

LGP: A robust Linear Genetic Programming implementation on the JVM using Kotlin.

Jed Simson¹

¹ University of Waikato, Waikato, New Zealand

DOI: [10.21105/joss.01337](https://doi.org/10.21105/joss.01337)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Submitted: 09 February 2019

Published: 27 October 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Summary

The desire for a system which can automatically craft computer programs has been known in the machine learning community for some time. Friedberg (1958) experimented with a system that solved problems by randomly changing instructions in a program and favouring those changes which most frequently achieved a positive result.

Linear Genetic Programming (LGP) (Brameier & Banzhaf, 2007) is a paradigm of genetic programming that employs a representation of linearly sequenced instructions in automatically generated programs.

There are two primary features which differentiate LGP from a traditional tree-based approach: first, LGP programs exhibit a unique graph-based data flow due to the way the contents of a particular register may be used multiple times during a program's execution. This leads to program graphs with higher variability thus enabling program solutions which are more compact in comparison to tree-based solutions to evolve.

Secondly, special non-effective code coexists with a program's effective code as a result of the imperative structure. Non-effective code refers to instructions within an LGP program which do not impact the program output. These non-effective instructions guard the effective instructions from disruption caused by the genetic operator application and allows variations to remain neutral in terms of a fitness change.

LGP is a Kotlin package for performing Linear Genetic Programming, with a focus on modern design, ease of use, and extensibility. The usage of the Kotlin language enables a functional and modern API that can make full usage of the Java Virtual Machine and rich Java package ecosystem. The LGP core package is designed for the generic case, offering a rich set of extensible components that can be adapted to the particular problem and a set of different core implementations of various LGP algorithms. A sub-package `LGP-lib` provides implementations for core components, including program generation, program instruction operations, and base program definitions. A set of example usages is available to aid users in getting started solving problems.

The creation of LGP was motivated by the apparent lack of open-source Linear Genetic Programming implementations in an attempt to promote its usage. The project has intentions for use at the University of Waikato for future student projects and is currently in the process of being used for a research project in the image processing domain at the Max Planck Institute of Molecular Cell Biology and Genetics. The goal of LGP is to facilitate the usage of Linear Genetic Programming in solving problems in machine learning, as an alternative to other techniques.

References

Brameier, M. F., & Banzhaf, W. (2007). *Linear genetic programming*. Springer Science & Business Media.

Friedberg, R. M. (1958). A learning machine: Part i. *IBM J. Res. Dev.*, 2(1), 2–13. doi:[10.1147/rd.21.0002](https://doi.org/10.1147/rd.21.0002)