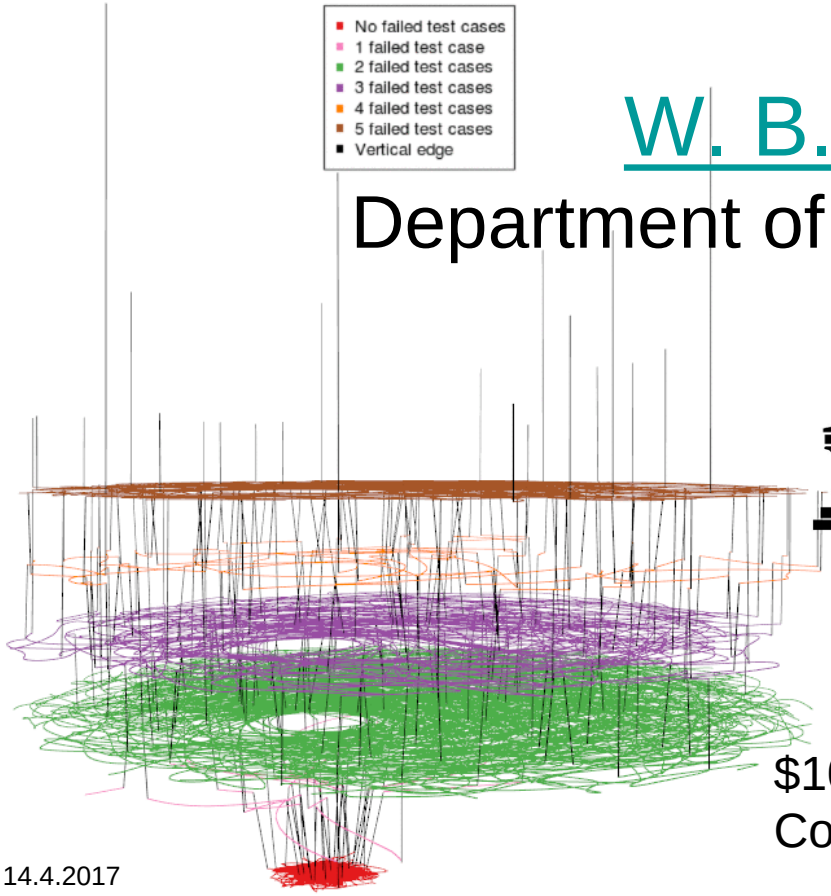
CREST

Visualising the Search Landscape of the Triangle Program

EuroGP 2017, Mauro Castelli *et al*., LNCS 10196, pages 96-113, Amsterdam



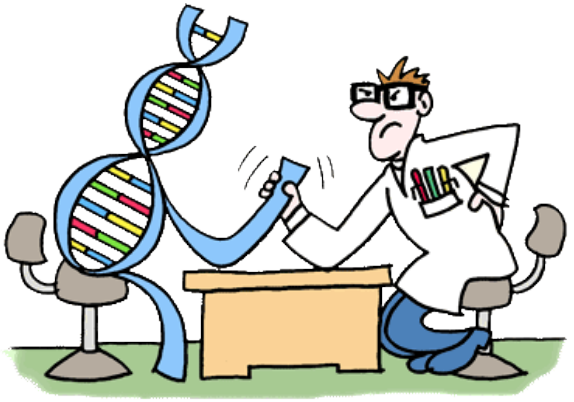
W. B. Langdon

Department of Computer Science

UCL

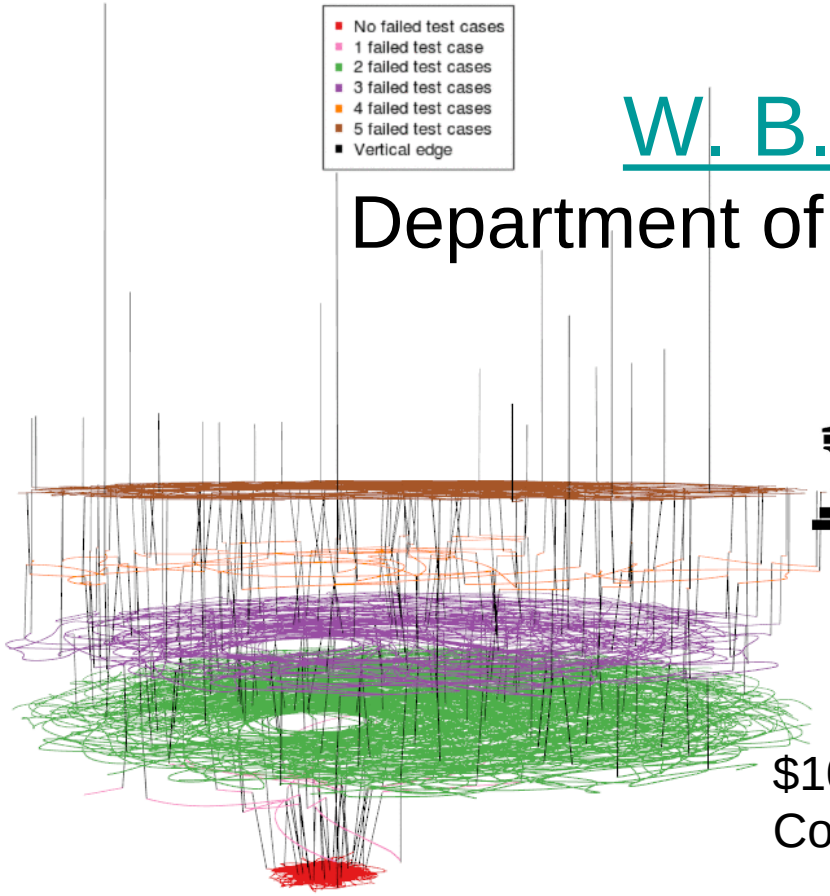
Humies

$10,000 Human-Competitive Results



CREST

Visualising the Search Landscape of the Triangle Program



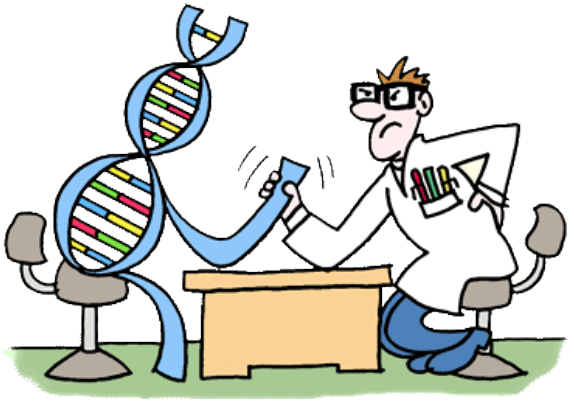
W. B. Langdon

Department of Computer Science

UCL

Humies

$10,000 Human-Competitive Results



Fitness Landscape of the Triangle Program

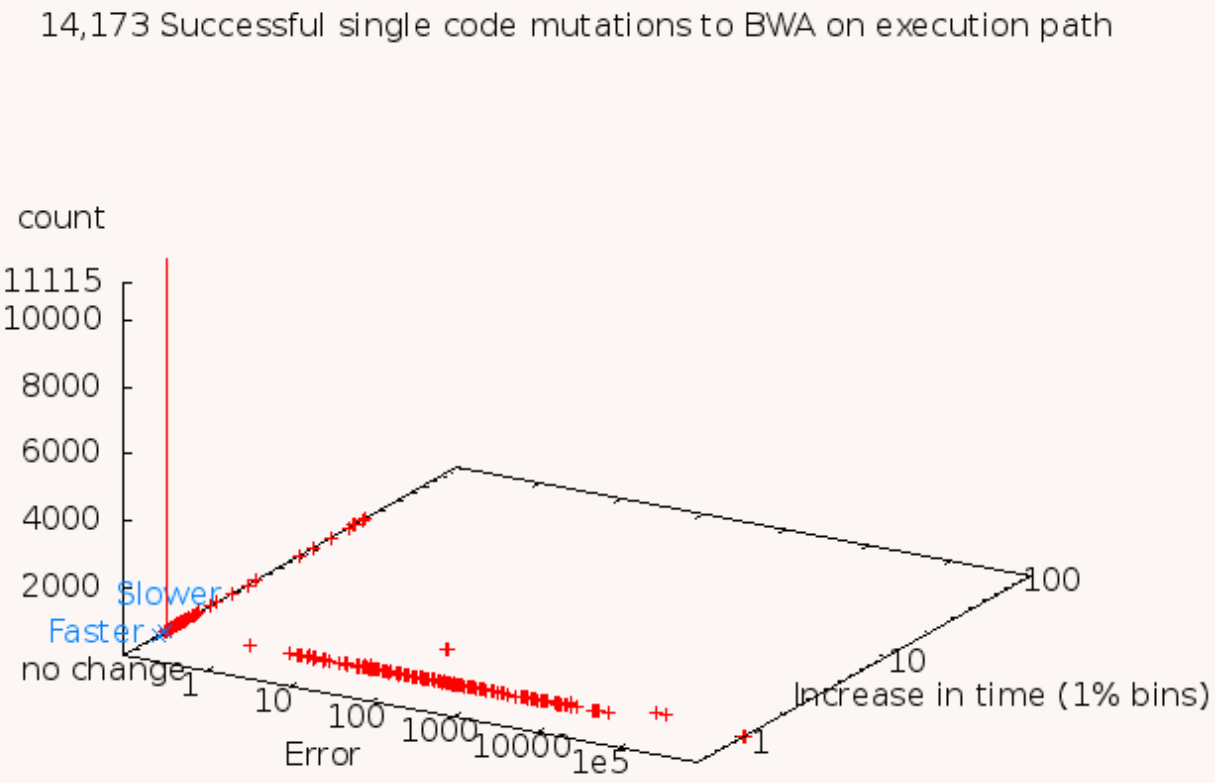
- Background, what is Genetic Improvement
  - Fitness landscapes of genetic improvement
  - the Triangle program.
  - Constructing Triangle's fitness landscape
    - Two versions: binary and **all comparison changes**
- Results
  - searchspaces
  - schema analysis
  - Hill climbing and local optima visualisation
- Insight into Genetic Improvement?

Genetic Improvement

