Applying Genetic Improvement to MiniSAT

Justyna Petke

Centre for Research on Evolution, Search and Testing

University College London

Summary (from SSBSE'13 talk)

Genetic Improvement Programming:

Automatically improves system behaviour

According to some desired criteria using GP

(Relies on a set of test cases)

Summary (from SSBSE'13 talk)

Genetic Improvement Programming:

Automatically improves system behaviour

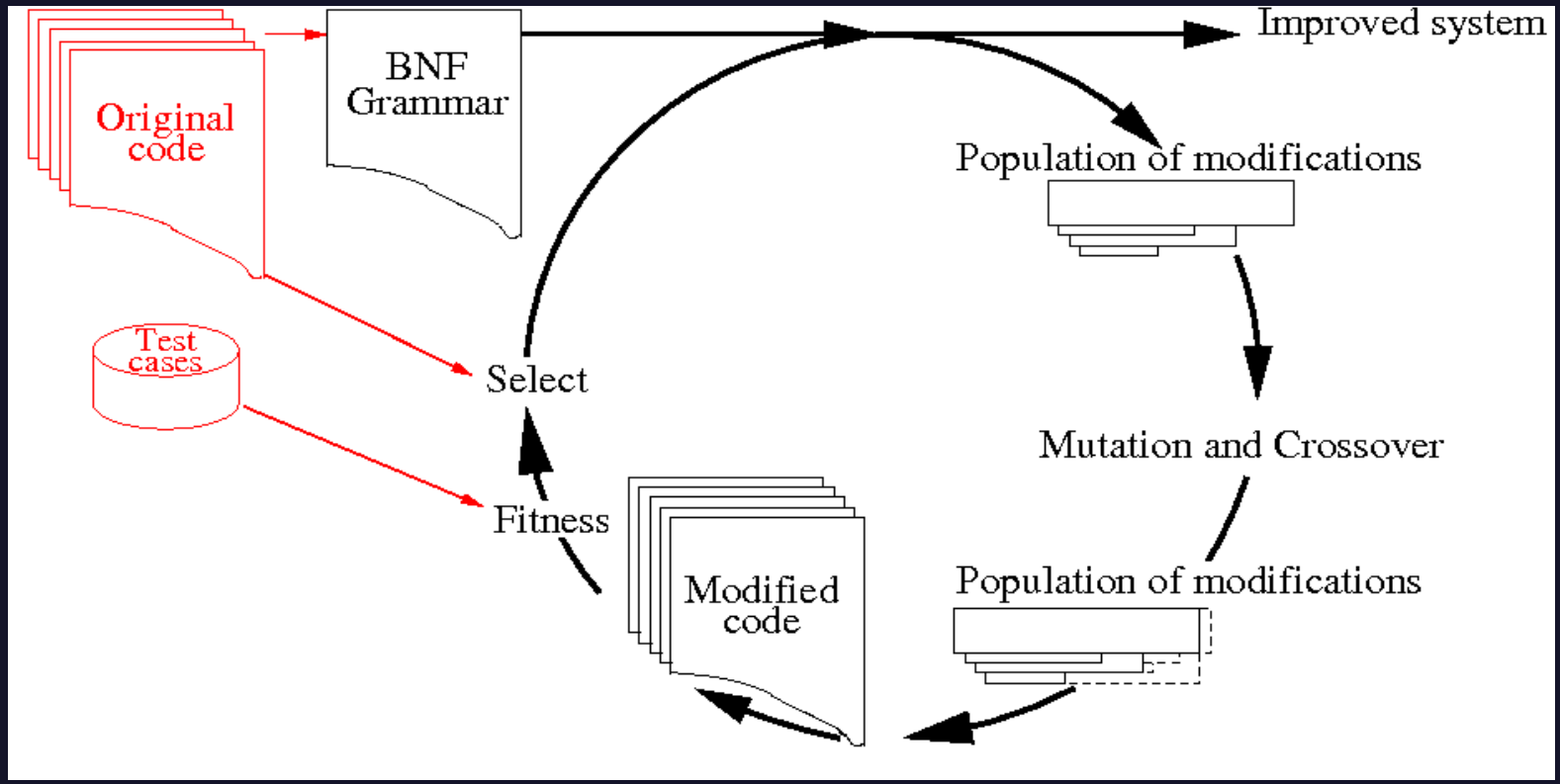
According to some desired criteria using GP

(Relies on a set of test cases)

Bowtie2 : big runtime improvement ("Optimising Existing Software with Genetic Programming", W.B.Langdon & M.Harman, IEEE Transact. on Evol. Comp.)

