# Finding Sufficient Mutation Operators via Variable Reduction

*Akbar Siami Namin and James H. Andrews*
*Computer Science Department*
*The University of Western Ontario*
*London, Ontario, Canada*
*{asiamina, andrews}@csd.uwo.ca*

*Second Workshop on Mutation Analysis (Mutation 2006)*
*in conjunction with the 17th IEEE International Symposium on Software Reliability Engineering (ISSRE 2006)*

*November 2006, Raleigh, North Carolina, USA.*

# Outline

- Introduction
- Related Work
- Experimental Procedure
- Statistical Analysis
- Preliminary Results
- Discussion
- Conclusion and Future Work

# Introduction

- Needs faulty versions for experiments
  - Applying mutation operators to programs
- Many mutation operators have been proposed
  - Enormous number of mutants $\Rightarrow$ Infeasible Computation
- Finding a reduced set which is sufficient
- Our approach
  - Applying variable reduction techniques

# Related Work

## Mutation Testing and Analysis

- Mutation Testing
  - Generating faulty versions: measuring the test suites adequacy (Offutt et al.)
- Mutants behave very similarly to real faults (Andrews et al.)
- Mutant generator tools: Mothra (Fortran), Proteum (C), and MuJava (Java)

# Related Work (cont)

## Sufficient Mutation Operators

- Sufficient mutation operators problem (Offutt et al.)
- Experiments to find the sufficient set for Fortran (Wong)

## Variable Reduction

- Decreasing the number of variables (Jolliffe)

# Experimental Procedure

## Setup

- Subject programs: Siemens programs
  - Seven subject programs
- Mutant generator: Proteum, C mutant generator (Maldonado et al.)
  - 108 mutation operators
- Code coverage tool: ATAC
  - Block, Decision, Computation Use, Predicate Use

# Experimental Procedure (cont)

## Definitions of Sets and Measurements

- Mutation operators: $\mu_i, 1 \leq i \leq 108$
- $NMNE_i$, ♯ of non-equivalent mutants generated using $\mu_i$
- $TNMNE$, total ♯ of mutants that are nonequivalent
- For each test suite $S$:
  - $kills_i(S)$, The set of mutants generated by $\mu_i$ and kille by $S$
  - $kills(S)$, The set of all mutants killed by $S$
  - $Am_i(S) = |kills_i(S)|/NMNE_i$, the adequacy ratio for mutants of operator $\mu_i$
  - $AM(S) = |kills(S)|/TNMNE$, the adequacy ratio for all mutants

# Experimental Procedure (cont)

## Test Suite Groups

- Sufficient operators may depend on test suite construction
- Test suite construction methods studied:
  - $SING$, group of singleton test suites
  - $RAND_i, i = \{10, 20, 50, 100\}$, groups of randomly-selected test suites with size $i$
  - $BLK, DEC, CUSE, PUSE$, coverage-based test suites

# Statistical Analysis

- All-Subset Regression (SUB)
- Elimination-Based Correlation Technique (EBC)
- Cluster Analysis (CA)

# Statistical Analysis

## All Subset Regression (SUB)

- Select a subset from a larger number of variables to predict another variable
- Finding subsets which fit well
- Number of variables in the precidtion formula can be identified
- Constructing all regression models (Exhustive)

# Statistical Analysis

## Elimination-Based Correlation Technique (EBC)

- Based on the correlation coefficients between variables
- constructing correlation matrix
- Eliminating one which has higher correlation with others
- Hgih correlation: $cor(X, Y) \geqslant 0.90$

# Statistical Analysis

## Cluster Analysis (CA)

- Based on similarities of two variables
- Constructing proximity matrix
- Distances of variables in multi-dimensional space (Nearest Neighbor)
- Producing dendrogram

# Preliminary Results

- Applied the techniques to tcas.c program so far.
- 300 test suites for each group
- $49$ operators: no generated mutants
- Applied only to $SING, RAND_i$
- Applied each technique to each group of test suites
- Merged data for groups $\Rightarrow$ Core set of operators from each technique

