Genetic Improvement @ ICSE 2023

William B. Langdon
UCL
London, UK
w.langdon@cs.ucl.ac.uk

Vesna Nowack
Imperial College
London, UK
v.nowack@imperial.ac.uk

Justyna Petke
UCL
London, UK
j.petke@ucl.ac.uk

Erik M. Fredericks
Grand Valley State University
Michigan, USA
frederer@gvsu.edu

Gabin An
KAIST
Korea
agb94@kaist.ac.kr

Aymeric Blot
Universite du Littoral Cote d'Opale
Calais, France
aymeric.blot@univ-littoral.fr

Markus Wagner
Monash University
Melbourne, Australia
markus.wagner@monash.edu

Hyeonseok Lee
KAIST
Korea
christmas@kaist.ac.kr

ABSTRACT

Following the formal presentations, which included keynotes by Prof. Myra B. Cohen of Iowa State University and Dr. Sebastian Baltes of SAP as well as six papers (which are recorded in the proceedings) there was a wide ranging discussion at the twelfth international Genetic Improvement workshop, GI-2023 @ ICSE held on Saturday 20th May 2023 in Melbourne and online via Zoom. Topics included GI to improve testing, and remove unpleasant surprises in cloud computing costs, incorporating novelty search, large language models (LLM ANN) and GI benchmarks.

1. SUMMARY 12TH GI WORKSHOP

The 12th International Workshop on Genetic Improvement (GI 2023) was co-located with the 45th International Conference on Software Engineering (ICSE 2023) in Melbourne, Australia, and ran in hybrid mode using Zoom.us. Despite GI's impressive findings (see next section), there remain many opportunities to improve the state-of-the-art. By bringing together researchers doing work in GI as well as GI enthusiasts, the workshop aids discussions (Section 5) and so moves the field forward by sharing experiences and exchanging ideas.

2. WHAT IS GENETIC IMPROVEMENT

Genetic Improvement is a branch of Artificial Intelligence (AI) and Software Engineering which applies optimisation [2] to improve existing programs. Rather than the long-term dream of evolving programs from scratch [3, 4], genetic improvement is the process of using automated search to improve existing software. It has successfully been used to fix bugs [5], transplant functionality from one system to another [6] improve predictions [7, 8], and reduce software's runtime [9], energy [10], and memory consumption [11]; all without the necessity of costly human labour. GI research has already won three "Humies", [12, 13, 14] prestigious cash prizes awarded for demonstrating human-competitive results at difficult-to-automate tasks. Figure 2 shows a recent summary of publications by type, conference and journal.

We can always compare the new code with the existing code (effectively treating the program as its own specification) allowing GI to make measurable improvements to today's software. Improvements may be functional: e.g., does the new code have fewer bugs? does it have a new feature? does it give more accurate answers? or non-functional: e.g., does it have better battery life? is it more reliable?

3. WORKSHOP FORMAT

As usual, this year the Genetic Improvement workshop was a full one-day event. It was held on Saturday May 20th, immediately after the main ICSE 2023 conference. The final workshop program and the recordings of the talks are available online at http://geneticimprovementofsoftware.com/events/icse2023. and in the ICSE 2023 workshop proceedings [15]. GI @ ICSE 2023 included two keynotes, three research paper talks, three position paper talks and a discussion session.

Keynotes. We were extremely fortunate that **Prof. Myra B. Cohen** and **Dr. Sebastian Baltes** agreed to give the keynote speeches at the GI workshop.

Prof. Cohen is a full professor at Iowa State University (USA) and the title of her keynote speech was "It's all in the semantics: When are genetically improved programs still correct?" [1].

Dr. Sebastian Baltes is a Principal Expert for Empirical Software Engineering at SAP SE in Germany and an Adjunct Lecturer at the University of Adelaide in Australia. His presentation was on "All about the money: Cost modeling and optimization of cloud applications" [16]. As expected, the keynote talks triggered many interesting questions and discussion points (Figures 3–5), which were followed up during the rest of the workshop.

Paper presentations Six papers were presented at the workshop [17, 18, 19, 20, 21, 22]. (They will be available online via the IEEE Digital Library shortly.) Fortunately, all authors agreed to release recordings of their presentations and these can be found online (see active video_url links in the GP bibliography and the GI workshop web page.

The workshop attracted 22 early registrations. Some of whom participated online via Zoom (Figure 6). In total, there were perhaps about 20 people in person plus seven or eight on Zoom. Most stayed for the whole workshop but a few came especially to see the two Keynote presentations. There were approximately 30 people in total.

