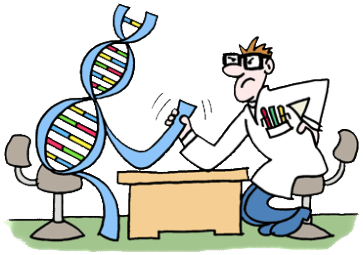
# Memory Fitness Landscapes

14th International Workshop on Genetic Improvement
Sunday 27 April 2025



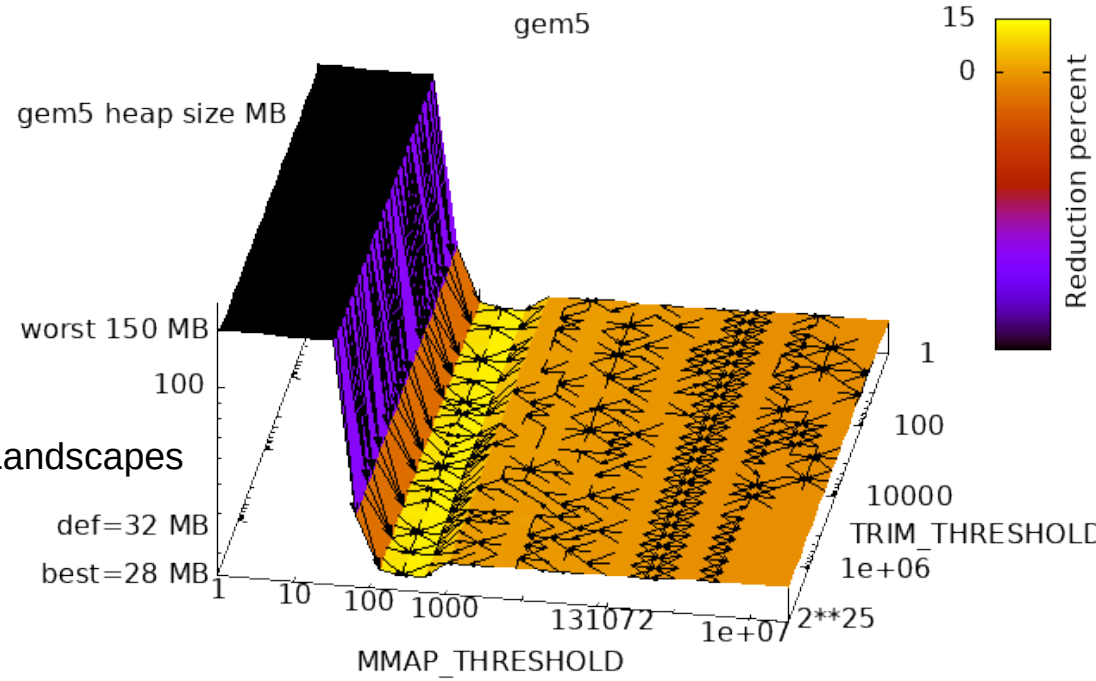
Humies $10000 prizes
Submit by **Friday 30 May**

W. B. Langdon, UCL

Can sometimes tune C++ dynamic memory.
Heap memory landscapes are smooth.

The gem5 C++ glibc Heap Fitness Landscape,
GI @ ICSE 2025, Ottawa
W.B. Langdon and B.R. Bruce.

gem5/Z3/gcc/Clang/Redis glibc Heap Fitness Landscapes
Evo* 2025 Late-Breaking Abstracts.
W.B. Langdon, J.Petke, D. Clark

