

Genetic Improvement

CREST 10th Anniversary Celebrations
UCL, Wednesday 7th Sept 2016

David R. White
david.r.white@ucl.ac.uk



Challenges in Software Development

Software development is
expensive and **imperfect**

expensive

\$10 billion estimated cost of redeveloping Linux

imperfect

13,499,074 open issues on Github

<https://www.linux.com/publications/estimating-total-cost-linux-distribution>

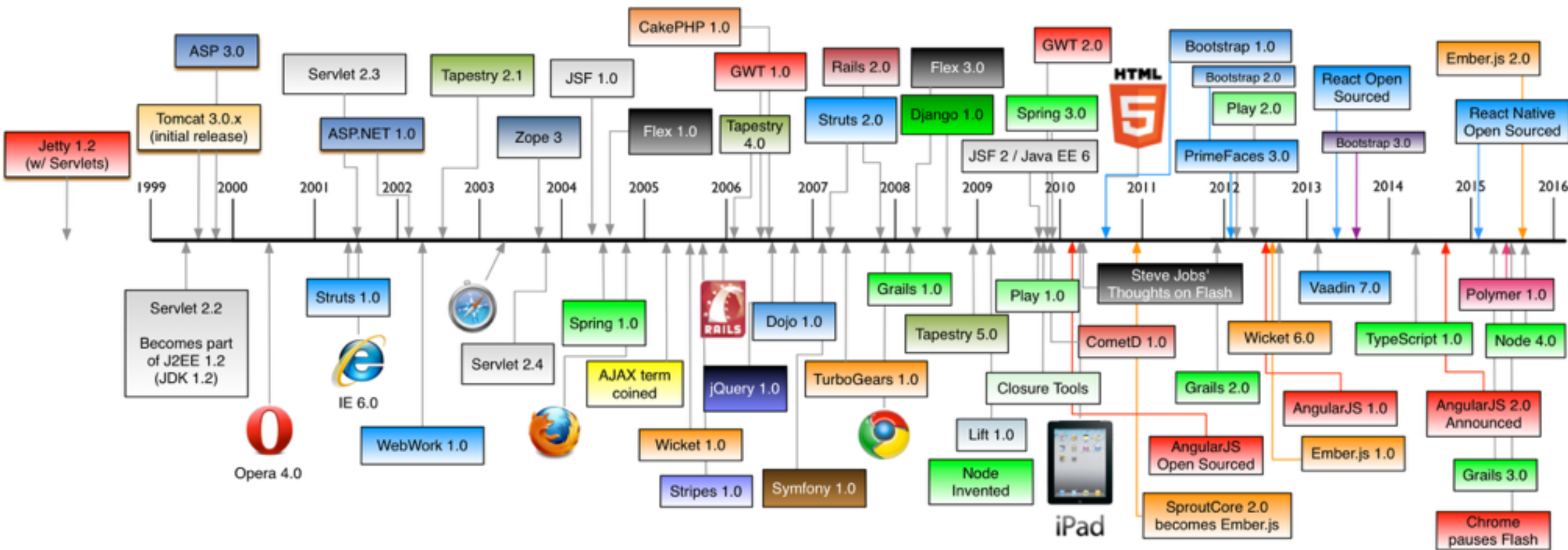
New Challenges



Google | google.com/datacenters

ios





What can we do?

Automate time-consuming tasks.

Assist programmers in difficult tasks.

How can we do it?

The (semi-)automated programming dichotomy:

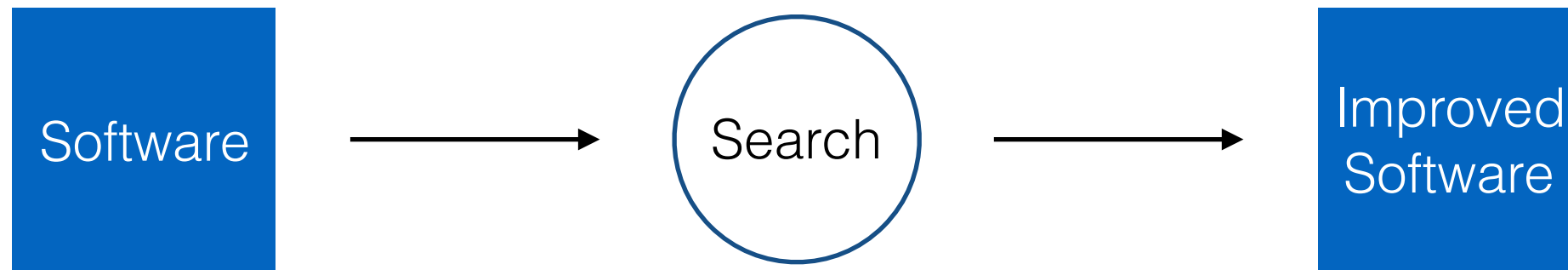
1. Methods that rely on some kind of **derivation**