Awards. Traditionally at the GI workshop, the best paper awards are given to the researchers for their outstanding contributions to the GI field. This year, we granted three awards, where the best presentation award was decided by a vote from the participants of the workshop, while the other two were given based on reviews:

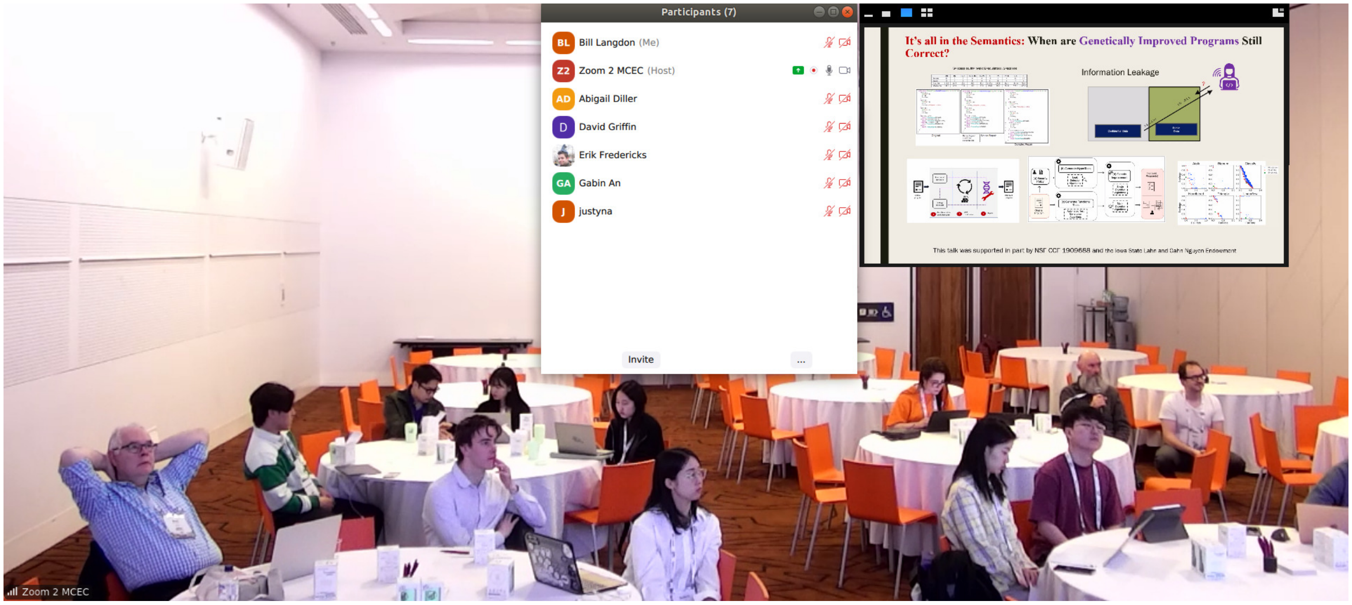

Participants (7)
BL Bill Langdon (Me)
Z2 Zoom 2 MCEC (Host)
AD Abigail Diller
D David Griffin
Erik Fredericks
GA Gabin An
J justyna
Invite
It's all in the Semantics: When are Genetically Improved Programs Still Correct?
Information Leakage
This talk was supported in part by NSF CCF 1909688 and the Iowa State Lahn and Gahn Nguyen Endowment
Zoom 2 MCEC

Figure 1: It's all in the Semantics: When are Genetically Improved Programs Still Correct [1]?

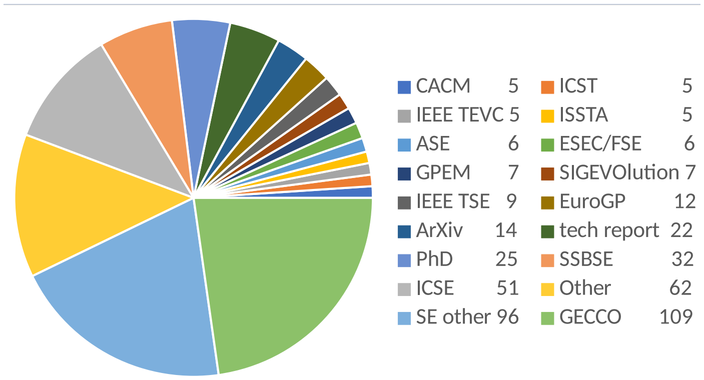


Figure 2: Genetic improvement papers by venue. Venues with less than 5 papers are split into software engineering (SE other, 96, blue) and other (62, yellow).

Figure 3: David Clark (UCL, with microphone) questioning the first keynote speaker, Prof. Cohen [1]. Right: Somin Kim (KAIST) raises a discussion point with Prof. Cohen [1]. (Note workshop souvenir, lower left.)

Figure 4: Gabin An (standing) moderated the discussion following Sebastian Baltes' keynote [16]. Myra Cohen (with microphone) starts the questions.

Best research paper award: "Generative Art via Grammatical Evolution" by Erik M. Fredericks, Abigail C. Diller, and Jared M. Moore of Grand Valley State University, Michigan, USA [17] (Figure 9).

Best position paper award: "Towards Objective-Tailored Genetic Improvement Through Large Language Models" by Sungmin Kang and Shin Yoo (KAIST) [19] (Figure 10).

Best presentation award: was also won by Sungmin Kang for "Towards Objective-Tailored Genetic Improvement Through Large Language Models" (Figure 11).