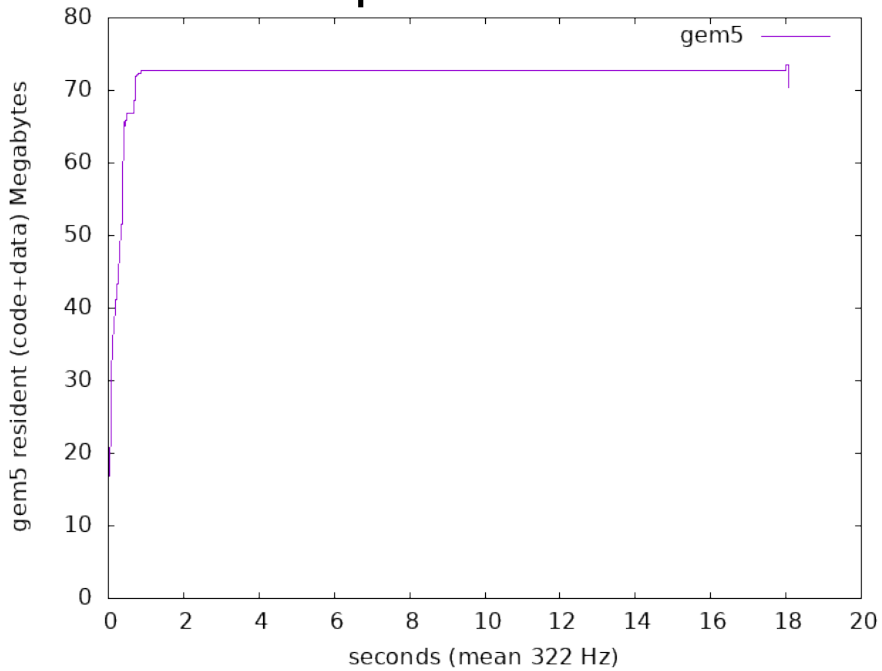


# The gem5 C++ glibc Heap Fitness Landscape

- GI 2025 gem5 landscape smooth saving 12%
  Evo* 2025 LBA try 4 more: smooth but little improvement
- What is gem5?
- What is GNU C malloc() Heap?
- What are Fitness Landscapes?
- Why glibc heap is a “nice” landscape
- Measuring memory: malloc_info(), massif, top
- Genetic Improvement: Magpie, CMA-ES
- Others
  - Z3 Microsoft theorem prover
  - GNU g++ compiler
  - Clang LLVM compiler
  - Redis key-value store
- Heap differences: smooth but little gain
- Why

# What is gem5?

- gem5 is a single threaded 1.3 million line C++ program
- It is a discrete time simulation for CPU, cache, memory, etc.
- It can look for performance bugs before the hardware is created.
- Eg to simulate a single machine code instruction, the instruction must be fetched, the data it needs and its outputs stored.
- gem5 creates timed events these.
- Millions of different C++ heap objects are created and deleted as events are started and complete.



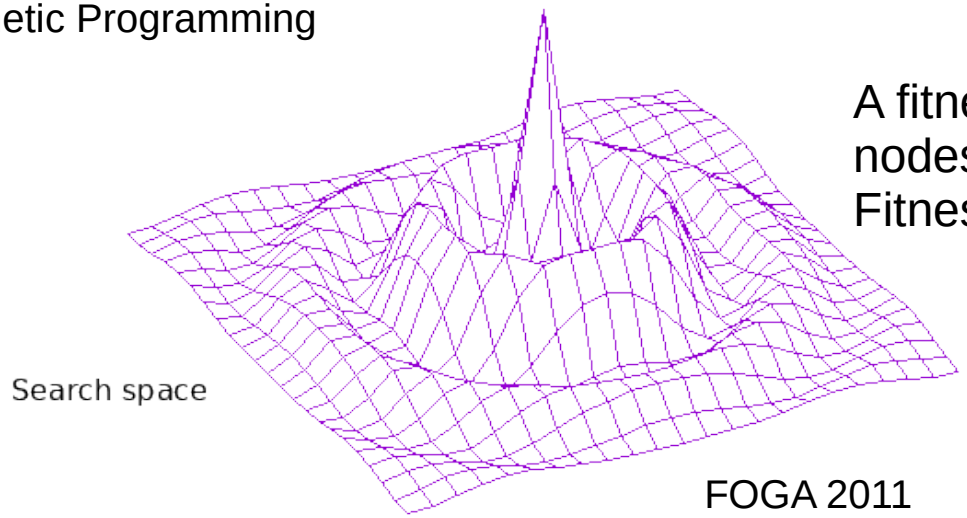
# What is the GNU C malloc Heap?

- In C dynamic memory is created by malloc() etc.
- In g++ malloc is also used
- 37 /lib64/ld-linux-x86-64.so.2 --list-tunables
- g++ malloc 7 tuning parameters:
  - M_ARENA_TEST, M_ARENA_MAX, M_PERTURB, M_TOP_PAD
  - **M_MMAP_MAX, M_TRIM_THRESHOLD M_MMAP_THRESHOLD**
- Only last three relevant to single thread programs
- Total space 2**(32+64+64) = $2^{160}$ = 1.5 $10^{48}$

# What are Fitness Landscapes?

- In genetic algorithms etc a search space is a graph where adjacent nodes (a,b) are potential solutions which are connected iff there is a genetic operation (mutation or crossover) which allows a move from node a to node b.
- Often
  - a,b is symmetric
  - probability of a given mutation/xo is ignored.
  - High dimensional graph shown only in two dimensions
- Performance of each node represented as altitude (z-axis)

