*. 43–44.

[10] Penny Faulkner Rainford and Barry Porter. 2022. Using Phylogenetic Analysis to Enhance Genetic Improvement. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 849–857.

[11] Halit Bener Suay and Sonia Chernova. 2011. Effect of human guidance and state space size on Interactive Reinforcement Learning. In *2011 RO-MAN*. 1–6.

[12] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems* 135 (2022), 364–381.

[13] Ruohan Zhang, Faraz Torabi, Garrett Warnell, and Peter Stone. 2021. Recent Advances in Leveraging Human Guidance for Sequential Decision-Making Tasks. *Autonomous Agents and Multi-Agent Systems* 35, 2 (oct 2021), 39 pages.