4. GENETIC IMPROVEMENT SPECIAL ISSUE AUTOMATED SOFTWARE ENGINEERING

Authors of accepted papers were invited to submit their extended work to the special issues on Genetic Improvement to be published in the Automated Software Engineering journal (Editor-in-Chief: Tim Menzies, Special Issue Editors: Justyna Petke & Markus Wagner): We already have interest from authors of four papers who said they intend to submit.

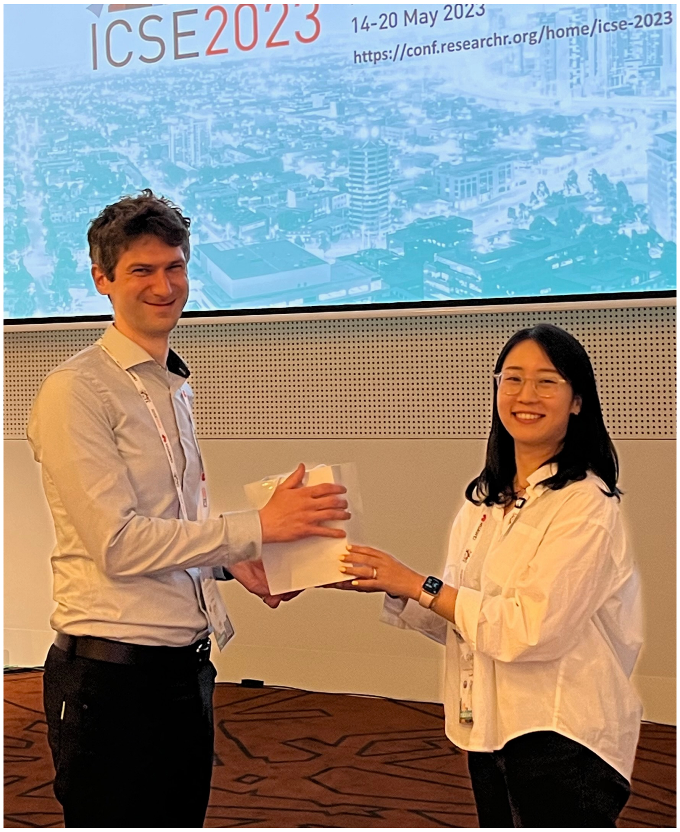


Figure 5: Gabin An thanking Sebastian Baltes (left) after his keynote [16].

Figure 6: Brad Alexander (Optimatics, with microphone) questioning remote speaker David Griffin (York University) [18] on Zoom screen (top left), whilst the meeting is chaired by Sungmin Kang (KAIST, standing). Others (seated at the same table) include Gabin An, Myles Watkinson (University of Adelaide) and Markus Wagner.

Figure 7: Juyeon Yoon (KAIST) discussing paper [22], following Myles Watkinson's presentation. (Despite the time zone differences, co-author, Sandy Brownlee, University of Stirling, participated in the discussion via Zoom.)

5. DISCUSSION

Although the full discussion, as recorded via Zoom will be available on YouTube, the following section condenses about an hour's flowing discussion into just a few subjects. Brad Alexander (Optimatics) led the discussion.

Erik Fredericks (GVSU) got the ball rolling by suggesting genetic improvement needs to include more research on automated testing, such as EvoSuite [23]. Brad Alexander suggested GI be applied to test suites (as opposed to the current emphasis of applying GI to program code [24]).

Erik Fredericks talked of experience with the SBFT workshop [25], where it has been recognised that there is a need to automatically improve aspects of test suites such as: ensuring fairness [26], auto prioritising test suites [27], auto generation of test cases [28], targeting tests to improve bug detection, (e.g. finding the most bugs) and improving tests to detect program failures [29]. Similar to a lot of GI work on optimising program performance, with the high cost of testing, GI might be applied to optimise the performance of the (regression) test suite, in particular, Brad suggested minimising its memory footprint. Fredericks suggested that GI could be applied to these research topics.

Myles Watkinson (Adelaide) suggested novelty search [30] for making test suits more diverse [18]. Diversity could include non-functional aspects of the software under test. This prompted Erik Fredericks to consider testing for non-functional properties, e.g. performance (particularly speed), security, usability and safety critical aspects.

Brad Alexander referred to Sebastian Baltes' keynote on measuring and controlling cloud computing costs (Section 3 [16]). Brad reaffirmed that cost was perhaps the most important non-functional property. And said at work, the costs of cloud computing, provoked the most animated discussions on the company's Slack channel[1]. Brad confirmed in industry, controlling cloud costs is a tough problem.

[1] Although Zoom chat can be used during Zoom meetings to pass text messages between online workshop participants, it is not ideal, since the chat disappears when the Zoom link is closed. This has oft been remarked upon. Indeed previously Emily Winter [31] recommended the use of Slack to continue conversations after scheduled talks/sessions had finished. She said that it was something that worked well at other conferences.