- Genetic Improvement is the application of search (often genetic programming) to improve existing software, e.g.
  - Fix bugs
  - Faster (CPU or on parallel hardware: GPU)
  - Less energy used
  - Less memory
- Real programs (10^4 to 10^6 of lines of code)

WIKIPEDIA
Genetic Improvement

GI Mutation Fitness Landscapes



89% mutations which compile make no change to test case CS-DC'15

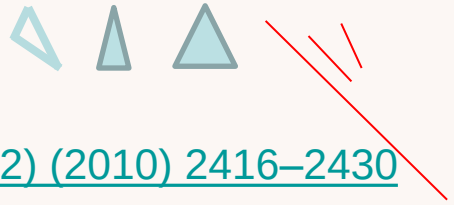
BWA 0.7.12-r1039
10958 lines of code

GI Fitness Landscapes

- Real software is resilient to mutations.
- Schema (crossover) analysis.
- The Triangle program is a software engineering benchmark

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
- 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



All 17 comparisons are potential mutation sites. Shown in red

39 lines of C code

```
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;
}
```

testcases_oracle.txt 14 tests

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.

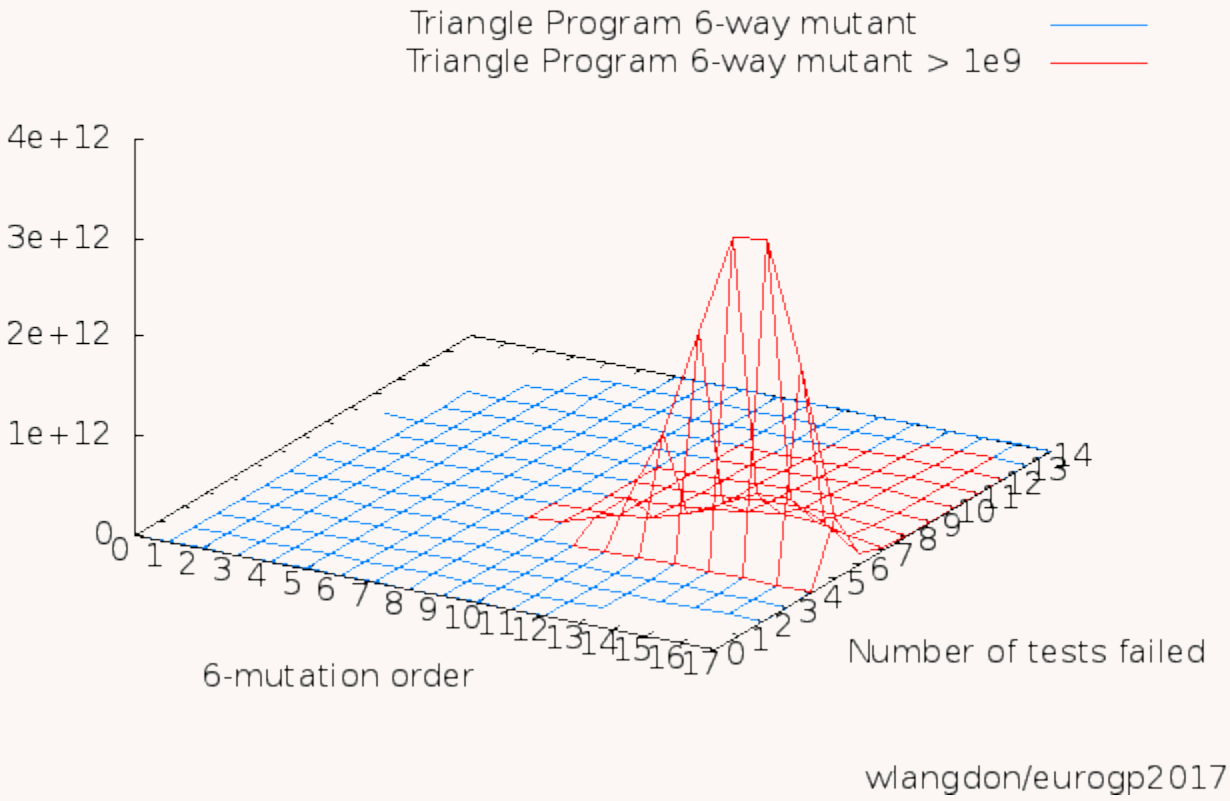
Dataset gives whole test equivalent fitness landscape for 2-way comparisons.

