

2024 IEEE/ACM International Workshop on Genetic Improvement

GI 2024

GI 2024

MESSAGE FROM THE CHAIRS

It is our great pleasure to welcome you to the 13th edition of the International Genetic Improvement workshop, GI-2024, to take place at ICSE-2024 in Lisbon, Portugal, on 16 April 2024. Genetic Improvement is essentially the application of search-based software engineering (SBSE) to the software itself. Be it for automatically repairing programs (APR, auto fix) or optimizing existing source, byte, assembler, intermediate, or machine code to improve its results or operation. Such as running faster or using less resources (e.g. energy). Since 2015, the GI workshop has been held annually as part of the Genetic and Evolutionary Computation Conference (GECCO) and/or the IEEE/ACM International Conference on Software Engineering (ICSE). We are very pleased that the workshop will also be held at the International Conference on Software Engineering (ICSE) for the sixth time. Its first edition at ICSE 2018 in Göteborg, Sweden (and 5th workshop edition overall), showed that there is great interest in genetic improvement in the software engineering community. Since starting to hold the GI workshop at ICSE, the workshop has also been run several times at GECCO (2018 Kyoto, Japan, 2019 Prague, Czech Republic, 2020 Cancún, Mexico, 2022 Virtual). In addition, several GI tutorials have been given at conferences, such as PPSN 2020, ASE 2020 and each year at GECCO from 2020-2024. Finally, there have been multiple CREST Open Workshops (<https://www.ucl.ac.uk/crest/crest-open-workshops>) on Genetic Improvement. COW65 and earlier are archived: http://crest.cs.ucl.ac.uk/cow/past_events/.

The GI workshops continue to bring together researchers from across the world to exchange ideas about using optimisation techniques, particularly evolutionary computation, such as genetic programming, to improve existing software. We invited short two-page position papers to encourage the discussion of new ideas and recent work in addition to longer and more concrete research submissions. The call for participation invited GI work on improving efficiency; decreasing memory consumption; reducing energy consumption; transplanting new functionality; specialising software; translating between programming languages; generating multiple versions of software and repairing bugs. Recently, GI papers that apply large language models (LLMs), currently being used in various code-related tasks, to GI are also invited. As you will see, half the papers are position papers proposing new ideas, and half are full-length research papers. Most of the submissions came from the UK or USA, but there were also submissions from Spain, Austria, etc.

Putting together GI-2024 was a team effort. Firstly, we thank the authors for providing the content of the program. We would like to express our gratitude to Dr. Shin Yoo and Dr. Aymeric Blot for their keynote talk and tutorial on MAGPIE, respectively. Finally, we are grateful to the program committee, who worked hard to review papers and provide great feedback for authors.

We hope that you will find these papers thought-provoking and that the workshop will provide you with an opportunity to share ideas with people across the globe. We hope that you will notice many areas of software engineering that are not yet covered. Our primary aim remains to encourage you to participate. Go one step beyond being a better programmer, get the AI to program for you!

Sincerely,
Gabin, Oliver, Vesna, Aymeric and Justyna



ORGANIZING COMMITTEE GI 2024

Workshop Chairs

Gabin An, *KAIST, South Korea*
Aymeric Blot, *Université of Rennes, France*
Vesna Nowack, *Imperial College London, United Kingdom*
Justyna Petke, *University College London, United Kingdom*
Oliver Krauss, *University of Applied Sciences Upper Austria, Austria*

Special Thanks

To Bill Langdon, *University College London, United Kingdom*, for helping us with advertising the workshop.

