2025 IEEE/ACM International Workshop on Genetic Improvement

GI 2025

MESSAGE FROM THE CHAIRS

It is our great pleasure to welcome you to the 14th edition of the International Genetic Improvement workshop, GI-2025, to take place at ICSE-2025 in Ottawa, the capital of Canada, on 27 April 2025. 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 seventh 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 2020-2025. 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,and more recently AI's Large Language Models 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, most of the papers are full-length research papers, with one position paper proposing new ideas. Most of the submissions came from the UK or China but there were also submissions from France and the USA.

Putting together GI-2025 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 Hwei Tan 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,
Vesna, Aymeric, Penn, and Oliver

ORGANIZING COMMITTEE GI 2025

Workshop Chairs

Aymeric Blot, *University of Rennes, France*
Vesna Nowack, *Imperial College London, United Kingdom*
Penn Faulkner Rainford, *University of York, United Kingdom*
Oliver Krauss, *University of Applied Sciences Upper Austria, Austria*

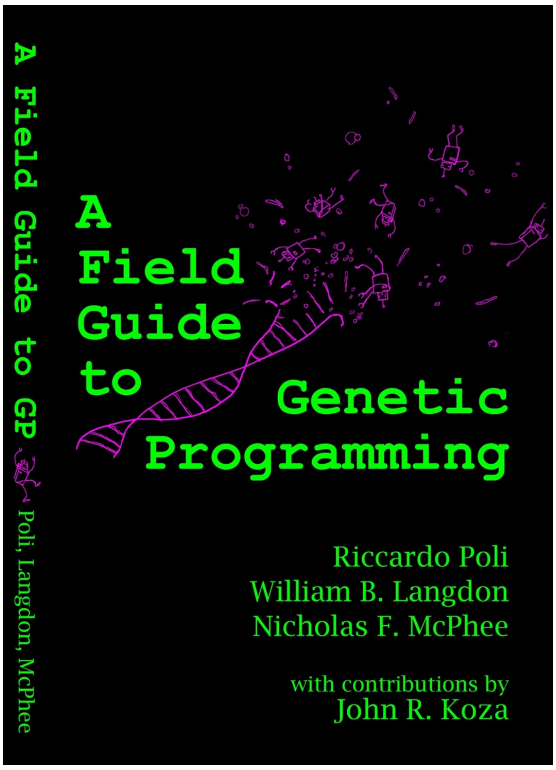
Special Thanks

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

PROGRAM COMMITTEE GI 2025

Brad Alexander, Optimatics, Australia
Nadia Alshahwan, Meta, UK
Gabin An, Roku, Korea
Marcio Barros, Universidade Federal do Estado do Rio de Janeiro, Brazil
Zishuo Ding, Hong Kong University of Science and Technology, China
Sophie Fortz, King's College London, UK
Alina Geiger, Johannes Gutenberg University Mainz, Germany
Anastasiia Grishina, Simula Research Laboratory, Norway
Carol Hanna, University College London, UK
Max Hort, Simula Research Laboratory, Norway
Yu Huang, Vanderbilt University, USA
Yusaku Kaneta, Rakuten Group Inc, Japan
Sungmin Kang, KAIST, Korea
Martin Nowack, Imperial College London, UK
Jeongju Sohn, Kyungpook National University, Korea
Sarah Thomson, Napier University, UK
Christopher Timperley, Carnegie Mellon University, USA
Michele Tufano, Google, USA
Jifeng Xuan, Wuhan University, China
Yuan Yuan, Michigan State University, USA

LIST OF ACCEPTED PAPERS GI 2025



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

Shin Hwei Tan (Concordia University, Montreal)

Put on Your Tester Hat: Improving programs for Automated Program Generation

Given either a specification written in natural language or an input program, automated program generation techniques produce a program according to the given specification or by modifying the input program. Automated program generation is a powerful technique that can be used for finding bugs in software systems that take programs as input or fixing bugs in the input programs. However, most existing techniques focus on automated program generation for automated program repair or program synthesis. This talk introduces a different perspective of automated program generation where we will discuss our latest results on automated program generation for testing static program analyzers by designing different types of program transformations. We will also explore and rethink about the automated program generation problem from the tester perspective. The new perspective could have huge potential for the design of new genetic improvement techniques to improve programs for automated program generation.

Shin Hwei Tan is an Associate Professor (Gina Cody Research Chair) in Concordia University. Before moving to Concordia University, she was an Assistant Professor in Southern University of Science and Technology in Shenzhen, China. She obtained her PhD degree from National University of Singapore and her B.S (Hons) and MSc degree from University of Illinois at Urbana-Champaign. Her main research interests are in automated program repair, software testing and open-source development. She is an Associate Editor for TOSEM and the Guest Editors-in-Chief for the New Frontier in Software Engineering track in TOSEM. She has also served as PCs for top-tier software engineering conferences, where she won three best reviewers award (FSE 2020, ASE 2020, ICSE 2022 NIER-track). She is also the general chair of FSE26 which will be held in Concordia University.

Aymeric Blot (University of Rennes, France)

Automated Software Performance Improvement with Magpie

In this tutorial, I will present Magpie (https://github.com/bloa/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 University of Rennes and a member of the IRISA research centre in the joint Inria/IRISA DiverSE team. Building on previous work at University College London on software specialisation and a doctorate from the University of Lille focused on automated algorithm design for multi-objective combinatorial optimisation, their research explores evolving and optimising software using genetic improvement, automated machine learning, algorithm configuration, and evolutionary computation. This includes leading the development and maintenance of the Magpie automated software improvement framework.