locoGP: Improving Performance by Genetic Programming Java Source Code

Brendan Cody-Kenny, Edgar Galván López, Stephen Barrett

{codykenb, edgar.galvan, stephen.barrett}@scss.tcd.ie

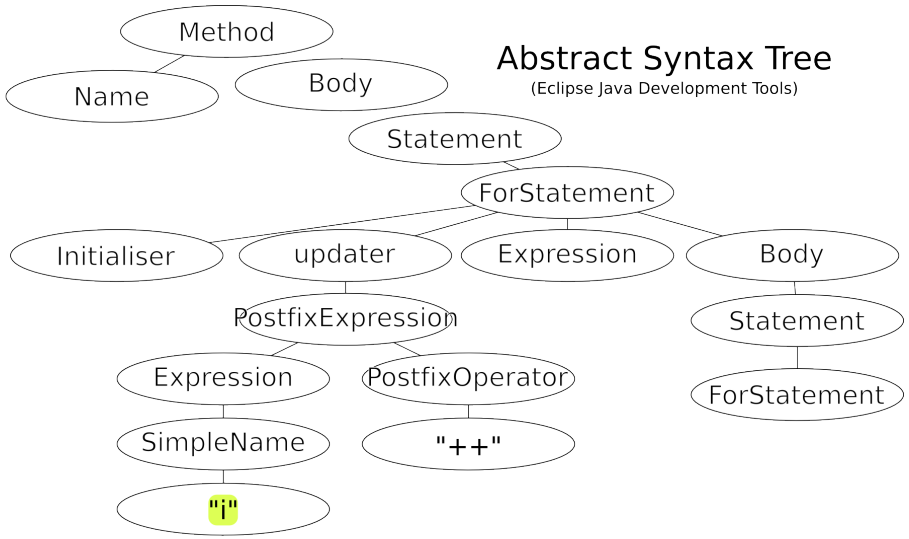
Distributed Systems Group,
School of Computer Science and Statistics,
Trinity College Dublin,
Ireland.

12 July 2015

Working with Java

```
public void sort( Integer[] a, Integer length){
  for (int i=0; i < length; i++) {
    for (int j=0; j < length - 1; j++) {
      if (a[j] > a[j + 1]) {
        int k=a[j];
        a[j]=a[j + 1];
        a[j + 1]=k;
    }}
}}
```

Edit




```
public void sort( Integer[] a, Integer length){
  for (int i=0; i < length; length++) {
    ....
```

