# Incremental Evaluation in Genetic Programming

# EuroGP 2021 pp 229-246 doi:10.1007/978-3-030-72812-0_15
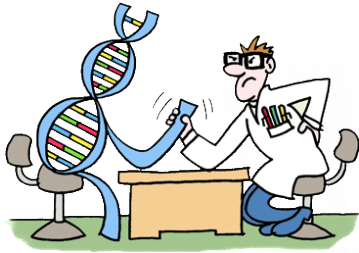
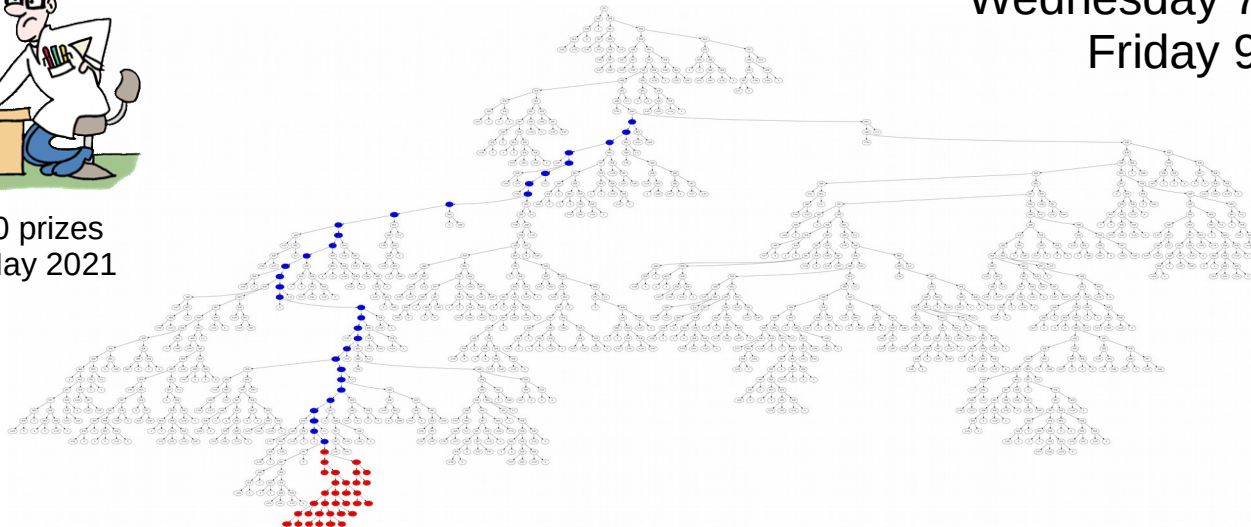## W. B. Langdon

Poster
 Wednesday 7 April 18:00
  Friday 9 April 7:45 am

Humies $10000 prizes
Submit by 28 May 2021

slide 1 of 8

1.4.2021

# Incremental Evaluation of Genetic Programming

- Many GP function and terminal sets have no side effects (they are pure functions).

- Such trees can be evaluated in any order.

- Evaluation from change point shows mum and child can be phenotypically similar even if they are genotypically different.

- Smooth landscape due to information loss

# Top Down = Bottom Up
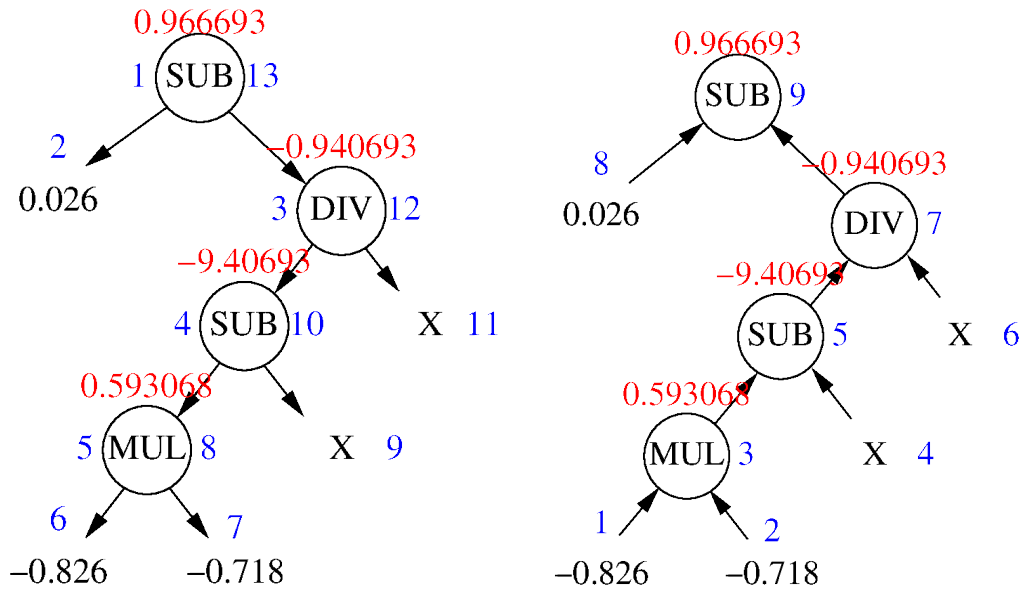
**Left**: Conventional top-down recursive evaluation of (SUB 0.026 (DIV(SUB (MUL -0.826 -0.718) X) X)). X=10.

