Pushing the Boundaries in Regression Testing

Shin Yoo & Mark Harman / King’s College London
Shmuel Ur / IBM Haifa
Paolo Tonella & Angelo Susi / FBK
Regression Testing

Large Test Suites  Regression Testing  Limited Resource
Selection

Large Test Suites

Limited Resource

Uses impact analysis to precisely identify the changed parts of the program and only test those parts

Often not the answer:
- requires static analysis
- not enough reduction
Minimisation

Seeks to reduce the size of test suites while satisfying test adequacy goals.
Prioritisation

Seeks to achieve test adequacy as much and as early as possible

Limited Resource

A-B-C-D-E
C-E-B-A-D
Potential Impact

Summary of 157 papers on regression testing techniques from a recent survey

- Purely Academic: 80%
- Author from Industry: 20%
- Toy Programs: 92%
- Industrial Scale Subjects: 8%
Research Output

- Pareto Efficient Multi-Objective Test Case Selection: S. Yoo & M. Harman, ISSTA 2007
- Mesuring and Improving Latency to Avoid Test Suite Wear-Out: S. Yoo, M. Harman & S. Ur, SBST 2009
- Clustering Test Cases to Achieve Effective and Scalable Prioritisation Incorporating Expert Knowledge: S. Yoo, M. Harman, P. Tonella & A. Susi, ISSTA 2009
Multi-Objectiveness: Problems

• “After performing minimisation, the test suite is still too big. What can I actually do in the next 6 hours?”

• “I care not just code coverage, but something else too. Can I also achieve X with the minimised suite?”
Single Objective

Choose test case with highest block per time ratio as the next one

1) T1 (ratio = 2.0)
2) T2 (ratio = 2 / 5 = 0.4)

∴ {T1, T2} (takes 9 hours)

“But what if you have only 7 hours...?”

Multi Objective
Latency : Problems

• “My regression testing seems to depend on the same test cases all the time; is this okay?”

• “Code coverage is necessary but not sufficient test adequacy; how do we make it safer?”
The Coverage Trap

T5 may detect an unknown fault. If the minimisation technique never picks T5, fault detection capability is compromised.

We should consider multiple subsets. If necessary, we should improve the remaining part of the test suite.
Designing Test Suites

![Graph showing average maximum statement coverage (%) over times that reduction technique has been applied. The graph includes lines for flex, grep, sed, space, gzip, and printtokens, with different markers for each tool.](image)
The image shows a graph titled "ComplexBranch" which plots the average maximum branch coverage (%) against the times that the reduction technique has been applied. The graph includes data points for different methods: Hill Climbing, EDA, Random, and Original. The x-axis represents the times the reduction technique has been applied, while the y-axis shows the average maximum branch coverage percentage.
Expertise : Problems

• “I have seen the results of automated test case prioritisation and I don’t agree. This is the way it should be!”

• “We need to prioritise a specific set of tests due to business priority.”
Interleaved Clusters
Prioritisation

Cluster
Intra-cluster Prioritisation
Inter-cluster Prioritisation
Interleaving Clusters
Experimental Setup

- Simple Agglomerative Hierarchical Clustering (k=14)
- Hamming distance between stmt. coverage as dissimilarity metric
- A human user model with controlled error rate
Tolerance

This is what we initially expected to see.

But...
Tolerance

Some test suites are very resilient to errors
Boundaries & Open Questions

• Scalability: not only quantitative but also qualitative scalability

• Complexity: set-up cost, oracle cost, dependency

• Effectiveness: are the traditional metrics good enough?
Summaries

• Regression testing is hard - not a single solution
• Multi-objective paradigm allows us to formulate a complex problem
• Code coverage needs to be re-thought
• Humans are a vast pool of knowledge