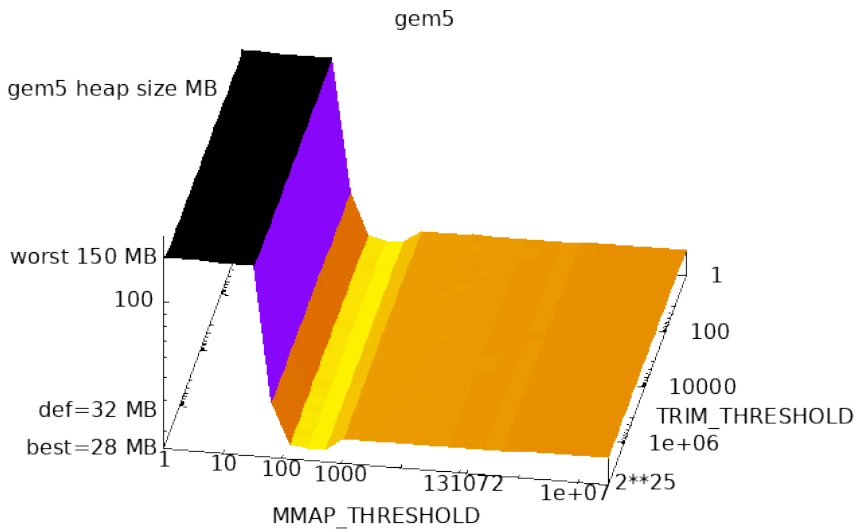
Foundations of Genetic Programming

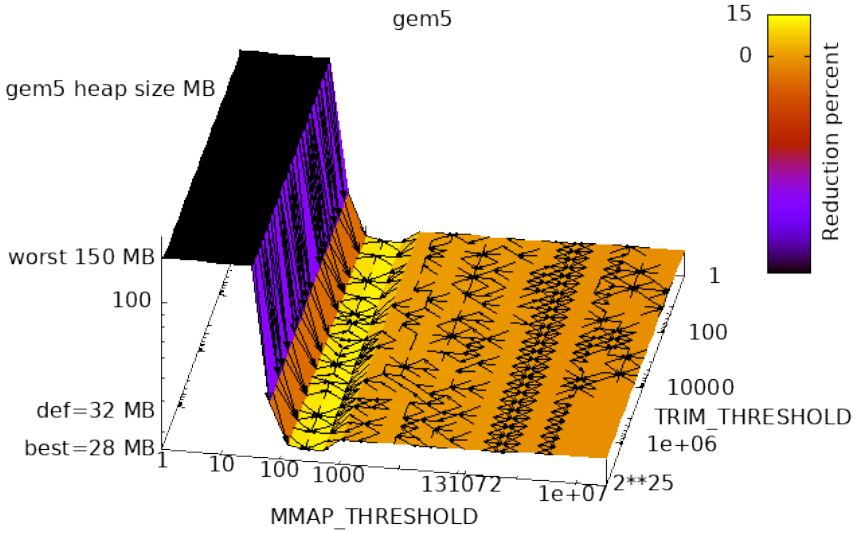



A fitness landscape where nodes have 4 neighbours. Fitness is plotted vertically.

# Why glibc heap is a “nice” landscape

- Although representation is 96 bits ($2^{96}$) only three continuous dimensions
- One dimension (M_MMAP_MAX) makes almost no difference, any reasonable value will do.
- Can navigate landscape near defaults ($2^{16}$,$2^{17}$,131072) assuming it is continuous
- Sample two dimensional grid at powers of two and half powers
- Add arrows to emphasis gradient to better (lower) values



# Measuring Memory, malloc_info(), massif, top

- malloc_info()
  - Precise heap data
  - Heap stats when called
  - Small code change to add calling malloc_info
- valgrind --tool=massif (heap profiler)
  --pages-as-heap=yes --peak-inaccuracy=0.0
  - High overhead (gem5 15x slow down, others less)
  - Gives max heap size
- Top
  - -b -d0.0
  - Limited sampling rate (max about 300 samples/second)
  - Gives same data as /proc/self/statm, ie total code+data (not heap only)
- Others, eg perf

