# A Data Structure for Efficient Analysis of Genetic Programs

*The iTree and its application*

Anikó Ekárt

Computer and Automation Research Institute

Hungarian Academy of Sciences

ekart@sztaki.hu

# Outline of talk

1. Introduction

# Outline of talk

1. Introduction

2. The information hyper-tree (iTree)

# Outline of talk

1. Introduction

2. The information hyper-tree (iTree)
   - The construction of the iTree

# Outline of talk

1. Introduction

2. The information hyper-tree (iTree)
   - The construction of the iTree
   - Population measures based on the iTree

# Outline of talk

1. Introduction

2. The information hyper-tree (iTree)
    - The construction of the iTree
    - Population measures based on the iTree

3. Case study

# Outline of talk

1. Introduction

2. The information hyper-tree (iTree)
   - The construction of the iTree
   - Population measures based on the iTree

3. Case study

4. Conclusions & future directions

# Introduction

Our goal is to make exploratory analysis beyond simple measures more accessible

We introduce a data structure (the iTree) that

- is efficient to maintain
- offers a compact view on a population of tree structured genetic programs
- allows for the efficient computation of many population measures

We use the iTrees in comparing simple GP with fitness sharing

# The information hyper-tree

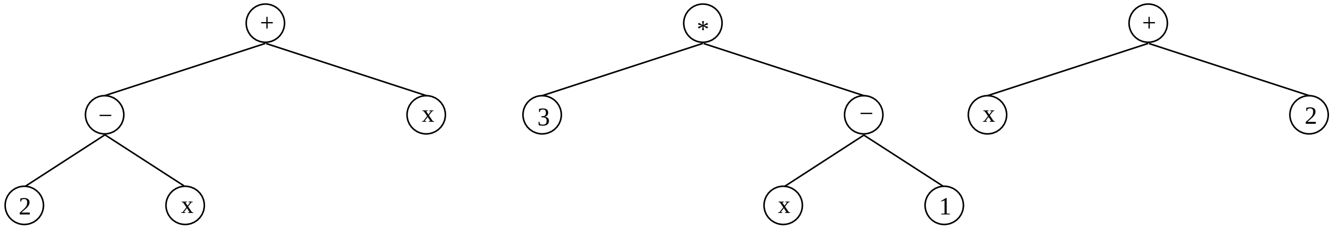
A data structure that collects important details of individuals in a population in one easily accessible place

1. The structure of the iTree must be such that it incorporate the structure of any tree in the population
2. Each node of the iTree should capture the population information related to that particular node position

The iTree can be constructed for any set of genetic trees

# Example iTrees

**Population $P_1$ :**

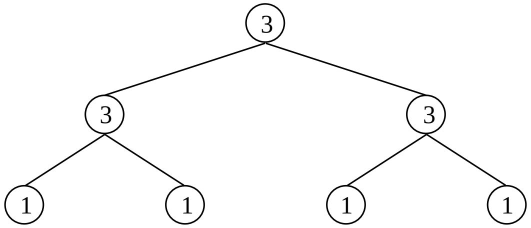


iTree:

$M_1$ =7

$M_2$ =13

$M_3$ =7



ED diversity=5.33

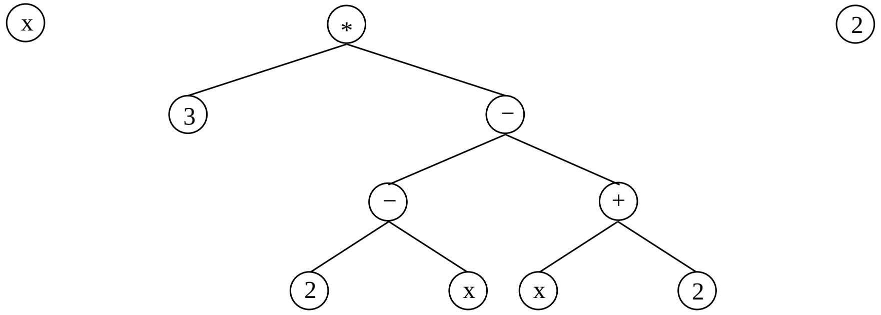
SC diversity=2.33

$IB_1$ =0

$IB_2$ =0

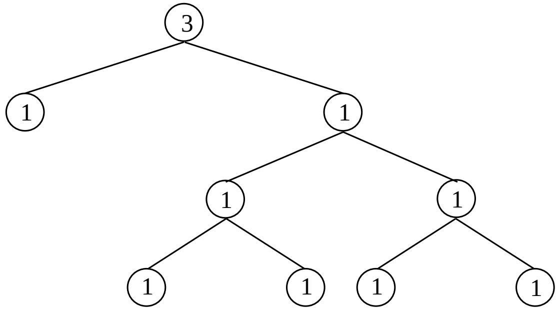
# Example iTrees

**Population $P_2$:**



iTree:

$M_1$ =9
$M_2$ =11
$M_3$ =5

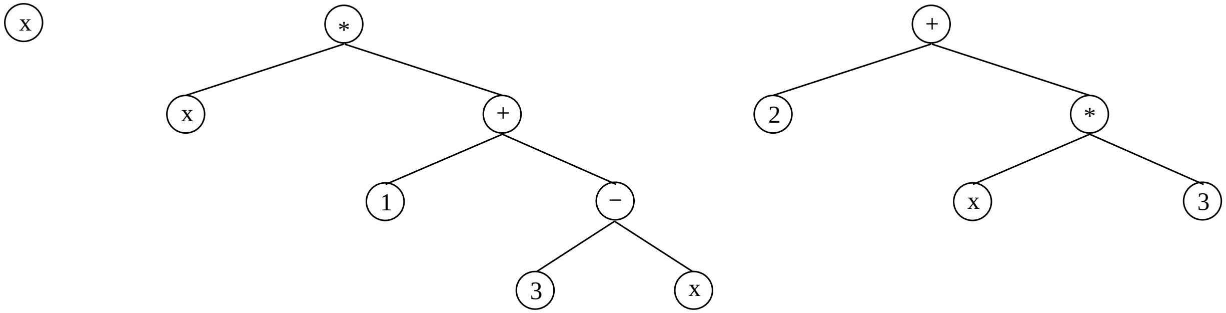


ED diversity=6.33
SC diversity=2.33

$IB_1$ =6
$IB_2$ =6

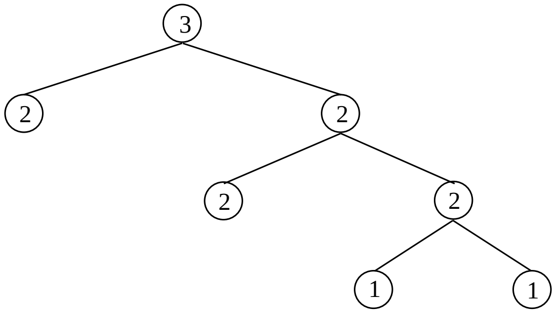
# Example iTrees

**Population $P_3$:**



iTree:

$M_1 = 7$
$M_2 = 13$
$M_3 = 6.25$



ED diversity=6.33

SC diversity=2.67

$IB_1 = 6$
$IB_2 = 8$

# Simple population measures

- Number of node positions explored in the tree search space
  $M_1(P) = \sum_{A \in iTree} 1$

# Simple population measures

- Number of node positions explored in the tree search space
  $M_1(P) = \sum_{A \in iTree} 1$

- Number of genetic nodes in the population
  $M_2(P) = \sum_{A \in iTree} n_A$

# Simple population measures

- Number of node positions explored in the tree search space
  $M_1(P) = \sum_{A \in iTree} 1$

- Number of genetic nodes in the population
  $M_2(P) = \sum_{A \in iTree} n_A$

- Degree of fullness
  $M_3(P) = \frac{1}{size(P)} \sum_{A \in iTree} \frac{1}{2^{depth(A)}} n_A$

# Simple population measures

- Number of node positions explored in the tree search space
  $M_1(P) = \sum_{A \in iTree} 1$
- Number of genetic nodes in the population
  $M_2(P) = \sum_{A \in iTree} n_A$
- Degree of fullness
  $M_3(P) = \frac{1}{size(P)} \sum_{A \in iTree} \frac{1}{2^{depth(A)}} n_A$
- Entropy of a node $A$
  $E(A) = - \sum_{s \in F \cup T} \frac{D(s)}{\sum_{v \in F \cup T} D(v)} log \frac{D(s)}{\sum_{v \in F \cup T} D(v)}$

