

Analysis of Ancestry in Genetic Programming with a Graph Database

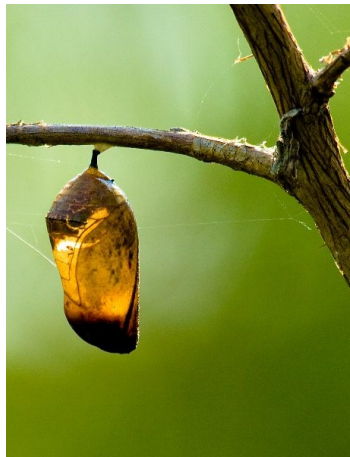
David Donatucci
M. Kirbie Dramdahl
Nicholas Freitag McPhee

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

25 April 2014
MICS, Verona, WI

The Big Picture

- The power of graph databases: analyzing internal data of GP
- ????????????
- Even fewer allow for plasticity during development????
- N-gram GP has natural developmental phase?????
- Can we find useful information about runs?



Bluedrakon

<http://tr.im/pWUi>

Outline

- 1 Genetic Programming
- 2 Graph Database
- 3 Experimental Setup
- 4 Results
- 5 Conclusions

Outline

- 1 Genetic Programming
 - GP Overview
 - Symbolic Regression and Fitness
- 2 Graph Database
- 3 Experimental Setup
- 4 Results
- 5 Conclusions

Genetic Programming Overview

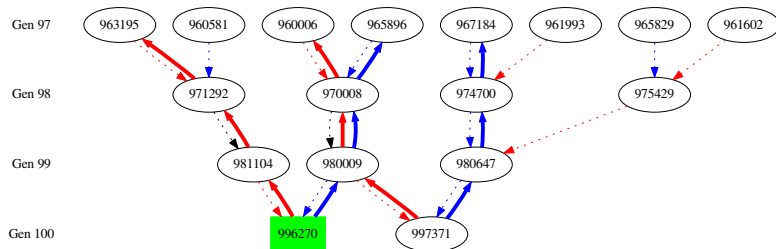
- Genetic Programmings is based upon biological principles.
- Individuals form a population.
- Transformations
 - Crossover (XO)
 - Mutation
 - Reproduction
 - Elitism
- Transformations occur over a specified amount of generations.
- Individuals are rated by their fitness.



Sam Fraser-Smith
<http://tr.im/pq71>

Transformations

- (XO) sexual reproduction (root and non-root)
- mutation subtrees altered
- reproduction asexual reproduction
- elitism reproduction based on fitness



Symbolic Regression and Fitness

We are focusing on symbolic regression problems.

- Measured data fitted to mathematical formula.
- Collection of test points to evolve individuals.

Fitness determines individual's distance from target function.

- Lower the fitness, the better the individual.
- A zero fitness would exactly match test data.
- Anything else to add??????????

The goal of GP is to evolve an individual with a fitness as low as possible.

Outline

1 Genetic Programming

2 Graph Database

- Neo4j
- Cypher

3 Experimental Setup

4 Results

5 Conclusions

Neo4j

Neo4j is a graph database.

- Relatively new tool (initial release 2007 popularized in 2010).
- Information is stored like a graph.
- Nodes and relationships.
- Efficient recursive queries.

Cypher

Neo4j's query language is Cypher.

Fundamental elements of Cypher queries:

- START
- RETURN
- MATCH
- WHERE

Outline

1 Genetic Programming

2 Graph Database

3 Experimental Setup

- Configurations
- Methods

4 Results

5 Conclusions

Run Configurations

Function $\sin(x)$

Variables x (range from 0.0 to 6.2, incremented by steps of 0.1)

Constants range between -5.0 and 5.0

Operations addition (+), subtraction (-), multiplication (*), protected division (/)

Number 100

Iteration 1000 (6 runs) and 10000 (1 run)

Stages Crossover (90%), Mutation (1%), Reproduction (9%)

Elitism best 1%

Methods

- fitness** Absolute error between target function and individual function.
- PTC2** Randomly adds operators to array of specified length (empty slots for arguments where appropriate). Empty slots divided between variables (63%) and constants (37%).
- Type** Subtree Mutation

Outline

1 Genetic Programming

2 Graph Database

3 Experimental Setup

4 Results

- Questions
- Improved Transformations
- Fitness Over Time
- Common Ancestor
- Modularity and repeated structures in IFD

5 Conclusions

Questions Asked

- 1 *How often do mutations improve fitness? Also, how often do crossovers improve fitness, where the root parent is more fit than the non-root parent, and vice versa?*
- 2
- 3 *Do a group of individuals have a common root parent ancestor and what is the latest generation where such an ancestor occurs?*
- 4 *How many individuals in the initial generation have any root parent descendants in the final generation?*

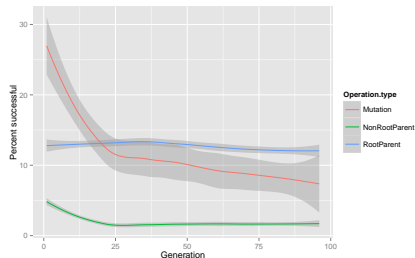
Percentage of Improved Transformations

How often do mutations and crossovers improve fitness?