# Genetic Improvement of Parameters with Magpie

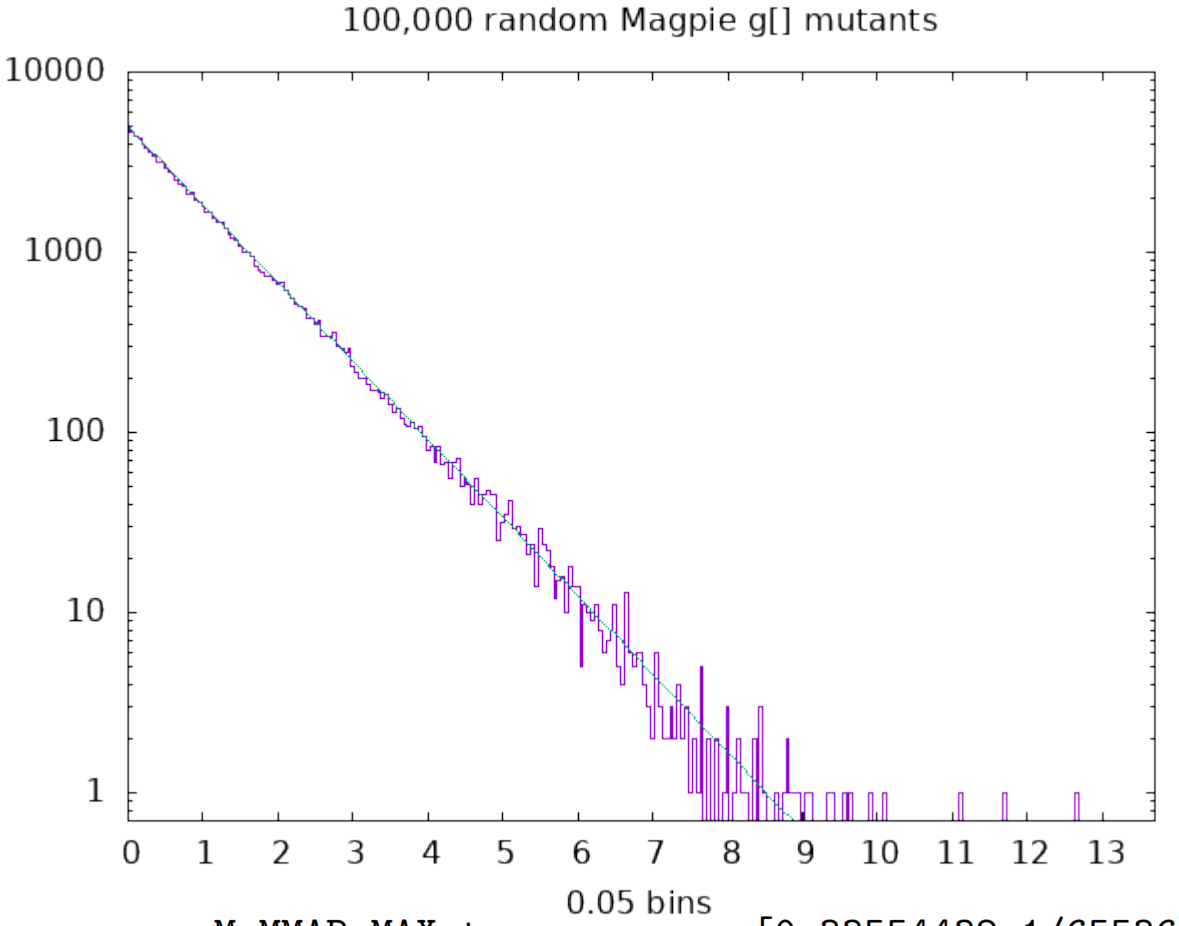
- Genetic Improvement typically applied to source code
- Improvements can be found by tuning parameters [EuroGP 2018]. Cf. deep parameter tuning
- Magpie can do both code and parameter tuning
- Magpie parameter file specify types of mutation
  - Geometric
    - set range 0 to 256x default,
    - set mutation default to be malloc default
    - mean: opps should have been default, was 2.56x default
- setenv environment variable to change malloc parameters
- Run gem5 with and without parameter changes
- Fitness = ratio reduction in peak heap (malloc_info)
- 1000 fitness trials
- 12% reduction

```
M_MMAP_MAX_tune          g[0,33554432,1/65536][65536]
M_TRIM_THRESHOLD_tune    g[0,33554432,1/131072][131072]
M_MMAP_THRESHOLD_tune    g[0,33554432,1/131072][131072]
```

Corrected magpie parameters

# Magpie g[] geometric mutation distribution

With g[] magpie generates mutant values with a geometric distribution: specify default, mean and lower and upper bounds

