



NANJING UNIVERSITY · SOFTWARE INSTITUTE
南京大学软件学院

自动化软件测试

陈振宇

南京大学软件学院

计算机软件新技术国家重点实验室

<http://software.nju.edu.cn/zychen>

2013/6/17

南京



NANJING UNIVERSITY · SOFTWARE INSTITUTE
南京大学软件学院

梦想



100% 自动化是软件测试的一个梦想！

Software Testing Research: Achievements, Challenges, Dreams, A.
Bertolino, In Future of Software Engineering @ ICSE 2007



NANJING UNIVERSITY · SOFTWARE INSTITUTE
南京大学软件学院

梦想与现实



软件测试

学术研究

- 顶级会议ICSE, FSE
 - 1/4-1/3测试相关论文
- 以测试为主的权威学术会议: ISSTA, ICST, ICSM, QSIC
- 软件测试相关论文中一般以上与自动化测试相关。

工业应用

- 2006年HP以45亿美元现金收购Mercury
 - Loadrunner, Qtp, Diagnostics, Qc
- 2011年SOASTA获得3300万美元投资, 基于云端测试应用
- 2012年Sauce Labs B轮融资中获得 300 万美元, 基于Selenium



自动化测试

学术研究

工业应用

自动化测试生成

自动化测试执行

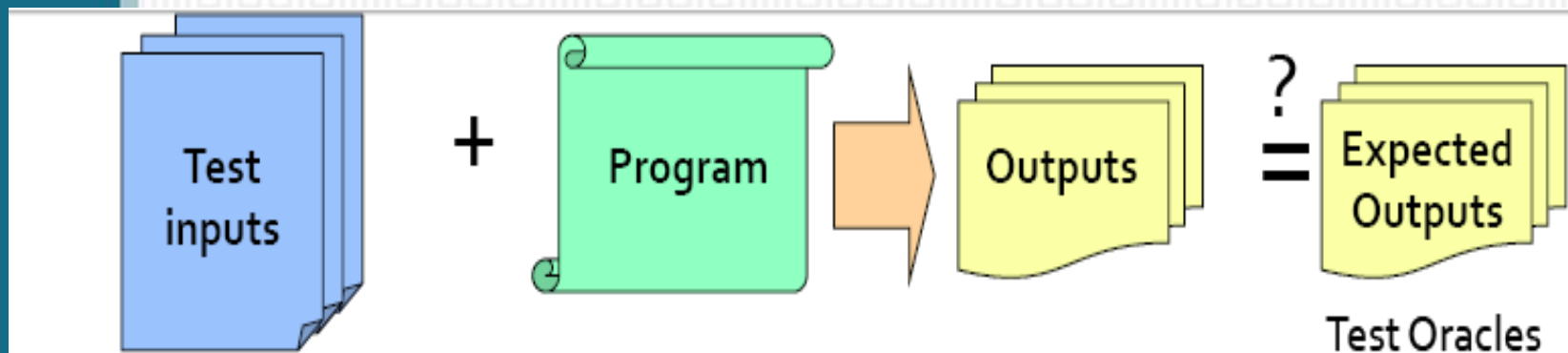
自动化测试选择

自动化测试修复

自动化测试扩增



测试用例生成



测试数据

+

测试预言

测试用例生成



自动化测试数据生成工具

工具名称	适用语言	特 点	是否工业化
JPF	Java语言	1.较为通用2.在执行过程中会存储参数的实际值，能够用于动态符号执行3.能够分析java字节码和状态图模型	目前已用于NASA的部分程序中，仍未完全工业化。
DART	C语言	1.结合了随机测试和模型检验技术，能够用于测试用例的生成2.采用了实际数值和随机化技术，增加了符号执行的精确度	目前已被BELL实验室和其他扩展工具使用。
CUTE	C语言	1.适用于单元测试2.基于DART的扩展，可以用于多线程程序3.基本解决了指针限制，避免了因指针分析而导致符号执行不准确的问题4.	目前主要用于几个开源程序检测，主要由学校的实验室进行研究，基本未实现工业化。
JCUTE	Java语言	结合了 concolic execution 和 dynamic partial order reduction 来产生测试用例	
CREST	C语言	1.是一个concolic testing的开源工具2.使用启发式方法来选择程序执行路径3.易于扩展	目前已提供开源下载，下载量有1500+且已用于多个团队的研究。
SAGE	C语言	1.适用于集合测试2.适用于大规模程序	目前已经用于工业程序中，例如在window 7的研发中
PEX	C#, VisualBasic, F#	1.用参数的实际数字以简化约束条件2.支持复杂数据类型的测试用例生成3.在寻找路径时使用不同的路径搜索方法以快速达到较高的覆盖。	目前已经下载了40000多次。已经被微软的部分小组所使用。可以同时用于学术研究或工业使用。
EXE.	C语言	1.采用了符合和实际值结合的符号执行方法，并且采用了位级存储2.使用启发式方法来选择程序执行路径3.适用于对复杂程序的测试4.包含了多个约束优化器，可以快速求解约束	目前已经集成在 Visual Studio 2011 中，已经工业化。
KLEE.	C语言	1.是基于EXE的一个扩展工具2.相对于目前已有的工具，可以存储大量的并发状态3.拥有处理外部环境交换的能力	目前已经提供开源下载，可以同时用于学术研究或工业使用。



