Species Scholarships 2023
Fixing Faults in Software via Trajectory-guided Mutation

Dr Colin Johnson, University of Nottingham, UK
Colin.Johnson@nottingham.ac.uk

Advisor CV

Dr. Colin Johnson (PhD, University of Kent, 2003)

Associate Professor, University of Nottingham

Colin Johnson has worked on a wide variety of research topics in Evolutionary Computation. One strand of work has included the development of new techniques in genetic programming, in particular semantic GP [1,2,3], where evolutionary operators are designed to act on the meaning of programs rather than just on program text, and on learned fitness functions [4]. Another strand of work has focused on empirical and statistical analysis of existing evolutionary algorithms [5]. A final interest is in applications of evolutionary computation, including work on data mining [6], bioinformatics [7], art [8], music [9] and finance [10].

[1] Alberto Moraglio, Krzysztof Krawiec, and Colin G. Johnson, Geometric Semantic Genetic Programming, in C.A. Coello Coello et al. (Eds.), *Parallel Problem Solving from Nature—PPSN XII*, pp. 21–31, Springer, 2012.

[2] Lawrence Beadle and Colin G. Johnson, Semantically Driven Mutation in Genetic Programming, *Proceedings of the IEEE Congress on Evolutionary Computation*, Trondheim, Norway, Spring 2009, pp 1336–1342.

[3] Lawrence Beadle and Colin G. Johnson, Semantically Driven Crossover in Genetic Programming, *IEEE World Congress on Computational Intelligence*, Hong Kong, June 2008.

[4] Colin G. Johnson, Solving the Rubik's Cube with Stepwise Deep Learning, *Expert Systems*, 38(3):e12665.

[5] Colin G. Johnson, Genetic Programming Crossover: Does it Cross Over? *Genetic Programming: 12th European Conference (EuroGP2009)*, Springer, Lecture Notes in Computer Science 5481, pp. 97–108, 2009.

[6] Fernando E.B. Otero, Alex A. Freitas and Colin G. Johnson, A New Sequential Covering Strategy for Inducing Classification Rules with Ant Colony Algorithms, *IEEE Transactions on Evolutionary Computation*, 17(1):64–76, February 2013.

[7] Mudassar Iqbal, Alex A. Freitas, Colin G. Johnson and Massimo Vergassola, Message-Passing Algorithms for the Prediction of Protein Domain Interactions from Protein-Protein Interaction Data, *Bioinformatics*, 24(18):2064–2070, September 2008.

[8] Colin G. Johnson, Jon McCormack, Iria Santos and Juan Romero, Understanding Aesthetics and Fitness Measures in Evolutionary Art Systems, *Complexity*, Volume 2019, Article ID 3495962, 2019

[9] Colin G. Johnson, Creating a Digital Mirror of Creative Practice, Proceedings of the 10th International Conference on Artificial Intelligence in Music, Sound, Art and Design (EvoMusArt 2021), Springer, Lecture Notes in Computer Science, April 2021

[10] Adesola Agedboye, Colin G. Johnson, and Michael Kampouridis, Evolving Directional Changes Trading Strategies with a New Event-based Indicator, *Proceedings of Simulated Evolution and Learning 2017*, November 2017.

Research at the Host Institution

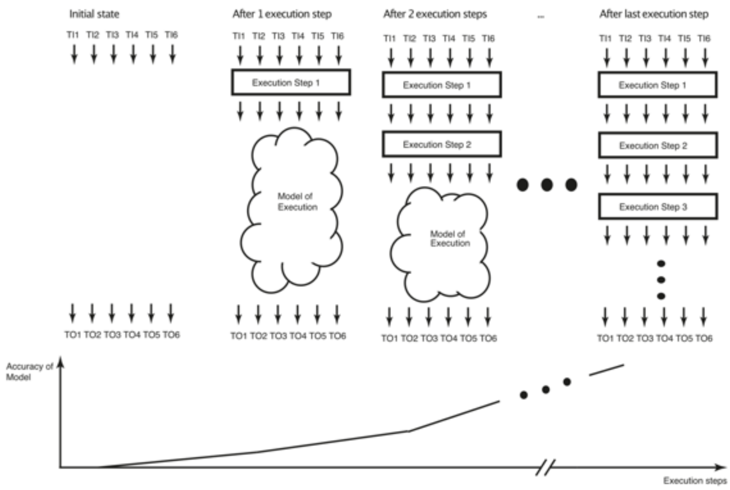
The University of Nottingham is a leading UK university, part of the Russell Group of research-intensive universities. The School of Computer Science has a thriving community of researchers in AI, optimisation, and evolutionary computation. This includes staff with interests in evolutionary optimisation with applications to planning and scheduling, visual image understanding, hybridising traditional approaches to operational research problems with optimisation and machine learning, and the use of fuzzy logic to build systems that can cope with uncertainty. These are applied to a wide range of domains, including transport, medicine, engineering, biosciences, and the digital economy.

