Prioritizing Test Cases for Resource Constraint Environments Using Historical Test Case Performance Data

Computer Science and Information Technology, 2009. ICCSIT

Y. Fazlalizadeh, A. Khalilian, M. Abdollahi Azgomi and S. Parsa

Speacker: Dong-Ah Lee

Contents

- Introduction
- Background
- The Proposed Approach
- Empirical Studies

Introduction

- Changing the software
  - the SW may regress
- Regression Testing (RT) ?
  - the re-running of all test cases
  - be costly or even infeasible
- To reduce the cost of RT
  - regression test selection (RTS)
  - test cases prioritization

Introduction (cont'd)

- Most prioritization techniques
  - memoryless
  - based only on the analysis of source code and test case
  - ignoring historical test case performance data
  - taking a one-time testing model
- History-based regression testing
  - considers only effect of last execution of test cases to calculate the selection probability of test cases
- A new equation
  - considers the environment time and resource constraints
  - historical effectiveness in fault detection, each test case's execution history, the last priority

Background

- Test Case Prioritization

The Test Case Prioritization Problem. Given: *T, a test suite; PT, the set of permutations of T; f, a function from PT to the real numbers.*

Problem: *Find* $T' \in PT$ *such that*

$$(\forall\, T''\ (T'' \in PT)\ (T'' \neq T')\ [f(T') \geq (T'')].$$

Search Algorithms for Regression Test Case Prioritization, TSE(2007)
Zheng Li at al,

  - NP-Hard basically
  - heuristic or sub-optimal solutions: **code-based** (coverage, fault-exposing potential prioritization), model-based prioritization, history-based test prioritization etc.

Background (cont'd)

- History-Based Test Case Prioritization
  - $P_{tc,t}(H_{tc},\alpha)$: selection probability of each test case based upon execution history
    - H_{tc}: a set of t, time-ordered $\{h_1, h_2, \ldots, h_t\}$ observations drawn from previous runs of TC

$$P_0 = h_1$$

$$P_k = \alpha h_k + (1 - \alpha)P_{k-1},\ 0 \le \alpha \le 1,\ k \ge 1$$

"A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Environments," ICSE (2002)
Jung-Min Kim et al.

  - Ways to define H_{tc}
    - Execution history
    - Demonstrated fault detection effectiveness
    - Coverage of program entities

The proposed Approach

- Priority factor 1
  - Number of regression test sessions the test case executes: fc_k
  - Number of test sessions it reveals fault(s): ec_k

$$PR_k \approx \frac{fc_k}{ec_k}$$

$$fc_k = \sum_{i=1}^{k-1} f_i \quad f_i = \begin{cases} 1 \; if \; tc \; has \; revealed \; fault \; in \; test \; session \; i \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

$$ec_k = \sum_{i=1}^{k-1} e_i \quad e_i = \begin{cases} 1 \; if \; tc \; has \; been \; executed \; in \; test \; session \; i \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

Historical Fault Detection Effectiveness

The proposed Approach

- Priority factor 2
  - A period of time that a test case is not being executed

$$h_0 = 0$$

$$h_k = \begin{cases} h_{k-1} + 1 & \text{if } tc \text{ has been executed in test session } i \\ 0 & \text{otherwise} \end{cases}$$

The proposed Approach

- Priority factor 3
  - the recent priority of each test case in each test suit during past executions

$$PR_{k-1}$$

The proposed Approach

- Equation of each test case priority in the k_{th} execution

$$PR_k = \alpha \frac{fc_k}{ec_k} + \beta PR_k + \gamma h_k \qquad 0 \leq \alpha, \beta, \gamma < 1, k \geq 1 \quad (8)$$

