



# 基于程序频谱的动态缺陷 定位方法研究

汇报人：王昭丹



# CONTENTS



01. 背景知识



02. SFL研究框架



03. 框架内影响因素分析



04. 已有实证研究分析

# 背景知识

## 静态缺陷定位方法

- 不需要执行测试用例
- 采用代码审查方法
- 分析被测程序的内在结构

## 动态缺陷定位方法

- 分析被测程序的内在结构
- 需要执行测试用例
- 基于特定模型

## 基于程序频谱的动态缺陷定位

spectrum based dynamic fault localization, SFL

# 4个基本假设

假设1

缺陷具有偶然正确属性

假设2

每个失败测试用例至少会覆盖到 1 个缺陷语句

假设3

无法事先预知到被测程序内缺陷分布的先验概率

假设4

完美缺陷检测假设

# SFL研究框架

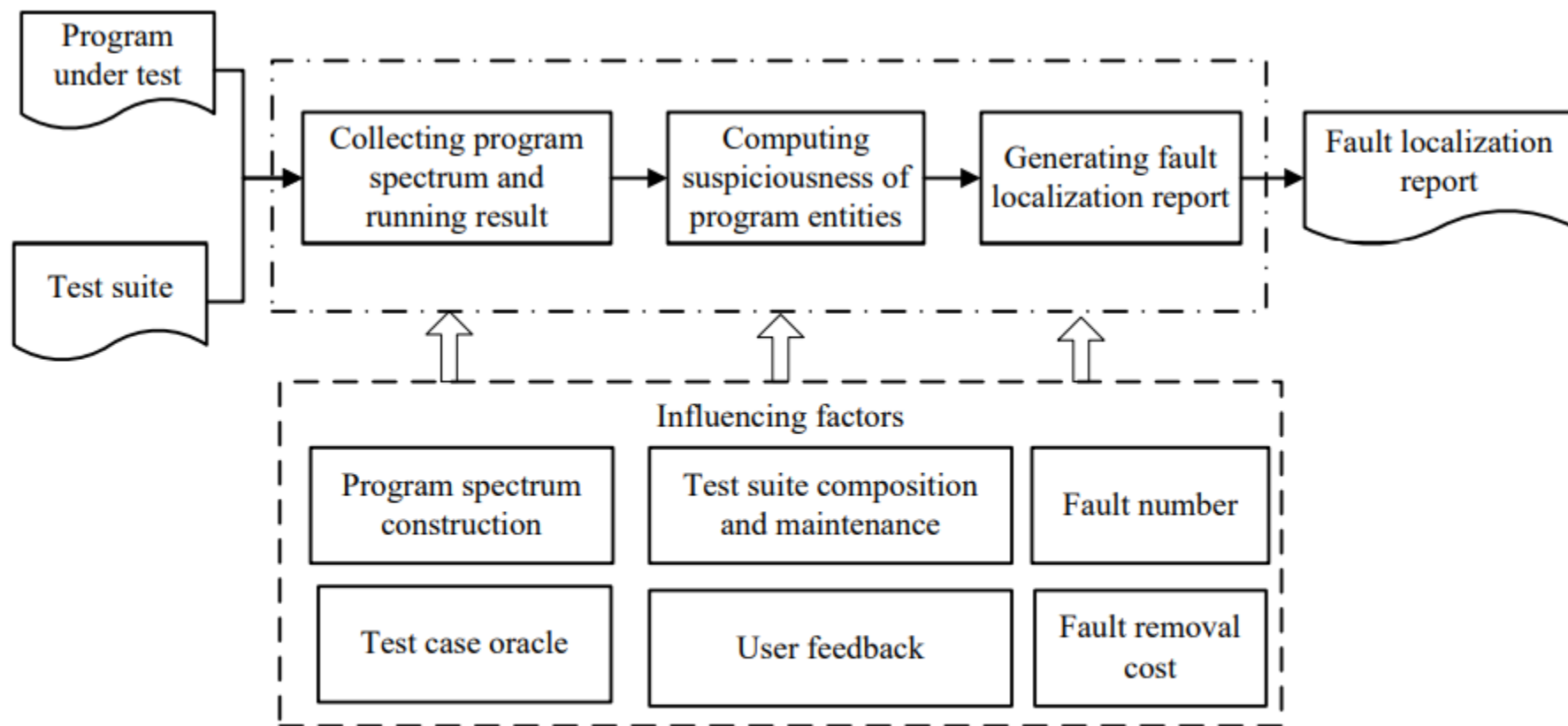


Fig.1 Research framework of SFL

图 1 SFL 研究框架

# 框架内影响因素分析

程序频谱构造方式

01

04

测试用例预言构造

测试套件构成和维护

02

05

用户反馈

缺陷数量

03

06

缺陷修复开销

# 程序频谱构造方式

$$Tarantula(s) = \frac{n_{ef}(s)/n_f}{n_{ef}(s)/n_f + n_{ep}(s)/n_p}$$

## 轻量级

仅需简单统计测试用例的**程序实体覆盖信息**。

## 重量级

需要进一步**分析程序实体间的控制或数据依赖关系**。

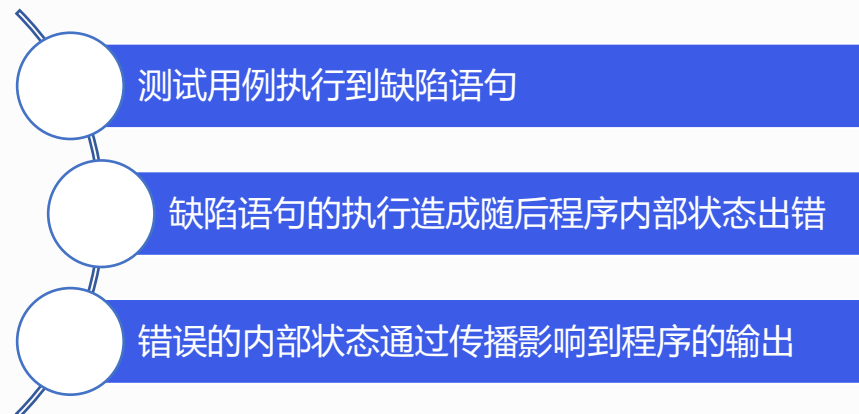
- 基于信息流的 Tarantula 方法 / DES方法 / 基于缺陷传播分析的 CP 模型 / LOUPE 模型 / 概率程序依赖图PPDG /基于执行切片和砍片的缺陷定位方法

## 其他

例如考虑面向对象程序特征、测试用例执行时间或同时考虑多种不同程序频谱等。

# 测试套件构成

触发软件失效的  
3个必要条件



## 偶然正确测试用例

仅保证满足必要条件(1)和(2)，而(3)不满足。

- Muffler 方法

1

## 相似测试用例

- SAFL 方法
  - 构造执行矩阵  $E = e_{ij}$