MiniSAT : up to $\sim$2.5% runtime improvement (SSBSE'13)

GP Improvement



Motivation for choosing a SAT solver

Bounded Model Checking

Planning

Software Verification

Automatic Test Pattern Generation

Combinational Equivalence Checking

Combinatorial Interaction Testing

and many other applications..

Motivation for choosing a SAT solver

Benchmarks available from SAT solver competitions

Motivation for choosing a SAT solver

Benchmarks available from SAT solver competitions

MiniSAT hack track in SAT solver competitions

Research directions (from SSBSE'13 talk)

Specialise test sets for GP

Change population and generation size

Allow to inject lines of code from other SAT solvers

Research directions (from SSBSE'13 talk)

Specialise test sets for GP

Change population and generation size

Allow to inject lines of code from other SAT solvers

Trying an older version of MiniSAT

Comparing random vs. fitness-based selection strategy

Allowing changes within expressions by using GenProg

(http://dijkstra.cs.virginia.edu/genprog/)

Experiments: Setup

Solvers used:

MiniSAT1.14 and MiniSAT2-070721

Experiments: Setup

Solvers used:

MiniSAT1.14 and MiniSAT2-070721

Test cases used:

from SAT competitions (application track)

from Combinatorial Interaction Testing field (generated)

Experiments: Setup

Solvers used:

MiniSAT1.14 and MiniSAT2-070721

Test cases used:

from SAT competitions (application track)

from Combinatorial Interaction Testing field (generated)

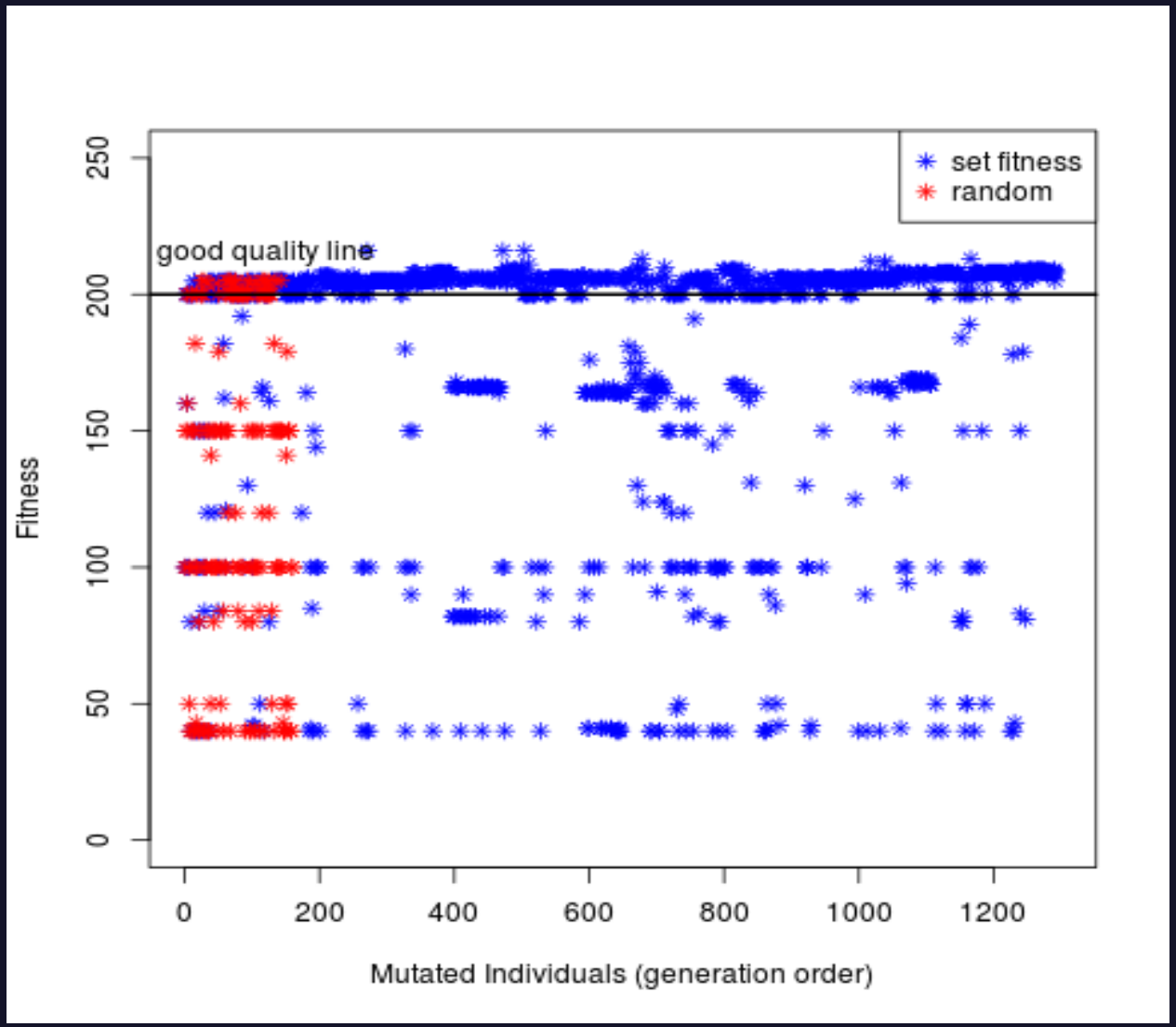
Scoring:

200 points for returning the correct answers

0-50 points for runtime improvement

MiniSAT1.14: Fitness Strength

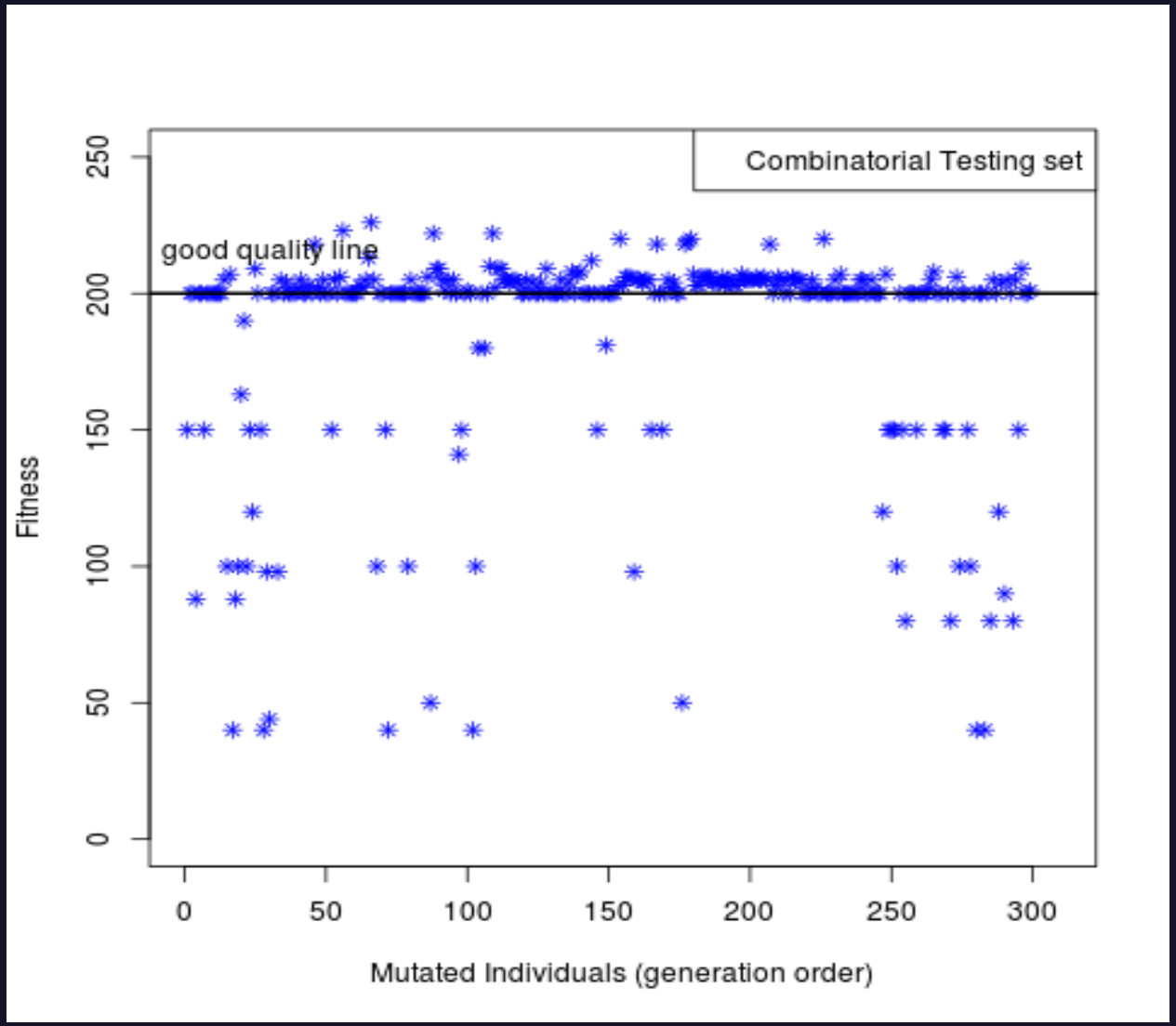
MiniSAT1.14: Fitness Strength



population size: 100, generations: 20, Application benchmarks

MiniSAT1.14: Specialised Test Set

MiniSAT1.14: Specialised Test Set



population size: 100, generations: 10, Combinatorial Testing benchmarks

MiniSAT1.14: Specialised Test Set

Test Cases from the Application track of SAT competitions:

65% produced non-zero fitness (47% of good quality)

highest fitness: 216

MiniSAT1.14: Specialised Test Set

Test Cases from the Application track of SAT competitions:

65% produced non-zero fitness (47% of good quality)

highest fitness: 216

Test Cases from Combinatiorial Testing application:

30% produced non-zero fitness (24% of good quality)

highest fitness: 226

MiniSAT1.14: Specialised Test Set

Test Cases from the Application track of SAT competitions:

improved on around 60% tests

MiniSAT1.14: Specialised Test Set

Test Cases from the Application track of SAT competitions:

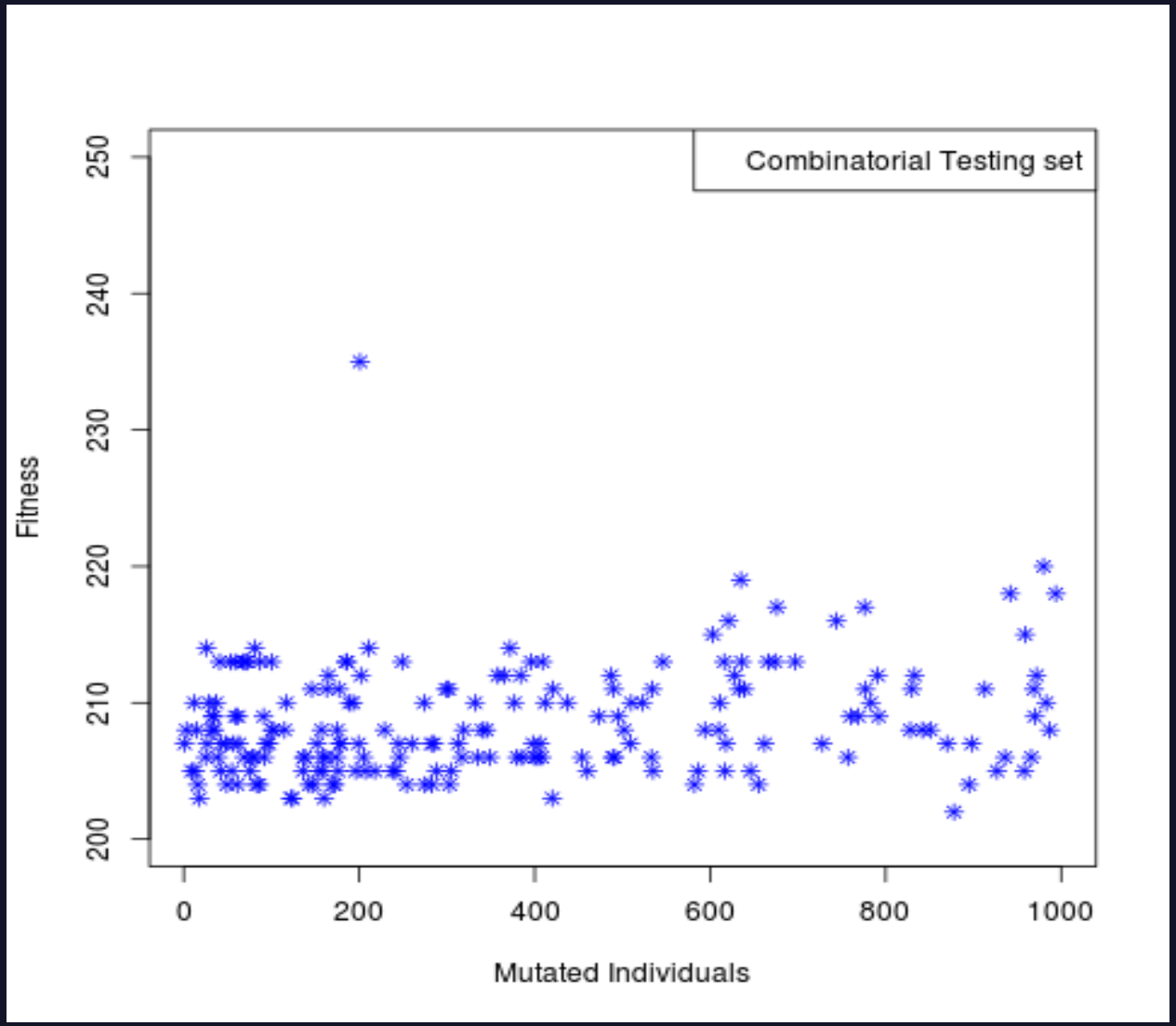
improved on around 60% tests

Test Cases from Combinatiorial Testing application:

improved on all test cases

MiniSAT1.14: Modifying Expressions (GenProg)

MiniSAT1.14: Modifying Expressions (GenProg)



population size: 100, generations: 10, Combinatorial Testing benchmarks

MiniSAT1.14: Modifying Expressions

Test Cases from Combinatiorial Testing application:

Original framework:

30% produced non-zero fitness (24% of good quality)

highest fitness: 226

changes: $<$ *IF_Solver*_632 $><$ *IF_Solver*_985 $>$

MiniSAT1.14: Modifying Expressions

Test Cases from Combinatiorial Testing application:

Original framework:

30% produced non-zero fitness (24% of good quality)

highest fitness: 226

changes: $<$ *IF_Solver*_632 $><$ *IF_Solver*_985 $>$

GenProg:

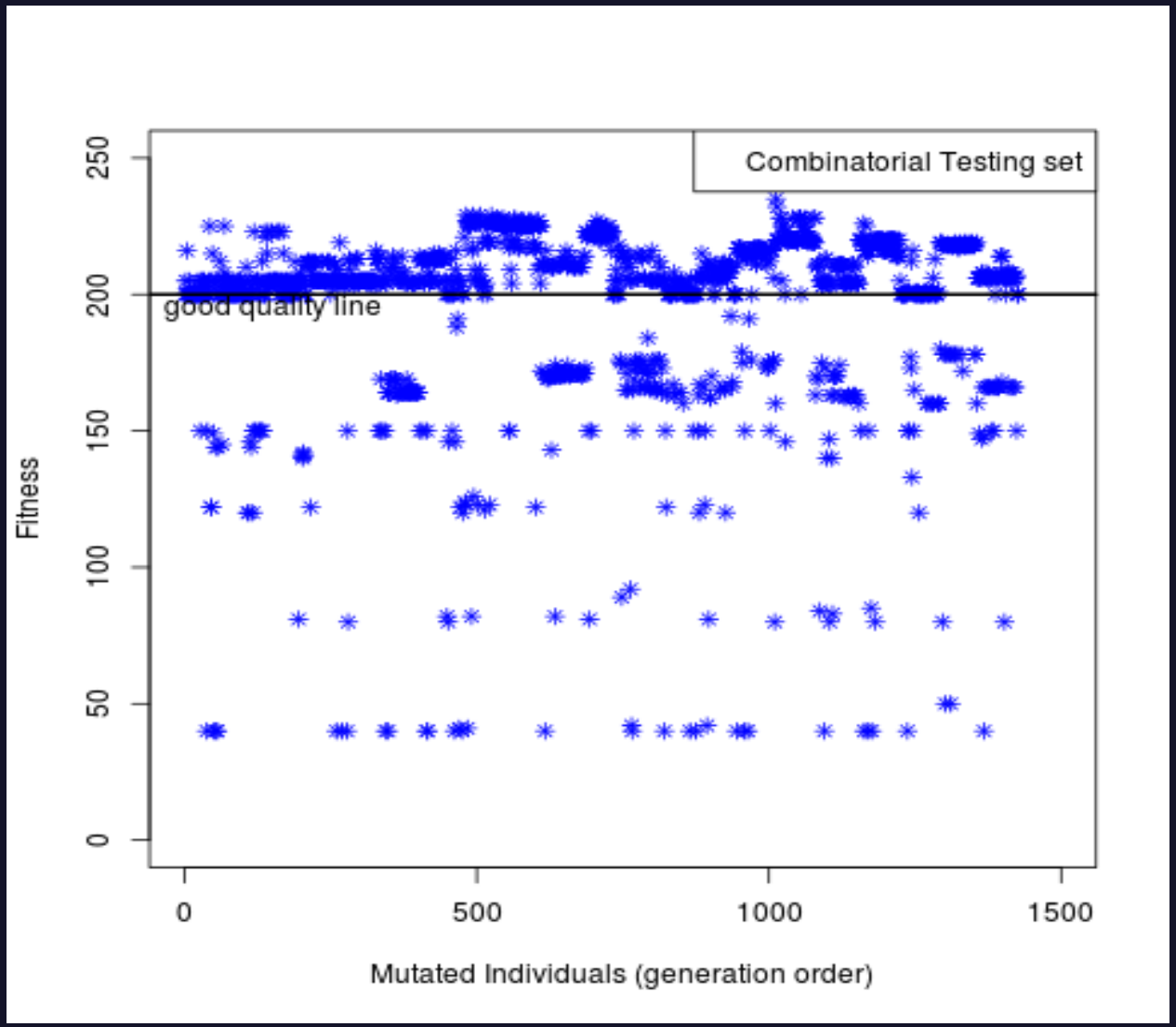
20% produced non-zero fitness (20% of good quality)