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

Triangle Fitness Landscape

- Only mutate C comparison operators (17)
- Run all tests (14). For how many does new code give the wrong answer? (0-14)

Triangle Program search space



- Fitness distribution of all 16,926,659,444,735 mutations
- 78% of mutants fail five tests (mean 5.42, sd 1.05)
- 9215 global optima
- fitness distance correlation -0.070837

Genetic Improvement Schema

Binary schema

011*0001*0001 * don't care

Triangle Program 17 locations

== == == * > ** == * == *

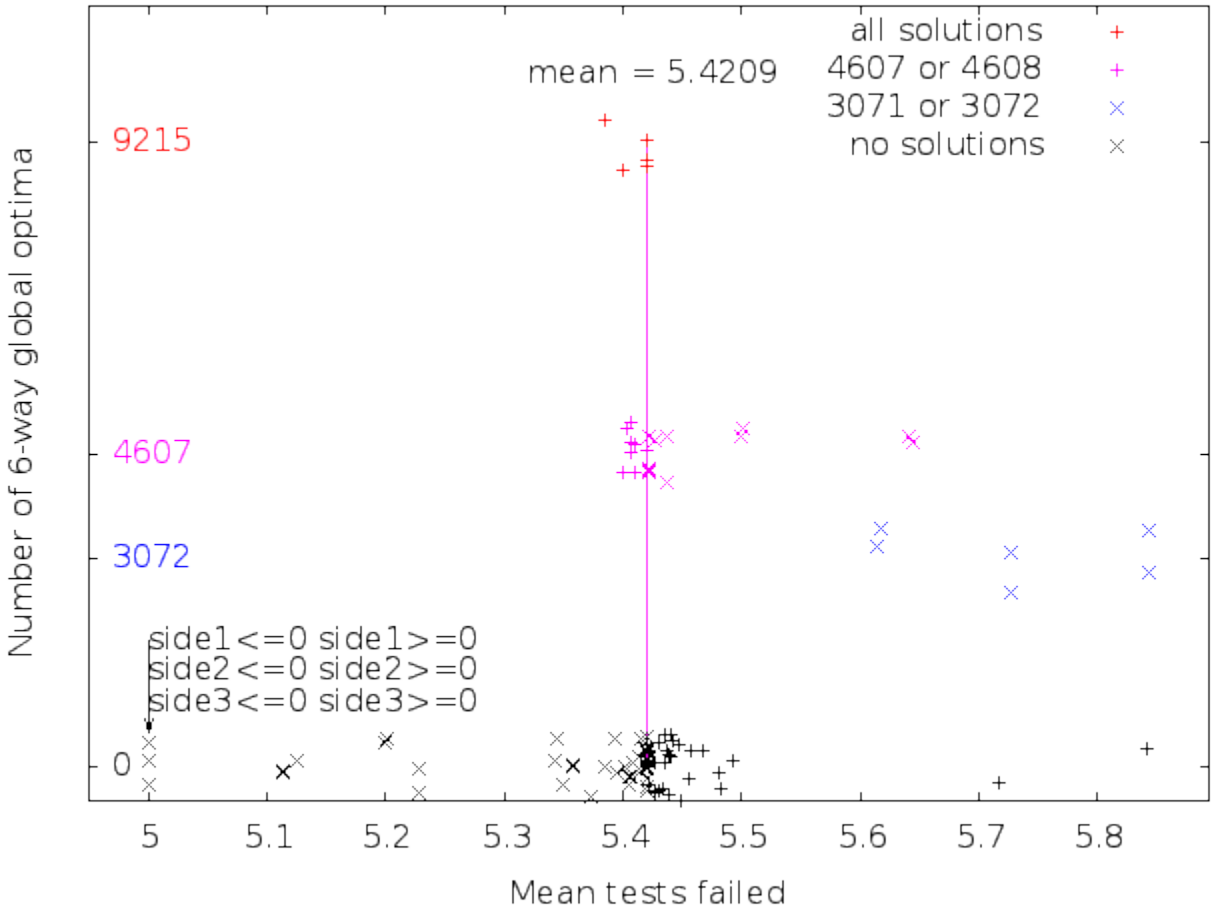
Example 1st order triangle schema

>= * ***** ***** *****

i.e. replace `if( side1 <= 0` with `if( side1 >= 0`

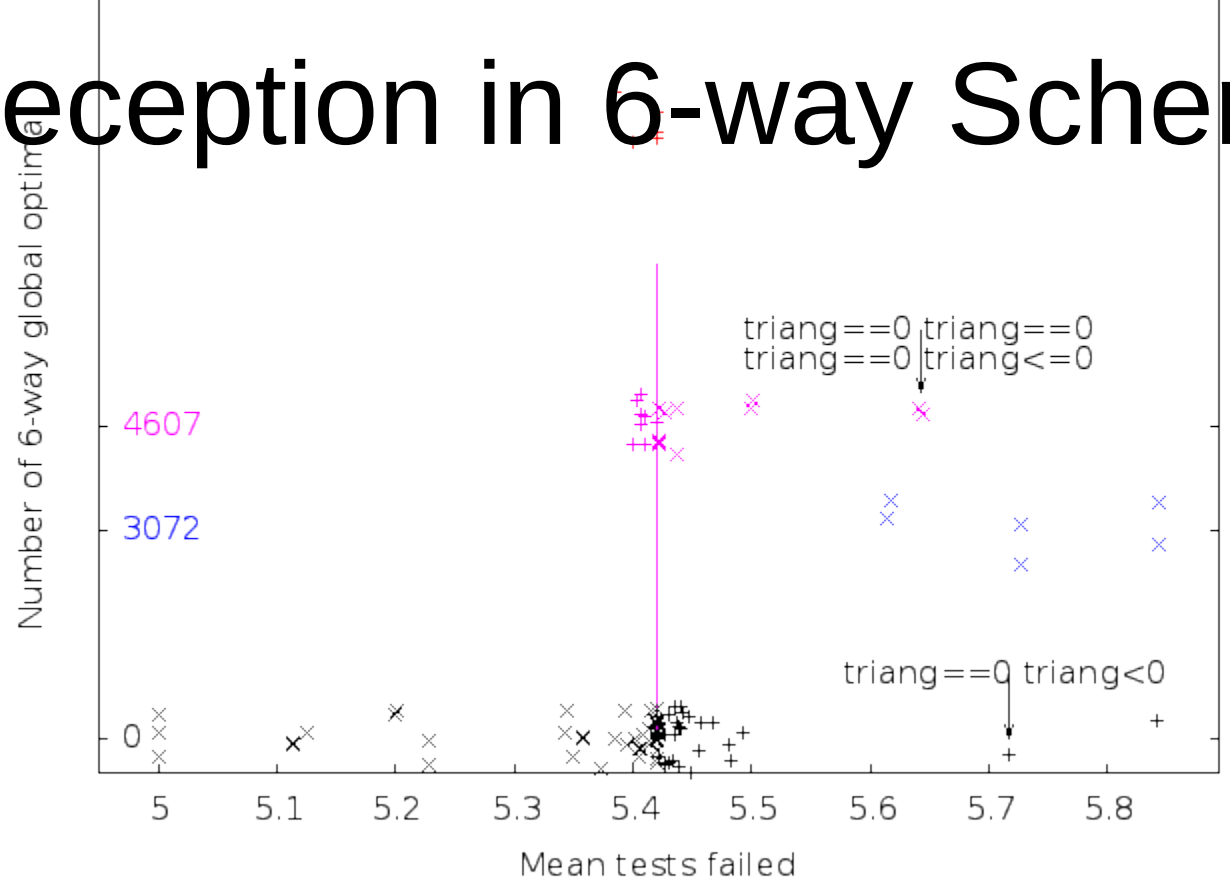
Deceptive schema has no solutions but its mean fitness is better than competing schema which contains solutions, vice-versa