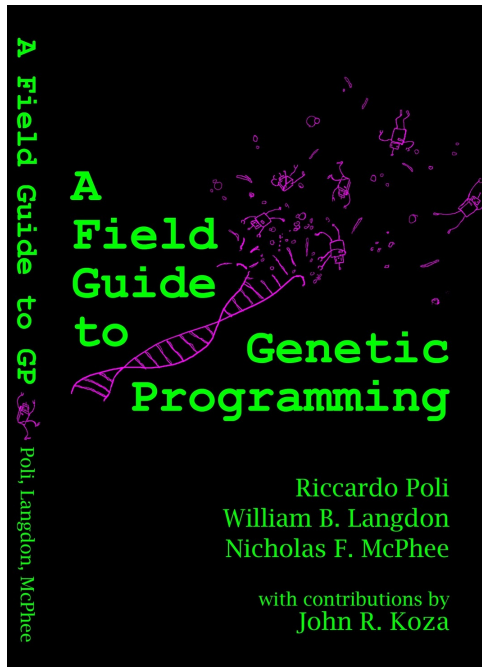
PROGRAM COMMITTEE GI 2024

Brad Alexander, *Optimatics, Australia*
Nadia Alshahwan, *Meta, UK*
Sandy Brownlee, *University of Stirling, UK*
Erik Fredericks, *Grand Valley State University, MI, USA*
Carol Hanna, *University College London, UK*
Max Hort, *Simula Research Laboratory, Norway*
Gunel Jahangirova, *King's College London, UK*
Yue Jia, *Meta, UK*
Yusaku Kaneta, *Rakuten Mobile, Inc., Japan*
Sungmin Kang, *KAIST, South Korea*
Maria Kechagia, *University College London, UK*
Penny Rainford, *Lancaster University, UK*
Eric Schulte, *Google, USA*
Dominik Sobania, *Johannes Gutenberg-Universität, Germany*
Jeongju Sohn, *Universite du Luxembourg, Luxembourg*
Christopher Timperley, *Carnegie Mellon University, USA*
Michele Tufano, *Microsoft, USA*
Marcus Wagner, *Monash University, Australia*
Jifeng Xuan, *Wuhan University, China*

LIST OF ACCEPTED PAPERS GI 2024

Deep Mutations have Little Impact – Bill Langdon
Ecosystem Curation in Genetic Improvement for Emergent Software Systems – Zsolt Nemeth, Penn Faulkner Rainford and Barry Porter
Genetic Improvement for DNN Security – Hunter Baxter, Yu Huang, and Kevin Leach
Grammar evolution and symbolic regression for astrometric centering of Hubble Space Telescope images – Beatriz R. Sarmiento, Marina de la Cruz, Alfonso Ortega, Roberto Baena-Galle, Terrence Girard, Dana Casetti-Dinescu and Alejandro Cervantes
Human Guidance Approaches for the Genetic Improvement of Software – Benjamin Craine, Penn Rainford and Barry Porter
On Reducing Network Usage with Genetic Improvement – James Callan, Bill Langdon and Justyna Petke

SPONSORS GI 2024



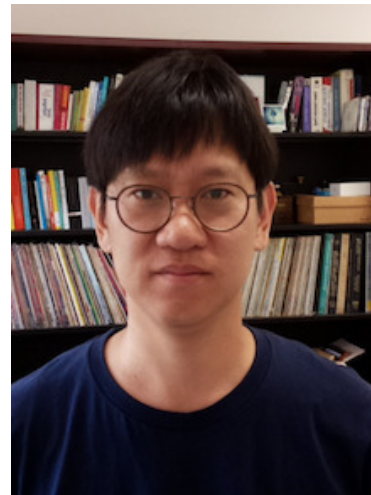
A Field Guide to Genetic Programming
<http://www.gp-field-guide.org.uk/>

Shin Yoo (KAIST, South Korea)

Executing One's Way out of the Chinese Room

One very attractive property of Large Language Models (LLMs) is their emergent in-context learning capability, which enables us to simply describe our requirements in natural languages and get the corresponding source code generated in programming languages. While LLMs as a generative model are known to hallucinate, i.e., generate factually incorrect contents, the fact that code can be executed can be used to fight this phenomenon. We briefly look at existing techniques designed to improve the correctness of code generated by LLMs, and will try to imagine the future of Genetic Improvement that is supported, enhanced, and driven by LLMs.

Prof. Shin Yoo gained his PhD under Mark Harman in the CREST group at King's College, London and was a lecture in UCL's Centre for Research on Evolution, Search and Testing before returning to his native Korea to join in the Korea Advanced Institute of Science and Technology (KAIST) in 2015. Shin has been active in software engineering research, particularly efficient mutation testing, code slicing, and fault localisation, for more than fifteen years. He is an expert in mining fixes and GI and recently has shown large language models (LLMs) may be used in advanced software testing. Dr. Yoo is an associate editor of ACM TOSEM and editorial board member of Empirical Software Engineering and Genetic Programming and Evolvable Machines. He has served on many conferences, including co-chair of ICST 2018, GI 2020, SSBSE 2022 and this year is the ICSE 2024 area-chair for testing and analysis.



INVITED TUTORIAL GI 2024

Aymeric Blot (Université of Rennes, France)

Automated Software Performance Improvement with Magpie

In this tutorial, we present Magpie, a powerful tool for both Genetic Improvement researchers and practitioners. Magpie stands at the forefront of software evolution, providing a streamlined approach to model, evolve, and automatically improve software systems. Addressing both functional and non-functional concerns, Magpie offers a user-friendly no-code interface that seamlessly integrates various search processes, as well as enabling easy Python code injection for advanced users to further tailor and specialise the improvement process to meet their specific needs. We will provide a concise overview of Magpie's internals before exploring diverse real-world scenarios.

Aymeric Blot is a Senior Lecturer at the Université of Rennes, France. He received in 2018 a doctorate from the University of Lille following work on automated algorithm design for multi-objective combinatorial optimisation, before moving to University College London to work on software specialisation using GI. His research focuses on strengthening GI techniques using knowledge from automated machine learning, algorithm configuration, and evolutionary computation.