highest fitness: 235

changes: s(992,1031) d(1082)

MiniSAT2-070721: Adding New Code

MiniSAT2-070721: Adding New Code



population size: 100, generations: 20, Combinatorial Testing benchmarks

MiniSAT2-070721: Adding New Code

Test Cases from Combinatiorial Testing application:

71% produced non-zero fitness (55% of good quality)

highest fitness: 235

MiniSAT2-070721: Adding New Code

Test Cases from Combinatiorial Testing application:

71% produced non-zero fitness (55% of good quality)

highest fitness: 235

New code used in generations 5, 8-20

None of the new code used in the ‘best’ individuals

MiniSAT2-070721: Adding New Code

Best individuals found in early generations

MiniSAT2-070721: Adding New Code

Best individuals found in early generations

Combining all ‘best ’ individuals gives:

Improvement of 4.9% in terms of CPU time

Improvement of 7.9% in terms of lines of code executed

40 changes in total

MiniSAT2-070721: Adding New Code

Best individuals found in early generations

Combining all ‘best ’ individuals gives:

Improvement of 4.9% in terms of CPU time

Improvement of 7.9% in terms of lines of code executed

40 changes in total

Better than a winning solver from MiniSAT hack track 2009

(which was worse than MiniSAT2-070721 on Combinatorial Testing set)

Summary

Applied genetic improvement to a SAT solver

By specialising the training set:

Achieved 4.9% improvement in terms of CPU time

Achieved 7.9% improvement in tems of lines of code executed

Summary

Fitness vs. Random:

Summary

Fitness vs. Random: fitness provides good guidance

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised:

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised: specialised

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised: specialised

Lines of code vs. plus expressions:

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised: specialised

Lines of code vs. plus expressions: inconclusive

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised: specialised

Lines of code vs. plus expressions: inconclusive

Original code vs. additional code:

Summary

Fitness vs. Random: fitness provides good guidance

Non-specialised vs. Specialised: specialised

Lines of code vs. plus expressions: inconclusive

Original code vs. additional code: inconclusive

Research Directions

Research Directions

Add separate file with new code to be added

Research Directions

Add separate file with new code to be added

Specify part of solver for GP

Research Directions

Add separate file with new code to be added

Specify part of solver for GP

Try to re-generate some functionality of MiniSAT

Research Directions

Add separate file with new code to be added

Specify part of solver for GP

Try to re-generate some functionality of MiniSAT

Modify number of generations and population size

Research Directions

Add separate file with new code to be added

Specify part of solver for GP

Try to re-generate some functionality of MiniSAT

Modify number of generations and population size

Modify fitness