2. Methods that are **feedback-driven** GI

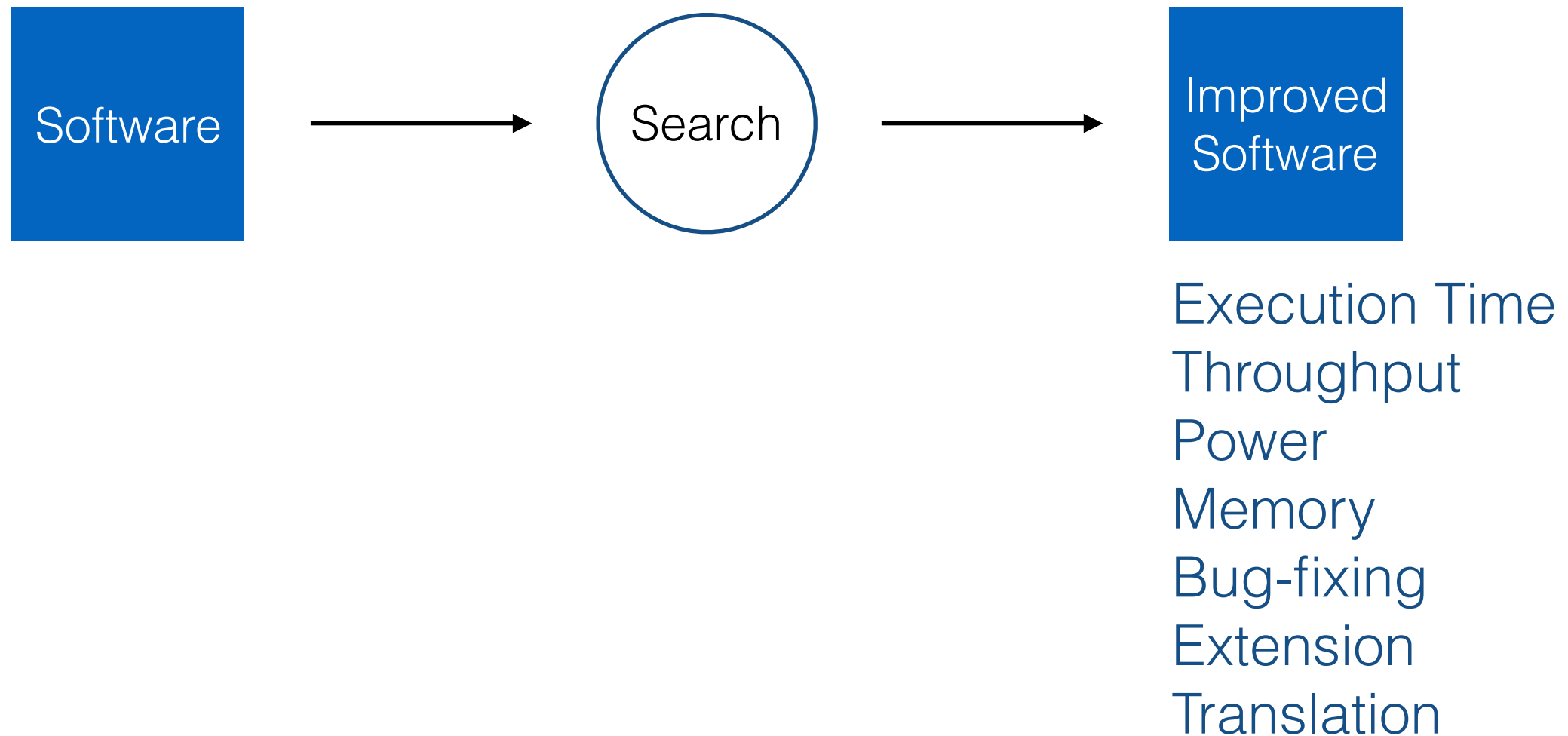
What is GI?

Improvement of Software through Search

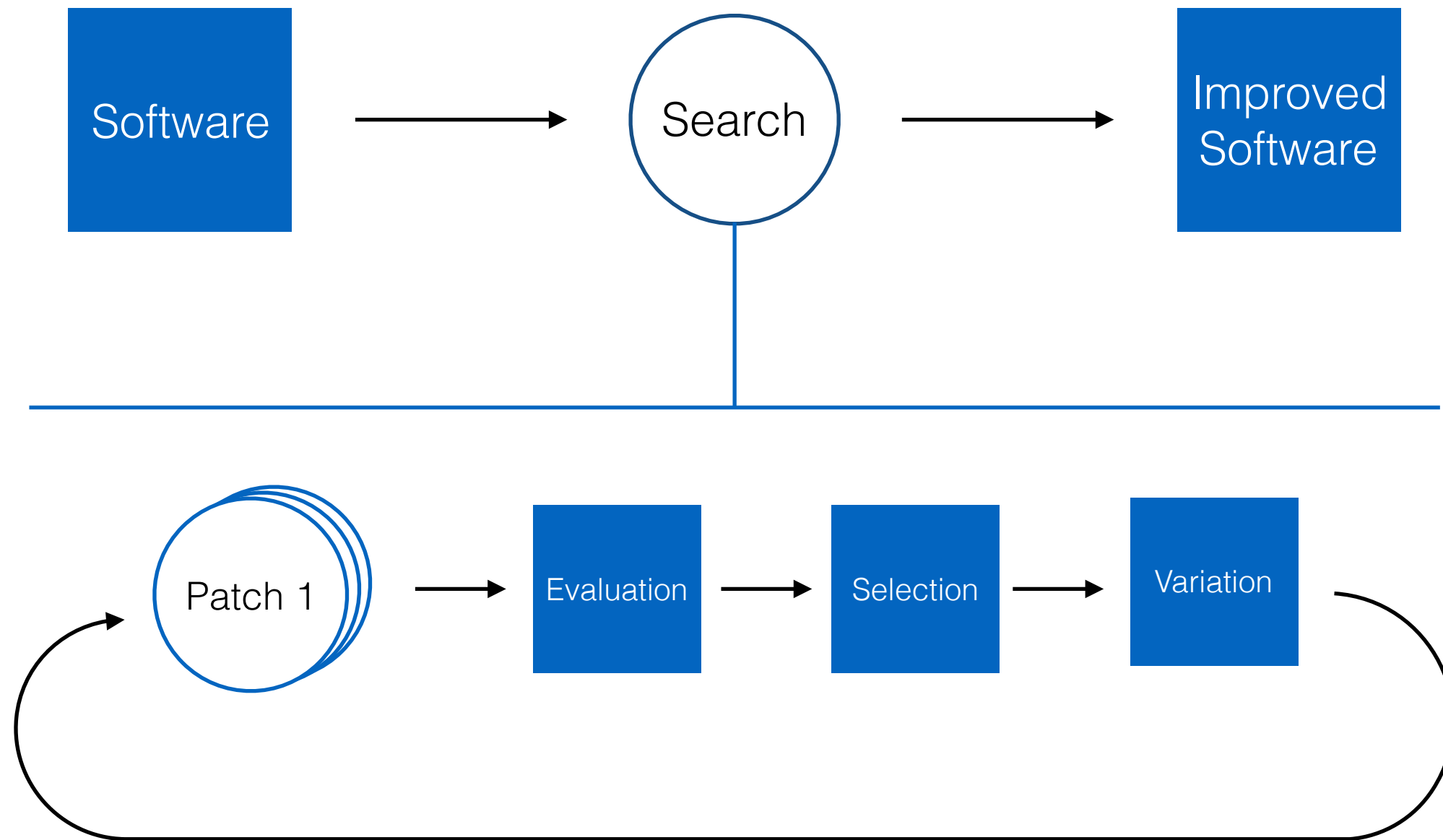
What is GI?