自动化测试数据生成研究

```
main(){  
    int t1 = randomInt();  
    int t2 = randomInt();  
    test_me(t1,t2);  
}  
  
int double(int x) {return 2 * x; }  
  
void test_me(int x, int y) {  
    int z = double(x);  
    if (z==y) {  
        if (y == x+10)  
            abort(); /* error */  
    }  
}
```

**Symbolic
Execution**

**Path
Constraint**

Solve: $(2 * x == y) \ \&\& \ (y == x + 10)$

Solution: $x = 10, y = 20$

create symbolic
variables x, y

$2 * x == y$

$y != x + 10$

$z = 2 * x$



测试用例选择

- Regression Test Selection
 - 针对修改部分的测试
- Test Suite Reduction
 - 针对测试需求的用例集约简
- Test Case Prioritization
 - 针对测试需求的用例排序
- **S. Yoo and M. Harman. Regression Testing Minimization, Selection and Prioritization : A Survey. STVR 2010.**



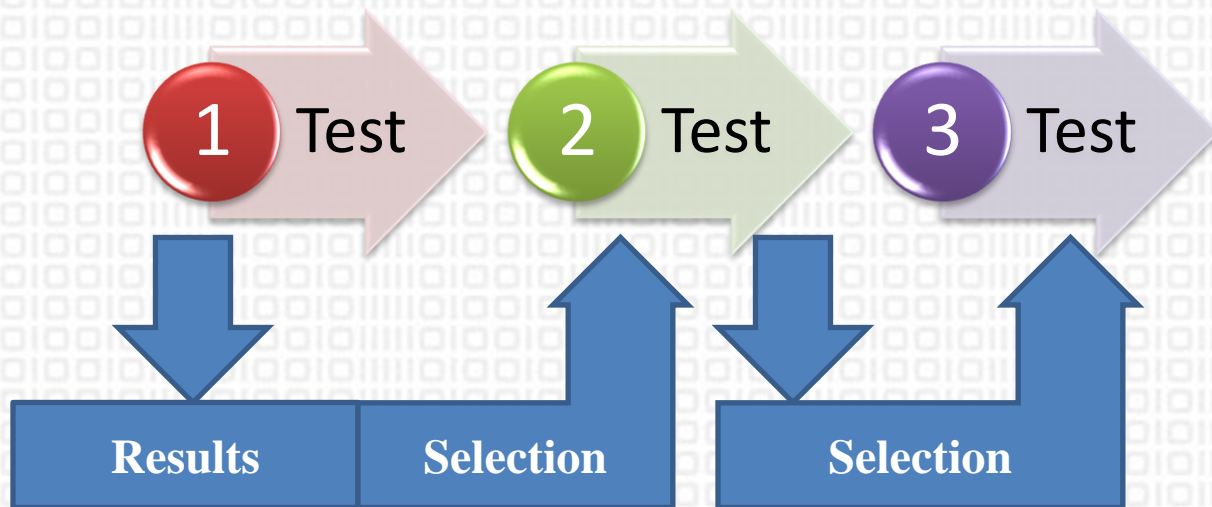
测试用例选择研究

	t1	t2	t3
r1	1	1	0	
r2	0	1	1	
r3	1	0	1	
r4	1	1	0	
r5	0	0	1	
.....				

数学? 算法? 软件工程?