Blue integers indicate evaluation order, red floats are node return values.

**Right**: an alternative ordering, starting with leaf -0.826 and working to root node.

Both return exactly the same answer.

# Genetic difference ≠> phenotypic difference

- Mum and child are identical except for inserted & removed subtrees.

- If by chance inserted & removed subtrees are identical:
  - mum and child are identical and so have the same fitness

- If inserted subtree evaluates to same value as removed subtree on every test case:
  - mum and child (at root node) evaluate to same value on every test case
  - genetic difference => identical fitness

- What if the inserted subtree evaluates to different values to that given by remove subtree?
  - If we evaluate both child and mum starting at the change, there is a progressive fall in the number of test cases where the change is visible as we move towards the root node.

# Evaluate both trees from change up

- Mum and child are identical above change.

- Fitness evaluation is identical except on route from change to root node.

- Evaluate both mum and child up this path.

- If they evaluate to identical values at any point then they evaluate to same value on the rest of the tree, including the root node:

  - semantic difference => identical fitness.
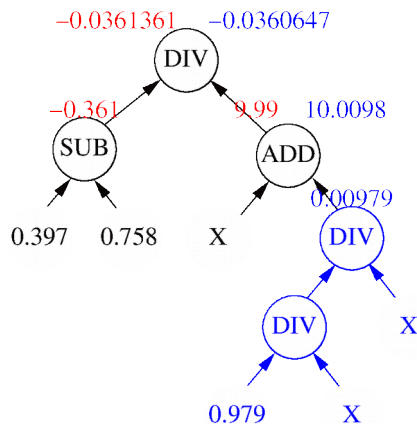


Evaluate mum in red. Evaluate child in blue. Inserted code (DIV (DIV 0.979 X) X) in blue. Here incremental evaluation proceeds 38 levels up the child tree before both mum and child evaluations are identical on all 48 test cases.

