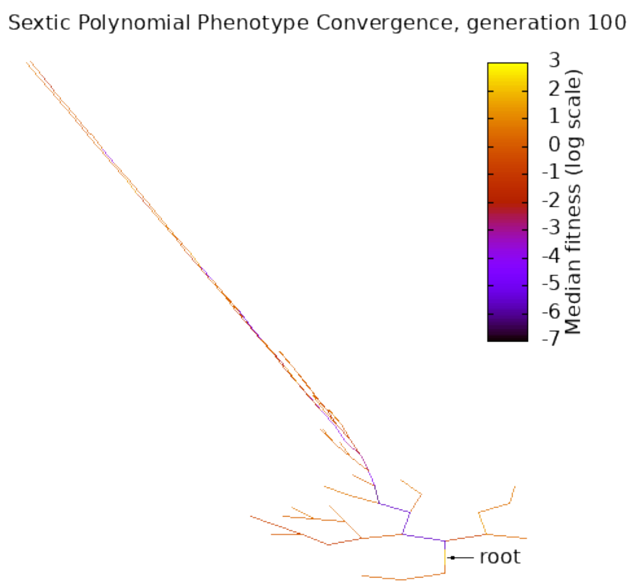
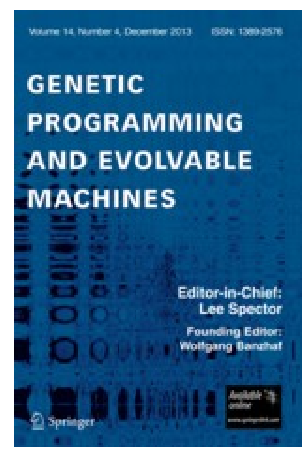
Genetic Programming Convergence

W. B. Langdon. 2022. GP & EM 23(1) 71–10

https://doi.org/10.1007/s10710-021-09405-9

W. B. Langdon





March 2022

22.6.2022

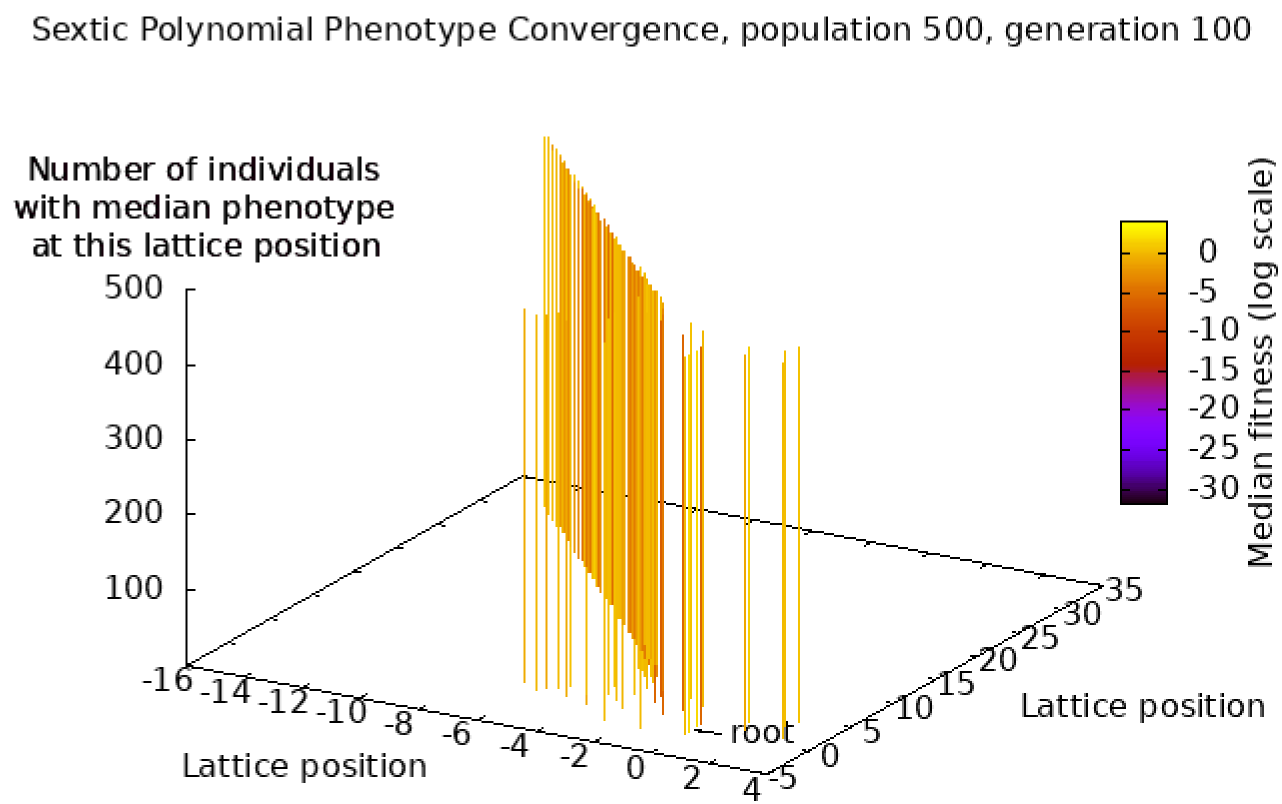
Genetic Programming Convergence

- Paper looks at symbolic regression tree over 100,000 generations.
- Evolution continues
  - Size and shape. Everyone is *unique* but
  - Convergence of fitness
  - Convergence of tree contents
  - **Convergence of tree node run time values**
  - Fraction introns 0.5% to 91%
  - Information theory applied inside GP
  - Very fast (trillion GP op/sec) fitness evaluation
  - Ideas for better crossover/mutation and representation