Figure 8: Some of GI @ ICSE 2023 workshop participants. Top: David Clark, Justyna Petke, Sandy Brownlee, Erik Fredericks, Abigail Diller, Bill Langdon, David Griffin. Keynotes: Myra Cohen, Sebastian Baltes (both keynote speakers travelled to Melbourne and gave their invited keynote presentations in person). In Melbourne: Markus Wagner, Gabin An, Myles Watkinson, Brad Alexander, Banseok Woo, Sungmin Kang, Juyeon Yoon, Jinhan Kim, Hyeonseok Lee, Naryeong Kim, Somin Kim

Brad Alexander referred to the two workshop papers on GI and today's artificial intelligence (AI) Large Language (Artificial Neural Network) Models [19, 20]. Brad suggested several ways Large Language Models (LLMs) might work with GI [32]. For example, they might be used to concentrate GI search on suitable subsets of the code base. Alternatively perhaps GI might broaden large language models; so that instead of LLM tools, such as GitHub's Copilot, concentrating on local context (such as the source code the software developer is currently modifying), LLMs might be broadened to consider the whole program.

Sungmin Kang (KAIST) suggested Large Language Models might be a way to make genetic improvement more powerful by providing access to other tools. In particular, LLMs might provide a way to integrate Natural Language Processing (NLP) tools into GI. Sungmin Kang suggested Large Language Models and program source code search might be a way to provide an extended global context (rather than Copilot's local context). Brad Alexander suggested Large Language Models might also be used to help GI work with both large programs and large source code patches. (See also [33].)

Erik Fredericks raised the question of where the data needed to train Large Language Models would come from. Although traditionally machine learning has used modest size datasets, in recent years, particularly in text processing and image processing, huge datasets have been published. Often datasets are freely available [9, 34]. Some of the larger software engineering datasets are the result of Mining Software Repositories (see the MSR conference series). For example, the CROP dataset was created by extracting program source code revision histories from GitHub [35]. Gabin An said KAIST had found the QuixBugs [36] benchmark useful. Some of the existing LLMs have been made available so it may not be necessary to train our own LLM. Indeed some public LLMs might be re-trained at a modest cost.

Myles Watkinson suggested GI might be used to address security issues [37], program specifications [38], models, non-function versus functional testing and safety critical domains [39]. Broadly to tackle problems at the system level rather than detailed level.

Brad Alexander also suggested that GI might be used to optimise hyperparameters. (Note there is some existing research on parameter tuning [7] and hypertesting [40].)

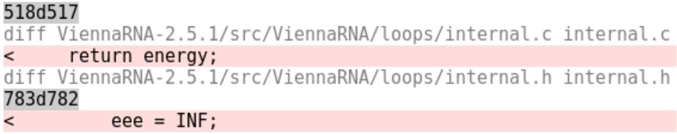
Although there is currently great interest in opaque Large Language Models, it should not be forgotten that software, at least source code, is its own model [41]. It is written to be human readable. One of the great strengths of GI operating on the program's source code is that bug fixes, code optimisation, etc., are visible to and often immediately accessible to software engineers. Sometimes as researchers we fall into the trap of talking only about performance. In science and engineering, the components of the model may be more important than its accuracy. How the model works may give practitioners in another field valuable insights or a different perspective on their problem. We should not be disappointed if they then streak ahead and solve it without our fancy AI tools.

Figure 9: Markus Wagner presenting the best research paper award [17]. Top: Zoom participants Erik Fredericks and Abigail Diller

Figure 10: Sungmin Kang and Shin Yoo (right), winners of the best position paper for [19].

Figure 11: Markus Wagner (left) and Gabin An handing Sungmin Kang (right) the best presentation award.

When presenting to the customer we should put the model in front of them, e.g. on a single slide. We should use their jargon (not ours). In GI's case, we can put the source code change itself on a slide. Perhaps even format it as a coloured "diff patch" (Figure 12).



```
518d517
diff ViennaRNA-2.5.1/src/ViennaRNA/loops/internal.c internal.c
<     return energy;
diff ViennaRNA-2.5.1/src/ViennaRNA/loops/internal.h internal.h
783d782
<         eee = INF;
```

Figure 12: Example patch generated by Magpie [42]

APPENDIX

A. FUTURE PLANS

We hope to hold the GI workshop again at ICSE next year (ICSE 2024 will be in Lisbon in Portugal). A few things to remember for next time:

We anticipate that the trend for "hybrid" (i.e. in person and on line) will continue. Although the time zones worked against us this year, we anticipate next year again a sizable fraction of participants, for any of a number of reasons (e.g. health, time, expense, visa issues), will want to participate remotely. Also, if not too expensive, the use of a European time zone (which may better suit some people in the USA), could well encourage a larger online audience. It would be great if ICSE 2024 makes it easy, e.g. advertised well in advance, to entice additional remote ad-hoc "walk in" participation, as well as more traditional formal registration.