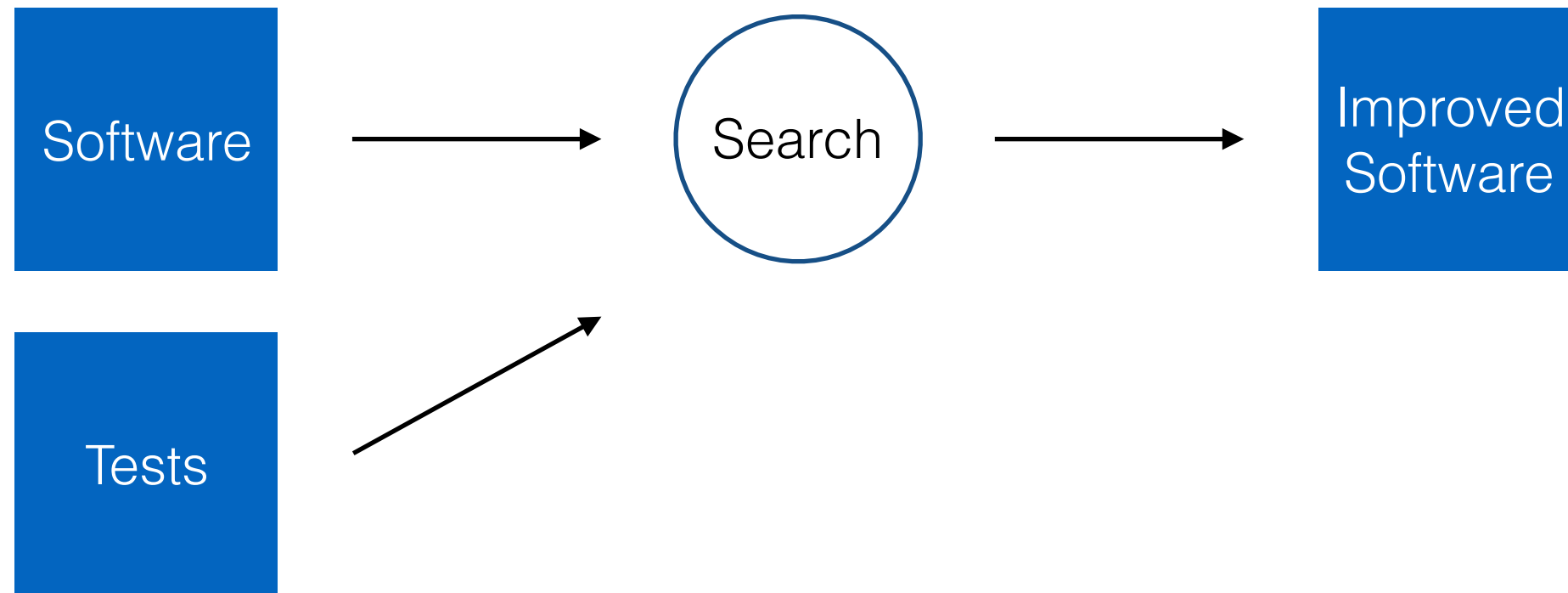
Target Properties



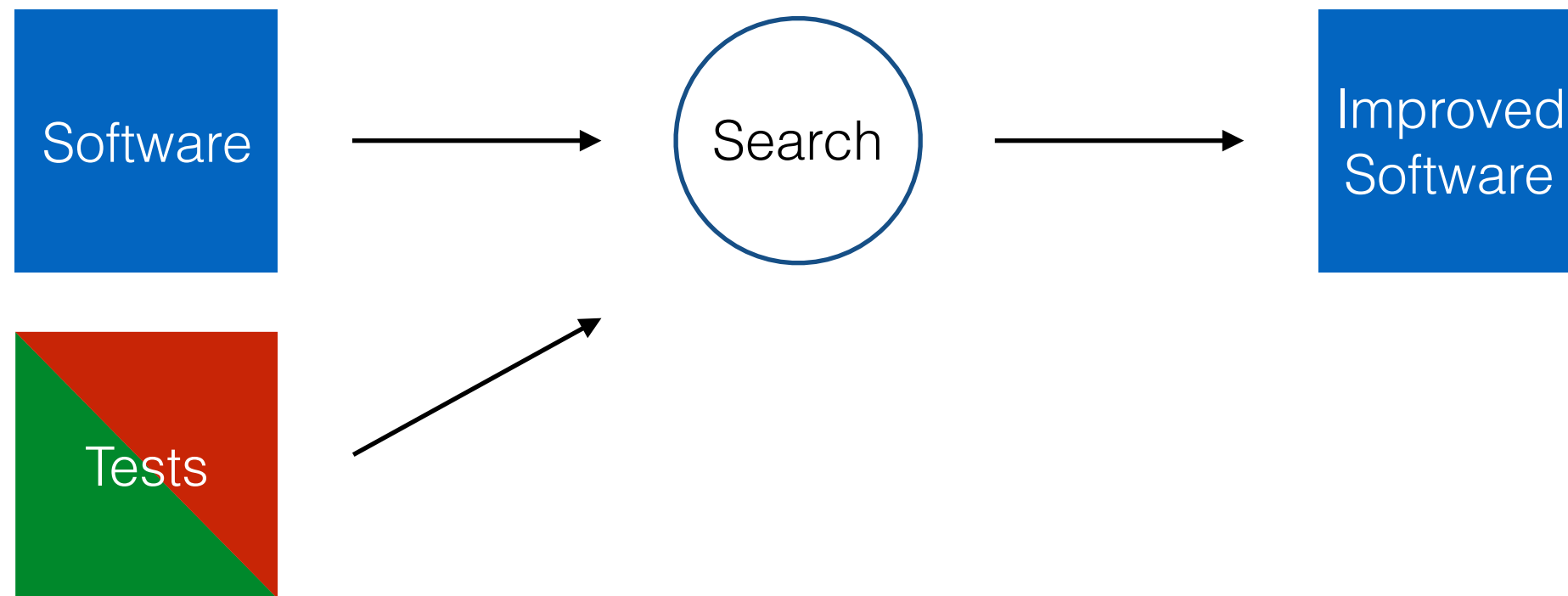
Search Process



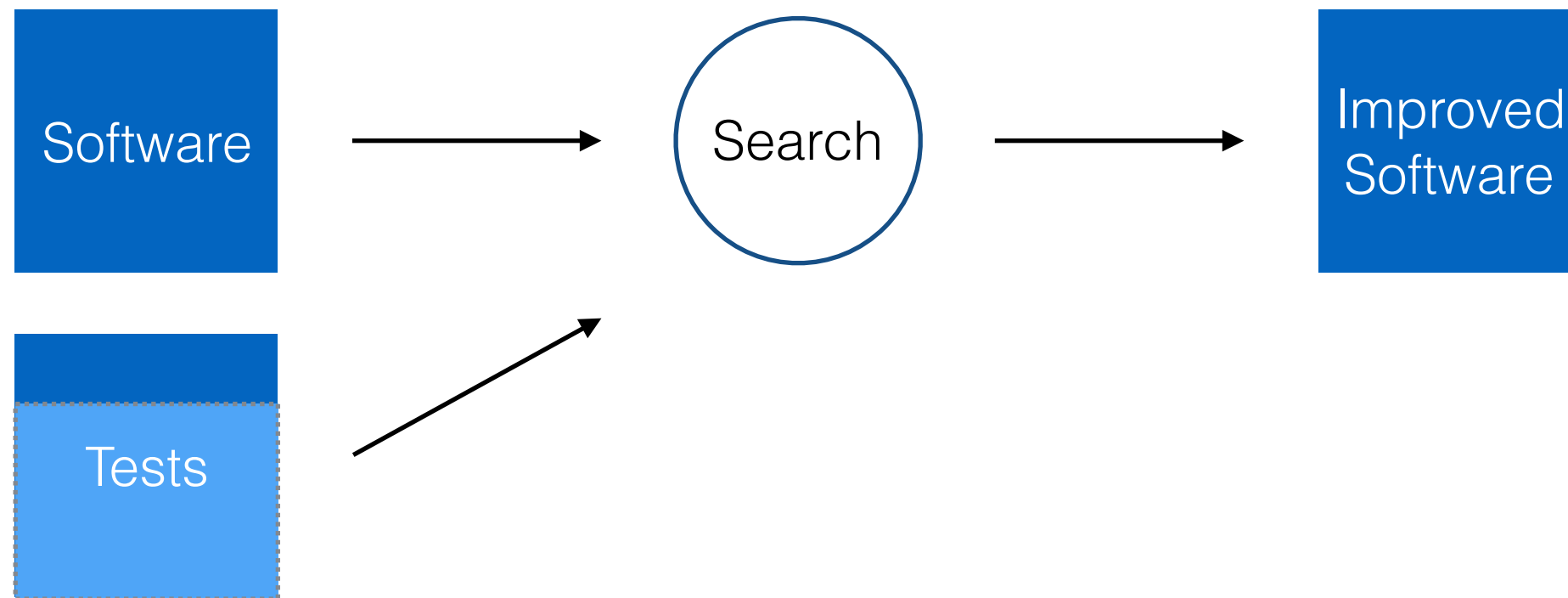
Test Cases



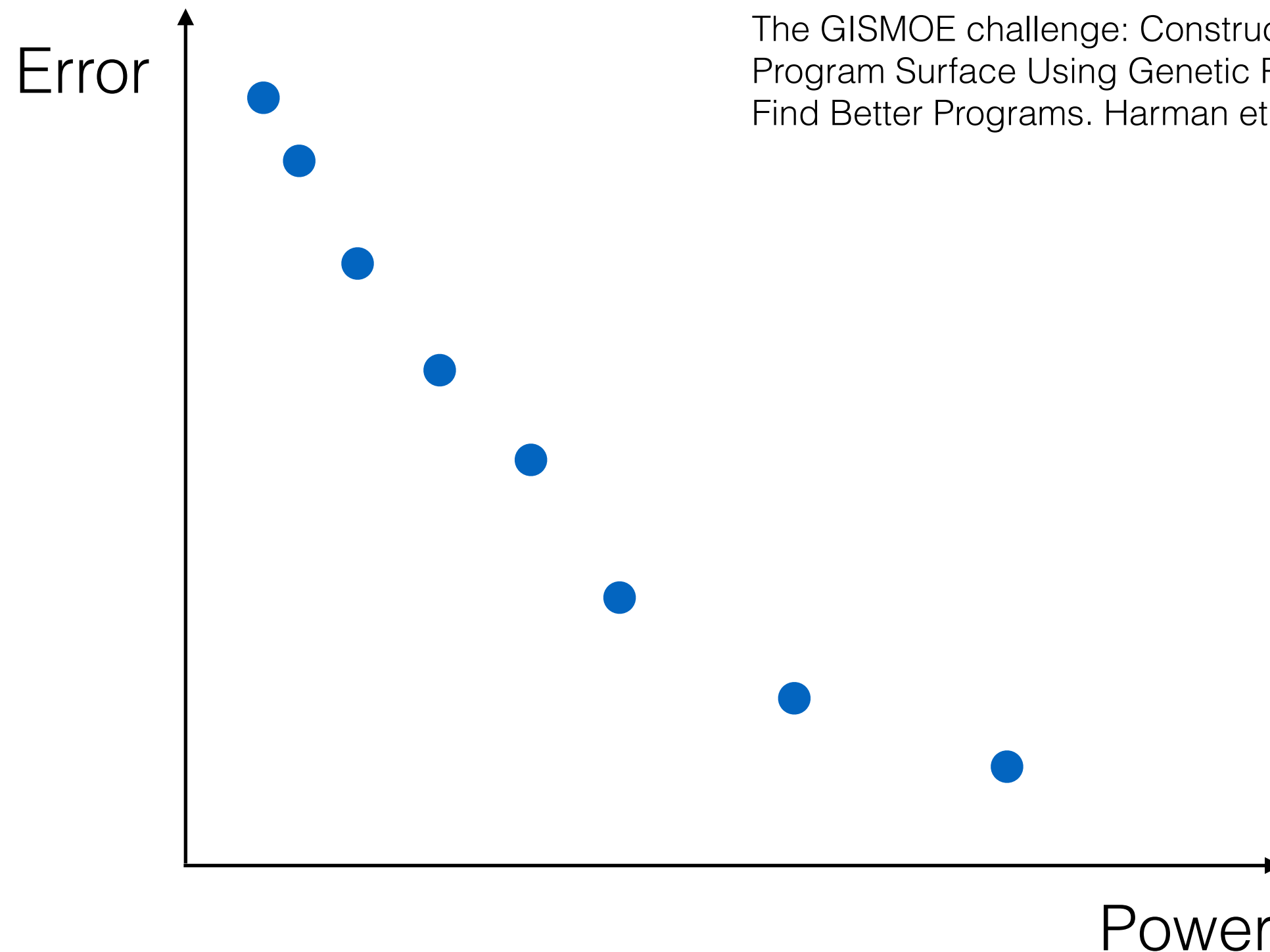
Test Cases: Bug-Fixing



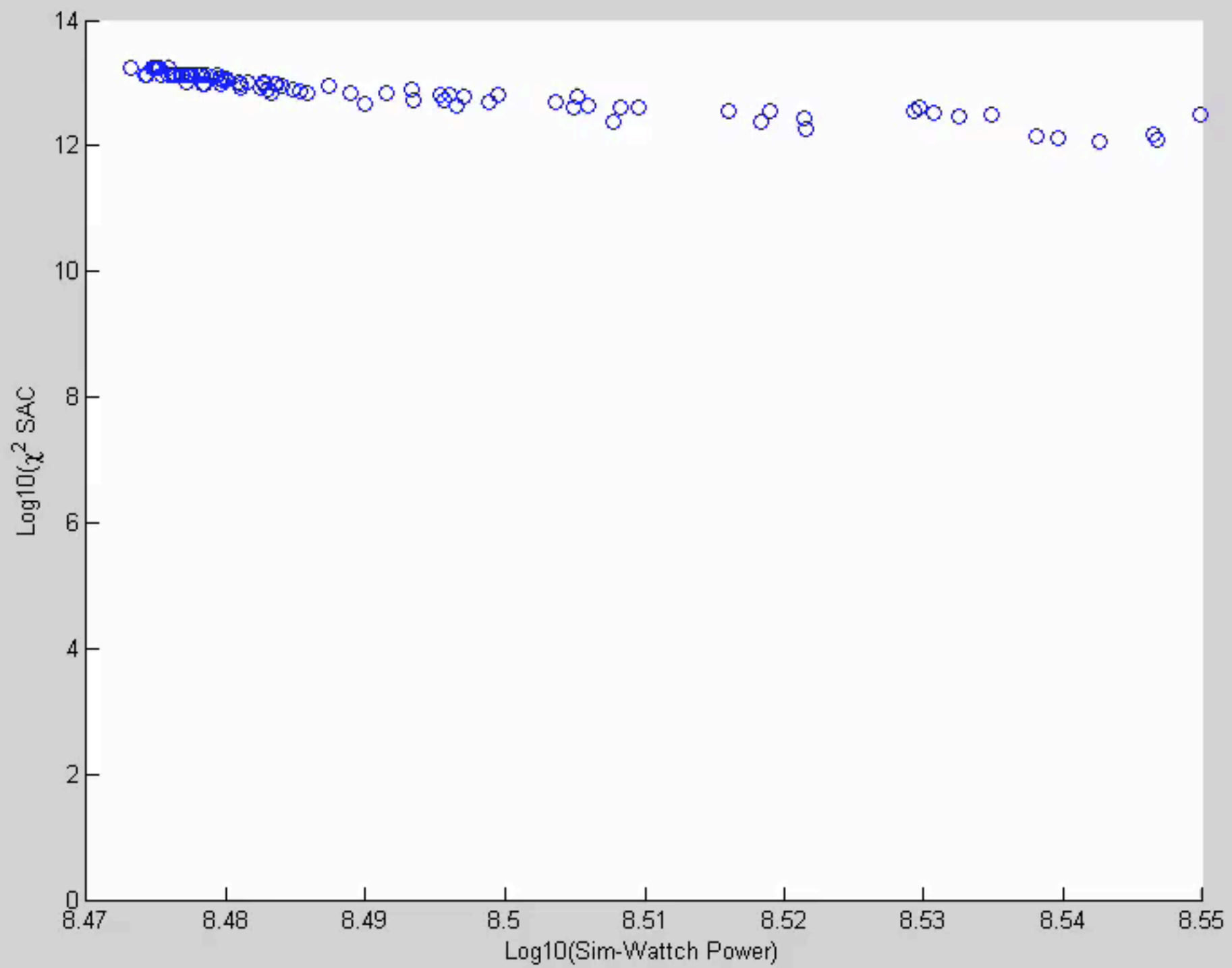
Test Cases: Specialisation



Multi-Objective Trade-offs

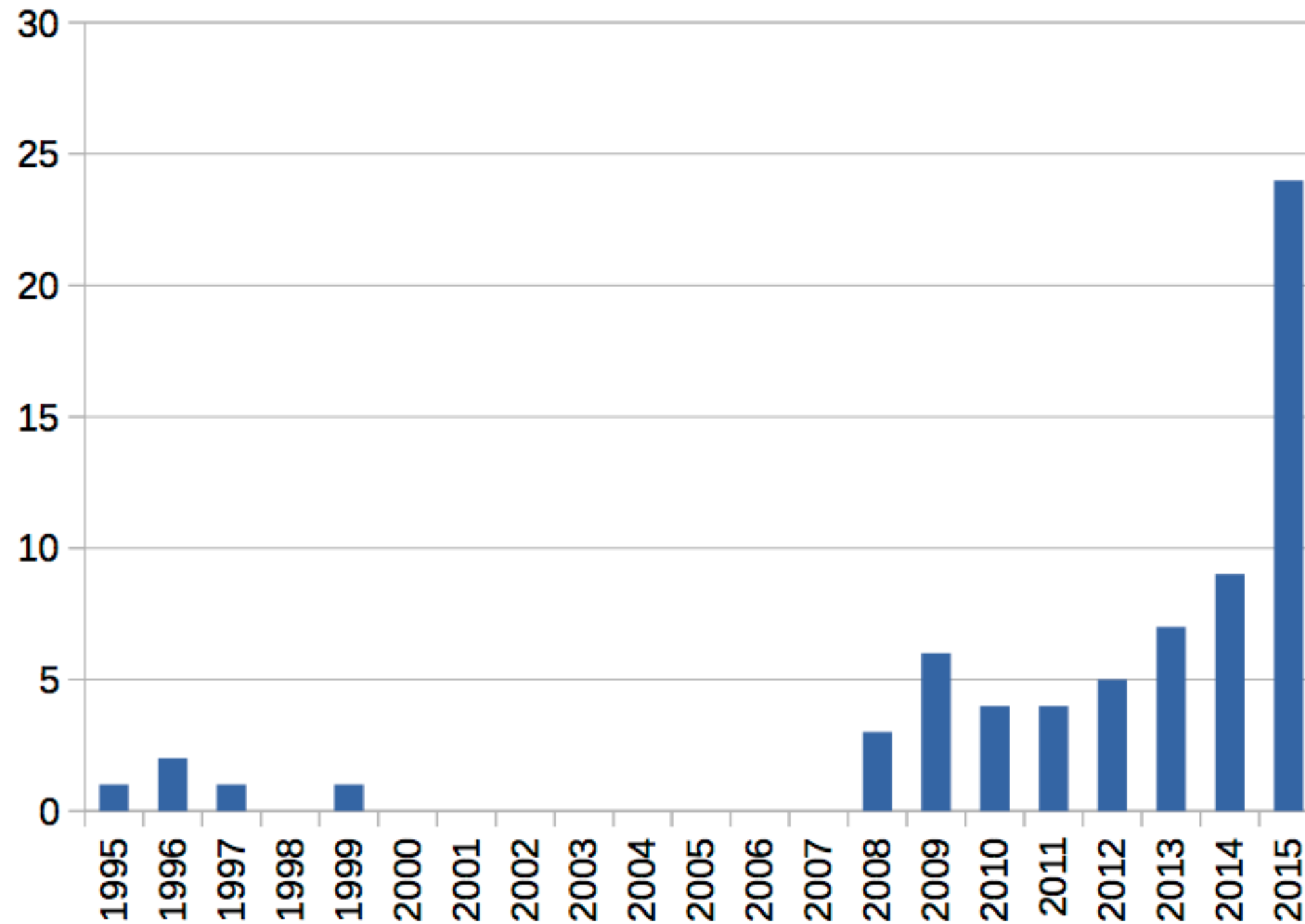


The GISMOE challenge: Constructing the Pareto Program Surface Using Genetic Programming to Find Better Programs. Harman et al. ASE 2012.