102 First Order Schema



- 17×6=102 schema. 56 deceptive(×)
- 3 “best” get five tests wrong. This makes then better than average but contain no solutions
- (2-way,two alternatives only, schema are not deceptive)

Deception in 6-way Schema

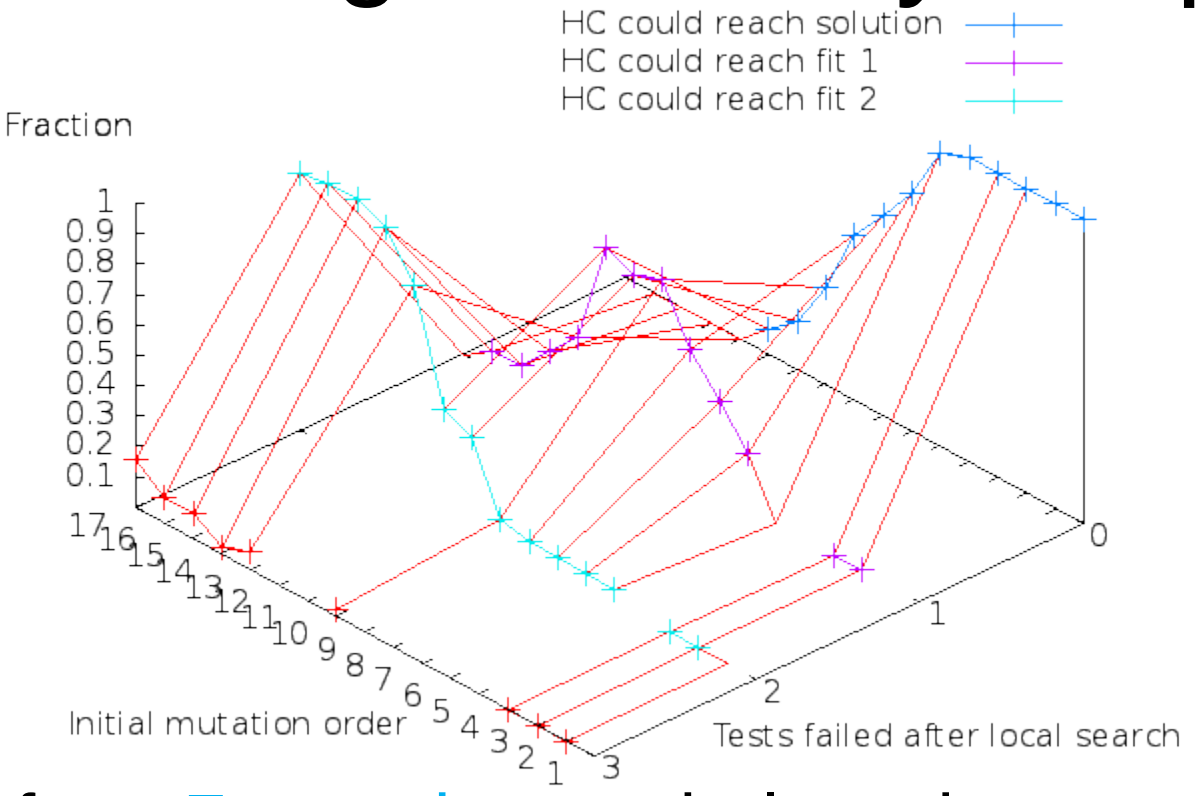


- 1st order cover the search space, i.e. far from start point. (two alternatives schema are closer to start point.)
- Strong epistasis so hard for naive GA and Hill Climbing.
- 1st order only have one defined position, often low signal to noise. $\sigma \approx 1$ so $\Delta/\sigma << 0.01$
- 20 schema where signal/noise $\Delta/\sigma > 0.1$, 18 are deceptive

Summary 6-way 1st order Schema

- 17×6=102 schema. 56 deceptive(×)
- None with strong signal towards any solution
- Strong epistasis, low fitness distance correlation
- Hard for global search GA and Hill Climbing but GI starts search near solution
- 1st order only have one defined position, often low signal to noise. $\sigma \approx 1$ so $\Delta/\sigma << 0.01$
- 20 schema where signal/noise $\Delta/\sigma > 0.1$, 18 are deceptive
- 2-way case not deceptive. (Closer to solution?)