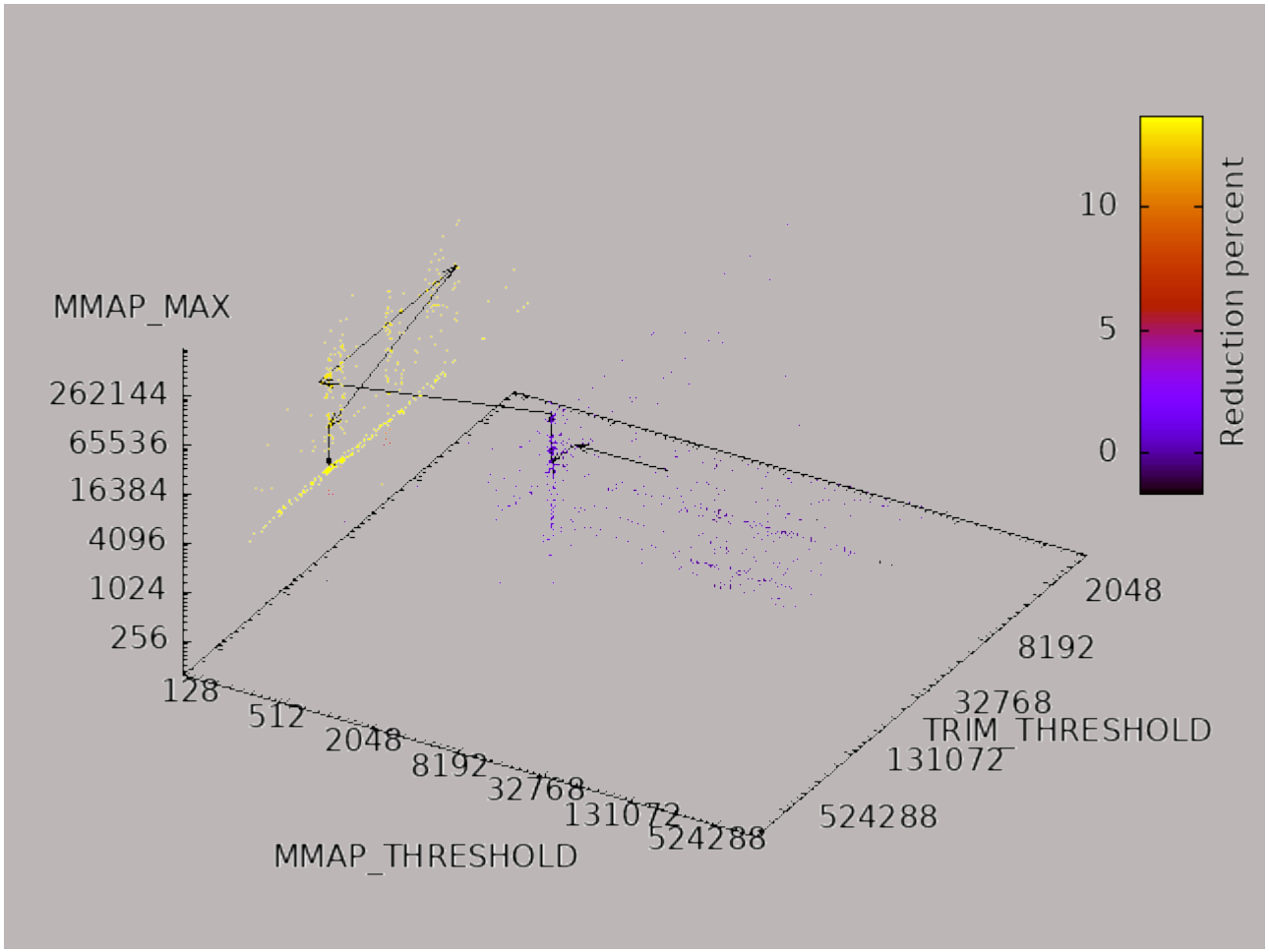



```
M_MMAP_MAX_tune          g[0,33554432,1/65536][65536]
M_TRIM_THRESHOLD_tune    g[0,33554432,1/131072][131072]
M_MMAP_THRESHOLD_tune    g[0,33554432,1/131072][131072]
```

Corrected magpie parameters

# gem5 Magpie search

Magpie starts 131072,131072,65536 0%. Arrows show new better place. Search is parallel to x,y,z axis. Best mmap=212