# Structural diversity

Based on pairwise distances between individuals in a population (average)

Very time-consuming:

$N \times (N-1)$ pairs

Edit distance for pair $T_1$, $T_2$ takes $O(|T_1| \times |T_2|)$

# Structural diversity

Based on pairwise distances between individuals in a population (average)

Very time-consuming:

$N \times (N - 1)$ pairs

Edit distance for pair $T_1$, $T_2$ takes $O(|T_1| \times |T_2|)$

Edit distance diversity can be calculated by traversing the iTree and summing up the nodes' contributions

A node's contribution – the number of pairs of non-identical symbols encountered in that position in the iTree

Time complexity: $O(|F \cup T| \times size(iTree))$

# Distance between populations

```
Pop_dist(iTree1, iTree2, N1, N2)
begin
   dist := 0;
   for each symbol s found D1(s) times in iTree1
                       and D2(s) times in iTree2
       dist := dist + D1(s) x (N2 - D2(s))
                    + (N1 - D1(s)) x D2(s);

   dist := dist / (2 x (N1 x N2));

   if at least one root has nonempty left child
       dist := dist + Pop_dist(iLeft1, iLeft2, N1, N2);

   if at least one root has nonempty right child
       dist := dist + Pop_dist(iRight1, iRight2, N1, N2);

   return dist;
end
```

# Imbalance of a population

Unbalanced iTree ⇒ biased sampling of nodes

Balanced iTree ⇒ uniform sampling of nodes

# Imbalance of a population

Unbalanced iTree ⇒ biased sampling of nodes

Balanced iTree ⇒ uniform sampling of nodes

Also indicates structural diversity

# Imbalance of a population

Unbalanced iTree $\Rightarrow$ biased sampling of nodes

Balanced iTree $\Rightarrow$ uniform sampling of nodes

Also indicates structural diversity

$IB_1 =$ sum of absolute differences between the sizes of the two subtrees of each node

# Imbalance of a population

Unbalanced iTree ⇒ biased sampling of nodes

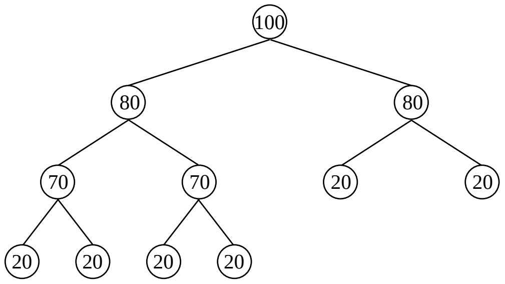
Balanced iTree ⇒ uniform sampling of nodes

Also indicates structural diversity

$IB_1 =$ sum of absolute differences between the sizes of the two subtrees of each node

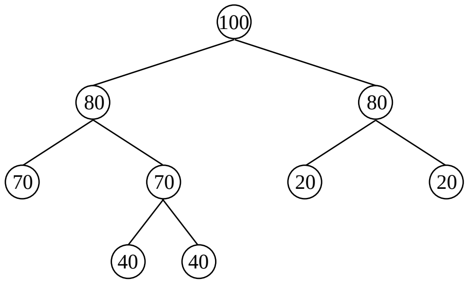
$IB_2$ uses total number of genetic tree nodes instead of size

