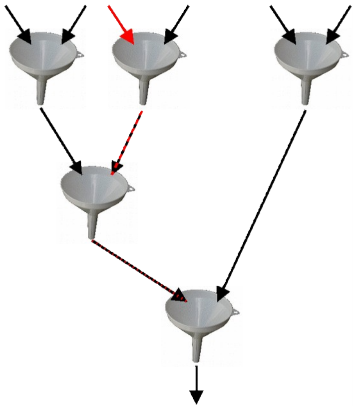
Failed Disruption Propagation in Integer Genetic Programming

W. B. Langdon Department of Computer Science, University College London

Failed Disruption Propagation (FDP) is when an error, defect, transient glitch etc., changes a running program internally but that disruption is *not spread* to the output, so the bug has *no effect*. Failed disruption propagation leads to software robustness and underlies coincidental correctness, correctness attraction, self-healing and anti-fragile software, code plasticity, software mutational robustness, and neutral variant rate. Information theory shows FDP is inherent in digital computation due to entropy loss. Fig 3 shows here the *optimal test oracle* should be within about *three* nested functions of the error, implying fault localisation must be good or many oracles will be required.

Failed disruption propagation means small changes typically do not have large impact. I.e. software is not chaotic.

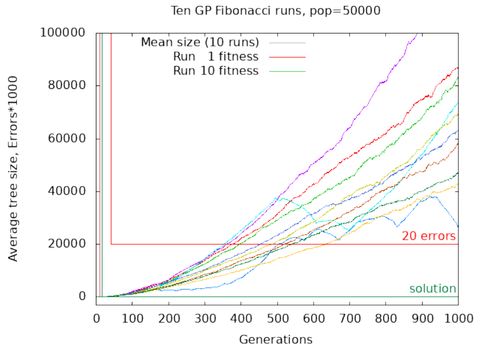
We detect failed disruption propagation in **deep GP** trees by **randomizing run time** evaluation at every node and tracing its impact. If unchecked, FDP prevents crossover and mutation changing GP outputs leading to fitness convergence and very **slow evolution**.



Information flow in five nested functions. A funnel takes two inputs each with up to 32 bits of information (total ≤64) but outputs at most 32 bits, giving potential information loss at each (irreversible) function. Disruption may fail to reach reach output. (No side effects.)

Table 1: GP to create deep fit Fibonacci trees for Failed Disruption Propagation (FDP) experiments

Terminal set:	J, 0, 1, 2, 3
Function set:	ADD SUB MUL SRF
Fitness cases:	First 20 members of the Fibonacci sequence.
Selection:	Fitness = $\sum_{J=0}^{19} \lvert GP(J) - Fibonacci_J \rvert$. I.e. the sum of the absolute error between GP's answer and the value of the J^{th} member of the Fibonacci sequence. Tournament size 7.
Population:	Panmictic, non-elitist, generational.
GP parameters:	Initial population of 50 000 trees created by ramped half and half with depth between 2 and 6. 100% unbiased subtree crossover. 1000 generations. No size or depth limit.



INFORMATION THEORY OF GP CONVERGENCE

All functions loose information. Without side effects, lost information cannot be restored. Disruption passes up tree but once lost on a test case cannot be restored. In deep trees impact does not reach root. Hence child behaves identically to its mother and therefore has the same fitness. Deep trees give GP a smooth landscape. Relatively insensitive, order log n, to number of test cases.

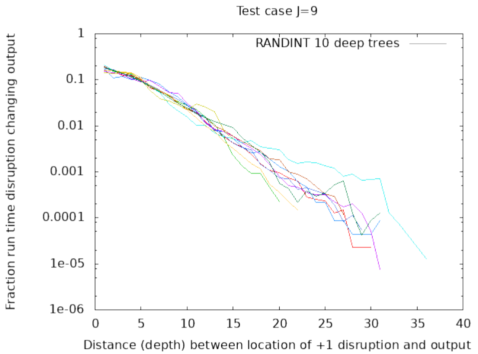


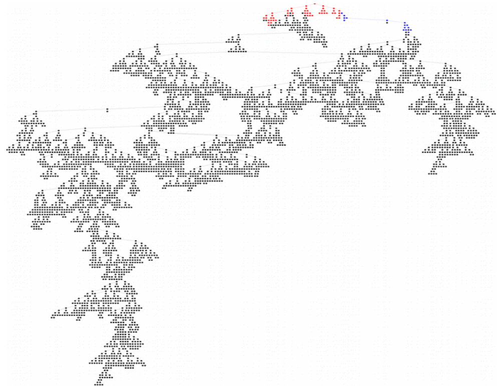
Figure 3 How far RANDINT disruption travels up tree for middle training case (J=9). Exponent -0.33 to -0.20

Table 2: Ten Deep Fit GP Fibonacci Trees

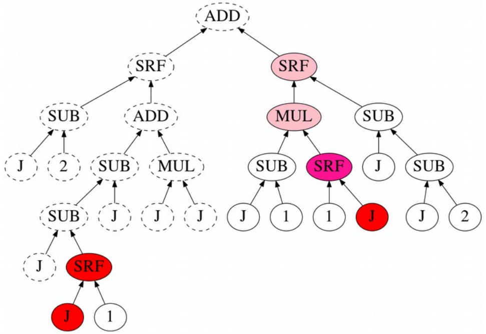
Size	Depth	Σ|error|	Disruption	exponent
86035	663	20	0.092 %	-0.31
4347	160	10	1.449 %	-0.33
23289	220	184	3.053 %	-0.27
131159	449	130	0.121 %	-0.29
77479	454	632	0.256 %	-0.20
51697	626	0	0.056 %	-0.27
771	33	0	7.523 %	-0.22
35727	425	0	0.073 %	-0.30
53305	485	0	0.032 %	-0.33
23377	360	0	0.137 %	-0.26

Speed *equivalent* of *up to* 5 billion GP operations per sec on 3.6 GHz i7 4790 desktop (1 core, no AVX)

1. W.B. Langdon & J. Petke. 2015. Software is Not Fragile
2. Justyna Petke, David Clark, W.B. Langdon. 2021. Software Robustness...FSE 2021. 1475–1478.
3. WBL EuroGP 2021, Incremental Evaluation in GP
4. Measuring Failed Disruption Propagation in GP. GECCO 2022, DOI:10.1145/3512290.3528738
5. 11 page version, ArXiv 2204.13997



Run 2, tree size 4347, depth 160. Randomised evaluation only changes fitness on coloured nodes: red 6-20 test cases changed, blue 1-2 test cases.



An evolved Fibonacci solution. The arrows show the information flow from the leafs to the root node (top oval). When the bottom most J node is artificially perturbed by +1, the disruption only reaches the calling SRF function, whose output does not change.

w.langdon@cs.ucl.ac.uk