测试用例选择假设



- 测试用例过多，无法在指定时间执行完。

云计算的出现

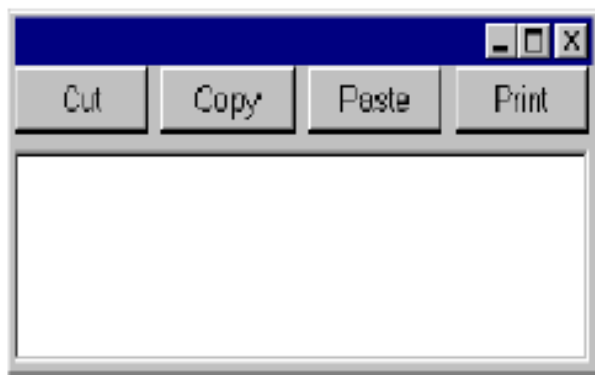
- 旧的测试用例能够在新版本软件执行。

测试用例修复

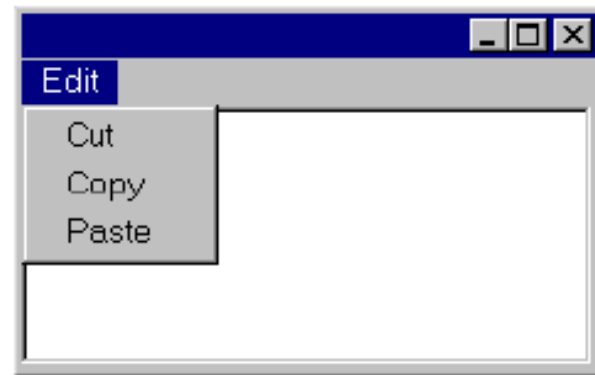
测试用例扩增



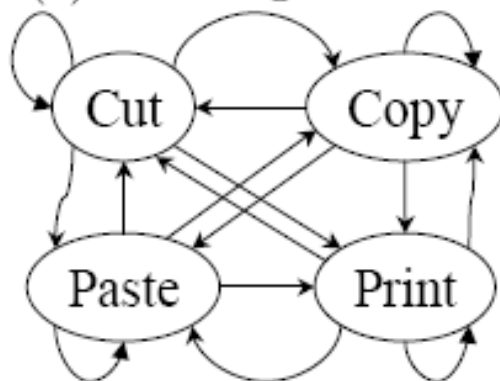
测试用例修复



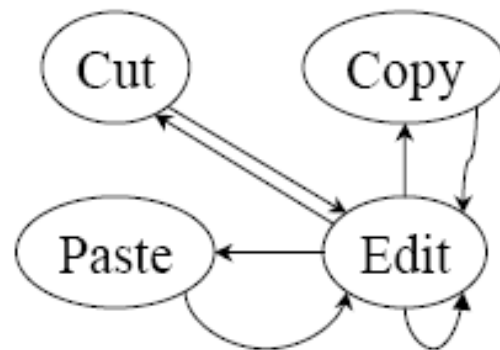
(a) The Original GUI.



(b) The Modified GUI.



(c) The Original EFG.



(d) The Modified EFG.



测试用例修复研究

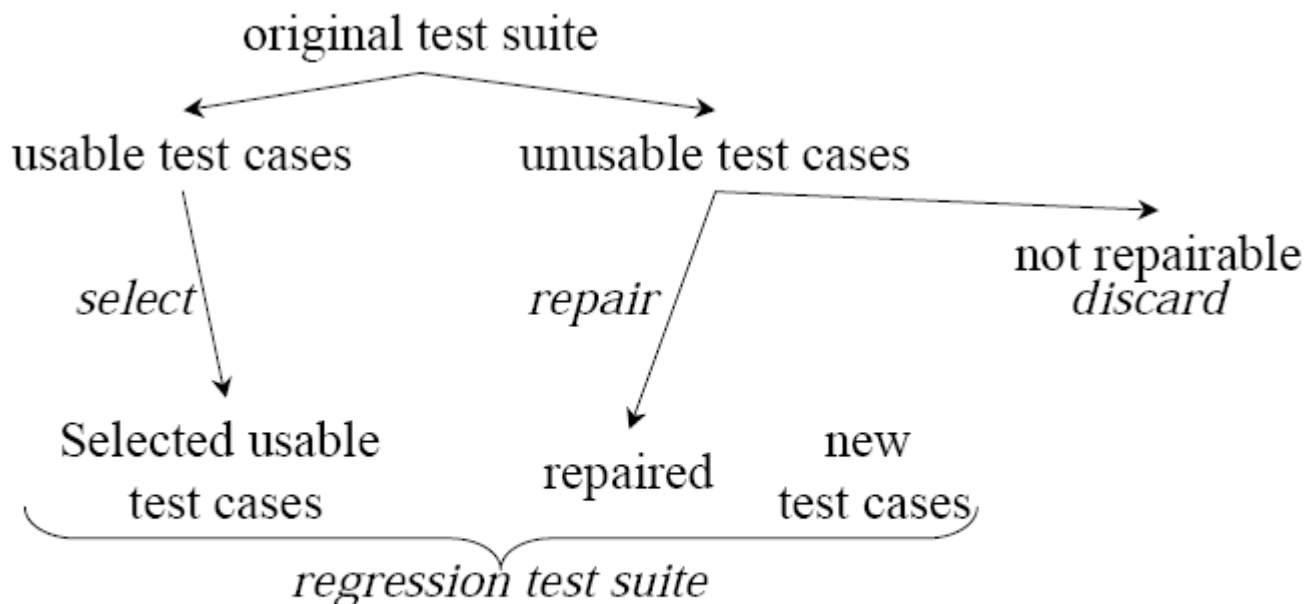


Fig. 2. The New Regression Testing Method.