Certainly this year there were no Zoom issues and the MCEC venue had copious bandwidth to support multiple internet video links. Also the on site workshop organisers took pains to ensure that both the speakers and members of the on site audience used microphones that were connected to the Zoom system and continuously monitored the Zoom channels to ensure overseas participants could hear ok and actively participate in the workshop, e.g. via Zoom's "chat" feature. Similarly, they took care to ensure people on site in the MCEC could follow the remote presentations. As in previous years, in case of problems at the remote speaker's end, we asked all presenters to pre-record their presentation. I think despite the time zone differences, on line participation of the hybrid GI workshop with Zoom went well. (However not all of ICSE was so well prepared for hybrid operation).

Perhaps next year we could more actively solicit input from members of the programme committee during the workshop itself.

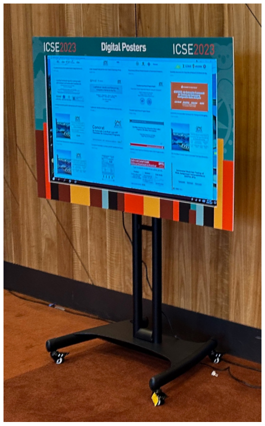


Figure 13: One of the 55" (≈ 4 feet wide) ICSE 2023 digital poster boards.

As always there were suggestions that we should allocate more time for "Questions and Answers". As usual, there is a trade-off between presentation time and discussion time. We feel that possibly the Q&A went more successfully than at many "regular" conference sessions, where there are often less than five minutes to raise and answer perhaps one or at most two questions. The GI workshop was much more flexible. E.g. in the second morning session we naturally consumed the 15 minutes that were meant as pre-lunch buffer.

As with recent GI workshops, we offered students presenting full papers some financial assistance. To continue this, the workshop will require further sponsorship next year.

This year ICSE offered authors of workshop papers the opportunity to additionally present their work in the main ICSE conference as a poster (Figure 13). Unfortunately, due to practical issue with travel, we were unable to take this up. If the opportunity is repeated by ICSE 2024, it would be nice to let potential authors know early, as it could be an additional inducement for people to submit to ICSE workshops.

B. GI WORKSHOP ORGANISERS

Vesna Nowack

Markus Wagner

Gabin An

Aymeric Blot

Justyna Petke

C. PROGRAMME COMMITTEE

Each submission received at least three independent reviews from the workshop's programme committee (see Figure 14).

In addition to providing feedback to the authors and deciding which submissions to accept, the best paper awards were decided by the organisers using the reviewers' comments. Whilst the best presentation was chosen by the audience on the day in Melbourne and on Zoom.us

Acknowledgement

The workshop received funding from the UKRI project EP/P023991/1 and the ICSE 2023 conference. Our thanks to UK Research and Innovation, ACM SIGSOFT and the IEEE Technical Community on Software Engineering.

D. REFERENCES

[1] Myra B. Cohen. It's all in the semantics: When are genetically improved programs still correct? In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, page ix, Melbourne, Australia, 20 May 2023. IEEE. Invited Keynote. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/Cohen_2023_GI.html`.

[2] Mark Harman and Bryan F. Jones. Search based software engineering. *Information and Software Technology*, 43(14):833–839, December 2001. URL: `http://dx.doi.org/10.1016/S0950-5849(01)00189-6`.

[3] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/koza_book.html`.