# Preliminary Results (cont)

Core sets - No operators common to all

| core-SUB | Mutants |
|---|---|
| I-DirVarAriNeg | 44 |
| I-DirVarRepReq | 220 |
| I-IndVarLogNeg | 19 |
| I-IndVarRepReq | 91 |
| u-OABN | 3 |
| u-OLSN | 34 |
| u-VDTR | 111 |
| u-VGSR | 794 |
| u-VTWD | 74 |
| II-ArgLogNeg | 3 |
| Total | 1393 |

| core-EBC | Mutants |
|---|---|
| I-IndVarAriNeg | 19 |
| I-IndVarLogNeg | 19 |
| II-ArgRepReq | 5 |
| u-OALN | 2 |
| u-OASN | 2 |
| u-OCNG | 8 |
| u-OLLN | 17 |
| Total | 72 |

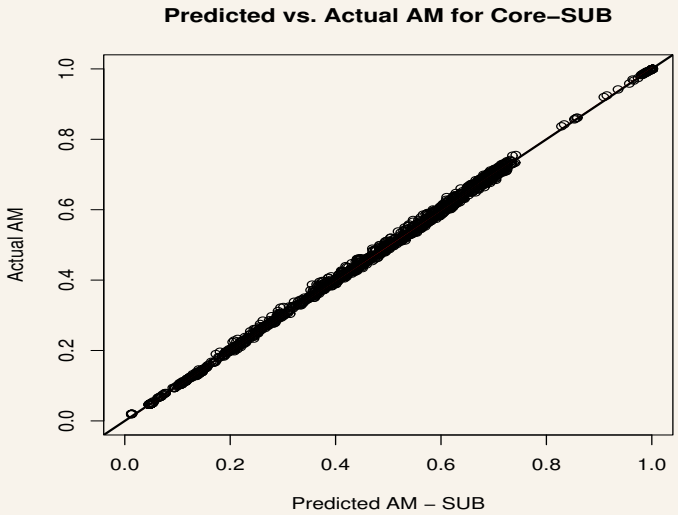
| core-CA | Mutants |
|---|---|
| I-CovAllEdg | 12 |
| I-DirVarRepCon | 38 |
| I-IndVarAriNeg | 19 |
| I-RetStaDel | 17 |
| II-ArgIncDec | 54 |
| u-OABN | 3 |
| u-OALN | 2 |
| u-OASN | 2 |
| u-OCNG | 8 |
| u-OLNG | 51 |
| u-ORSN | 30 |
| u-SRSR | 60 |
| u-STRP | 70 |
| Total | 366 |

# Preliminary Results (cont)

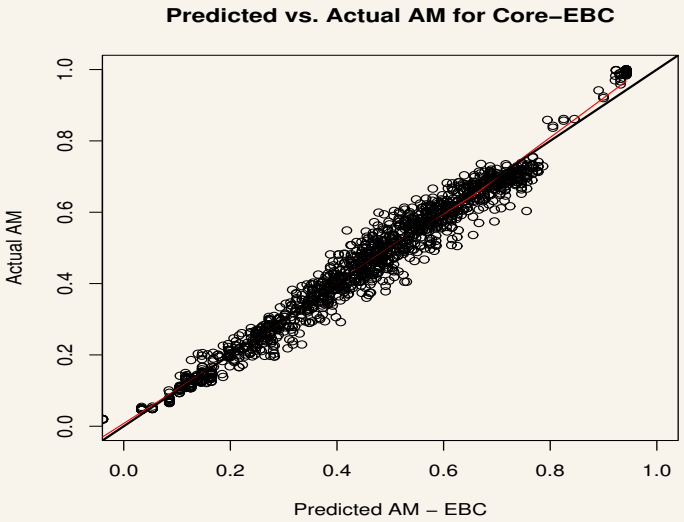
## Comparison of core sets

| | **Core-SUB** | **Core-EBC** | **Core-CA** |
|---|---|---|---|
| Mutants | 1393 | 72 | 366 |
| %Total | 28.22% | 1.45% | 7.41% |
| %Saving | 71.78% | 98.55% | 92.59% |
| Operators | 10 | 7 | 13 |
| %Total | 9.25% | 6.48% | 12.03% |
| %Saving | 90.75% | 93.52% | 87.97% |