Atif Memon. Automatically Repairing Event Sequence-Based GUI Test Suites for Regression Testing, TOSEM, 2008



测试用例修复

- EFG模型的测试脚本修复

工业化测试工具脚本
对象映射问题

- 完备的EFG模型

非完备EFG，无法判断对象 是漏识别还是已删除



自动化错误定位

```
mid() {  
    int x,y,z,m;  
1:  read("Enter 3 numbers:",x,y,z);  
2:  m = z;  
3:  if (y<z)  
4:      if (x<y)  
5:          m = y;  
6:      else if (x<z)  
7:          m = y;  
8:  else  
9:      if (x>y)  
10:         m = y;  
11:     else if (x>z)  
12:         m = x;  
13: print("Middle number is:", m);  
}
```

Test Cases

	3,3,5	1,2,3	3,2,1	5,5,5	5,3,4	2,1,3
	•	•	•	•	•	•
	•	•	•	•	•	•
	•	•	•	•	•	•
		•				
	•				•	•
	•					•
	•		•	•		
			•			
			•			
	•	•	•	•	•	•
	P	P	P	P	P	F

Results:

Name	Formula	Name	Formula
Jaccard	$\frac{a_{ef}}{a_{ef}+a_{nf}+a_{ep}}$	Anderberg	$\frac{a_{ef}}{a_{ef}+2(a_{nf}+a_{ep})}$
Sørensen-Dice	$\frac{2a_{ef}}{2a_{ef}+a_{nf}}$	Dice	$\frac{2a_{ef}}{a_{ef}+a_{nf}+a_{ep}}$
Kulczynski1	$\frac{a_{ef}}{a_{nf}+a_{ep}}$	Kulczynski2	$\frac{1}{2} \left(\frac{a_{ef}}{a_{ef}+a_{nf}} + \frac{a_{ef}}{a_{ef}+a_{ep}} \right)$
Russell and Rao	$\frac{a_{ef}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$	Hamann	$\frac{a_{ef}+a_{np}-a_{nf}-a_{ep}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$
Simple Matching	$\frac{a_{ef}+a_{np}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$	Simple Matching	$\frac{2(a_{ef}+a_{np})}{2(a_{ef}+a_{np})+a_{nf}+a_{ep}}$
M1	$\frac{a_{ef}+a_{np}}{a_{nf}+a_{ep}}$	M1	$\frac{a_{ef}}{a_{ef}+a_{np}+2(a_{nf}+a_{ep})}$
Rogers & Tanimoto	$\frac{a_{ef}+a_{np}}{a_{ef}+a_{np}+2(a_{nf}+a_{ep})}$	Goodman	$\frac{2a_{ef}-a_{nf}-a_{ep}}{2a_{ef}+a_{nf}+a_{ep}}$
Hamming etc	$\frac{a_{ef}+a_{np}}{a_{ef}+a_{nf}+a_{ep}+a_{np}}$	Euclid	$\frac{a_{ef}}{\sqrt{a_{ef}+a_{np}}}$
Ochiai	$\frac{a_{ef}}{a_{nf}+a_{ep}}$	Ochiai	$\frac{a_{ef}}{a_{nf}+a_{ep}}$
Tarantula	$\frac{\frac{a_{ef}}{a_{ef}+a_{nf}}}{\frac{a_{ef}}{a_{ef}+a_{nf}} + \frac{a_{ep}}{a_{ep}+a_{np}}}$	Zoltar	$\frac{a_{ef}}{a_{ef}+a_{nf}+a_{ep} + \frac{10000a_{nf}a_{ep}}{a_{ef}}}$
Ample	$\left \frac{a_{ef}}{a_{ef}+a_{nf}} - \frac{a_{ep}}{a_{ep}+a_{np}} \right $	Wong1	a_{ef}

Xiaoyuan Xie, Tsong Chen, Fei-Ching Kuo, Baowen Xu
A Theoretical Analysis of the Risk Evaluation Formulas for Spectrum-Based Fault Localization, accepted by TOSEM



错误定位假设

```
mid() {  
    int x,y,z,m;  
1:   read("Enter 3 numbers:",x,y,z);  
2:   m = z;  
3:   if (y<z)  
4:       if (x<y)  
5:           m = y;  
6:       else if (x<z)  
7:           m = y;  
8:   else  
9:       if (x>y)  
10:           m = y;  
11:      else if (x>z)  
12:           m = x;  
13:  print("Middle number is:", m);  
}
```

单Bug假设

完美Debugger假设



NANJING UNIVERSITY · SOFTWARE INSTITUTE
南京大学软件学院

学术研究

怎么破？

工业应用



博士和博士后研究

- 博士：关于模型检测中的极小化抽象
- 博士后：布尔逻辑测试研究
 - 逻辑故障模型
 - TOSEM 2011, SAC 2008
 - 测试用例集约简
 - SAC2008, SEKE2008, QSIC2008