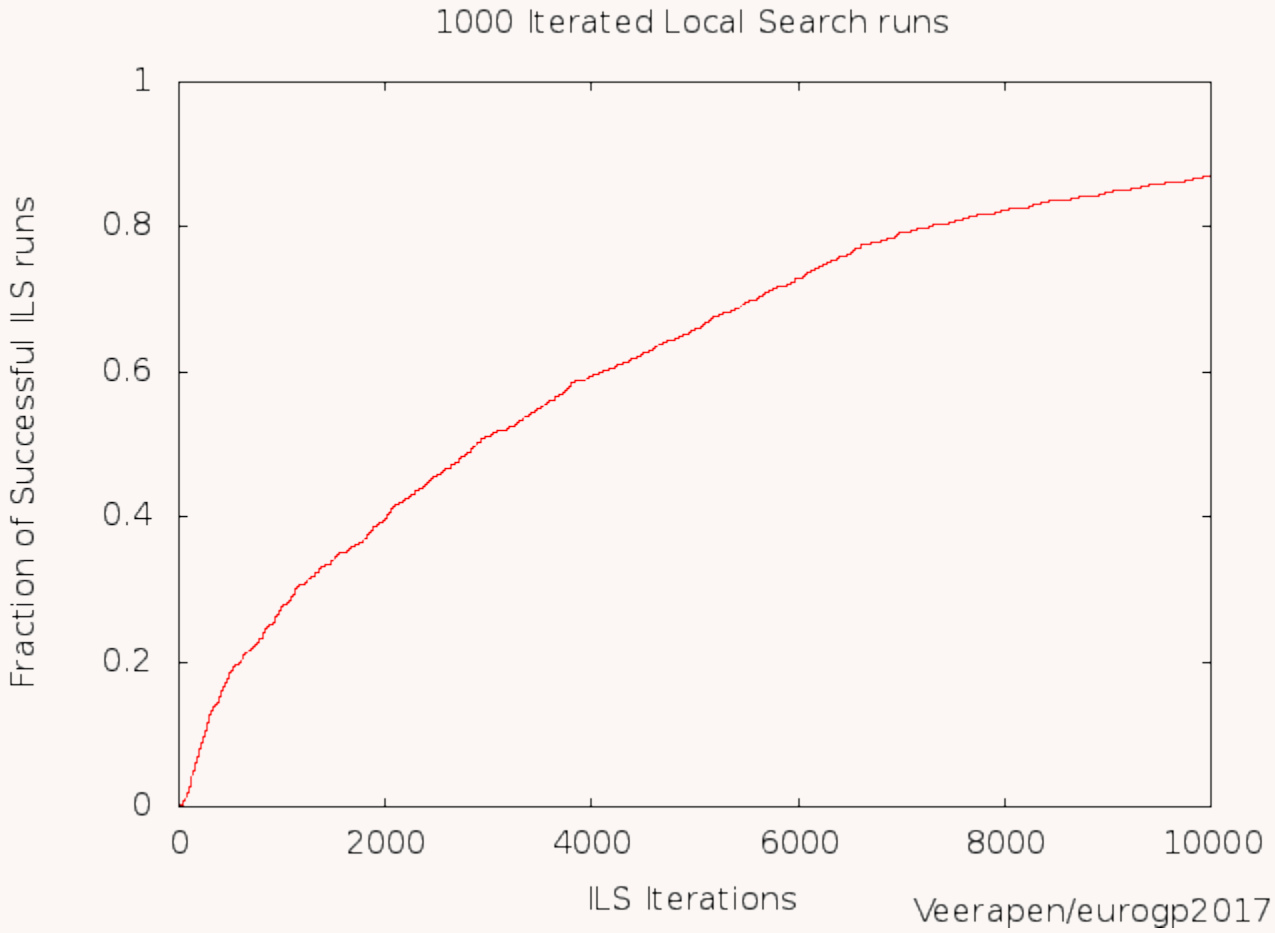
Hill Climbing can always Improve



- Start from 7 mutations or below, there probably is a route to solution.
- Start between 7 and 14 route to fail just one test
- Start between 11-17 fail at most two tests

Iterated Local Search

87% of Iterated Local Search (ILS) find a program which passes all the tests.

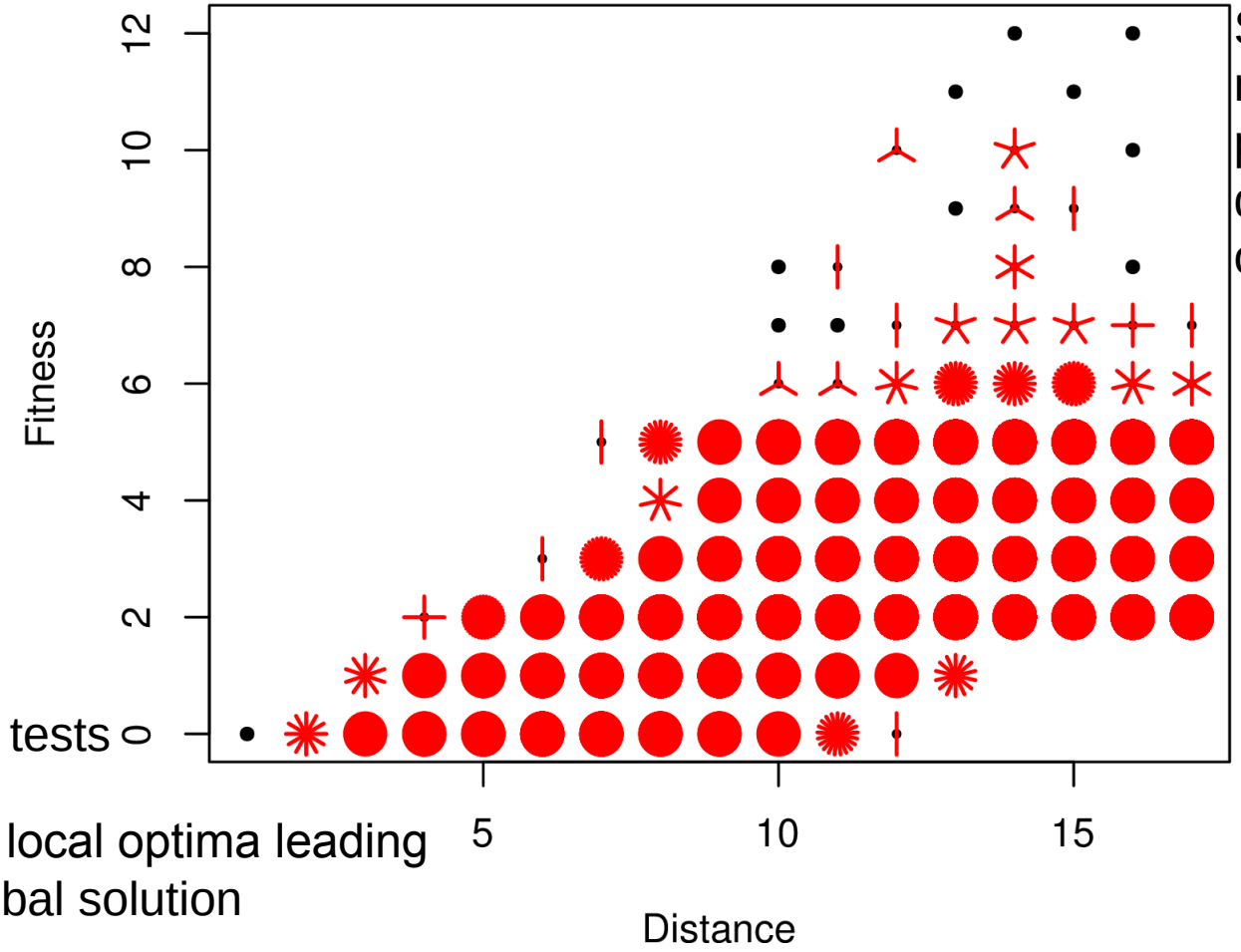


Local Optima

- A solution is a local optimum if none of its neighbours have a better fitness value.
- In Triangle Program each mutant has 5×17=85 neighbours.
- E.g. a mutants which fails 5 tests is a local optima if all of its 85 neighbours fail 5 or more tests.