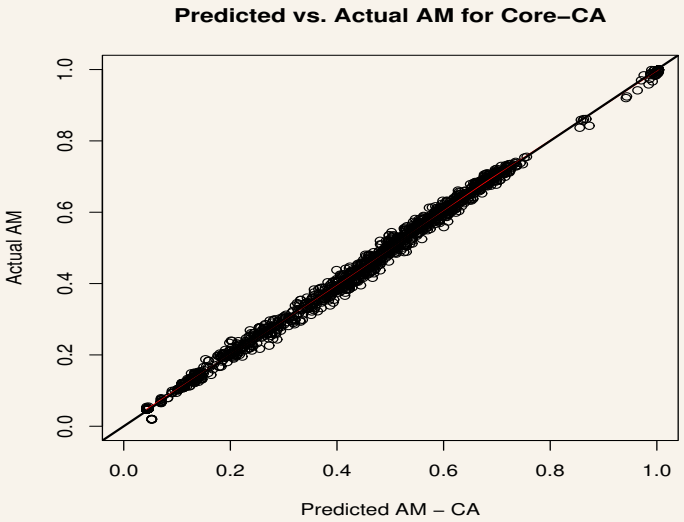
# Preliminary Results (cont)



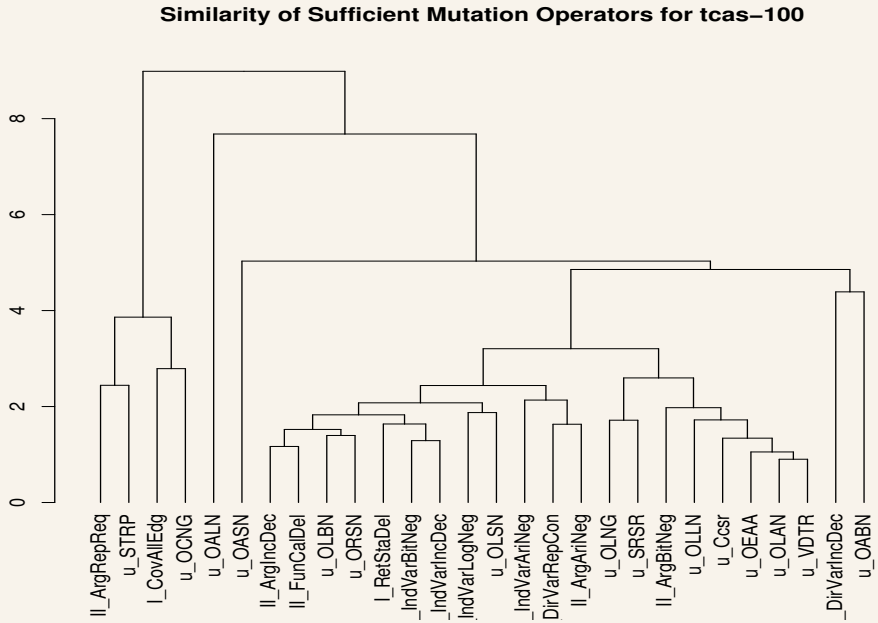
# Preliminary Results (cont)



# Preliminary Results (cont)



# Preliminary Results (cont)



# Discussion

## Threats to Validity

- Analysis for tcas.c so far
- The correctness of the mutant generation and scripting processes
  - Rely on Proteum and our shell scripts
- Use of C programs

## Data Combination and Analysis

- The order of activities:
  1. Combining Data from subject program
  2. Statistical Analysis
  3. Combine data from test suite groups
  4. Fit linear models

# Conclusion and Future Work

## Conclusion

- Applying variable reduction techniques to address sufficient mutation operators problem
  - Subset selection analysis considers AM as target
  - Correlation and cluster analyses try to identify the most distinct set of operators
- Our experiment reveals valuable information

## Future Work

- Extend to all seven Siemens programs
- Study of possibilities of finding non-linear relationships among mutation operators
- Identifying one or several sets of mutation operators as sufficient

# Thanks

*Questions*
*and*
*Answers*