# Examples of imbalance



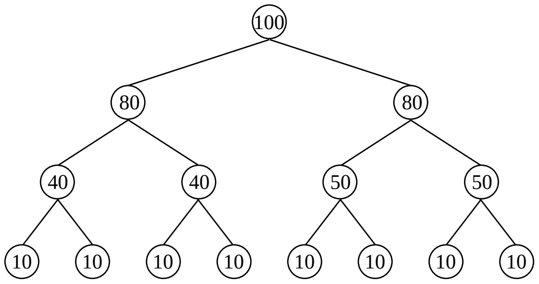
$IB_1 = 4$

$IB_2 = 180$



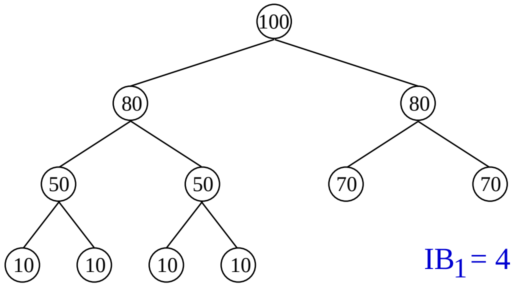
$IB_1 = 4$

$IB_2 = 260$



$IB_1 = 0$

$IB_2 = 20$

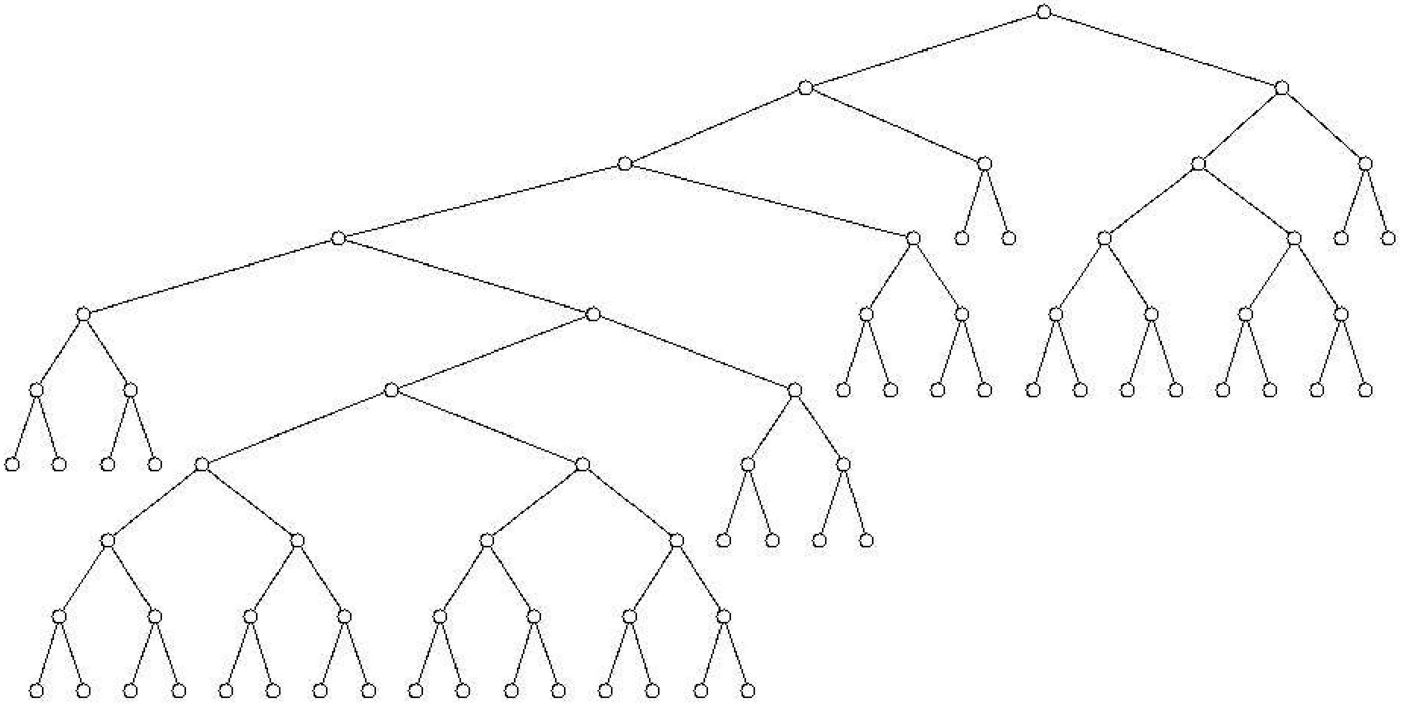


$IB_1 = 4$

$IB_2 = 0$

# Visualisation of the iTree

What are the most common structures in the best genetic programs encountered during a run?



# Visualisation of the iTree

What are the most common structures in the best genetic programs encountered during a run?

What makes a good program different from a bad program?

# Visualisation of the iTree

What are the most common structures in the best genetic programs encountered during a run?

What makes a good program different from a bad program?

When do the good structures emerge?