CREST and GI

GI Survey



Award-Winning Research

Automatic Software Transplantation

Gold Humie 2016:

ISSTA 2015 ACM Distinguished Paper Award.

Specialising SAT Solver

Silver Humie 2014

Automated Software Transplantation. Barr et al. ISSTA 2015.

Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class.
Petke et al. EuroGP 2014.

EPSRC GI Projects

GISMO: Genetic Improvement of Software for Multiple Objectives.

EP/I033688/1

GGGP: Grow and Graft Genetic Programming

EP/M025853/1

DAASE: Dynamic Adaptive Automated Software Engineering

EP/J017515/1

Major GI Events

Genetic Improvement 2016 @ GECCO

CEC Genetic Improvement Track 2016

CREST Open Workshop on GI January 2016

Genetic Improvement 2015 @ GECCO

Keynotes: ASE, SSBSE, SYNASC, SPLC, SEAMS...

CREST Work in GI

Example Publications

Optimising Existing Software with Genetic Programming

Langdon et al. TEVC 2014. Cited by 83.

Using Genetic Improvement & Code Transplants to
Specialise a C++ Program to a Problem Class

Petke et al. EuroGP 2014. Cited by 51.

Improving CUDA DNA Analysis Software with Genetic
Programming

Langdon et al. GECCO 2015. Cited by 8.

Scalable GI

Optimising Existing Software with Genetic Programming

Langdon et al. TEVC 2014. Cited by 83.

Software under optimisation:

Bowtie2: Aligns genome sequences to longer sequences
50,000 lines of C. 117 files.

BNF grammar to preserve syntactical correctness.

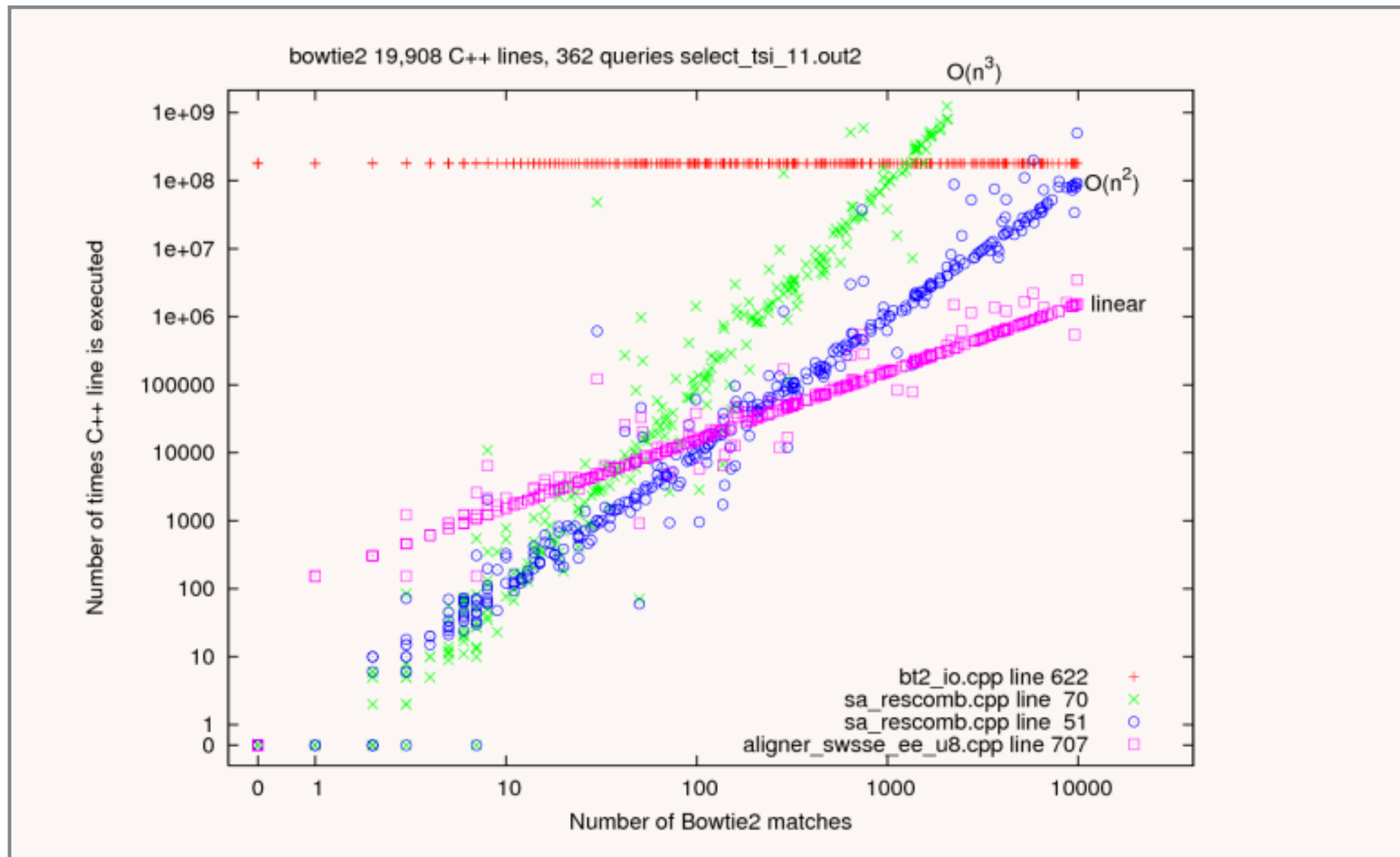
Local search for cleanup post-evolution.