# gem5 Genetic Improvement with CMA-ES

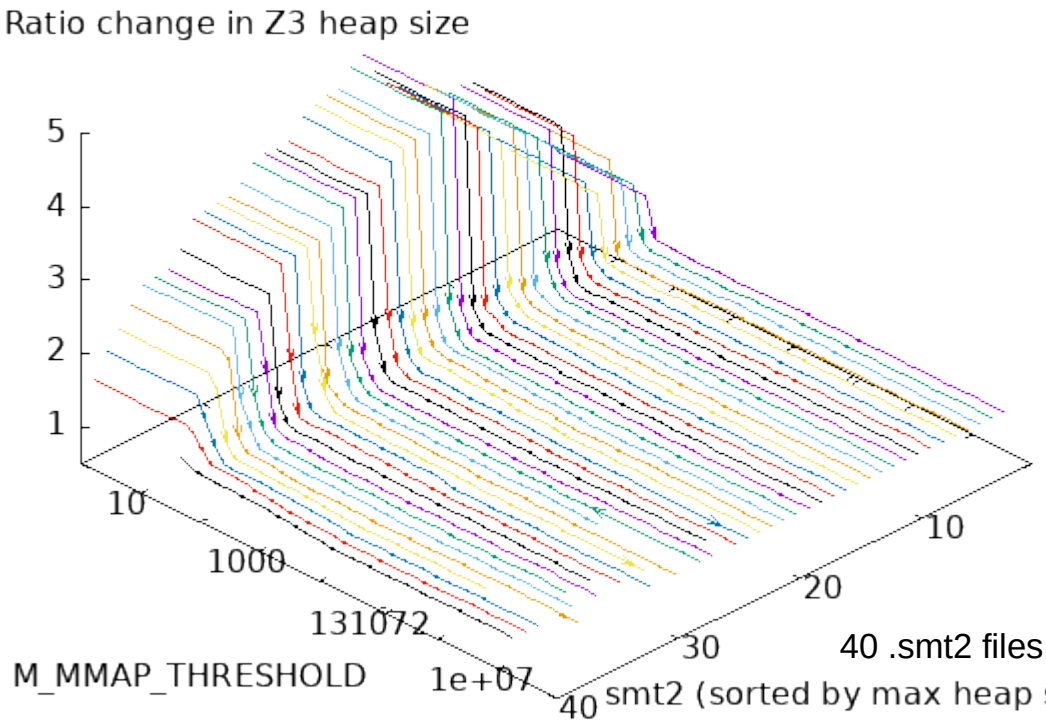
- CMA-ES is the state of the art search for continuous problems (you can use your favourite optimiser).
- Gaussian mutations with distribution's mean and standard deviation σ adapted during search
- Fitness function as Magpie
- x,y,z double values round to int. Negative values converted to 0
- Both Magpie and CMA-ES find improvements > 10%

# Other C++ Examples

- Use Valgrind's Massif (no code changes)
- Z3 Microsoft SMT theorem prover
  - 600,000 lines of code
  - Bench mark Certora Prover example from SMT competition
- GNU g++ compiler
  - Compile largest MySQL source code
- Clang LLVM compiler
  - Fitness function as g++
- Redis key-value store (make MALLOC=libc)
  - 150,000 lines of code
  - Use redis-benchmark
- Heap landscape again smooth but little gain
- Is small improvement because they do not stress heap?

# Microsoft Z3 on Certora Prover benchmarks

- Z3 Massif peak heap
- Smooth fitness landscape (plot one dimension mmap)
- Best 1.5% improvement



