CREST

Genetic Improvement:
A Key Challenge for Evolutionary Computation

Key Challenges and Future Directions of Evolutionary Computation Workshop, Yun Li *et al*. IEEE CEC-2016

W. B. Langdon
Department of Computer Science

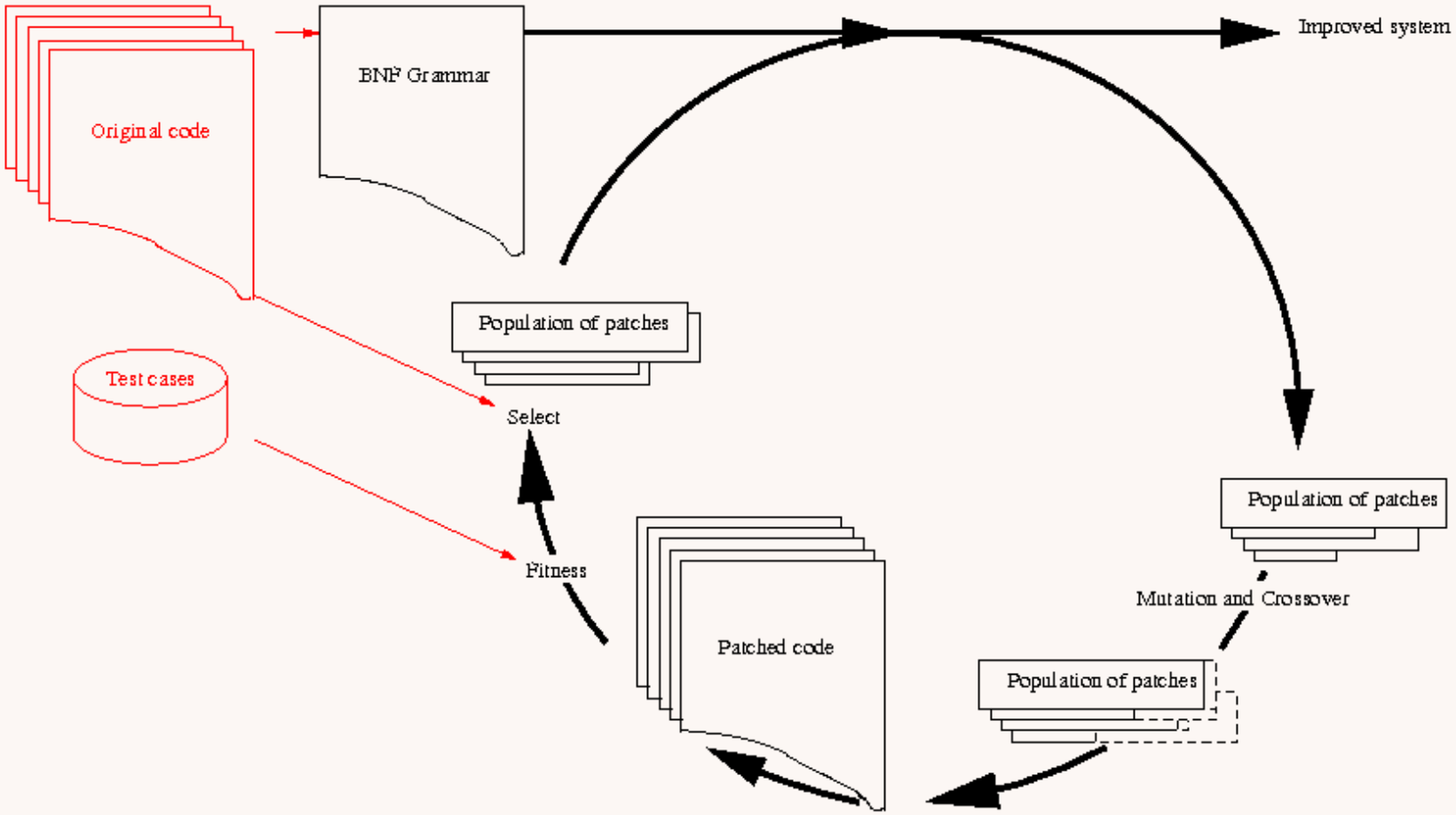
Genetic Improvement of Software

- Background
  - What is Genetic Improvement
  - Recent successes: automatic bug repair, better programs, code transplant.
  - Pragmatic: EC and human, sematic search
- Challenges to Evolutionary Computing
  - Better scaling
  - Automated programming,
  - Evolving artificial intelligence
  - Landscape analysis
  - Benchmarks

What is Genetic Improvement

- Application of evolutionary computation (often genetic programming) to existing software.
- Real programs (10^4 to 10^6 of lines of code)
- Key advantages
  - We know if mutated code is better/worse
  - Compare with original (hand made) code
  - Use existing software to create mutation
  - Small changes: delta debugging, incremental compilation, user acceptance