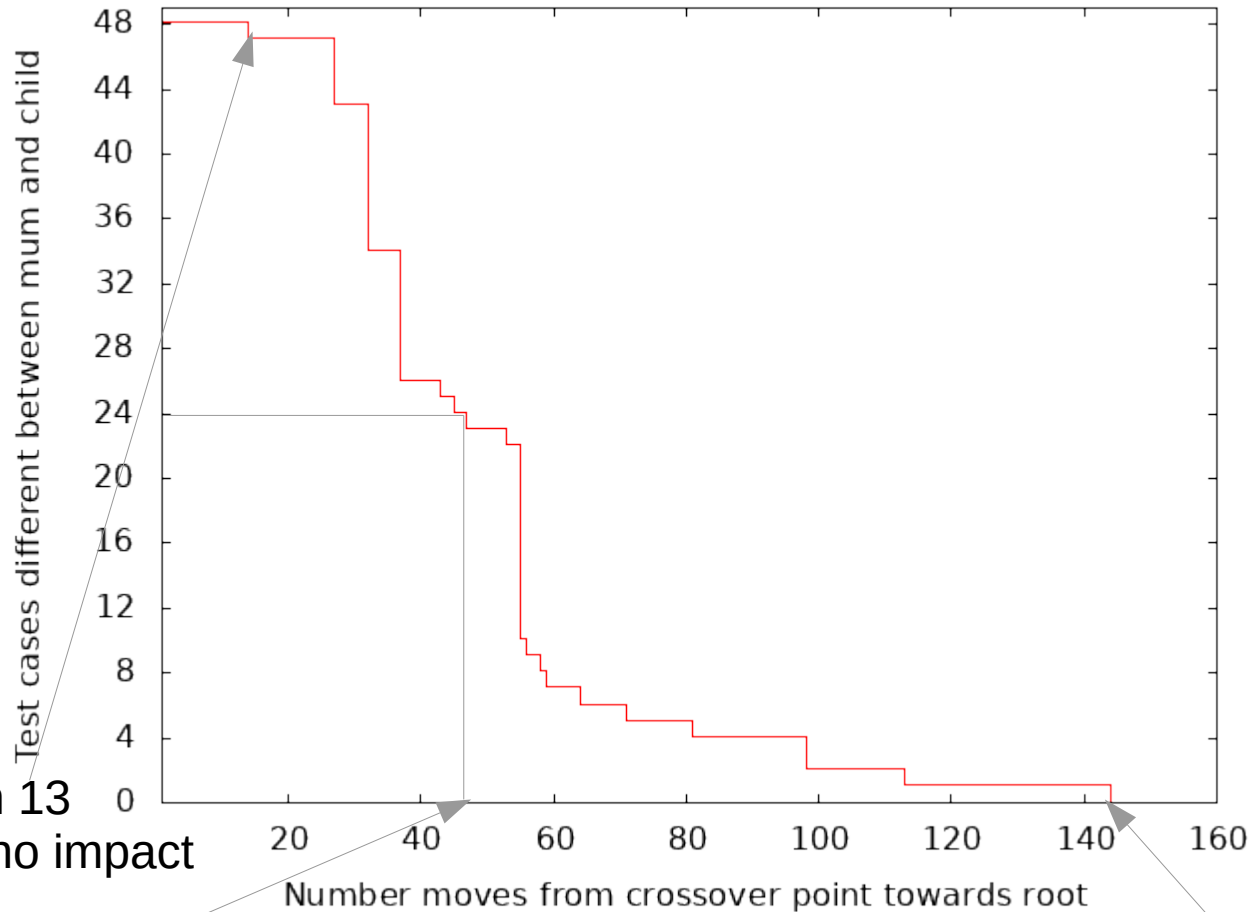Functions lose information and so can give same output even with different inputs.

# Evaluate both trees from change up

New code in red. Can stop fitness evaluation early as mum and child are phenotypically identical on all test cases.
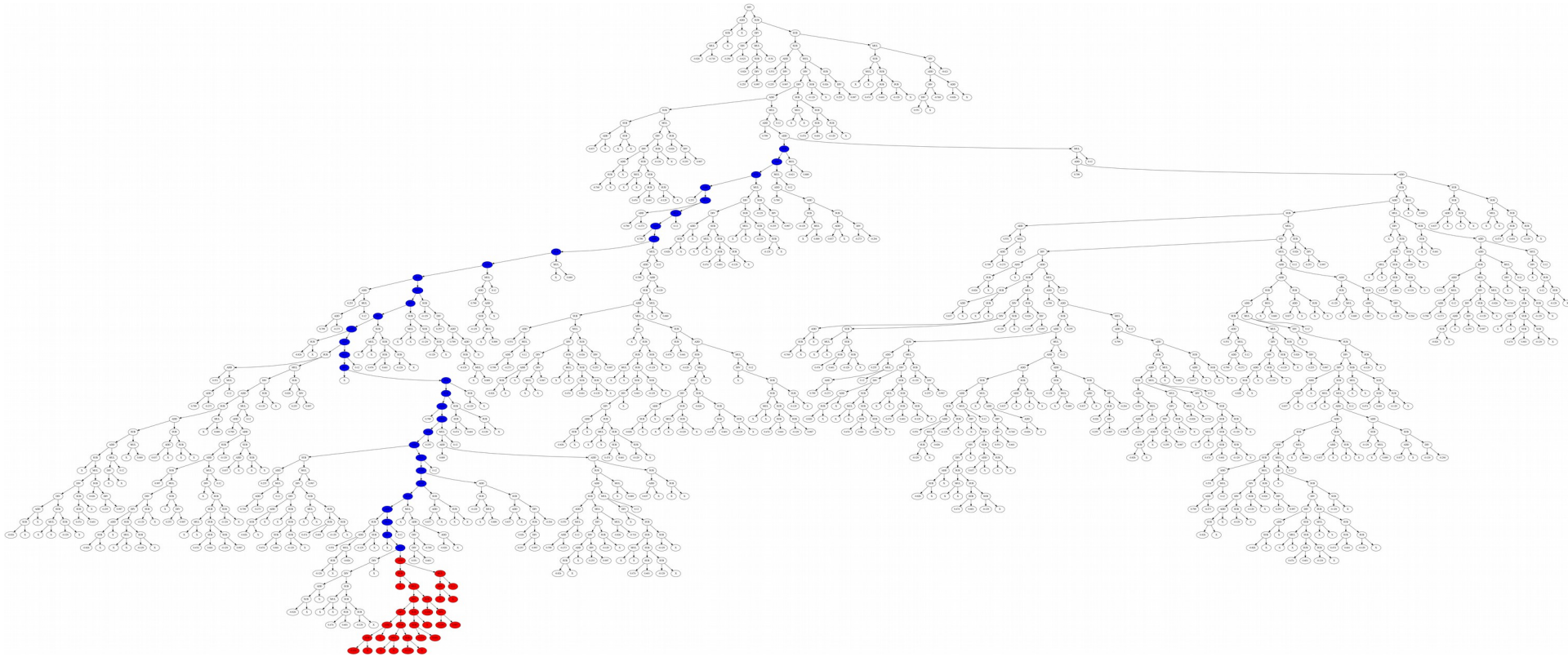
# Disruption Falls Monotonically