Convergence of tree node run time values

- Consider every node in every tree in the population
- Each is evaluated once per test case. Summarise this phenotypic information via the fitness function one value
- Every tree is *unique* but often in the population nearly all trees have a node at a given position.
- Often the contents of that node is the same in many trees
- Often the run time phenotype is the same in many trees.
- Graph height shows number of such trees. Colour gives nodes' median subfitness. Often interquartile range is zero.
- Some genetic/phenotypic variation (eg unique nodes) remains but have little impact on fitness.

Convergence of tree node run time values



10,000 generations video's url

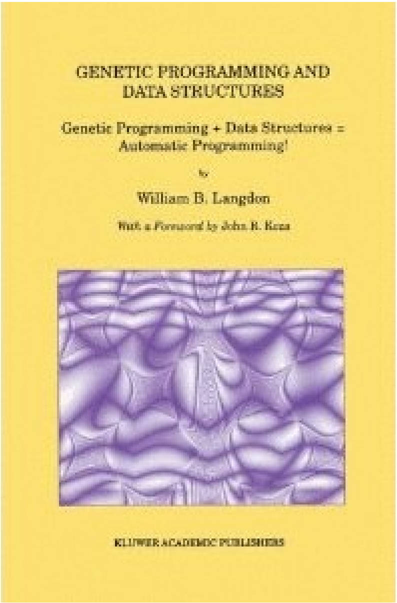
Conclusion Deep nesting hides crossover

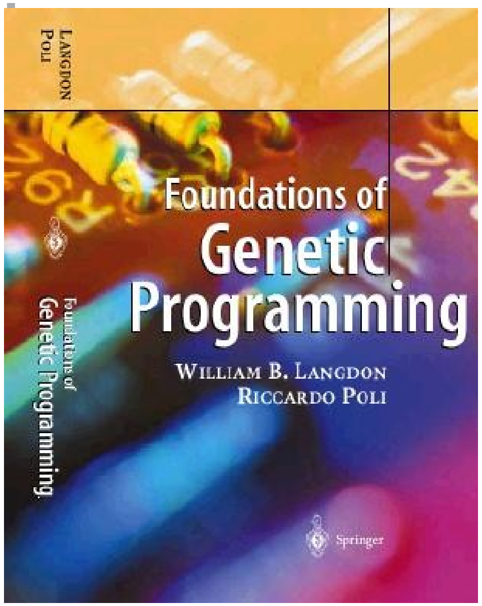
1) Strong selection drives convergence despite each program in the population being unique
2) Because evolved trees become deeper making crossover points deeper, giving longer paths from crossover disruption site to fitness effecting root node
3) Information theory shows longer paths are more susceptible to failed disruption propagation. So more daughters have same fitness as their mum's.
   - Design your new crossover & mutation operators
   - Design test set (here |x| > 1 more effective)
   - Consider function set as information flow (division losy)
4) Information loss gives smoother fitness landscape and evolution may slow but still continue

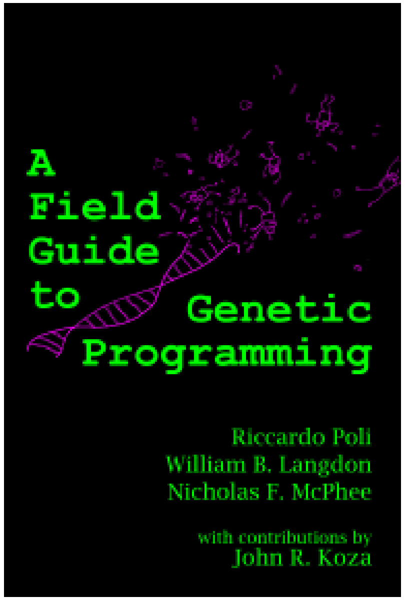
CREST EPSRC

Genetic Programming

W. B. Langdon







Issues

- Exponential decay in number of disrupted test cases suggests effectiveness of test suite of n tests rises only slowly with number of tests, Log(n)
  - With reasonable assumption this can be proved

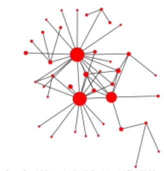
    [Measuring Failed Disruption Propagation in Genetic Programming, GECCO 2022]
- Some mutations not being totally concealed
  - Can we characterise them?
  - Should we use them more or less in GP?
  - Can we characterise the tests needed to find them
- How much does this generalise to other types of GP
- Can lessons on mutations and testing be used in Software Engineering

The Genetic Programming Bibliography

15582 references, [15000 authors]

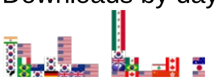
Make sure it has all of your papers!

E.g. email W.Langdon@cs.ucl.ac.uk or use | [Add to It] | web link



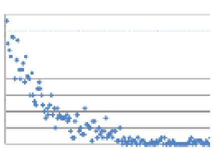
Co-authorship community.
Downloads

Downloads by day



A personalised list of every author's GP publications.

[blog]



Your papers

Googling GP bibliography, eg:
Development and learning site:gpbib.cs.ucl.ac.uk