  - 构造出量化矩阵  $F = (f_{ij})$

$$SAFL(j) = \frac{\sum_{k=1}^m \max(\{f_{ik} \mid e_{ij} > 0 \wedge e_{i(m+1)} = 0 \wedge 1 \leq i \leq n\})}{\sum_{k=1}^m \max(\{f_{ik} \mid e_{ij} > 0 \wedge 1 \leq i \leq n\})}$$

## 测试用例权重设置

- 成功测试用例的权重
- 失败测试用例的权重
- 增大边际权重

3



# 测试套件维护

## 测试用例优先级排序

- 设定特定排序准则;
- 工作主要在两个方向:
  - (1) 分析已有的 TCP 方法;
  - (2) 以提高缺陷定位效果为优化目标来指导测试用例的排序。

## 测试用例生成方法

- 传统测试用例生成方法
- 面向缺陷定位的测试用例生成方法
- 面向调试的测试(test-for-diagnosis)准则
- 多种用例相似性准则
- ENTBUG 方法

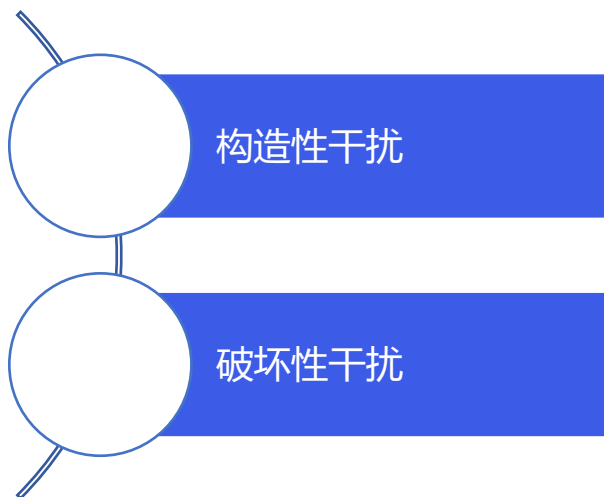
## 测试套件缩减

在满足对指定测试需求的覆盖前提下, 通过识别并移除冗余测试用例来降低测试开销。

- 基于语句的 TSR 策略
- 基于向量的 TSR 策略

# 缺陷数量

缺陷间干扰现象



## 失效聚类

- 将失败测试用例聚类
- 双聚类算法，可迭代的集体投票机制
- 新的失败测试用例相似性度量指标
- 并行计算的思想

## 基于模型的调试

- 基于模型的调试技术与 SFL 方法相结合，提出基于贝叶斯推理方法
- 结合程序切片技术，提出基于条件执行切片谱的多缺陷定位方法

# 测试用例预言

## 问题

测试用例预言(test case oracle)通过提供测试用例的预期结果与实际结果的对比机制来判断测试用例的执行结果是否成功。

## 解决方法

**蜕变测试**(metamorphic testing)是部分解决测试用例预言问题的一种有效方法。

- 分析被测程序特征, 获得一组蜕变关系;
- 根据蜕变关系生成一组新的相关测试用例;
- 执行这些测试用例, 并检查是否满足特定蜕变关系;
- 蜕变切片
- 基于多样性最大化加速(diversity maximization speedup)的策略