GI Evolving 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)

Evolution and Human Programmers

- EC and human
  - Grow and Graft GP, e.g.bi-translation for Pidgin
- Evolution to suggest new code to human
  - User in control
  - Less skilled (user) programmer, e.g. Flash fill.
  - Reject and/or re-code solution suggested by evolution
  - Evolution as code invention machine
- Sematic search

  “google” for code to do X, automate insertion [ASE2015]

Need for Improved Scaling

- Bugzilla 300+ bug reports per day [FSE-2009]
- Crowd source (volunteers)
  - Do we have that many programmers.
    Should we use people to do it. Do we trust them
  - Machine to generate 300 fixes per day?

Challenges to Evolutionary Computation

Challenges: Better Scaling

- Multiple aspects to better scaling
- Tackle bigger problems, rather than solving existing problems faster.
- EC well poised to exploit parallel hardware What of programming in parallel, eg crowd
- Hybridise EC with other techniques
  - Test generation
  - Using source code allows GI to be mixed with many existing software engineering tools?
  - Program validation, SAT solvers

Automated Programming

- Research on Evolutionary Computing started at the start of Artificial Intelligence.
- Given recent success of AI (driving cars) does EC need to redouble its efforts in AI
- Does recent success of Genetic Improvement suggest a EC route to automated programming?
- Or should GI only move humans up a level from source code to saying what programs should do, not how to do it?

Evolving Artificial Intelligence

- Is it as far away as it seems?
- Raj Reddy 1988 AAAI Presidential Address
  - ~~Chess~~, speech, vision, expert systems, accident avoiding car, math discovery (EC discovering quantum computing algorithms), theorem proving, motor processes (EC humanoid robotics), translating telephone
  - Excluding self-replicating systems, in 28 years considerable progress on all areas.