Local Optima

1000 Iterated Local Searches (ILS) find 2,372,805 unique local optima



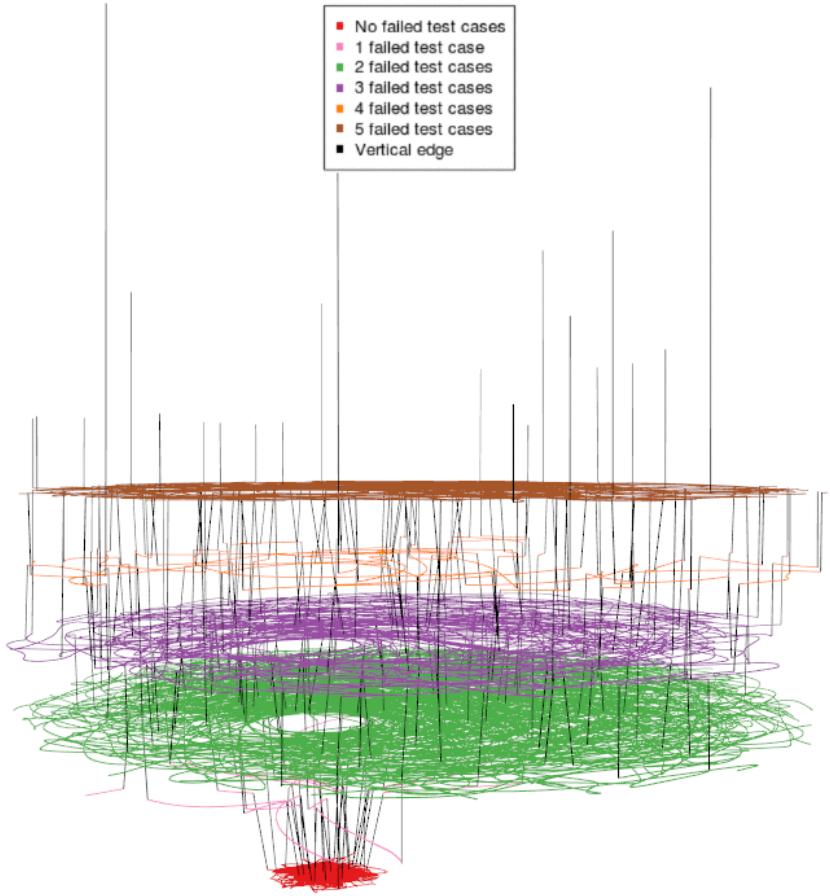
Sunflower plot: number of petals gives density of local optima

Best, fails 0 tests

"Big valley" local optima leading towards global solution

Path of 100 Iterated Local Searches

- First 1000 ILS iterations
- Only the edges shown.
- Black if fitness improves.
- Edges are coloured no fitness change.



Conclusions

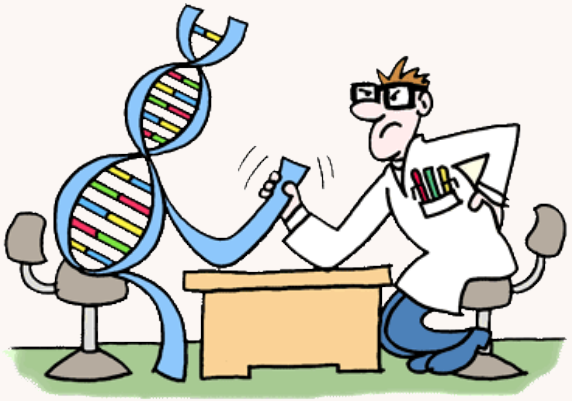
- Whole fitness landscape 16,926,659,444,735
  - 9215 pass all tests. Neutral moves & plateaus
- Global schema analysis
  - Low signal to noise, deceptive, epistasis.
  - 2-way closer to solution, no deception.
- “big valley” with many local optima, particularly close to global optima
- Hill climbing can always improve
- Iterated Local Search often finds solution.
- **Start near human code**

White Queen

In Lewis Carroll's “Through the Looking-Glass…” the White Queen says she could believe "six impossible things before breakfast”.

Do something impossible
Evolve something never attempted before

Humies
$10000
Human-Competitive Results



END

http://www.cs.ucl.ac.uk/staff/W.Langdon/

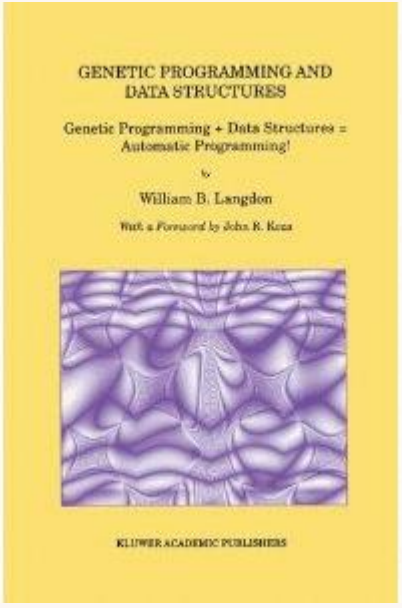
http://www.epsrc.ac.uk/ EPSRC

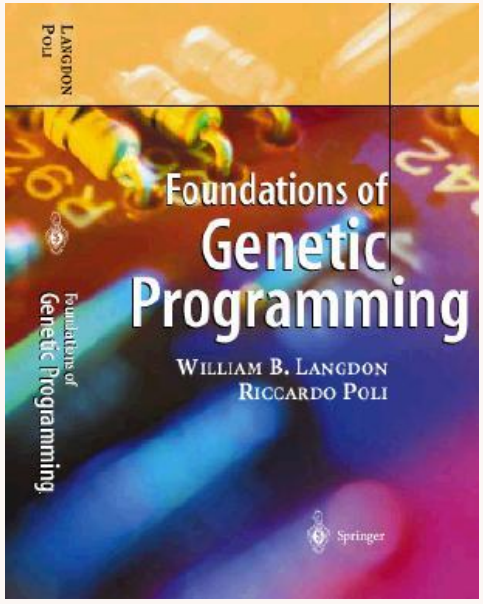
Genetic Improvement

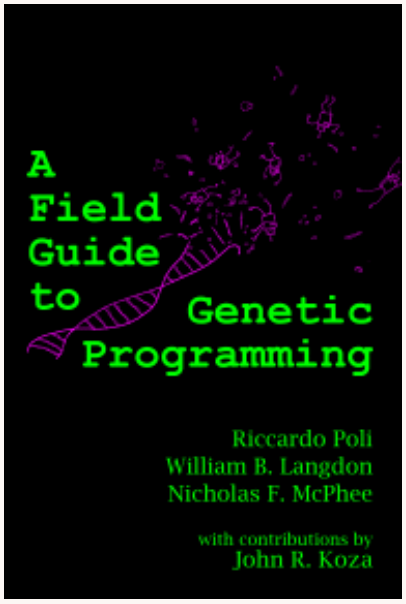
W. B. Langdon

CREST

Department of Computer Science







Discussion: Lessons for GI?

