COMP390: Evolving a Sorting Algorithm with SNGP

Robin Lockyer

University of Liverpool

03/2019

Introduction

Overview of Genetic Programming

Overview of Single Node Genetic Programming

Reproducing Kinnear's Results

Attempting SNGP

Conclusion

Bibliography

Project Description

Aims:

- Replicate K. E. Kinnear's work [2] in evolving a sorting algorithm using Genetic Programming

Project Description

Aims:

- Replicate K. E. Kinnear's work [2] in evolving a sorting algorithm using Genetic Programming
- Re-implement Kinnear's work using Single Node Genetic Programming, a variant of GP invented by Dr David Jackson [1]

Project Description

Aims:

- Replicate K. E. Kinnear's work [2] in evolving a sorting algorithm using Genetic Programming
- Re-implement Kinnear's work using Single Node Genetic Programming, a variant of GP invented by Dr David Jackson [1]
- Compare the effectiveness of the two approaches to evolving a sorting algorithm

What Was Achieved?

Successfully replicated Kinnear’s work

What Was Achieved?

Successfully replicated Kinnear's work

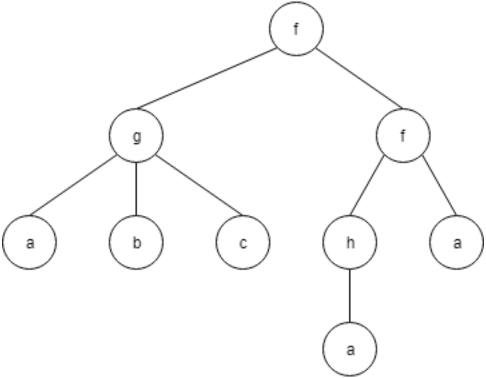
Unable to evolve a sort using SNGP

What is genetic programming?

Genetic programming is applying genetic algorithms to programmes in order to generate a programme that performs well in a given problem domain.

How Does GP Work? - 1

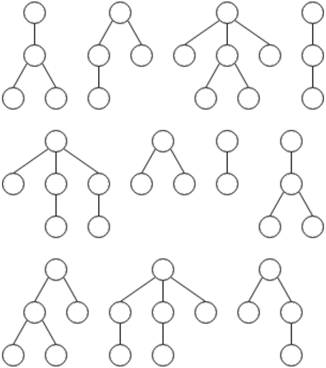
Programmes are encoded as a tree of primitive functions and terminals



This tree encodes the programme f(g(a, b, c), f(h(a), a)), where f, g, and h are functions and a, b, and c are terminals.

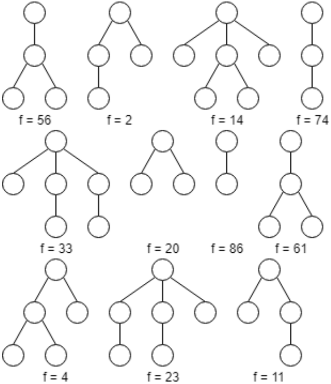
How does GP Work? - 2

An initial population of random programmes is created



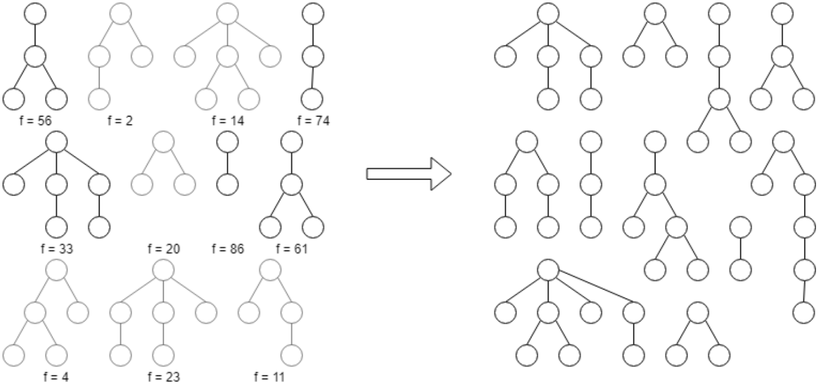
How Does GP Work? - 2

An initial population of random programmes is created



Each member of the population is executed, evaluated, and given a fitness score

How Does GP Work? - 3



A new population is created by selecting some of the most fit members of the initial population and performing genetic operations on them to create new programmes

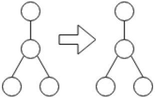
How Does GP Work? - 4

There are three main genetic operators:

How Does GP Work? - 4

There are three main genetic operators:

Reproduction

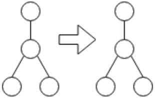


A programme is copied over to the new population without any changes

How Does GP Work? - 4

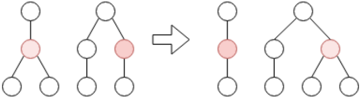
There are three main genetic operators:

Reproduction



A programme is copied over to the new population without any changes

Crossover

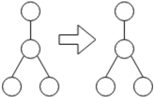


A random node is selected in each of the chosen programmes. The subtrees rooted at the selected nodes are swapped.

How Does GP Work? - 4

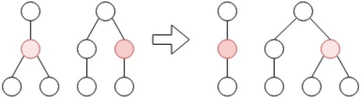
There are three main genetic operators:

Reproduction



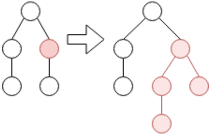
A programme is copied over to the new population without any changes

Crossover



A random node is selected in each of the chosen programmes. The subtrees rooted at the selected nodes are swapped.