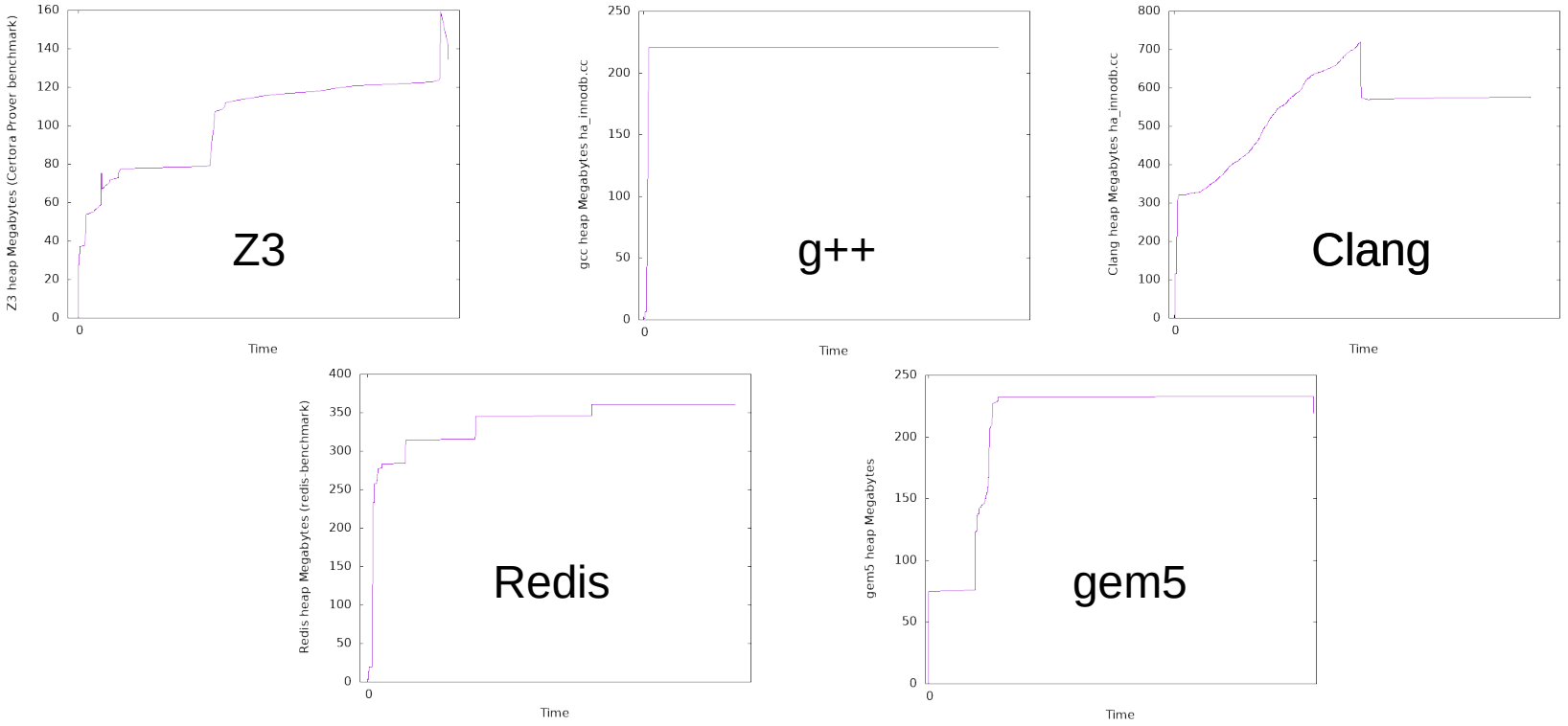

40 .smt2 files from SMT-COMP 2024

# Is gem5 typical?

- After an initial start up phase, gem5 created and deletes C++ objects of various sizes in an apparently random order during the simulation. Little time is spent writing results and shutting down.
- Seems typical? But in 4 cases tuning gives little benefit *Guess*:
- Z3, g++, Clang, process input file then stop.
  - *Whilst processing input file, heap is only added to*
  - *When done heap is discarded.*
  - *Ie simple use of heap? No problem with heap fragmentation?*
  - *Heap manager has little to do?*
- redis-benchmark *all key-values are same size, so no problem with heap fragmentation?*

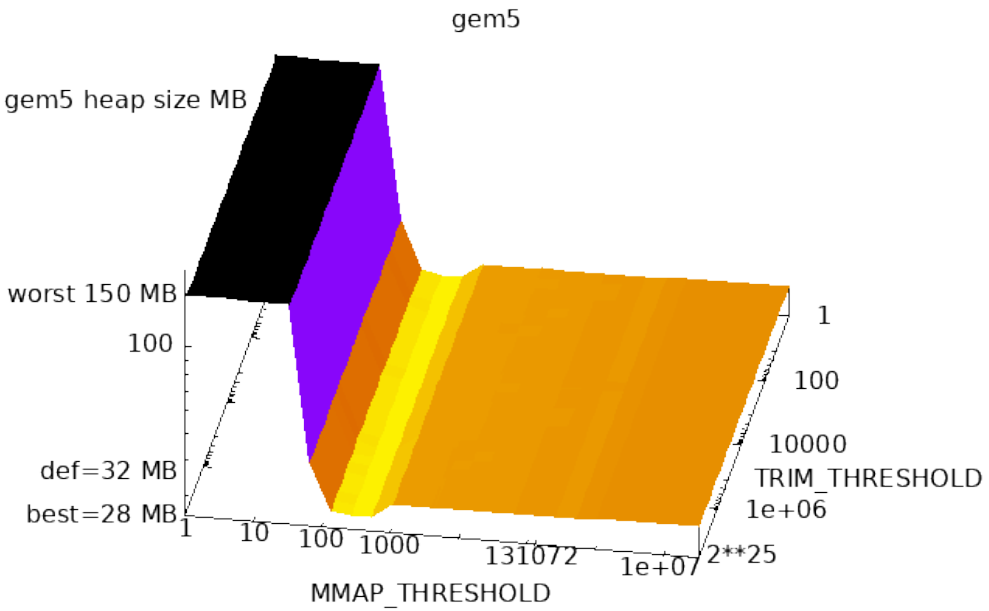
# Z3, g++, Clang, Redis, why flat landscapes

- All fitness landscapes (1 dimension mmap) smooth
- But only Z3 makes small improvement, others almost flat
- Heap use different from gem5
  - gem5 continuous random re-use.
  - Z3, compilers, redis-benchmark only add to heap?