- With simple scalar fitness, almost everyone fails 5 tests.
- Need more fitness levels?
- Is “fitness” function too simple?
- Would a novelty reward help?
- Would multi-objective approach help?

Discussion: Lessons for GI?

- Global search dominated by mutants which fail five of fourteen tests.
- 1st order, schema says global search hard
  - Epistasis, deception.
- ILS more practical than simple hillclimbing (HC)
- ILS shows big valley of local optima near global optima (human code)
- Even simple HC may work close to solution
- START NEAR SOLUTION

Insight into Genetic Improvement?

- Triangle program is software engineering benchmark
  - small but akin to unit testing
- **Whole** fitness landscape
  - Much of search space passes most tests
  - Very few pass all tests
  - Large areas where neighbours pass 5 tests. **Neutral networks, plateaus**.
- **Global** schema analysis.
  - Low signal to noise, **deceptive**, epistasis
  - But (2-way mutation) closer to solution no deception.
- Many local optima, particularly close to global optima.
- Hill climbing can **always improve** (if allow neutral moves) but if start far from solution HC sometimes gets stuck.
- Iterated Local Search **often finds solution**.
- For global and hill climbing search **starting near human code** helps

W.Langdon egp2017 triangle.tar.gz

All 17 comparisons are potential mutation sites. Shown in red

== replaced by <=
> replaced by !=
<= replaced by ==

(Chosen as they are the hardest to detect mutations.)

```
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;
}
```

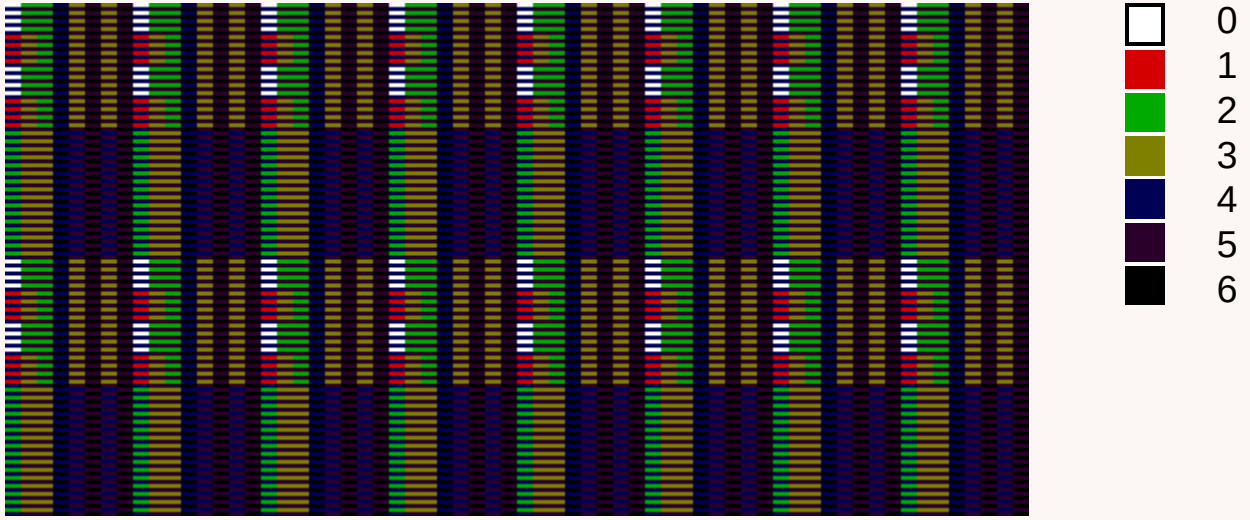
First Order Schema

Schema id	mean	sd	pop size
-4	3.719	±1.328	1.9
4	4.969	±1.075	
-5	4.062	±1.478	4.7
5	4.625	±1.166	
-6	3.812	±1.509	2.4
6	4.875	±0.927	
-11	3.438	±1.273	1.1
11	5.250	±0.661	
-14	4.312	±1.424	43.5
14	4.375	±1.293	
-16	4.188	±1.550	8.6
16	4.500	±1.118	

Pop size = $2sd_{12} / |mean_1 - mean_2|$

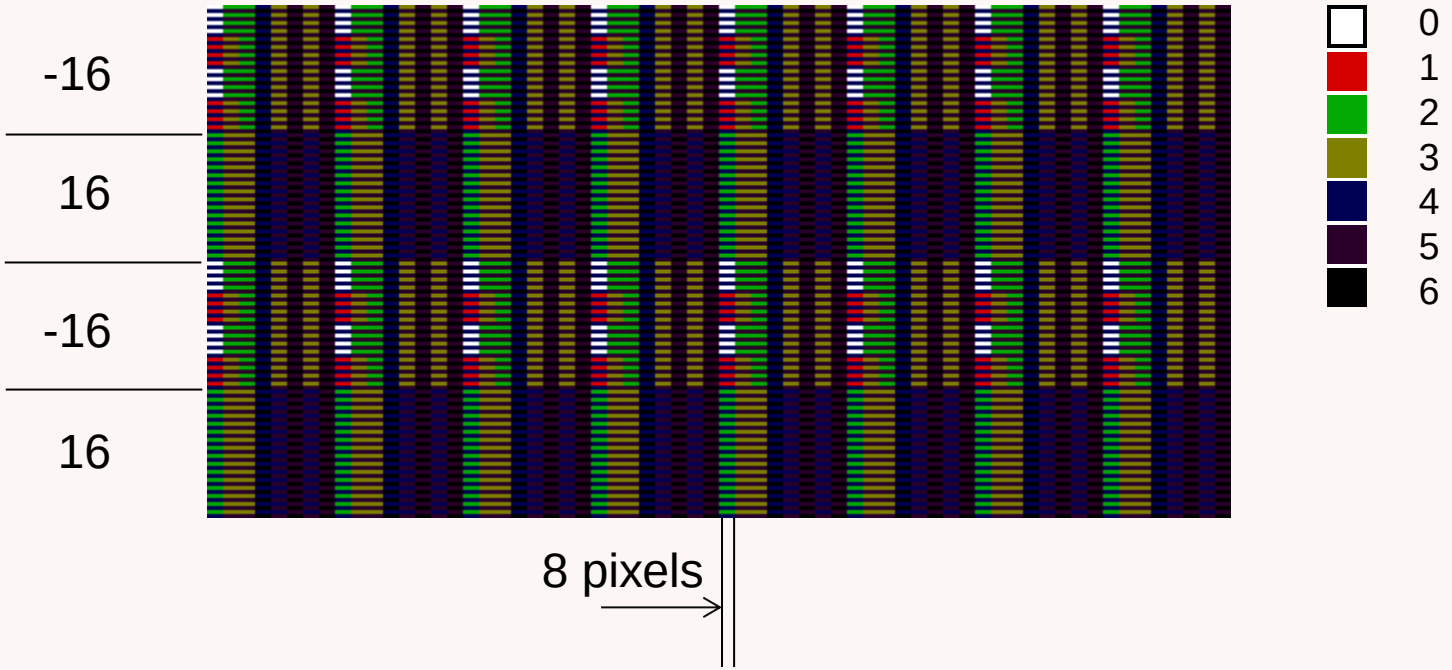
Table 2: Mean and standard deviation of number of tests failed for first order schema (excluding 22 with average means).