教学

科研

社会服务

2008年8月



寻找新研究方向

- 容易发论文

跨领域研究

- 有一定基础

软件测试

- 有一些兴趣

机器学习+测试用例选择

2009年

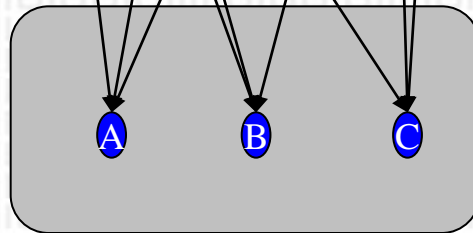
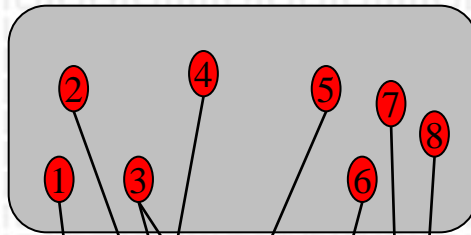
研究生

经费



Failure Proximity

● Test Data



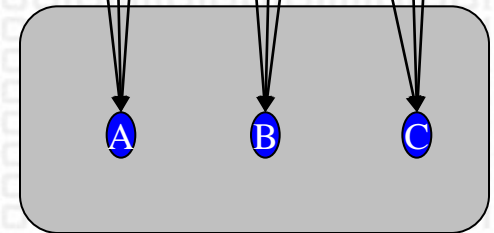
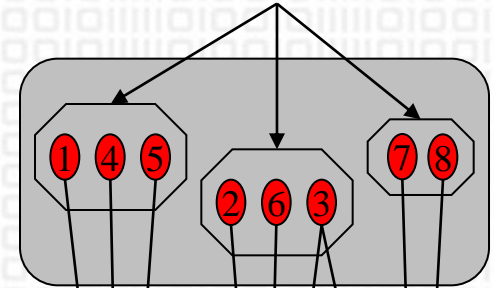
Faults in
Software