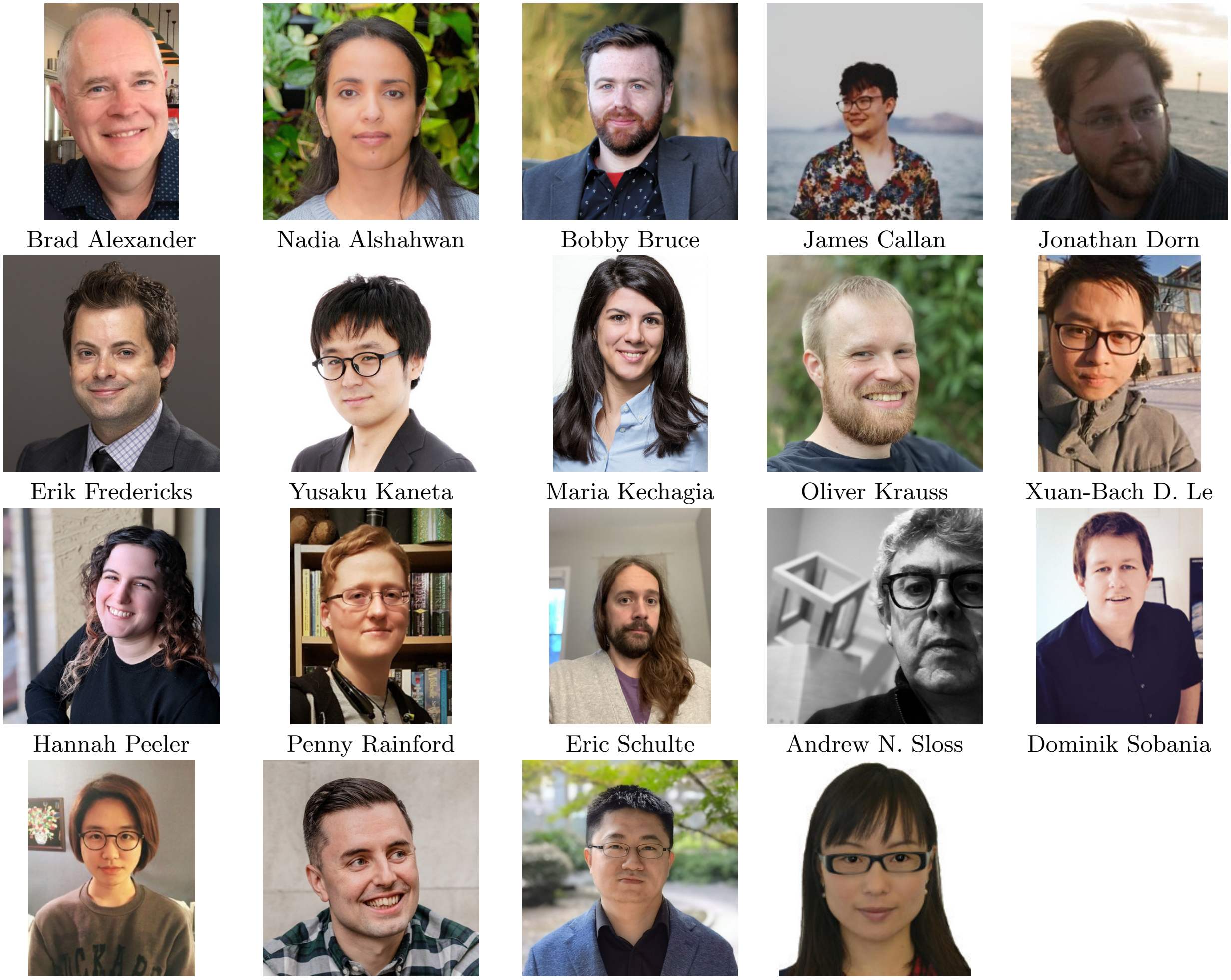
[4] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A field guide to genetic programming*. Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008. (With contributions by J. R. Koza). URL: `http://gpbib.cs.ucl.ac.uk/gp-html/poli08_fieldguide.html`.

[5] Claire Le Goues, Michael Pradel, and Abhik Roychoudhury. Automated program repair. *Communications of the ACM*, 62(12):56–65, December 2019. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/cacm2019_program_repair.html`.

[6] Alexandru Marginean, Earl T. Barr, Mark Harman, and Yue Jia. Automated transplantation of call graph and layout features into Kate. In Yvan Labiche and Marcio Barros, editors, *SSBSE*, volume 9275 of *LNCS*, pages 262–268, Bergamo, Italy, September 5-7 2015. Springer. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/Marginean_2015_SSBSE.html`.

[7] William B. Langdon, Justyna Petke, and Ronny Lorenz. Evolving better RNAfold structure prediction. In Mauro Castelli, Lukas Sekanina, and Mengjie Zhang, editors, *EuroGP 2018: Proceedings of the 21st European Conference on Genetic Programming*, volume 10781 of *LNCS*, pages 220–236, Parma, Italy, 4-6 April 2018. Springer Verlag. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/langdon_2018_EuroGP.html`.

[8] William B. Langdon and Oliver Krauss. Genetic improvement of data for maths functions. *ACM Transactions on Evolutionary Learning and Optimization*, 1(2):Article No.: 7, July 2021. URL: `http://gpbib.cs.ucl.ac.uk/gp-html/Langdon_TELO.html`.



Brad Alexander
Nadia Alshahwan
Bobby Bruce
James Callan
Jonathan Dorn
Erik Fredericks
Yusaku Kaneta
Maria Kechagia
Oliver Krauss
Xuan-Bach D. Le
Hannah Peeler
Penny Rainford
Eric Schulte
Andrew N. Sloss
Dominik Sobania
Jeongju Sohn
Christopher Timperley
Jifeng Xuan
Bing Xue

Figure 14: GI @ ICSE 2023 Reviewers

[9] William B. Langdon and Mark Harman. Optimising existing software with genetic programming. *IEEE Transactions on Evolutionary Computation*, 19(1):118–135, February 2015. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Langdon_2013_ieeeTEC.html.

[10] Mahmoud Bokhari and Markus Wagner. Optimising energy consumption heuristically on Android mobile phones. In Justyna Petke, David R. White, and Westley Weimer, editors, *Genetic Improvement 2016 Workshop*, pages 1139–1140, Denver, July 20-24 2016. ACM. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Bokhari_2016_GI.html.

[11] Fan Wu, Westley Weimer, Mark Harman, Yue Jia, and Jens Krinke. Deep parameter optimisation. In Sara Silva et al., editors, *GECCO '15: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 1375–1382, Madrid, 11-15 July 2015. ACM. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Wu_2015_GECCO.html.