Fitness in bit order ($2^8 \times 2^9$)



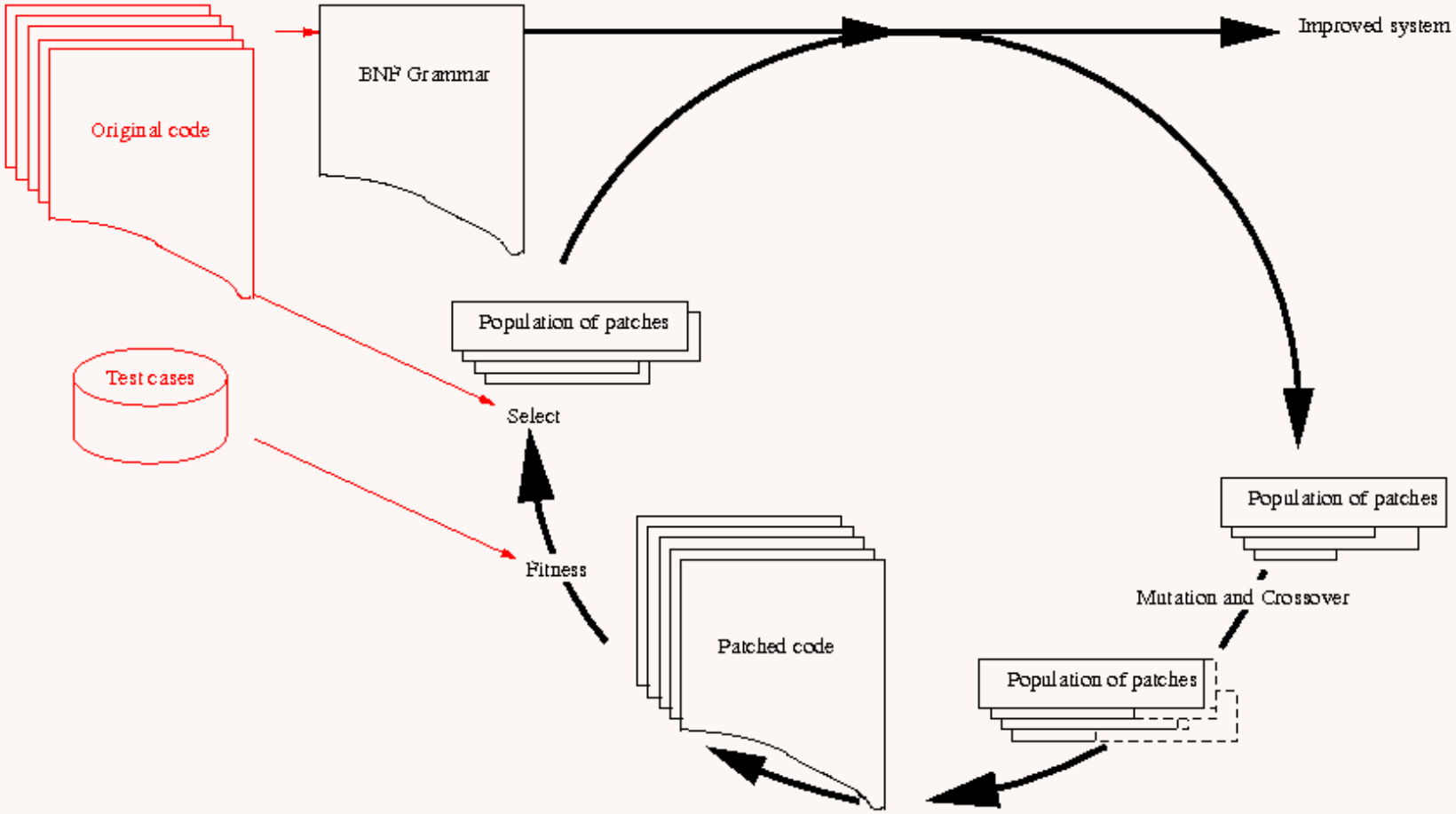
- 2048 global optima in white.
- Regular patterns indicate small building blocks.
- Vertical strips 8 pixels wide says first three bits do not impact fitness.
- Last but one bit gives four horizontal stripes:
  - two contain 50176 mutants fail ≥4 tests (dark)
  - others hold all the solutions (white)

Fitness in bit order ($2^8 \times 2^9$)



- Vertical strips 8 pixels wide says first three bits do not impact fitness.
- Last but one bit gives four horizontal stripes:
  - two contain 50176 mutants fail ≥4 tests (dark)
  - others hold all the solutions (white)

Genetic Improvement evolves code patches



Recent Successes of Genetic Improvement

- Automatic bug repair
  - GenProg, e.g.105 bugs fix most (multiple best papers, IFIP TC2 Manfred Paul Award, 2 Humies)
- Better programs
  - 70x Bowtie2, BarraCUDA, pknots 10000x
  - Less energy, less memory
  - MOGA speed v. quality, e.g. [SIGGRAPH]
- Code transplant [Marginean, e.g. best paper ISSTA 2015]
  - E.g. C++, code indent, call graph layout into Kate editor (we *can* evolve an editor) Humie

Genetic Improvement Benchmarks

- Bugs to be fixed
  - GenProg http://dijkstra.cs.virginia.edu/genprog/
- Software Engineering

  Many, e.g. SIR http://sir.unl.edu
- Fitness landscape

  Mutation testing/GA fitness landscape for the Triangle Program, UCL CS RN/16/05

  http://www.cs.ucl.ac.uk/staff/W.Langdon/ppsn2016/triangle/

  Code and datasets online

  http://www.cs.ucl.ac.uk/staff/W.Langdon/egp2017/triangle/

The Genetic Programming Bibliography

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

11484 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

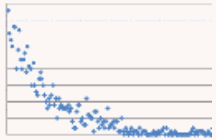
Co-authorships

RSS Support available through the Collection of CS Bibliographies.

Co-authorship community.
Downloads

Downloads by day

A personalised list of every author's GP publications.



Your papers

blog

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