传统的手工调试方式 → 交互式测试框架

- (1) 框架重新计算各个语句的怀疑率并继续推荐新的可疑语句。
- (2) 提出一种鲁棒性方法，可确保该框架在推荐时不会遗漏真正缺陷语句。

# 缺陷修复开销

程序自动化修复技术借助特定修改规则可以完成对一些可疑语句的修复，因此，可以借助 SFL 方法来提高程序修复效率。

- 评测指标 NCP
- 借助变异分析

# 已有实证研究分析——评测指标

- 基于程序依赖图的评测指标 T-Score
- 基于语句排序的评测指标 Score

针对语句  $n$ ，首先定义  $k$  依赖集  $DS_k(n)$ 。该集合包括从语句  $n$  出发，采用广度优先搜索策略，前向或后向遍历依赖边，且所有距离为  $k$  的语句集。假设  $DS^*(R)$  为包含缺陷语句的最小依赖集，则该评测指标的计算公式为

$$1 - \frac{|DS^*(R)|}{|PDG|}$$

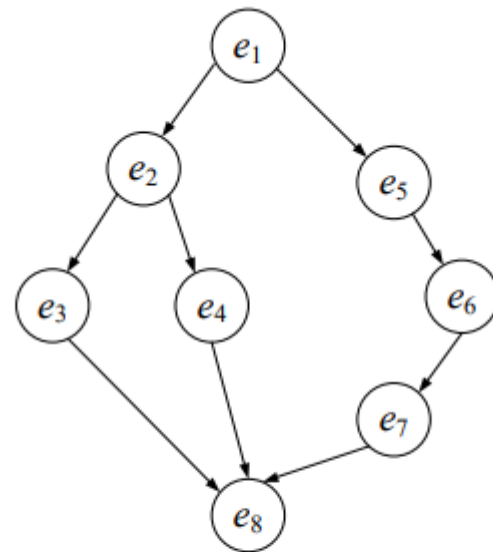


Fig.2 Program dependence graph of a faulty program

图 2 缺陷程序对应的程序依赖图

# 已有实证研究分析——评测程序

**Table 2** Subjects used in previous empirical studies

**表 2** 已有实证研究中使用的评测程序

| Name          | Size   | Description  | First used | Cumulative number |
|---------------|--------|--|------------|-------------------|
| Siemens suite | Small  | Small programs developed by Siemens                                  | 2003       | 65                |
| space         | Medium | An interpreter for an array definition language                      | 2002       | 33                |
| grep          | Large  | Searches input files for a pattern                                   | 2006       | 28                |
| gzip          | Medium | Data compression/decompression                                       | 2009       | 25                |
| sed           | Large  | Text processor   | 2009       | 24                |
| flex          | Large  | A lexical analyzer generator   | 2009       | 18                |
| nanoXML       | Medium | A small XML parser for Java  | 2005       | 8                 |
| ant           | Large  | A Java-based build tool  | 2008       | 8                 |
| XMLsecurity   | Large  | Implements security standards for XML                                | 2008       | 8                 |
| Unix utility  | Small  | Unix utilities   | 2008       | 6                 |
| make          | Large  | Build manager for C programming language                             | 2010       | 5                 |
| Jtopas        | Medium | Facilitates users to tokenize and parse arbitrary text data          | 2011       | 5                 |
| Jmeter        | Large  | Performance testing tool   | 2011       | 5                 |
| BC            | Large  | A calculator program that accepts scripts written in the bc language | 2006       | 4                 |
| DC            | Medium | Desk calculator  | 2005       | 2                 |
| TCC           | Large  | Tiny C compiler  | 2005       | 2                 |
| Jaligner      | Medium | An algorithm for biological local pair-wise sequence alignment       | 2010       | 2                 |
| seqmap        | Medium | A bioinformatics tool in C++   | 2011       | 2                 |
| SimpleJavaApp | Small  | A displaying and editing book listing application                    | 2012       | 2                 |
| Tetris        | Medium | Tetris game  | 2012       | 2                 |
| JHSA          | Large  | A JAVA hierarchical slicing tool                                     | 2012       | 2                 |



学 习 进 展 & 暑 期 计 划

# 感谢您的聆听

汇报人：王昭丹