Mutation



A random node is selected in the chosen programme. A new, random subtree is generated to replace the subtree rooted at the selected node.

How Does GP Work - 5

This process is repeated until a programme with high enough fitness is generated

What is SNGP?

What is SNGP?

- SNGP is a variation of GP that organises the whole population into a single interlinked graph

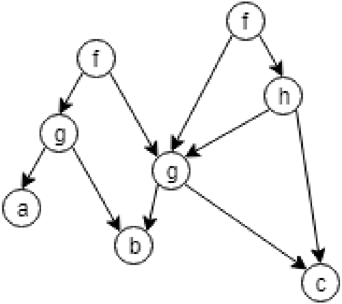
What is SNGP?

- SNGP is a variation of GP that organises the whole population into a single interlinked graph
- The subtree rooted at each node in the graph is considered to be an individual programme

What is SNGP?

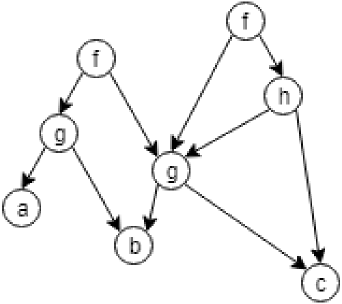
- SNGP is a variation of GP that organises the whole population into a single interlinked graph
- The subtree rooted at each node in the graph is considered to be an individual programme
- The graph is structured in such a way that a form of dynamic programming can be used to increase the efficiency of evaluating the population

SNGP Population - 1



	p
p[7]	f
p[6]	f
p[5]	h
p[4]	g
p[3]	g
p[2]	a
p[1]	b
p[0]	c

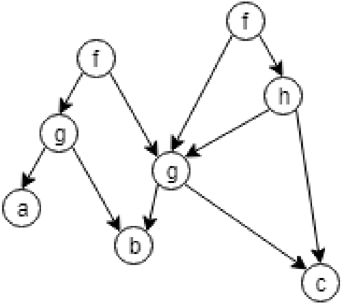
SNGP Population - 1



	p
p[7]	f
p[6]	f
p[5]	h
p[4]	g
p[3]	g
p[2]	a
p[1]	b
p[0]	c

- Graph nodes are stored in an array

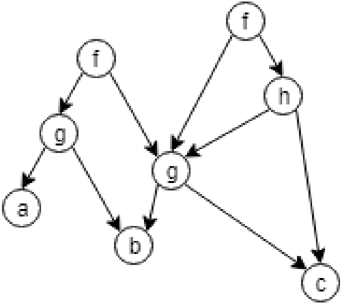
SNGP Population - 1



	p
p[7]	f
p[6]	f
p[5]	h
p[4]	g
p[3]	g
p[2]	a
p[1]	b
p[0]	c

- Graph nodes are stored in an array
- Terminals are stored in lowest elements

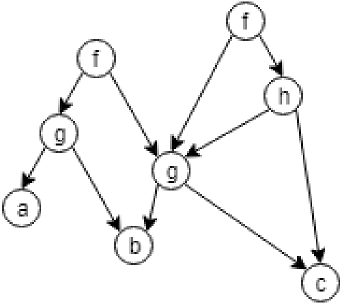
SNGP Population - 1



	p
p[7]	f
p[6]	f
p[5]	h
p[4]	g
p[3]	g
p[2]	a
p[1]	b
p[0]	c

- Graph nodes are stored in an array
- Terminals are stored in lowest elements
- Remaining elements store random function

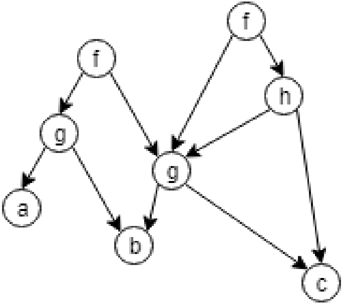
SNGP Population - 1



	p
p[7]	f
p[6]	f
p[5]	h
p[4]	g
p[3]	g
p[2]	a
p[1]	b
p[0]	c

- Graph nodes are stored in an array
- Terminals are stored in lowest elements
- Remaining elements store random function
- Each functions operands are chosen from elements with a smaller index

SNGP Population - 2



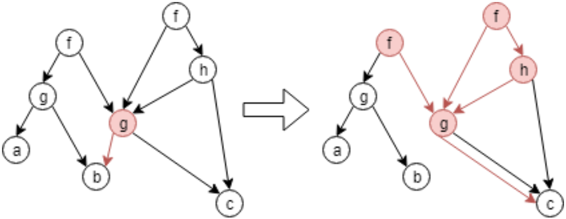
This graph contains the following programmes:

- a
- b
- c
- g(b, c)
- g(a, b)
- h(g(b, c), c)
- f(g(a, b), g(b, c))
- f(g(b, c), h(g(b, c), c))

SNGP Operators

SNGP has only one genetic operator:

Successor Mutate



Bibliography

[1] Jackson, D.
A new, node-focused model for genetic programming.
In *Genetic Programming* (2012), A. Moraglio, S. Silva, K. Krawiec, P. Machado, and C. Cotta, Eds., Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp. 49–60.

[2] Kinnear, K. E.
Evolving a sort: Lessons in genetic programming.
In *in Proceedings of the 1993 International Conference on Neural Networks* (1993), IEEE Press, pp. 881–888.