<http://bowtie-bio.sourceforge.net/bowtie2/>
<http://www.cs.ucl.ac.uk/staff/W.Langdon/gismo/>

Extracted BNF Grammar

```
<bowtie_main_42> ::= "int main(int argc, const char **argv) {\n"
<bowtie_main_43> ::= "{Log_count64++;/*29823*/} if" <IF_bowtie_main_43>
" {\n"
#"if
<IF_bowtie_main_43> ::= "(argc > 2 && strcmp(argv[1], \\"-A\\") == 0)"
<bowtie_main_44> ::= "const char *file = argv[2];\n"
<bowtie_main_45> ::= "ifstream in;\n"
<bowtie_main_46> ::= "" <_bowtie_main_46> "{Log_count64++;/*29826*/}\n"
#other
<_bowtie_main_46> ::= "in.open(file);"
<bowtie_main_47> ::= "char buf[4096];\n"
<bowtie_main_48> ::= "int lastret = -1;\n"
<bowtie_main_49> ::= "while" <WHILE_bowtie_main_49> " {\n"
#WHILE
<WHILE_bowtie_main_49> ::= "(in.getline(buf, 4095))"
<bowtie_main_50> ::= "EList<string> args;\n"
<bowtie_main_51> ::= "" <_bowtie_main_51> "{Log_count64++;/*29831*/}\n"
```

Empirical LOC Complexity



Bowtie2: Speedup

Over 70x increase of average throughput.

4 **hours** vs 12 **days** on unseen data.

Specialisation using GI

Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class

Petke et al. EuroGP 2014. Cited by 51

Specialisation to an expected input class, with human-competitive results.

Software under optimisation:

MiniSAT Solver.

2419 lines of C++, focused to 478

Expected use case: Combinatorial Interaction Testing.

Greedy algorithm to find synergistic patches.

Speedup: 17% speedup from original program; 4% faster than any human-modified version (achieved in hours). 56 changes.

GPGPU Acceleration using GI

Improving CUDA DNA Analysis Software with Genetic Programming

Langdon et al. GECCO 2015. Cited by 8.

Target Software:

BarraCUDA: DNA Matching Software.

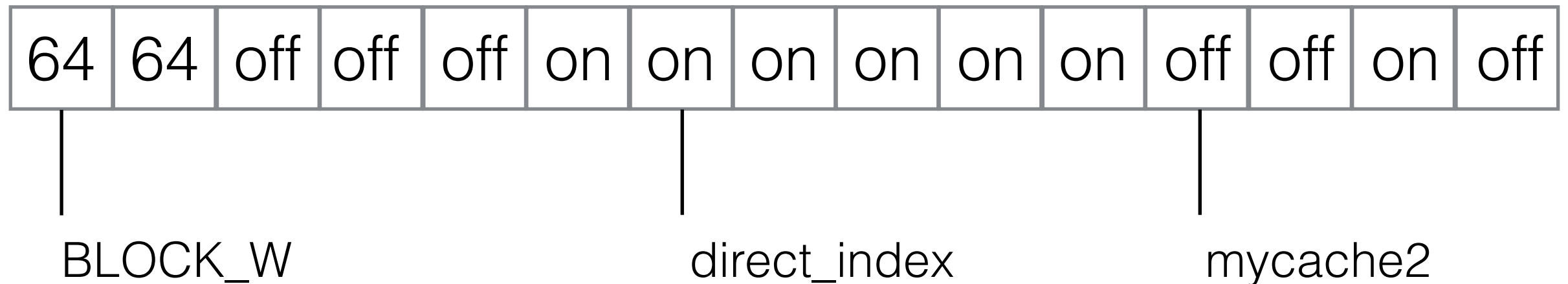
Aligns millions of short noisy DNA strings to a reference genome.

Handwritten CUDA Port of existing tool since 2009.

8000 lines of C, six kernels.