Transform

Mapping
Function

机器学习

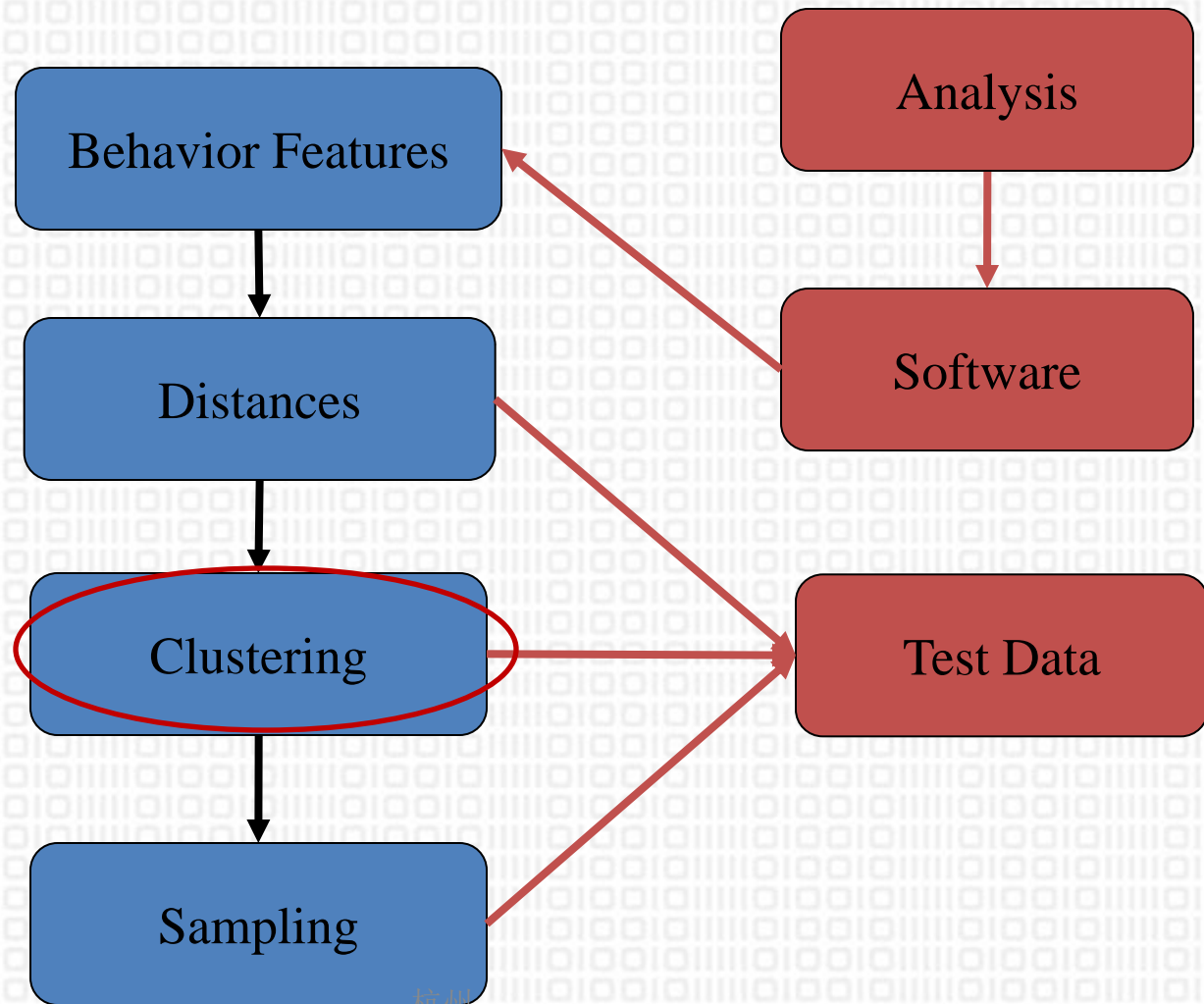
Cluster Sampling



Faults in
Software



基于软件行为聚类分析的 测试用例选择





简单示例

- Distance
 - Hamming distance
 - $D(t1, t2)=4$
 - $D(t1, t3)=3$
 - $D(t1, t2) > D(t1, t3)$
 - $[t1, t3]$ is more likely to be in same cluster than $[t1, t2]$
- The failure proximity of each test is computed by the function call trace.

	t1	t2	t3	t4
f1	1	0	1	0
f2	1	1	0	1
f3	0	1	0	1
f4	1	1	1	0
f5	1	0	1	1
f6	0	0	1	1
f7	1	1	0	1



研究问题

- 测试用例选择：根据软件的某种属性度量进行多样性选择
 - 属性：语句，分支，定义-引用等
 - 度量：覆盖
- 挑战：
 - 错误检测能力
 - 可伸缩性
- 属性简单：如何构造更丰富的属性？
- 度量简单：如何构造更丰富的度量？