# Conclusions

- Not all SBSE search problems are hard.
- Genetic Improvement can be applied to software parameters as well as code. Eg glibc 37 run time parameters. Use any optimiser
- gem5 is a million+ lines but C++ 7 dimensional new/delete landscape is smooth, collapses to essentially one dimension broad good fitness valley (4 $10^{17}$ solutions) large basin of attraction. Gives 11% heap reduction without loss of speed
- Other non-trivial C++ programs have similar smooth landscapes but tuning GNU glibc malloc gives only marginal improvement
- Magpie can tune parameters as well as multi-language code

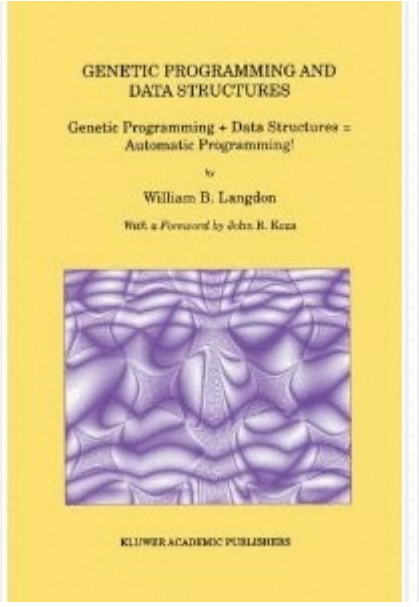


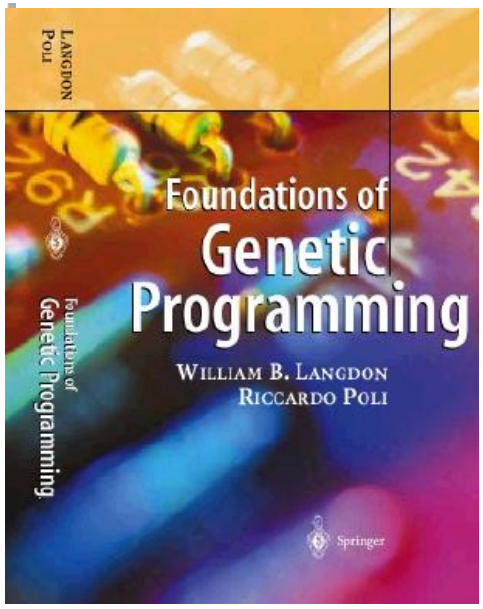


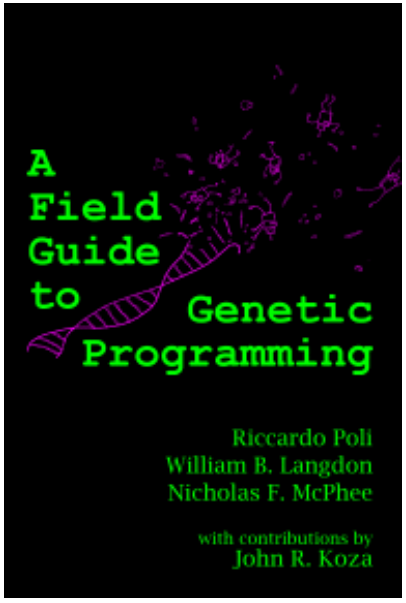

4e17 = 400 Quadrillion

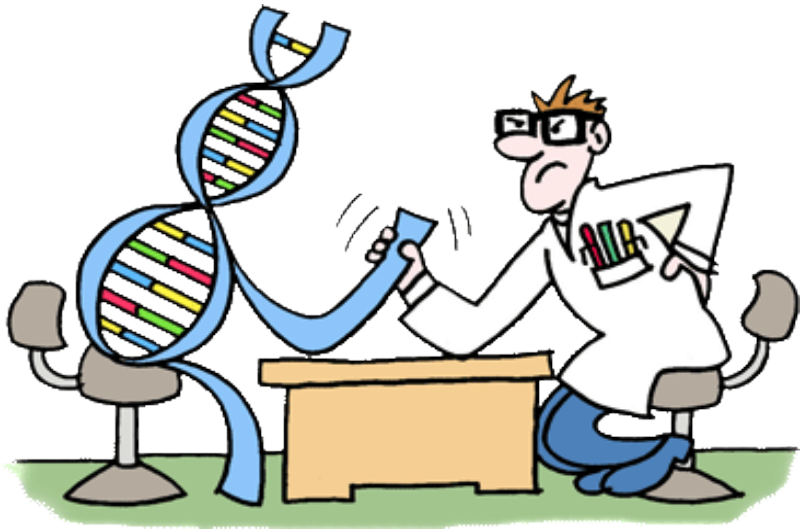
# Genetic Programming

## W. B. Langdon





Human-Competitive results $10,000 prizes
https://www.human-competitive.org

Email your entry to goodman@msu.edu
by **Friday 30 May**

# The Genetic Programming Bibliography

http://gpbib.cs.ucl.ac.uk/

**17586** references, 17000 authors

**Make sure it has all of your papers!**

E.g. email W.Langdon@cs.ucl.ac.uk or use | Add to It | web link

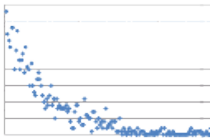

Co-authorship community.
Downloads

A personalised list of every author's GP publications.

blog

Googling GP bibliography, eg:
Development and learning site:gpbib.cs.ucl.ac.uk

Downloads by day



Your papers


Text search