# Visualisation of the iTree

What are the most common structures in the best genetic programs encountered during a run?

What makes a good program different from a bad program?

When do the good structures emerge?

If there are common structures, do they heavily depend on the initial population?

# Case study

Symbolic regression for

$$(x+0.2)^2\,(x-0.5)\,(x+0.5)\,(x-0.7)$$

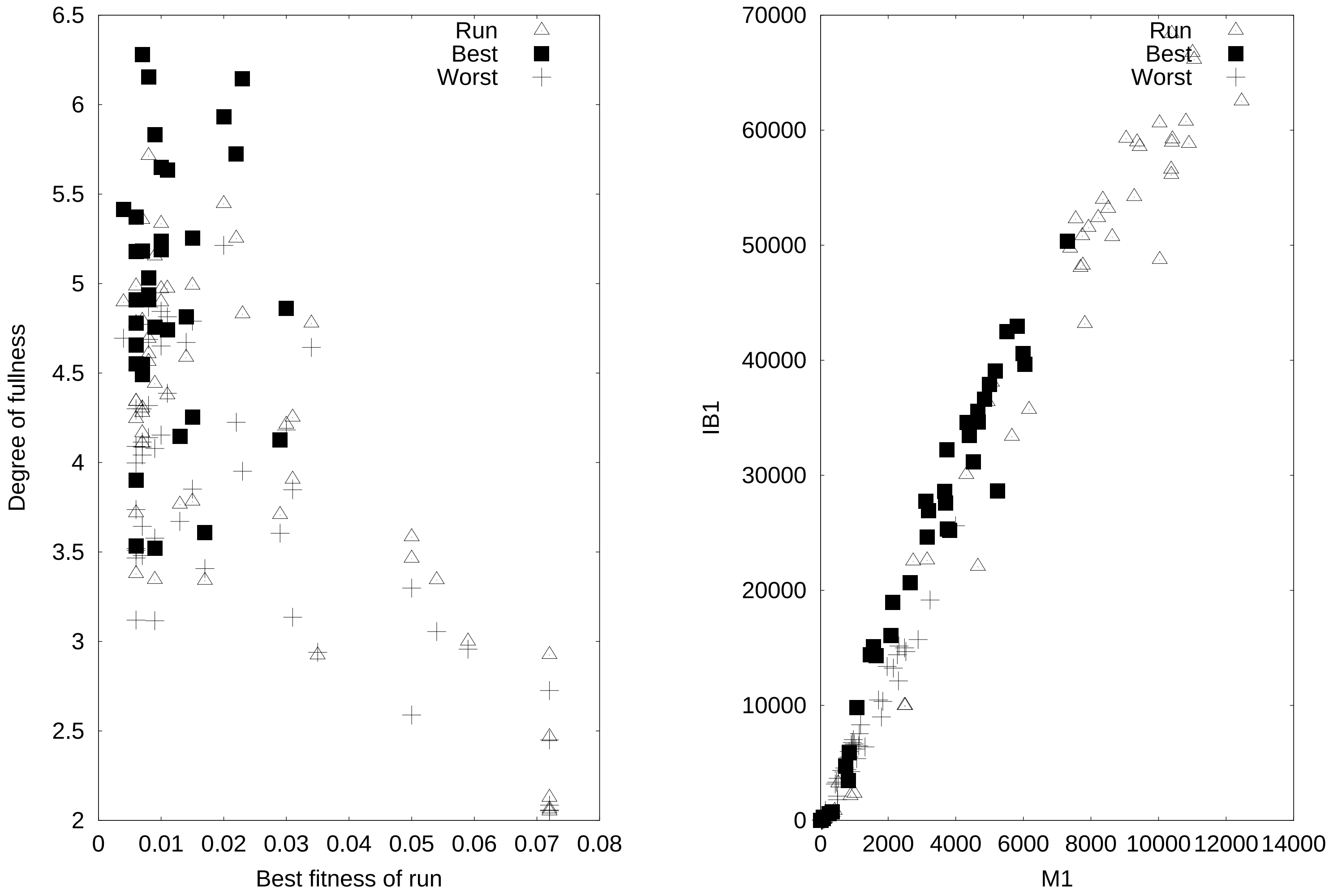
with $x \in [-1, 1]$

Analysis for three iTrees:

run's, best and worst genetic programs'

|  | $M_1$ | $M_2$ | $M_3$ | $IB_1$ | $IB_2$ |
|---|---|---|---|---|---|
|  | $[\times 10^2]$ | $[\times 10^3]$ |  | $[\times 10^3]$ | $[\times 10^4]$ |
| R | $59.7 \pm 11.4$ | $389 \pm 73.1$ | $4.24 \pm 0.26$ | $35.6 \pm 6.8$ | $185.6 \pm 42.4$ |
| B | $31.8 \pm 6.5$ | $249.9 \pm 53.9$ | $4.93 \pm 0.23$ | $23.5 \pm 4.7$ | $143.3 \pm 33.6$ |
| W | $10.6 \pm 2.6$ | $12.8 \pm 4.6$ | $3.78 \pm 0.22$ | $6.3 \pm 1.6$ | $5.1 \pm 2.3$ |

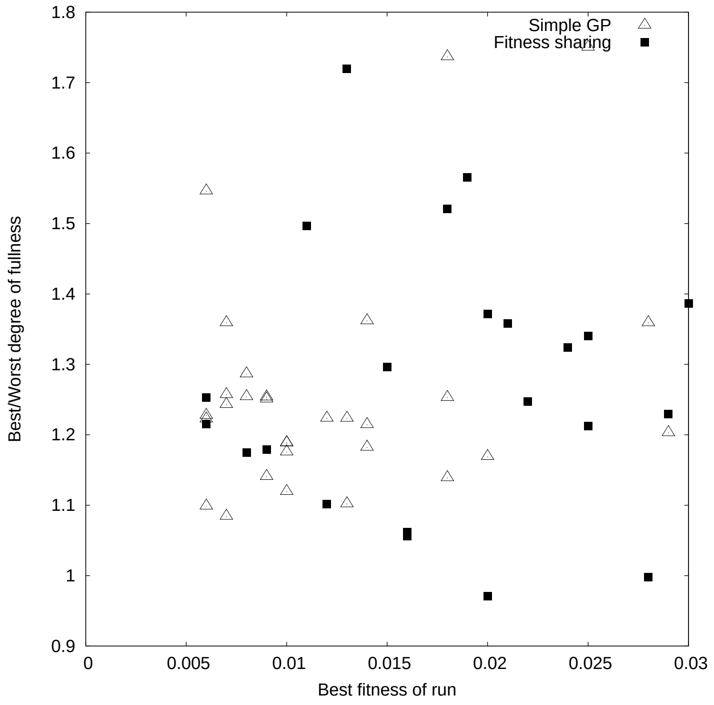
# Population measure plots

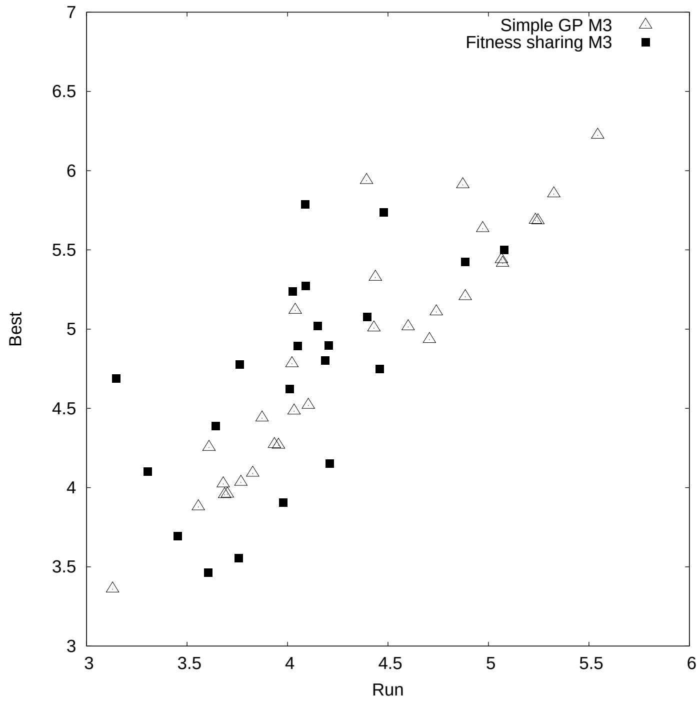


# Simple GP vs fitness sharing

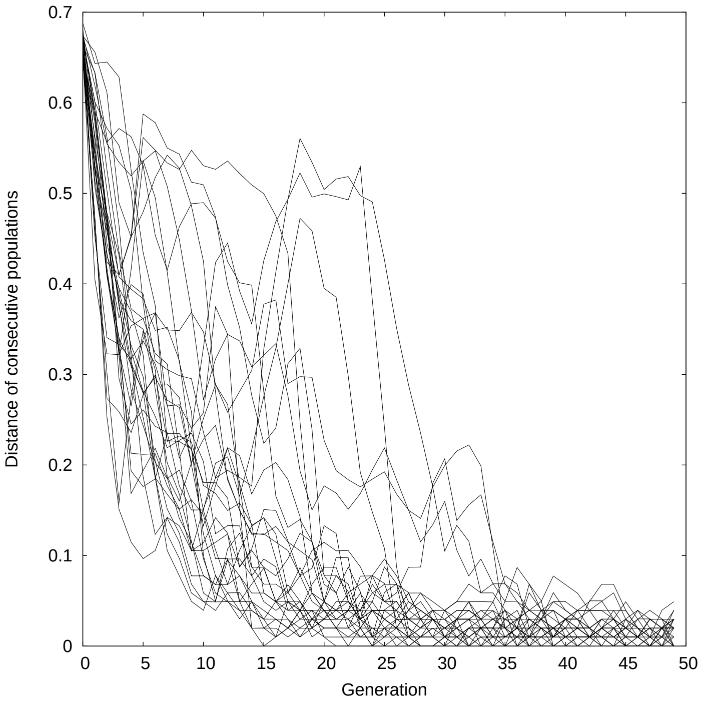
- Simple GP explores more nodes and the trees are more unbalanced
- Fitness sharing produces less, but smaller and more balanced solutions