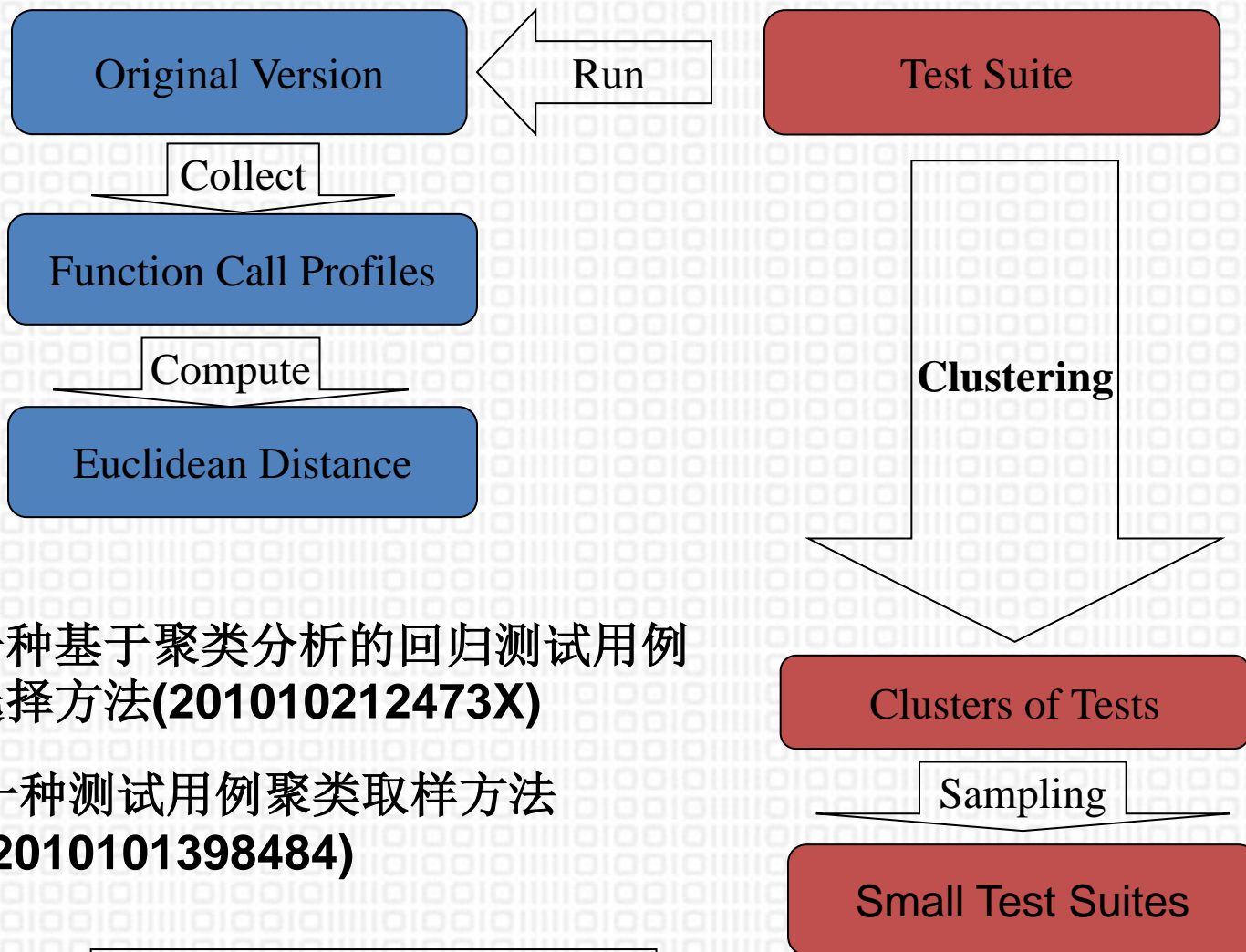


该方向的研究成果

- 2009年启动
- 2010年获资助（赵志宏）
- 2010年：ICST, QSIC, SEKE
- 2011年：ICST, IJSEKE
- 2012年：ICSE, SEKE, SERE
- 2013年：IJSEKE, 《计算机学报》...
- 5项发明专利（授权2项）
- 软件著作权



早期研究成果



一种基于聚类分析的回归测试用例选择方法(201010212473X)

一种测试用例聚类取样方法
(2010101398484)

工业应用



实证研究

- 不同距离是否影响结果？
 - 海明距离 vs. 欧式距离
- 不同聚类算法是否影响结果？
 - Hierarchy algorithms
 - Partition algorithms
 - Density algorithms

关于聚类测试选择方法的实证研究. 《计算机学报》



实证研究

- 不同距离是否影响结果？
 - 海明距离 vs. 欧式距离
- 不同聚类算法是否影响结果？
 - Hierarchy algorithms
 - Partition algorithms
 - Density algorithms



实证对象

表 1：目标程序相关属性

Program Name	Number of functions	Line of code	Test suite size	Modified version
Flex	148	10459	567	5
Space	136	6199	13585	33