Compile

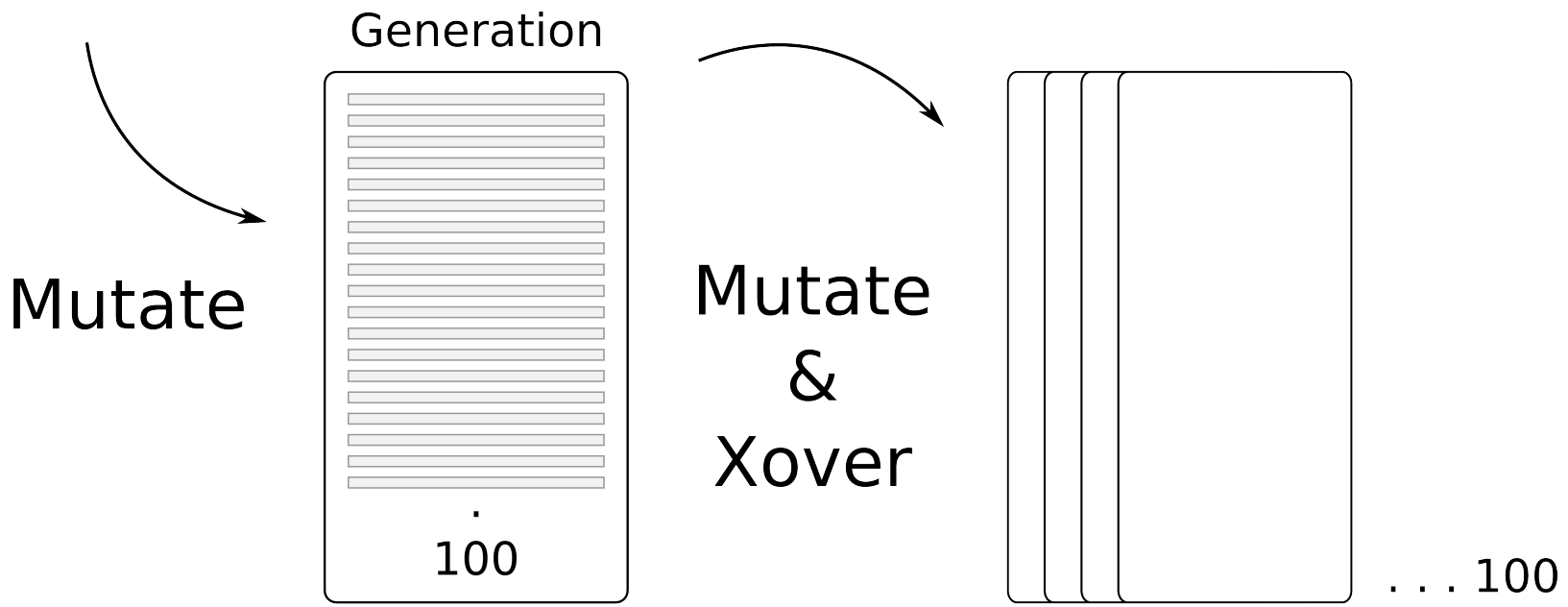
```
 0: iconst_0
 1: istore_2
 2: goto          69
 5: iconst_0
 6: istore_3
 7: goto          56
10: aload_0
11: iload_3
12: aaload
13: invokevirtual #16;
```

Instrument
(bycounter)

```
   0: ldc           #198;
   2: pop
   3: invokestatic  #204;
   6: astore_w      #65534;
  10: ldc2_w        #205;
      ....
1480: iconst_0
1483: lload_w       #65480;
1487: lconst_1
```