[12] Earl T. Barr, Mark Harman, Yue Jia, Alexandru Marginean, and Justyna Petke. Automated software transplantation. In Tao Xie and Michal Young, editors, *International Symposium on Software Testing and Analysis, ISSTA 2015*, pages 257–269, Baltimore, Maryland, USA, 14-17 July 2015. ACM. ACM SIGSOFT Distinguished Paper Award. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Barr_2015_ISSTA.html.

[13] Justyna Petke, Mark Harman, William B. Langdon, and Westley Weimer. Using genetic improvement and code transplants to specialise a C++ program to a problem class. In Miguel Nicolau, Krzysztof Krawiec, Malcolm I. Heywood, Mauro Castelli, Pablo Garcia-Sanchez, Juan J. Merelo, Victor M. Rivas Santos, and Kevin Sim, editors, *17th European Conference on Genetic Programming*, volume 8599 of *LNCS*, pages 137–149, Granada, Spain, 23-25 April 2014. Springer. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Petke_2014_EuroGP.html.

[14] Michail Basios, Lingbo Li, Fan Wu, Leslie Kanthan, and Earl T. Barr. Darwinian data structure selection. In Gary T. Leavens, Alessandro Garcia, and Corina S. Pasareanu, editors, *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, ESEC/FSE 2018*, pages 118–128, Lake Buena Vista, FL, USA, 4-9 November 2018. ACM. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Basios_2018_FSE.html.

[15] Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors. *12th International Workshop on Genetic Improvement @ICSE 2023*, Melbourne, Australia, 20 May 2023. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Nowack_2023_GI.html.

[16] Sebastian Baltes. All about the money: Cost modeling and

optimization of cloud applications. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, page x, Melbourne, Australia, 20 May 2023. IEEE. Invited Keynote. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Baltes_2023_GI.html.

[17] Erik M. Fredericks, Abigail C. Diller, and Jared M. Moore. Generative art via grammatical evolution. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 1–8, Melbourne, Australia, 20 May 2023. IEEE. Best paper. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Fredericks_2023_GI.html.

[18] David Griffin, Susan Stepney, and Ian Vidamour. DebugNS: Novelty search for finding bugs in simulators. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 17–18, Melbourne, Australia, 20 May 2023. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Griffin_2023_GI.html.

[19] Sungmin Kang and Shin Yoo. Towards objective-tailored genetic improvement through large language models. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 19–20, Melbourne, Australia, 20 May 2023. IEEE. Best position paper. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Kang_2023_GI.html.

[20] Oliver Krauss. Exploring the use of natural language processing techniques for enhancing genetic improvement. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 21–22, Melbourne, Australia, 20 May 2023. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Krauss_2023_GI.html.

[21] William B. Langdon and Bradley J. Alexander. Genetic improvement of OLC and H3 with Magpie. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 9–16, Melbourne, Australia, 20 May 2023. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/langdon_2023_GI.html.

[22] Myles Watkinson and Alexander Brownlee. Updating Gin's profiler for current java. In Vesna Nowack, Markus Wagner, Gabin An, Aymeric Blot, and Justyna Petke, editors, *12th International Workshop on Genetic Improvement @ICSE 2023*, pages 23–28, Melbourne, Australia, 20 May 2023. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Watkinson_2023_GI.html.

[23] Gordon Fraser and Andrea Arcuri. Evosuite: automatic test suite generation for object-oriented software. In 8^{th} *European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC/FSE '11)*, pages 416–419, Szeged, Hungary, September 5th - 9th 2011. ACM. URL: http://dx.doi.org/10.1145/2025113.2025179.

[24] Justyna Petke, Saemundur O. Haraldsson, Mark Harman, William B. Langdon, David R. White, and John R. Woodward. Genetic improvement of software: a comprehensive survey. *IEEE Transactions on Evolutionary Computation*, 22(3):415–432, June 2018. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Petke_gisurvey.html.

[25] Alessio Gambi, Giovani Guizzo, and Panichella Sebastiano, editors. *The Search-Based & Fuzz Testing (SBFT) Workshop*, Melbourne, 10 May 2023. Co-located with ICSE 2023. URL: https://sbft23.github.io/.

[26] Max Hort, Rebecca Moussa, and Federica Sarro. Multi-objective search for gender-fair and semantically correct word embeddings. *Applied Soft Computing*, 133:109916, January 2023. URL: http://dx.doi.org/10.1016/j.asoc.2022.109916.

[27] S. Yoo and M. Harman. Regression testing minimization, selection and prioritization: A survey. *Journal of Software Testing, Verification & Reliability*, 22(2):67–120, March 2012. URL: http://dx.doi.org/10.1002/stvr.430.

[28] Nadia Alshahwan and Mark Harman. Automated web application testing using search based software engineering. In *2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011)*, pages 3–12, Lawrence, KS, USA, 6-10 November 2011. URL: http://dx.doi.org/10.1109/ASE.2011.6100082.