Beyond this, the School has research expertise in programming languages, mixed reality, computer security, visualisation, robotics and cyber-physical systems, providing many opportunities for collaborative work.

Proposed Project

Fixing faults in software is one of the most important parts of the coding process, with estimates suggesting that around half of programmer's time is spent on finding and fixing faults. The recent growth of AI-aided programming makes fault finding and fixing even more salient. LLM systems such as Chat-GPT can produce good first drafts of program code, but usually with errors. Therefore, the combination of an LLM-based first draft generation together with a genetic improvement system to find and fix those bugs, has the potential to revolutionise automated programming.

In a recent paper [1] we proposed a novel way of identifying the location of faults in software. This is based on the idea that the execution of a correct line of code contributes some useful information to the solution of the problem. We can build a dataset by running the program on large numbers of tests cases and recording the state of the variables after each line has been executed. Then, we take a slice of this dataset for each line, and use a machine learning model to "bridge the gap" between the current state of the variables and the desired output from the program. We then measure the accuracy of this model; the key idea is that if the program is correct, this accuracy will steadily increase, because the problem-relevant information available as input to the model gets larger line-by-line as the program executes. This is illustrated in the diagram below.



If the program contains a fault, however, misleading information will be generated. Therefore, the accuracy of the model with cease to increase around the faulty line, or even decrease because of distracting information.

The aim of this project is to take this approach and add an additional feature, which is to correct the faults using a genetic improvement model [2,3]. The fitness function will be a measure of the monotonicity of the accuracy curve. As a stretch goal, this will be applied to code generated by Chat-GPT or similar LLMs. Code generated by the LLM for a particular problem will be analysed and faults corrected using the approach above.

Applicants should have good programming skills (ideally in Python), and a good understanding of core evolutionary computing concepts, particularly genetic programming. Some knowledge of ideas from software testing and genetic improvement of software would be useful, but is not necessary as it can be learned during the project.

[1] Colin G. Johnson, Software Fault Localisation via Probabilistic Modelling, *Artificial Intelligence XL, (proceedings of the Fortieth SGAI International Conference on Artificial Intelligence)*, Springer, Lecture Notes in Computer Science, December 2020. http://www.colinjohnson.me.uk/research/papers/Johnson_SGAI2020.pdf

[2] C. Le Goues, Dewey-Vogt, M., Forrest, S. and Weimer, W.: A systematic study of automated program repair: Fixing 55 out of 105 bugs for $8 each. In: Glinz, M., Murphy, G.C., Pezzè, M. (eds.) *International Conference on Software Engineering*, 2012 pp. 3–13. IEEE

[3] J. Petke, S. O. Haraldsson, M. Harman, W. B. Langdon, D. R. White and J. R. Woodward, "Genetic Improvement of Software: A Comprehensive Survey," in *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 415-432, June 2018, doi: 10.1109/TEVC.2017.2693219.

Host Institution—Practicalities

The scholarship will be held at the University of Nottingham, at our campus in Nottingham, in the midlands of England. Nottingham is a medium-sized city with a vibrant cultural, social and sporting life, and ready access the countryside and to other cities around the UK. Getting to Nottingham from around Europe is straightforward. The East Midlands and Birmingham airports are close-by, and London is around 1h40m from Nottingham, making access by air or international rail straightforward.

It is likely that residential accommodation will be available at reasonable cost on the campus.

The University has good facilities for research, including computing facilities (a GPU server and extensive HPC facilities), a library with subscriptions to a wide range of relevant publications in computer science, and extensive training programmes for researchers.