Landscape of the Triangle Program

W. B. Langdon
Department of Computer Science

Humies
$10,000 Human-Competitive Results
Fitness Landscape of the Triangle Program

- Software is not fragile
- the Triangle program
  - Two versions: binary and all C code comparisons
- Existing analysis EuroGP-2017
- Benchmark, with no run time analysis
  - Population based
  - Hill climbing
  - Empirical
- variable interaction graph
- Theory for Genetic Improvement
Software is not fragile

• Automatic bugfixing
  – suggests many repairs are easy

• Equivalent mutants in mutation testing
  – suggests many moves are neutral

• Failure of error propagation
  – suggests many moves are neutral

• Success of genetic improvement
  – software landscape is not so hard to search

• Empirical study suggests software is robust
89% mutations which compile make no change to test case. 

CS-DC'15

BWA 0.7.12-r1039

10958 lines of code
Triangle Program

• Given length of three sides what type is triangle? (Software Engineering benchmark)

• Test suite (14) covers all paths JSS 83(12) (2010) 2416–2430
  Quantified information flow $3 \times 32 = 96 \rightarrow 2$ bits

• Mutate conditionals

• Fitness is number of tests that fail (minimize)

• Code and datasets online http://www.cs.ucl.ac.uk/staff/W.Langdon/egp2017/triangle/

• whole landscape
int gettri(int side1, int side2, int side3)
{
    int triang ;
    if( side1 <= 0 || side2 <= 0 || side3 <= 0){
        return 4;
    }
    triang = 0;
    if(side1 == side2){
        triang = triang + 1;
    }
    if(side1 == side3){
        triang = triang + 2;
    }
    if(side2 == side3){
        triang = triang + 3;
    }
    if(triang == 0){
        if(side1 + side2 <= side3 ||
           side2 + side3 <= side1 ||
           side1 + side3 <= side2){
            return 4;
        } else {
            return 1;
        }
    }
    if(triang > 3){
        return 3;
    } else if ( triang == 1 && side1 + side2 > side3 ) { return 2; }
    else if (triang == 2 && side1 + side3 > side2) { return 2; }
    else if (triang == 3 && side2 + side3 > side1) { return 2; }
    return 4;
}
Three inputs | expected output
---|---
0 0 0 | 4
1 0 0 | 4
1 1 0 | 4
1 1 1 | 3
2 2 1 | 2
1 1 2 | 4
2 1 2 | 2
1 2 1 | 4
2 1 1 | 4
3 2 2 | 2
3 2 1 | 4
4 3 2 | 1
2 3 1 | 4
2 1 3 | 4

Inputs are the three sides of the triangle. Output is correct classification of the triangle. Test suite covers all paths but is not strong enough to detect all mutations. Dataset gives whole test equivalent fitness landscape for 2-way comparisons.

W. B. Langdon, UCL
int gettri(int side1, int side2, int side3) 
{
    int triang;
    if( side1 <= 0 || side2 <= 0 || side3 <= 0){
        return 4;
    }
    triang = 0;
    if(side1 == side2){
        triang = triang + 1;
    }
    if(side1 == side3){
        triang = triang + 2;
    }
    if(side2 == side3){
        triang = triang + 3;
    }
    if(triang == 0){
        if(side1 + side2 <= side3 || side2 + side3 <= side1 || side1 + side3 <= side2){
            return 4;
        } else {
            return 1;
        }
    }
    if(triang > 3){
        return 3;
    } else if ( triang == 1 && side1 + side2 > side3) {
        return 2;
    } else if (triang == 2 && side1 + side3 > side2){
        return 2;
    } else if (triang == 3 && side2 + side3 > side1){
        return 2;
    } return 4;
}
int triang;

if( 4 <= 0 || 3 <= 0 || 2 <= 0){
    return 4;
}

triang = 0;

if(4 == 3){
    triang = triang + 1;
}
if(4 == 2){
    triang = triang + 2;
}
if(3 == 2){
    triang = triang + 3;
}

if(triang == 0) return 1;

Simpliest example
test case 12: 4,3,2
correct answer scalene(1)

Depends on loci {1,2,3,4,5,6,7} only

Loci {1,2,3} only <, <= or ==
Loci {4,5} only <, <=, ==
Locus 6 only <, <=, ==
Locus 7 only ==
Loci {8…17} anything

4.40798 \times 10^{10} solutions

W. B. Langdon, UCL
```c
int triang;

if( 4 <= 0 || 3 <= 0 || 2 <= 0){
    return 4;
}

triang = 0;

if(4 == 3){
    triang = triang + 1;
}
if(4 == 2){
    triang = triang + 2;
}
if(3 == 2){
    triang = triang + 3;
}

if(triang == 0) return 1;

What are the variables?
C code: side1, side2, side3, triang

But search variable are 17 loci <=, == etc.
Test cases, C code, variables and constants tell us how our search variables interact.

Simple non-deceptive building blocks.
2048 solutions:
***== == == ***** >**==*==*
* two options

How do we use variable interaction graph?
Guide crossover.
For scalene, graph says loci 1-6 interact with 7
Build new graph for each test case?
Union too simple? Weight edges by number of test cases where they apply?
```
Conclusions

Triangle Program fitness landscape

- available and tractable (up to 16,926,659,444,735)
- Realistic
- Scalable (small $2^{17}$ and larger versions)
- Unsolved

Next: Runtime analysis?
Variable interaction graph
Abstract features, generalise to other software?
Other programs?
Analyse information flow?

W. B. Langdon, UCL
Humies
$10000
Human-Competitive Results

END

http://www.cs.ucl.ac.uk/staff/W.Langdon/
http://www.epsrc.ac.uk/
The Genetic Programming Bibliography

http://www.cs.bham.ac.uk/~wbl/biblio/

11504 references

Make sure it has all of your papers!
E.g. email W.Langdon@cs.ucl.ac.uk or use | Add to It | web link

RSS Support available through the Collection of CS Bibliographies.

Co-authorship community.
Downloads

A personalised list of every author’s GP publications.

blog

Search the GP Bibliography at
http://liinwww.ira.uka.de/bibliography/Ai/genetic.programming.html