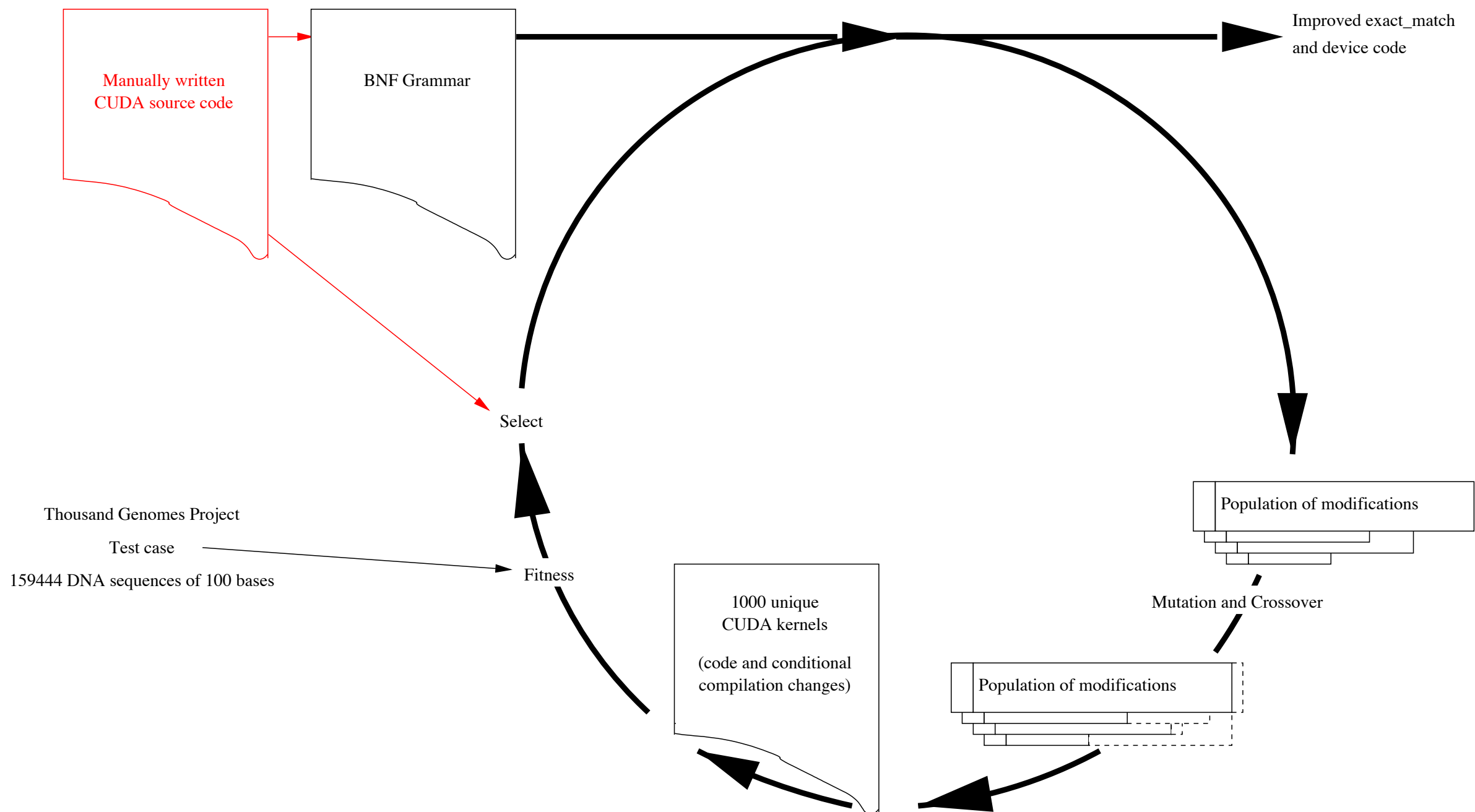
Representation

1. Architectural Config:



2. Patch:

```
<_Kkernel_bnf.cu_912> delete line 912  
<_Kkernel_bnf.cu_948><_Kkernel_bnf.cu_927> replace  
<_Kkernel_bnf.cu_852>+<_Kkernel_bnf.cu_922> insert after  
...
```



BarraCUDA: Results

Original: **15000** sequences per second

Optimised: more than a **million!** (on test set)

How:

- Architectural tuning (register use; threads; memory)

- Removing inefficient cache (but why?)

- Eliminate redundant tests

- Add unroll pragmas

Improving CUDA DNA Analysis Software with Genetic Programming

Langdon et al. GECCO 2015. Cited by 8.

The BarraCUDA Project

What's New

Using genetic programming, v0.7.x now offers a
>100% boost in speed compared to v0.6.2

Supports new BWTs built for BWA Version 0.6.x+



<http://seqbarracuda.sourceforge.net>

Summary

GI @ CREST

CREST is one of the leading GI institutions

Award winning research

Industrial uptake

What's next?

Deep Parameter Tuning

Concurrency

Power Optimisation

Dreaming Smartphone

Deep Parameter Optimisation. Wu et al. GECCO 2015.

Reducing Energy Consumption Using Genetic Improvement. Bruce et al. GECCO 2015.

Genetic Improvement for Adaptive Software Engineering. Harman et al. SEAMS 2014.

GI Researchers @ CREST



Mark



Bill



Earl



Yue



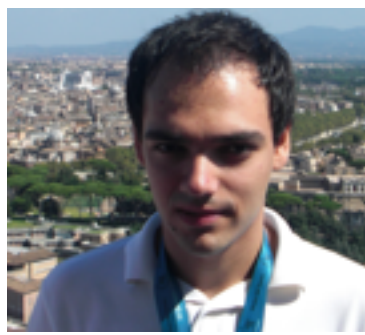
Justyna



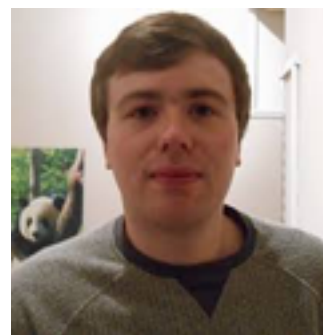
David



Fan



Alex



Bobby



Leo

```
19 bool again = true;
20
21 while (again) {
22     iN = -1;
23     again = false;
24     getline(cin, sInput);
25     system("cls");
26     stringstream(sInput) >> dblTemp;
27     ilength = sInput.length();
28     if (ilength < 4) {
29         again = true;
30         continue;
31     } else if (sInput[ilength - 3] != '.') {
32         again = true;
33         continue;
34     } while (++iN < ilength) {
35         if (isdigit(sInput[iN])) {
36             continue;
37         } if (iN == (ilength - 3)) {
```

Computer Programming To Be Officially Renamed “Googling Stackoverflow”

103k
SHARES

f Facebook

101k

Twitter

16

G+ Google+

1.1k

in LinkedIn

835

Reddit

Washington DC – The IEEE have produced a report today where they strongly recommend that from now on, the discipline of Computer Programming should be officially renamed to “Googling Stackoverflow”.

“We are recommending a root-and-branch name change to this discipline”, said President of the IEEE, Thomas M. Conte. “We are even going to change the official name of the IEEE Computer Society to the IEEE Quick Look At StackOverflow Society”.

“We are furthermore recommending that all universities across the planet should cease to award Bachelors degrees, Masters or PhDs in computer programming or software engineering and instead they should award these degrees in Googling Stackoverflow.”

“ultimately, genetic improvement looks forward to a world in which our successors regard human programmers as a ‘quaint anachronism of the past’ in the same way that we now regard the human *computers* of our nineteenth and twentieth century forbearers...”

Langdon et al.

Optimising Existing Software with Genetic Programming. TEVC 2012.

Optimising Existing Software with Genetic Programming

Langdon et al. TEVC 2014.

Using Genetic Improvement & Code Transplants to Specialise a C++ Program to a Problem Class

Petke et al. EuroGP 2014.

Improving CUDA DNA Analysis Software with Genetic Programming

Langdon et al. GECCO 2015.

Image Credits

iOS image

User Jcdriodch

https://commons.wikimedia.org/wiki/File:IOS7_Logo.png

Android

Google, Inc.

https://commons.wikimedia.org/wiki/File:Android_robot.svg

Web Framework timeline from:

<https://github.com/mraible/history-of-web-frameworks-timeline>

Stack Overflow story from:

<http://www.theallium.com/engineering/computer-programming-to-be-officially-renamed-googling-stackoverflow/>