Deeper disruption tends to have less impact on fitness



Deeper than 13
3 tests see no impact

Deeper than 44
½ tests observe no impact

Deeper than 144
No impact at all

# **Conclusions**

## Phenotypic Disruption Falls Monotonically

- Deeper crossover or mutation may have less effect
  - Design your new crossover & mutation operators
  - Same should hold for mutation.
- If no disruption reaches root
  - fitness is identical => convergence
- If on some test cases, disruption does not reach root, genetic change may have less impact on fitness.
- (simple) model suggests adding more fitness test cases has only small effect, $O(\log(n))$
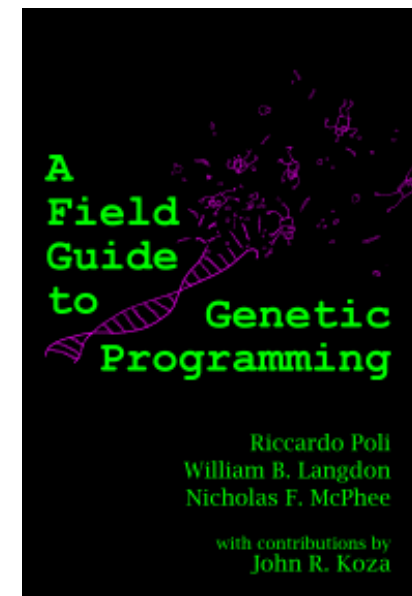- Dissipation of disruption (due to information loss) tends to give smoother fitness landscape.
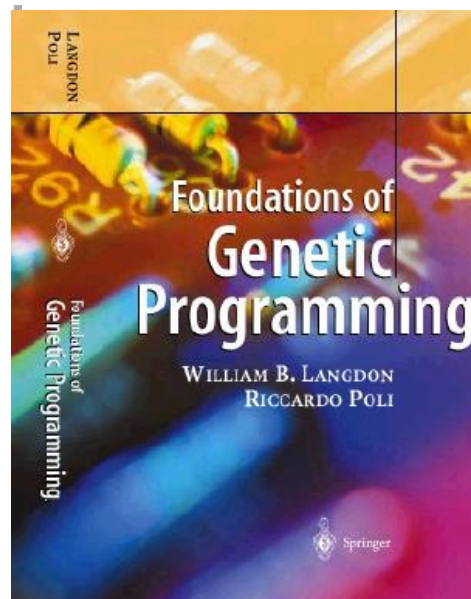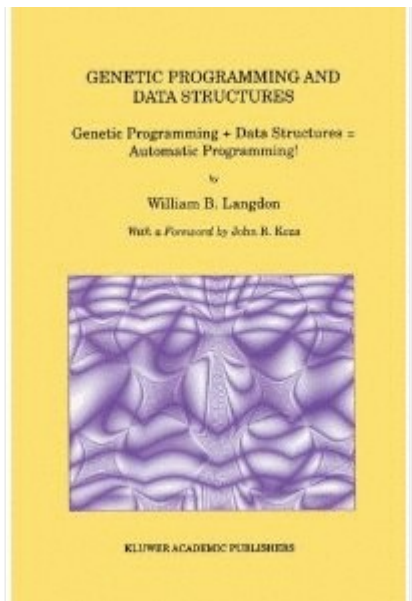
# END