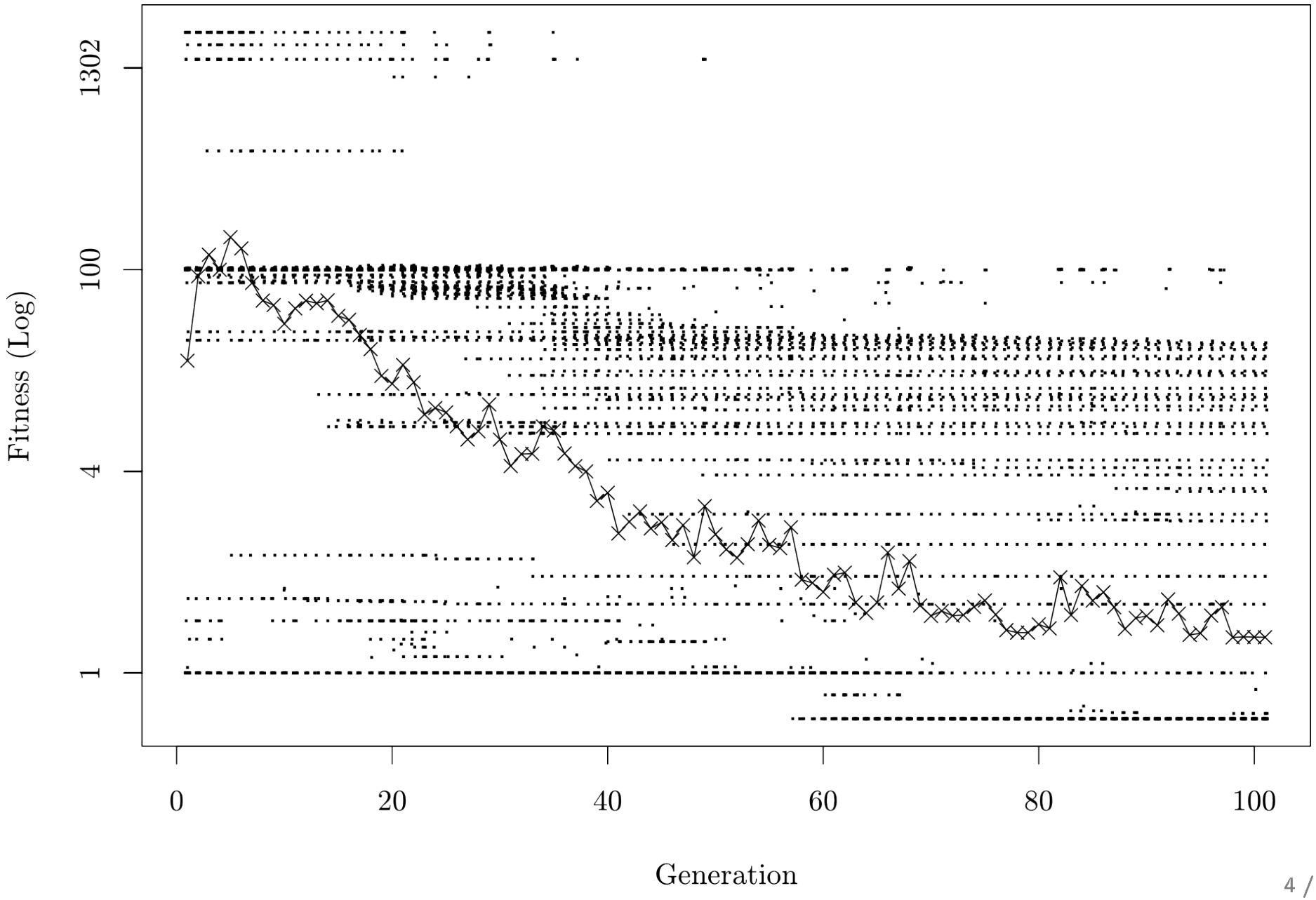
GP setup

```
public void sort( Integer[] a, Integer length){
  for (int i=0; i < length; i++) {
    ....
```



$$F = \frac{ExecutionCount_{ind}}{ExecutionCount_{seed}} + 100\left(\frac{Error_{seed} - Error_{individual}}{Error_{seed}}\right)$$

Improving Bubblesort



Results - all runs

Problem Name	*AST Nodes*	*Imp Nodes*	*LOC*	*Best Fit*	*% Disc*
Insertion Sort	60	3	13	0.91	73.3
Bubblesort	62	5	13	0.55	71.4
BubbleLoops	72	8	14	0.29	71.8
Selection Sort 2	72	1	16	0.99	70.9
Selection Sort	73	1	18	0.98	71.2
Shell Sort	85	3	23	0.95	71.4
Radix Sort	100	3	23	0.99	80.5
Quick Sort	116	2	31	0.46	72.7
Cocktail Sort	126	1	30	0.85	73.7
Merge Sort	216	1	51	0.95	73.2
Heap Sort	246	2	62	0.59	71.1
Huffman Codebook	411	5	115	0.57	83.8

Issues

- GP can not grow programs
- Discarded programs
  - difficulty of getting from one program to another
  - local optima
  - "Compilation Trap"
- Close to random search
  - small number of changes (<10%)

Potential Future Work

New problems

- fitness functions

Comparison

- different granularities/representations

Search algorithm

- reduce redundancy (tabu-search)
- evolve patches
- building blocks?

Infrastructure

- build servers, SPL's
- libraries of code (reuse)
  - online sources
- Patchwork of SE techniques
  - source code analysis

Questions, Comments, Contact

https://www.scss.tcd.ie/~codykenb
https://github.com/codykenb/locoGP

codykenb@tcd.ie

?