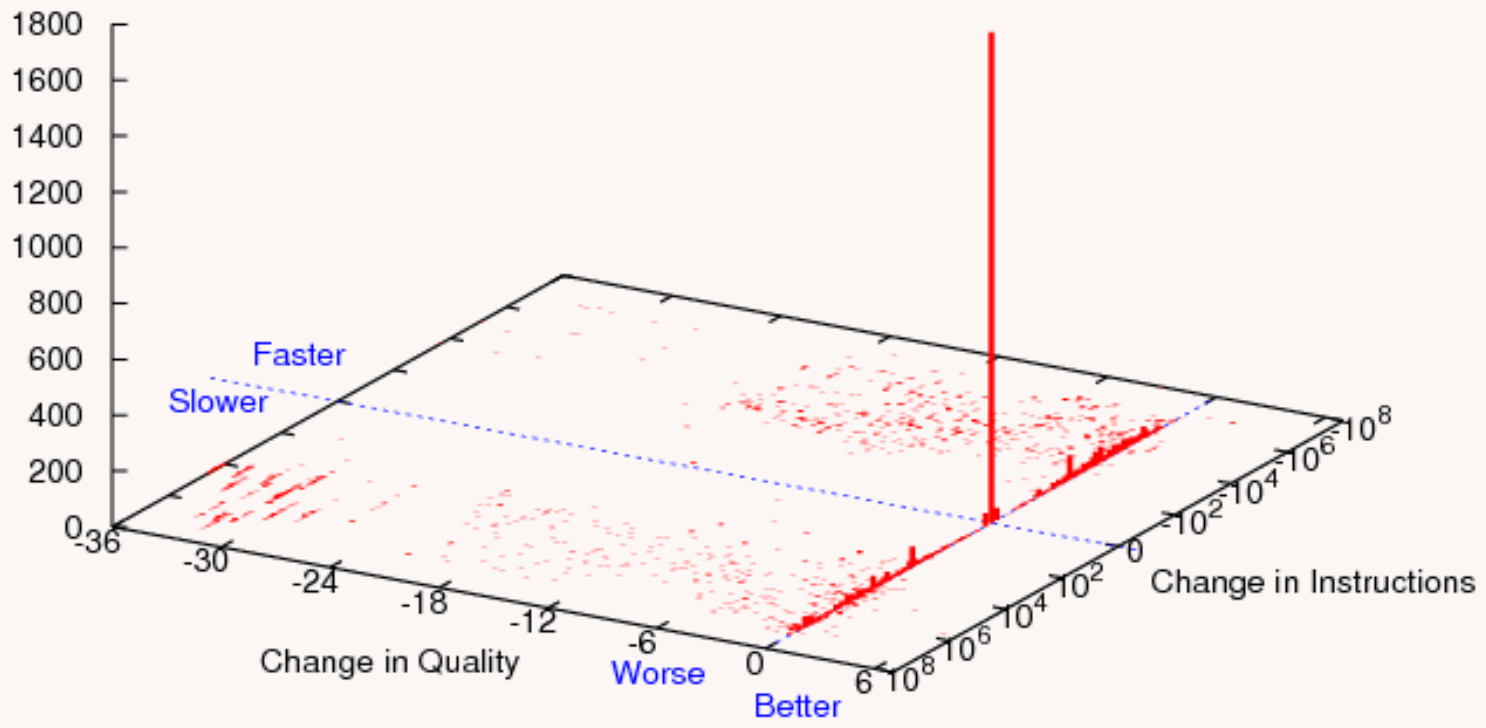
Evolving Artificial Intelligence

- Is it as far away as it seems?
- Huge stockpiles of source code free for use
- Huge (parallel) computation available.
- The DARPA Grand Challenge autonomous vehicles, something similar for EC-AI? Not an AI but a step (a jump)

Fitness Landscape Analysis

C++ is not fragile
Trading performance v speed

10000 random mutation runs GISMO bowtie2, WBL 3 May 2012



Program Mutation Landscape

- Not as hard as often claimed
- Triangle Program benchmark
- EC researcher already have many fitness landscape analysis tools. Can they be applied to searching real programs?

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/

Conclusions

- Recent successes of Genetic Improvement
- Challenges to Evolutionary Computing

  Better scaling: to tackle bigger problems

  Automated programming: GI as aid to user

  Evolving Artificial Intelligence: is it so far away

  Landscape analysis: C++ is not fragile

  Benchmarks: GenProg and Triangle Program
- EC as 2nd best. Red Queen. White Queen. Do the impossible

END

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

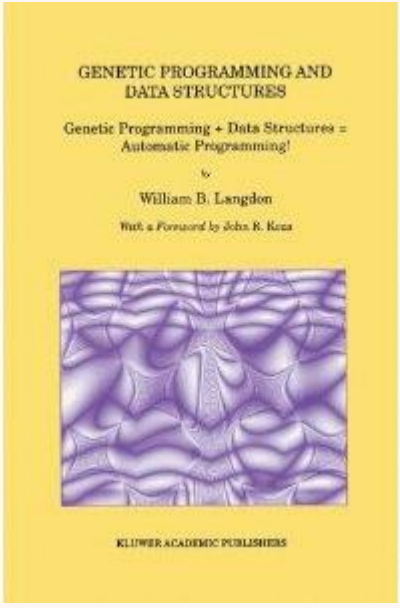
http://www.epsrc.ac.uk/ EPSRC

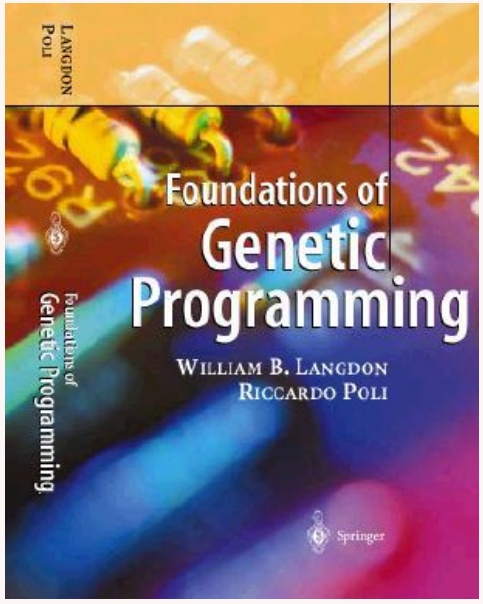
Genetic Improvement

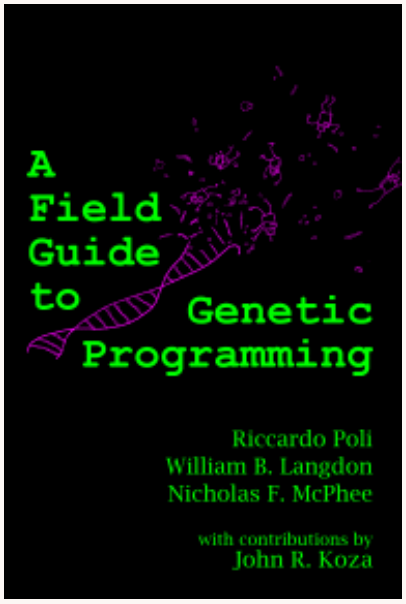
W. B. Langdon

CREST

Department of Computer Science







The Genetic Programming Bibliography

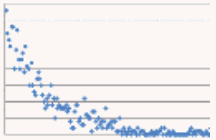
http://www.cs.bham.ac.uk/~wbl/biblio/

11138 references

RSS Support available through the Collection of CS Bibliographies. XML RSS

A web form for adding your entries. Co-authorship community. Downloads

A personalised list of every author's GP publications.



blog

Google scholar citations

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