  - PR_0: the percentage of code coverage of the test case

Empirical Studies

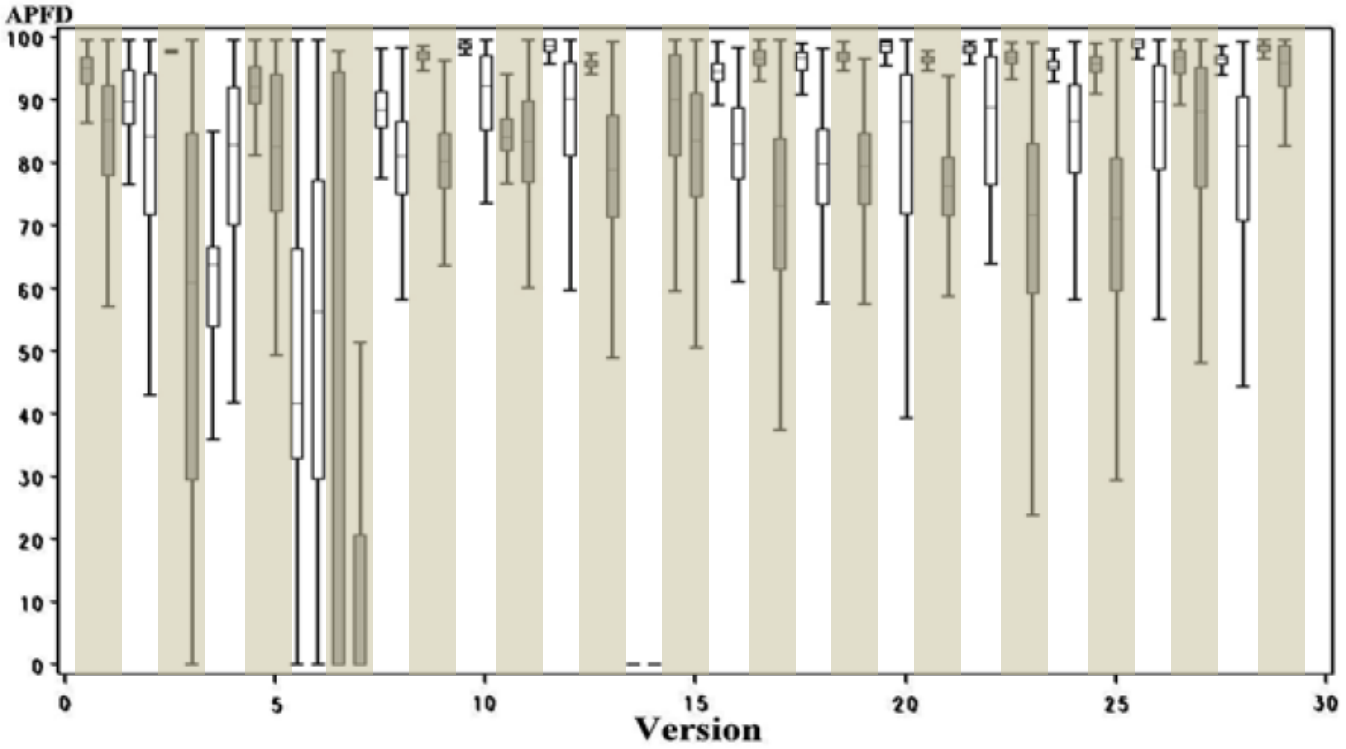
- Proposed approach *vs* random ordering approach
  - 7 programs from Siemens suite, one program from Space benchmark program
  - 1000 test suits

- Evaluation Metric: Average Percentage of Fault Detection

$$APFD = 1 - \frac{TF_1 + TF_2 + \ldots + TF_m}{nm} + \frac{1}{2n}$$

  - n: the number of test cases
  - m: the number of existing faults
  - TF_i: the place of a test case in ordered suite which first reveals the fault i

Empirical Studies



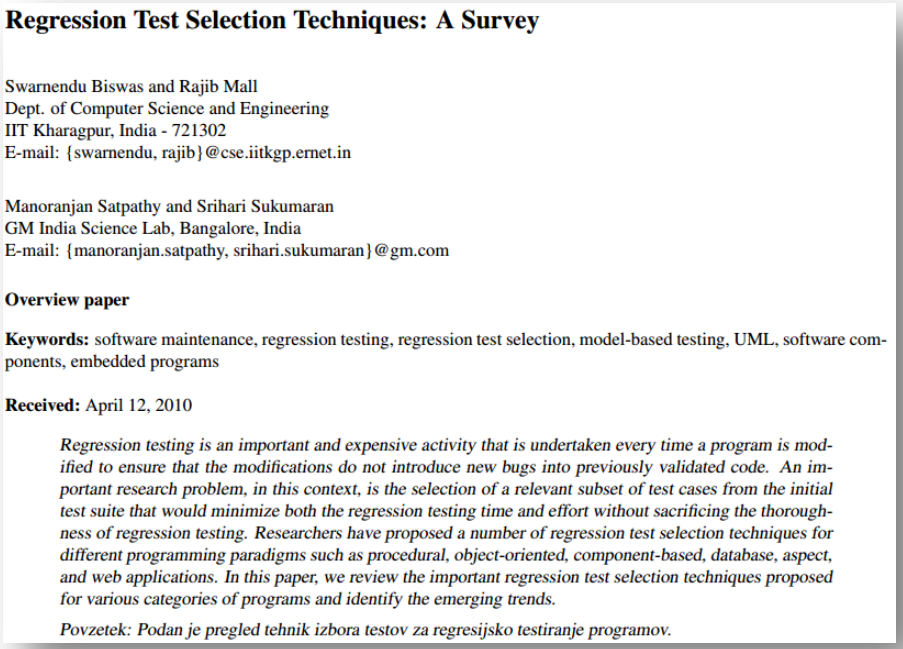
H. space

References

- Research papers
  - Fazlalizadeh, Y.; Khalilian, A.; Azgomi, M.A.; Parsa, S.; , "Prioritizing test cases for resource constraint environments using historical test case performance data," Computer Science and Information Technology, 2009. ICCSIT 2009.
  - Emelie Engström, Per Runeson, Andreas Ljung, " Improving regression testing transparency and efficiency with history-based prioritization – an industrial case study," IEEE Fourth International Conference on Software Testing, Verification and Validation (ICST), 2011
  - Jung-Min Kim et al. "A History-Based Test Prioritization Technique for Regression Testing in Resource Constrained Environments," ICSE (2002)
  - Gregg Rothermel, Roland H. Untch, Chengyun Chu, and Mary Jean Harrold. 1999. Test Case Prioritization: An Empirical Study. In Proceedings of the IEEE International Conference on Software Maintenance (ICSM '99)