[29] Mark Harman, Yue Jia, and Yuanyuan Zhang. Achievements, open problems and challenges for search based software testing. In Gordon Fraser and Darko Marinov, editors, *8th IEEE International Conference on Software Testing, Verification and Validation, ICST 2015*, pages 1–12, Graz, Austria, April 14-16 2015. IEEE. Keynote. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Harman_2015_ICST.html.

[30] Joel Lehman and Kenneth O. Stanley. Novelty search and the problem with objectives. In Rick Riolo, Ekaterina Vladislavleva, and Jason H. Moore, editors, *Genetic Programming Theory and Practice IX*, Genetic and Evolutionary Computation, chapter 3, pages 37–56. Springer, Ann Arbor, USA, 12-14 May 2011. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Lehman_2011_GPTP.html.

[31] William B. Langdon, Westley Weimer, Justyna Petke, Erik Fredericks, Seongmin Lee, Emily Winter, Michail Basios, Myra B. Cohen, Aymeric Blot, Markus Wagner, Bobby R. Bruce, Shin Yoo, Simos Gerasimou, Oliver Krauss, Yu Huang, and Michael Gerten. Genetic improvement @ icse 2020. *SIGSOFT Software Engineering Notes*, 45(4):24–30, October 2020. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Langdon_2020_SEN.html.

[32] Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, and Kenneth O. Stanley. Evolution through large models. ArXiv, 17 June 2022. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Lehman_2022_ELM.html.

[33] Miltiadis Allamanis, Earl T. Barr, Premkumar Devanbu, and Charles A. Sutton. A survey of machine learning for big code and naturalness. *ACM Computing Surveys*, 51(4), jul 2018. URL: http://dx.doi.org/10.1145/3212695.

[34] William B. Langdon. Large scale bioinformatics data mining with parallel genetic programming on graphics processing units. In Shigeyoshi Tsutsui and Pierre Collet, editors, *Massively Parallel Evolutionary Computation on GPGPUs*, Natural Computing Series, chapter 15, pages 311–347. Springer, 2013. URL: http://gpbib.cs.ucl.ac.uk/gp-html/langdon_2013_ecgpu.html.

[35] Matheus Paixao, Jens Krinke, Donggyun Han, and Mark Harman. CROP: Linking code reviews to source code changes. In *Proceedings of the 15th International Conference on Mining Software Repositories*, page 46–49, Gothenburg, Sweden, 2018. Association for Computing Machinery. Co-located with ICSE. URL: http://dx.doi.org/10.1145/3196398.3196466.

[36] Derrick Lin, James Koppel, Angela Chen, and Armando Solar-Lezama. QuixBugs: A multi-lingual program repair benchmark set based on the Quixey challenge. In *SPLASH Companion 2017*, pages 55—56, Vancouver, 2017. URL:

http://dx.doi.org/10.1145/3135932.3135941.

[37] Jason Landsborough, Stephen Harding, and Bryan Beabout. Evolutionary fuzzing for genetic improvement: Toward adaptive software defense. In Justyna Petke, Kathryn Stolee, William B. Langdon, and Westley Weimer, editors, *GI-2018, ICSE workshops proceedings*, pages 45–46, Gothenburg, Sweden, 2 June 2018. ACM. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Landsborough_2018_GI.html.

[38] Linsey Kitt and Myra B. Cohen. Partial specifications for program repair. In Justyna Petke, Bobby R. Bruce, Yu Huang, Aymeric Blot, Westley Weimer, and W. B. Langdon, editors, *GI @ ICSE 2021*, pages 19–20, internet, 30 May 2021. IEEE. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Kitt_2021_GI.html.

[39] Marouane Tlili, Stefan Wappler, and Harmen Sthamer. Improving evolutionary real-time testing. In Maarten Keijzer et al., editors, *GECCO 2006: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, volume 2, pages 1917–1924, Seattle, Washington, USA, 8-12 July 2006. ACM Press. URL: http://dx.doi.org/10.1145/1143997.1144316.

[40] Ibrahim Mesecan, Daniel Blackwell, David Clark, Myra B. Cohen, and Justyna Petke. HyperGI: Automated detection and repair of information flow leakage. In Hourieh Khalajzadeh and Jean-Guy Schneider, editors, *The 36th IEEE/ACM International Conference on Automated Software Engineering, New Ideas and Emerging Results track, ASE NIER 2021*, pages 1358–1362, Melbourne, 15-19 November 2021. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Clark_2021_ASE-NIER.html, arXiv:2108.12075.

[41] John R. Woodward, Justyna Petke, and William Langdon. How computers are learning to make human software work more efficiently. *The Conversation*, page 10.08am BST, June 25 2015. URL: http://gpbib.cs.ucl.ac.uk/gp-html/Woodward_2015_hclhswe.html.

[42] Aymeric Blot and Justyna Petke. MAGPIE: Machine automated general performance improvement via evolution of software. arXiv, 4 August 2022. URL: http://gpbib.cs.ucl.ac.uk/gp-html/blot_2022_corr_1.html.