

# Analysis of Ancestry in Genetic Programming with a Graph Database

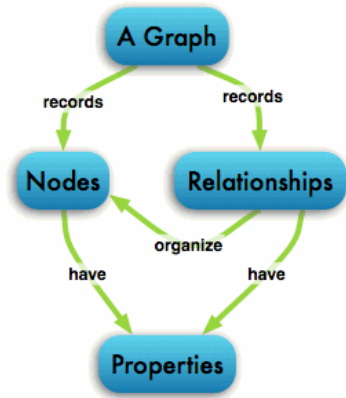
David Donatucci  
M. Kirbie Dramdahl  
Nicholas Freitag McPhee

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

25 April 2014  
MICS, Verona, WI

# The Big Picture

- Genetic programming demonstrated to be effective for a variety of applications.
- Difficult to determine how this process works.
- Databases allow examination of the internal interactions of a run.
- Graph databases more efficient at this task than relational databases.
- This knowledge may be used to improve genetic programming algorithms.



Neo4j

<http://www.neo4j.org/learn/graphdatabase>

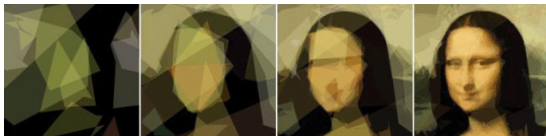
# Outline

- 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



Roger Alsing

<http://io9.com/5106124/a-computer-program-that-taught-itself-to-draw-the-mona-lisa>

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

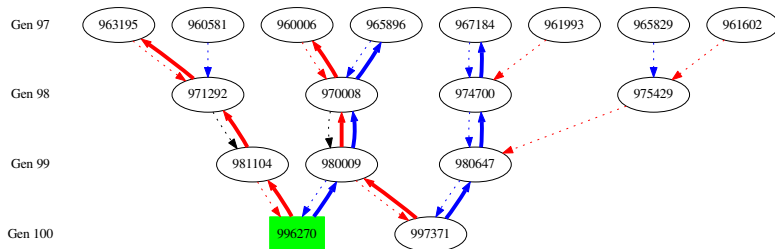
# Transformations

Crossover (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.

- Collection of test points as input.
- Evolve mathematical formula to fit data.

Fitness determines individual's distance from target function.

- Lower the fitness, the better the individual.
- A fitness of zero would exactly match test data.

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

# Outline

1 Genetic Programming

2 Graph Database

- Neo4j
- Cypher

3 Experimental Setup

4 Results

5 Conclusions



# Neo4j

Neo4j is a graph database.

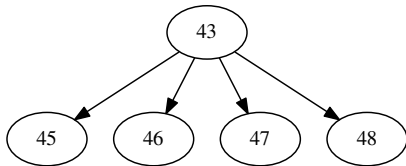
- relatively new tool
  - initial release 2007
  - popularized in 2010
- information is stored using a graph
- nodes and relationships
- efficient recursive queries compared with traditional databases

# Cypher

Neo4j's query language is Cypher.

Fundamental elements of  
Cypher queries:

- START
- RETURN
- MATCH
- WHERE



```
START parent=node(43)  
MATCH (parent)-[:PARENTOF]->(child)  
RETURN parent, child;
```

# Outline

1 Genetic Programming

2 Graph Database

3 Experimental Setup

- Configurations

4 Results

5 Conclusions

# Run Configurations

Target Function  $\sin(x)$

Variables  $x$  (range 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 (/)

Generation Number 100

Population Size Per Gen 1000 (6 runs) and 10000 (1 run)

Transform Percentages crossover (90%), mutation (1%), reproduction (9%)

Elitism best 1%

Fitness absolute error between target function and  
individual function

# Outline

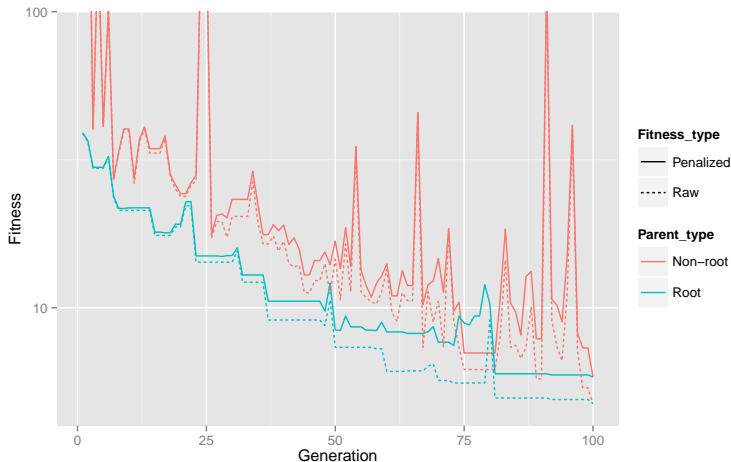
- 1 Genetic Programming
- 2 Graph Database
- 3 Experimental Setup
- 4 Results**
  - Questions Asked
  - Fitness Over Time
  - Improved Transformations
  - Common Ancestor
- 5 Conclusions

# Questions Asked

- 1 *What does the fitness of the “winning” root parent ancestry line look like over time?*
- 2 *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?*
- 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?*

# Fitness Over Time

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



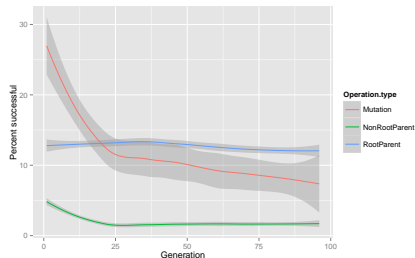
# Percentage of Improved Transformations

*How often do mutations and crossovers improve fitness?*

Results for 10000 Individual Run



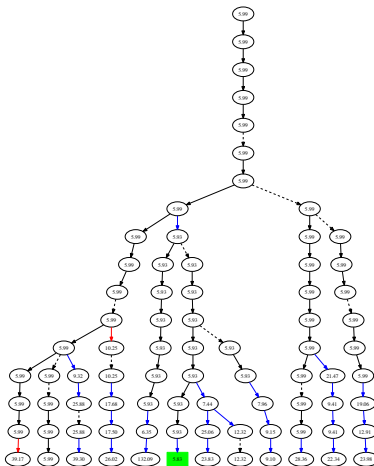
Results for Three 1000 Individual Runs





# Common Ancestor

*Do a group of individuals have a common root parent ancestor and what is the latest generation where such an ancestor occurs?*



# 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!

Contacts:

- `donat056@morris.umn.edu`
- `dramd002@morris.umn.edu`
- `mcphee@morris.umn.edu`

## Questions?

# References



LUKE, S.

*Essentials of Metaheuristics*, second ed.

Lulu, 2013.

Available for free at [http://cs.gmu.edu/~sim\\$sean/book/metaheuristics/](http://cs.gmu.edu/~sim$sean/book/metaheuristics/).



MCPHEE, N. F., AND HOPPER, N. J.

Analysis of genetic diversity through population history.

In *Proceedings of the Genetic and Evolutionary Computation Conference* (1999), vol. 2, Citeseer, pp. 1112–1120.



POLI, R., LANGDON, W. B., AND MCPHEE, N. F.

*A Field Guide to Genetic Programming*.

Published via <http://lulu.com> and freely available at <http://www.gp-field-guide.org.uk>, 2008.

(With contributions by J. R. Koza).



ROBINSON, I., WEBBER, J., AND EIFREM, E.

*Graph Databases*.

O'Reilly Media, Inc., 2013.