- Survey papers
  - Swarnendu Biswas and Rajib Mall, "Regression Test Selection Techniques: A Survey," 2011
  - Yoo, "Regression testing minimization, selection and prioritization: a survey," STVR(2010)

Appendix



Regression Test Selection Techniques: A Survey

Swarnendu Biswas and Rajib Mall
Dept. of Computer Science and Engineering
IIT Kharagpur, India - 721302
E-mail: {swarnendu, rajib}@cse.iitkgp.ernet.in

Manoranjan Satpathy and Srihari Sukumaran
GM India Science Lab, Bangalore, India
E-mail: {manoranjan.satpathy, srihari.sukumaran}@gm.com

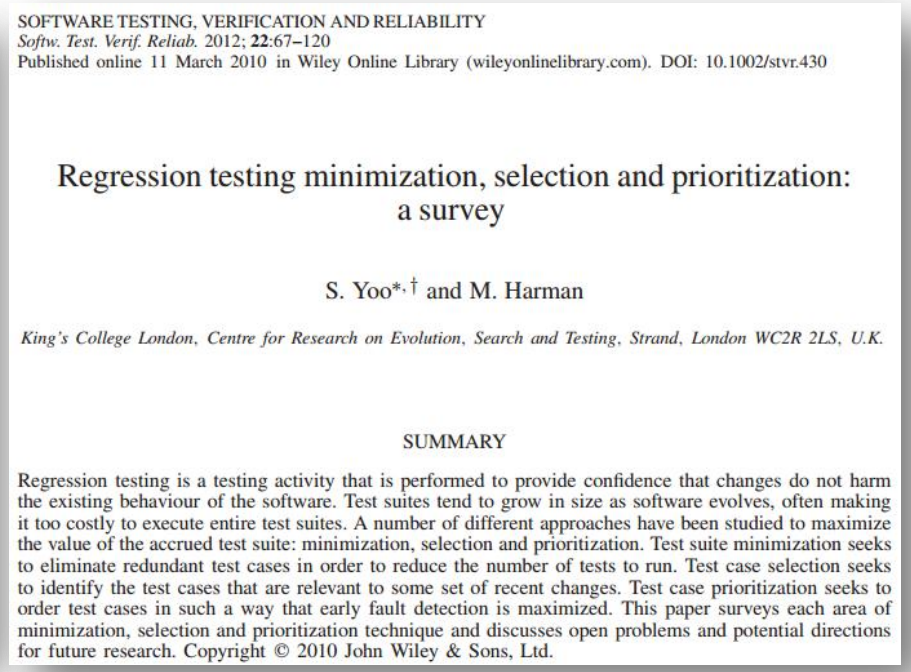
Overview paper

Keywords: software maintenance, regression testing, regression test selection, model-based testing, UML, software components, embedded programs

Received: April 12, 2010

Regression testing is an important and expensive activity that is undertaken every time a program is modified to ensure that the modifications do not introduce new bugs into previously validated code. An important research problem, in this context, is the selection of a relevant subset of test cases from the initial test suite that would minimize both the regression testing time and effort without sacrificing the thoroughness of regression testing. Researchers have proposed a number of regression test selection techniques for different programming paradigms such as procedural, object-oriented, component-based, database, aspect, and web applications. In this paper, we review the important regression test selection techniques proposed for various categories of programs and identify the emerging trends.

Povzetek: Podan je pregled tehnik izbora testov za regresijsko testiranje programov.



SOFTWARE TESTING, VERIFICATION AND RELIABILITY
Softw. Test. Verif. Reliab. 2012; **22**:67–120
Published online 11 March 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/stvr.430

Regression testing minimization, selection and prioritization: a survey

S. Yoo*,† and M. Harman

King's College London, Centre for Research on Evolution, Search and Testing, Strand, London WC2R 2LS, U.K.

SUMMARY

Regression testing is a testing activity that is performed to provide confidence that changes do not harm the existing behaviour of the software. Test suites tend to grow in size as software evolves, often making it too costly to execute entire test suites. A number of different approaches have been studied to maximize the value of the accrued test suite: minimization, selection and prioritization. Test suite minimization seeks to eliminate redundant test cases in order to reduce the number of tests to run. Test case selection seeks to identify the test cases that are relevant to some set of recent changes. Test case prioritization seeks to order test cases in such a way that early fault detection is maximized. This paper surveys each area of minimization, selection and prioritization technique and discusses open problems and potential directions for future research.