W. B. Langdon YouTube videos
https://www.youtube.com/channel/UChebBvv66dPOcElWk6ht4OA

W. B. Langdon

# Genetic Programming

## W. B. Langdon

# Genetic Programming Benchmark



48 Fitness Cases, Sextic Polynomial x^2(x+1)^2(x-1)^2
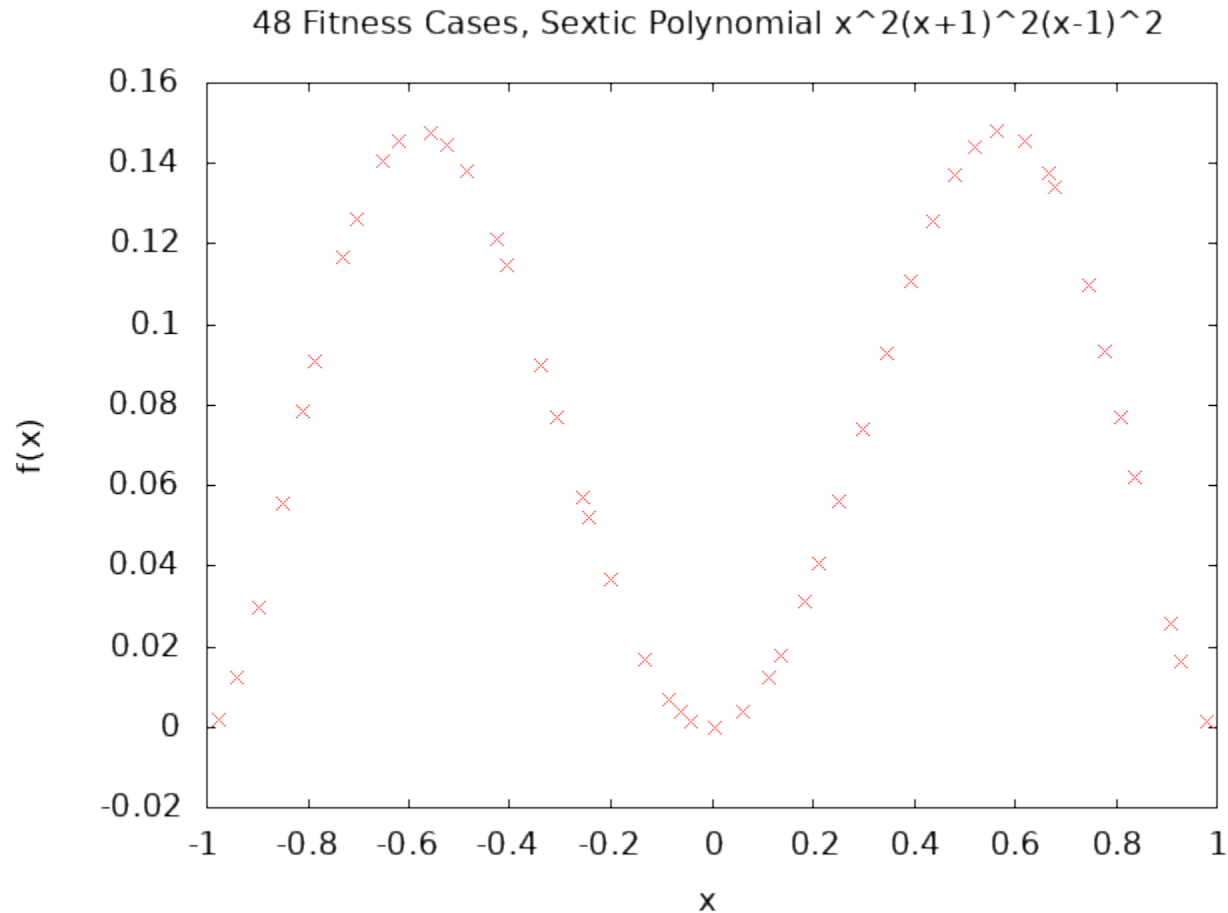
Sextic polynomial: match curve at 48 points

# The Genetic Programming Bibliography

**14414** references, <u>13000 authors</u>

**Make sure it has all of your papers!**
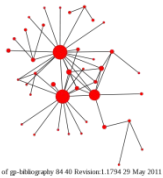E.g. email W.Langdon@cs.ucl.ac.uk   or   use | <u>Add to It</u> | web link

Part of gp-bibliography 84 40 Revision:1.1794 29 May 2011

Co-authorship community.
Downloads

Downloads by day

A personalised list of every author's
GP publications.

<u>blog</u>

Your papers

Googling GP Bibliography, eg:
steel site:gpbib.cs.ucl.ac.uk