

Guest Editorial

Genetic Programming: An emerging engineering tool

Bob McKay^a, Shu-Heng Chen^b and Xuan Hoai Nguyen^b

^a*Department of Economics, National Chengchi University, Taiwan*

^b*School of Computer Science and Engineering, Seoul National University, Korea*

It is now 15 years since Genetic Programming (GP) burst on the research scene in Koza's seminal book [3], though it had been pre-figured by a number of previous explorations of similar ideas [1,7]. Koza first showed that GP could be a credible engineering tool, and subsequent volumes [4–6] and the human-competitive research competitions [2] have drawn attention to its wide range of achievements.

The papers in this issue were submitted in response to the Asia-Pacific Symposium on Genetic Programming (www.aspgp.org), which aims to bring together researchers and practitioners in Genetic Programming from the Asia-Pacific region and internationally, and promote research in and application of GP in the region. The papers both confirm the wide applicability of GP, and illustrate the continuing ferment in its development. Two papers are directly concerned with real-world applications, one uses GP to improve a general engineering method, while the final two emphasise algorithm improvements aimed at improving the applicability and scalability of GP.

Polimon et al. use GP to evolve block ciphers for cryptographic applications. They demonstrate the automated evolution of powerful ciphers. Their work demonstrates the strength of the ciphers that GP can generate, suggesting that GP could potentially generate competitive state-of-the-art ciphers. Equally important, the results suggest that GP can readily generate new, high-quality ciphers, inspiring a scenario in which ciphers might be application-specific, reducing the payoff for cracking any particular cipher, thereby

reducing the incentive to undertake cracking at all. Beyond the particular problem under consideration, the paper demonstrates the ability of GP to generate results comparable with those generated by competent human practitioners.

Andreae et al. use GP for automated stress detection in spoken English. The stress detection system is intended as a component of an automated learning support system for spoken English, aimed at detecting stress errors in the student's speech production, and providing feedback. However it potentially has much wider application, as stress detection is important not only for detecting errors, but also for interpreting the meta-linguistic content of speech utterances in natural language understanding. In this task, GP evolved rules which detected stress significantly more reliably than either decision trees (DT) or support vector machines (SVM), presumably because they were able to automatically make use of interactions between attributes, which were beyond the expressive reach of DT or SVM representation languages.

Pujol and Poli use GP to transform the decision space for parameter optimisation and tuning (POT) problems. Compared with standard methods, on a wide range of test sets, the GP-based approach yields better results, is more robust to perturbation of the test functions (though at cost of slower convergence), and more scalable than current methods. While this research is in its early stages, it appears that the use of GP to transform the space may represent a significant improvement in POT methods.

Briggs and O'Neill present a new GP representation based on typed combinatory expressions. This representation, not widely known outside the higher-order-function community, has very substantial advantages for GP. They demonstrate impressive performance on a range of tough problems. Especially noteworthy is their system's success on recursive problems – because of the difficulty of synchronising the use of variables in terminating and recursive expressions, these are extremely difficult for most GP systems. Their work promises to open up solutions for a whole range of problems which were previously viewed as insoluble – where the solution consists of a repeated or recursive structure whose exact form cannot be predicted a priori. Their use of typed combinatory expressions promises to revolutionise the field.

Hoang et al. argue that evaluation during development is the key difference between natural evolutionary systems and current artificial systems, permitting the natural systems to generate regular and scalable solutions to the problems of survival. In their developmental GP system, evaluation on a family of problems of steadily increasing difficulty leads to scalable solutions to difficult problems in the family, and to more regularly structured solutions. Regularly structured, and scalable, solutions will enable GP to tackle larger and more complex problems than have been possible in the past.

Together, these papers provide an insight into the current state of both research and practice in Genetic Programming, and demonstrate the added value it can bring to engineering and related fields.

References

- [1] N. Cramer, *A Representation for the Adaptive Generation of Simple Sequential Programs*, Proc. of an Intl. Conf. on Genetic Algorithms and their Applications, Carnegie-Mellon University, July 24–26, 1985
- [2] Human-Competitive Awards, <http://www.genetic-programming.org/hc2005/main.html>, retrieved on 19/10/2007.
- [3] J.R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA, USA, 1992.
- [4] J.R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, Cambridge Massachusetts, May 1994.
- [5] J.R. Koza, F.H. Bennett III, D. Andre and M.A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, 1999.
- [6] J.R. Koza, M.A. Keane, M.J. Streeter, W. Mydlowec, J. Yu and G. Lanza, *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Kluwer Academic Publishers 2003, ISBN 1-4020-7446-8.
- [7] D. Dickmanns, J. Schmidhuber and A. Winklhofer, *Der genetische Algorithmus: Eine Implementierung in Prolog*. Fortgeschrittenenpraktikum, Institut f. Informatik, Lehrstuhl Prof. Radig, Tech. Univ. Munich, 1987.