Results for 10000 Individual Run

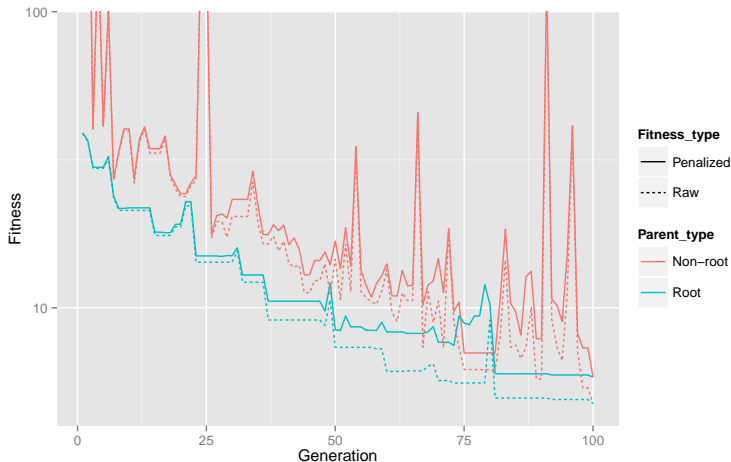


Results for Three 1000 Individual Runs

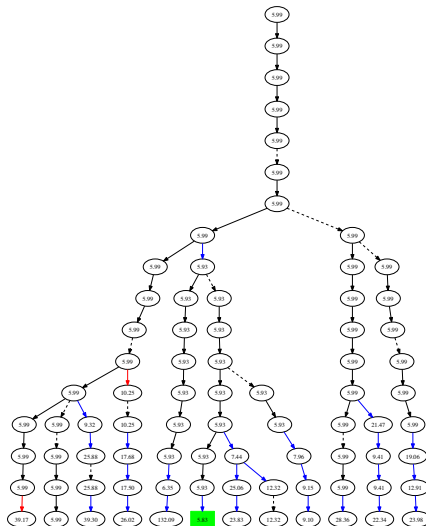


Fitness Over Time

What does the fitness of the “winning” root parent ancestry line look like over time?



Common Ancestor



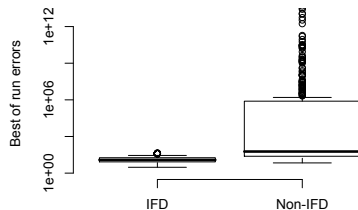
Success rates on 11 test problems

Label	Function	<i>Successes out of 100 runs</i>		
		TinyGP	N-gram	IFD
P1	$x + x^2 + x^3 + x^4 + x^5$	100	100	100
P2	$-x - 2x^2 + x^3$	100	100	100
P3	$1.009 + 1.419x + x^2$	100	61	100
P4	$6 + x^2 + 3x^3 + 8x^5$	0	0	0
P5	6	100	100	100
P6	$6 + x^2$	100	10	94
P7	$6 + x^2 + 3x^3$	85	0	1
P8	$8x^5$	100	100	100
P9	$3x^3 + 8x^5$	22	55	100
P10	$x^2 + 3x^3 + 8x^5$	100	7	80
Sine	$\sin(x)$	0	1	63

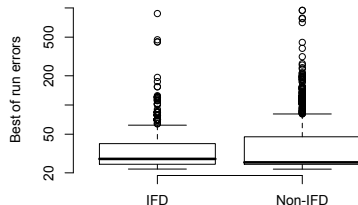
IFD wins either way

- IFD generates low-error individuals from tables evolved with IFD **and without IFD**.
- IFD's local search is valuable in all phases of the process, even if it wasn't used previously.
- N-gram GP isn't able to work effectively with the more complex probability tables that IFD generates.

Errors for programs generated from IFD matrix using both methods

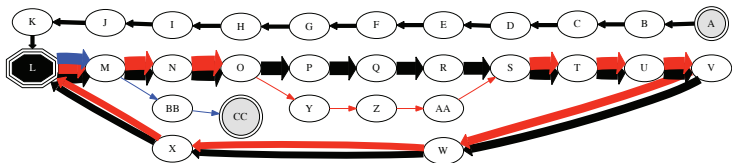


Errors for programs generated from Non-IFD matrix using both methods

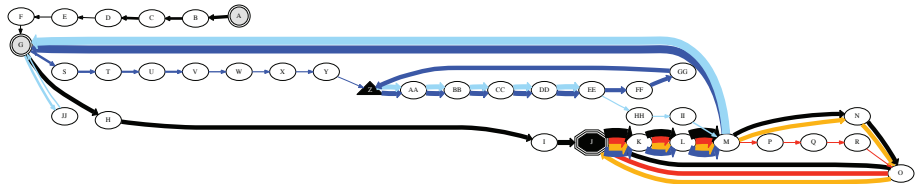


Structural differences and modularity

Standard N-gram GP tends to converge to a small set of loops with high probability edges.



With IFD there is less convergence, more variety and complexity in the modular structure, & greater use of low probability edges.



Outline

- 1 Genetic Programming
- 2 Graph Database
- 3 Experimental Setup
- 4 Results
- 5 Conclusions**

Conclusions

- We can gather internal data!
- Provides more in depth information than statistical summaries.
- Support for hypotheses.

Future work

- Trying different setup configurations.
- Enforcing the root parent to have better fitness in XO.
- Dynamically change parameters.

Thanks!

Thank you for your time and attention!

Contact:

- `mcphee@morris.umn.edu`
- `http://www.morris.umn.edu/~mcphee/`

Questions?

References



N. F. McPhee, E. Crane, S. Lahr, and R. Poli.

Developmental Plasticity in Linear Genetic Programming.

In Günther Raidl, *et al*, editors, *GECCO '09*, pages 1019–1026, Montréal, Québec, Canada, 2009.



R. Poli and N. McPhee.

A linear estimation-of-distribution GP system.

In M. O'Neill, *et al*, editors, *EuroGP 2008*, volume 4971 of *LNCS*, pages 206–217, Naples, 26–28 Mar. 2008. Springer.

See the GECCO '09 paper for additional references.