- For fitness sharing the good tree structures are more distinguishable from the bad ones

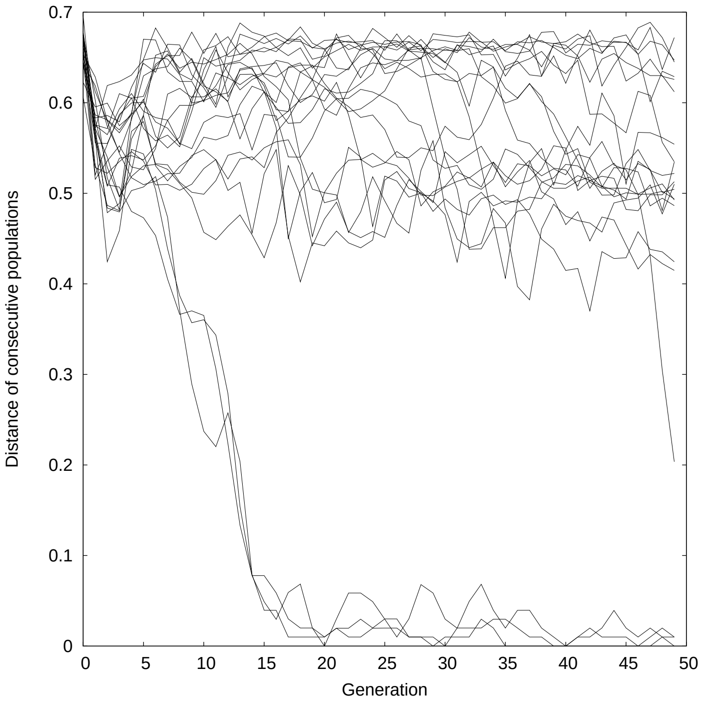
# Population measure plots





# Distance between populations





Simple GP

Fitness sharing

# Conclusions & future directions

- We introduced an intermediate data structure for more efficient complex population measures and visualisations
- The iTree-based analysis showed that for fitness sharing subsequent populations remain equidistant throughout evolution leading to less frequent convergence

- A methodology for efficiently analysing population dynamics will be built
- By providing feedback to the GP system throughout evolution we hope to both shorten evolution time and obtain better solutions