实证设计

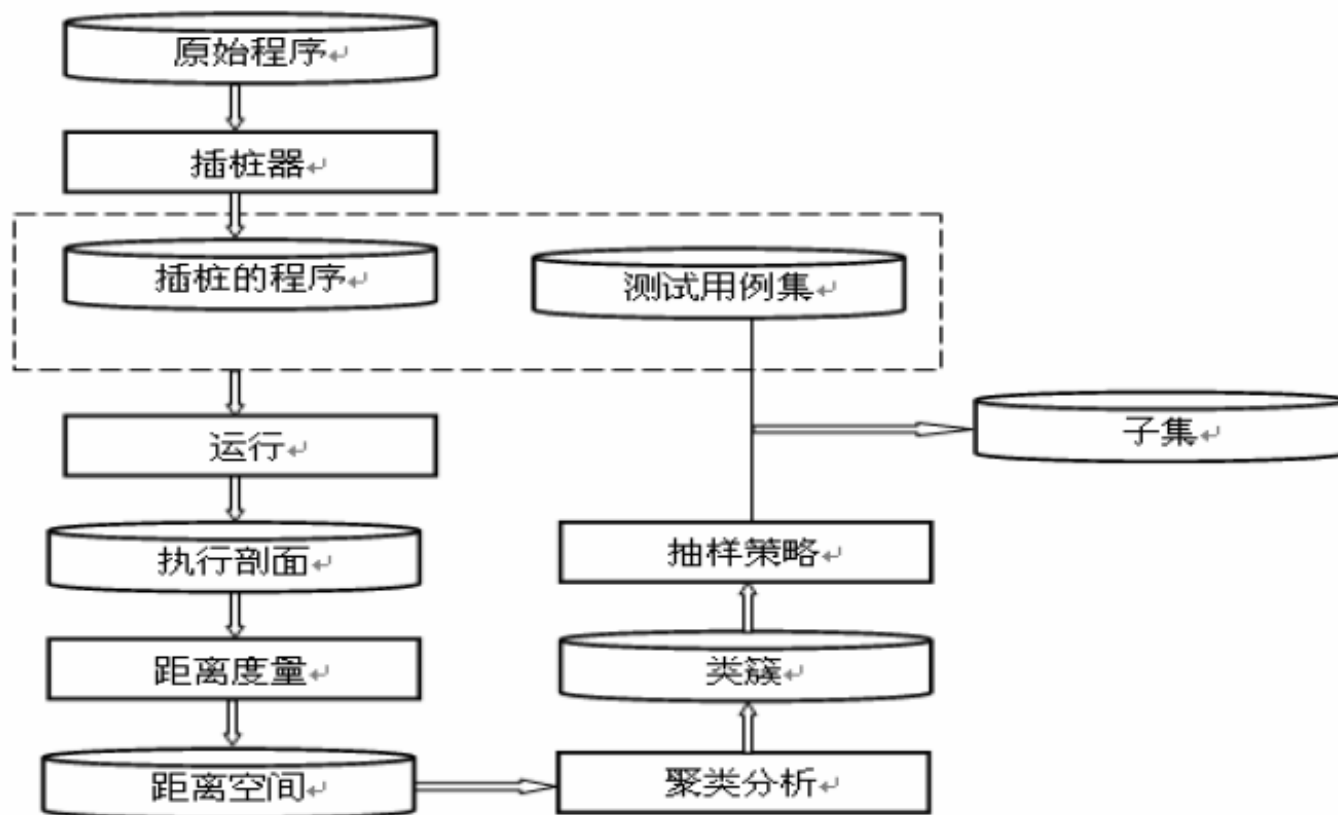


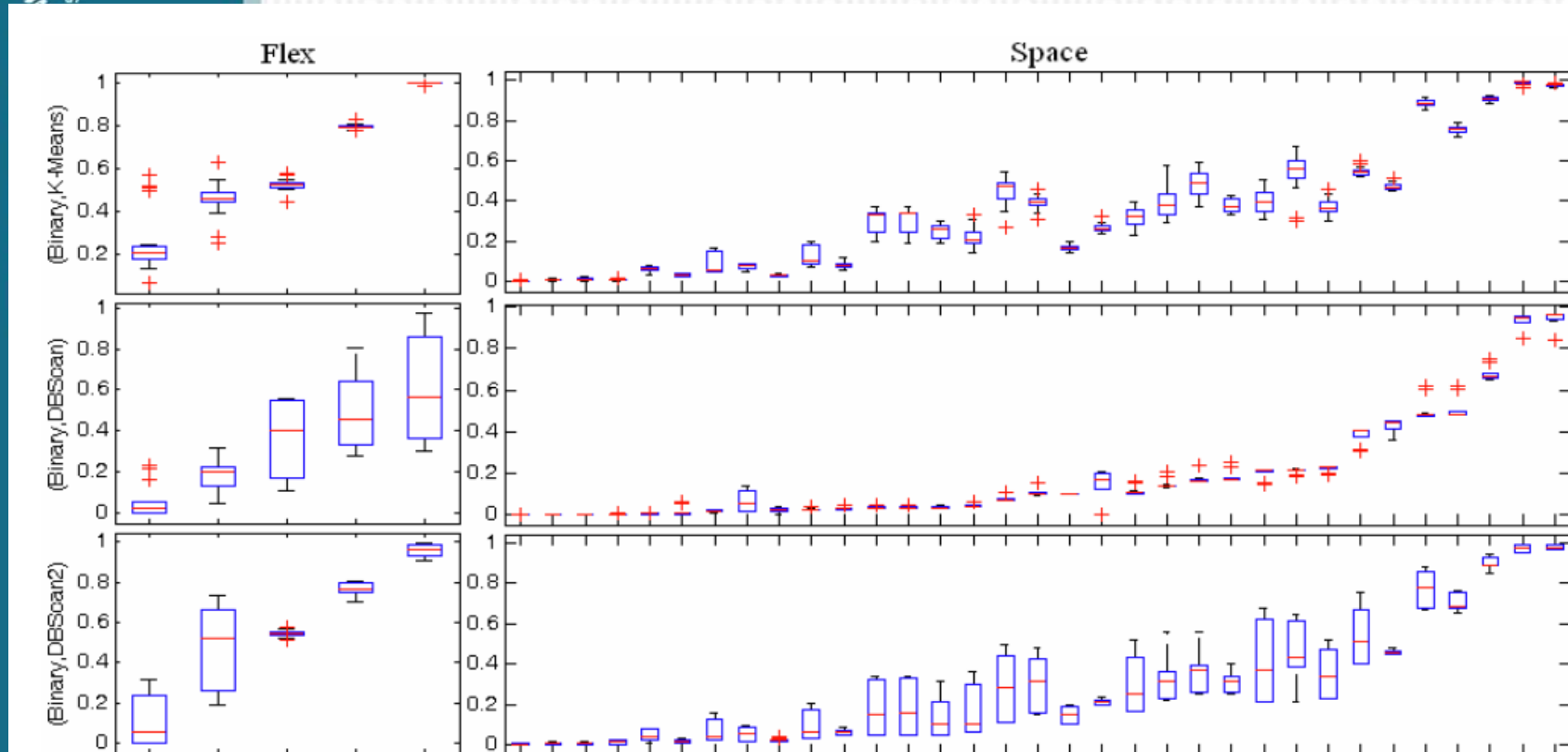
图 1：聚类测试选择流程



南京
UNICAVN

实证结果

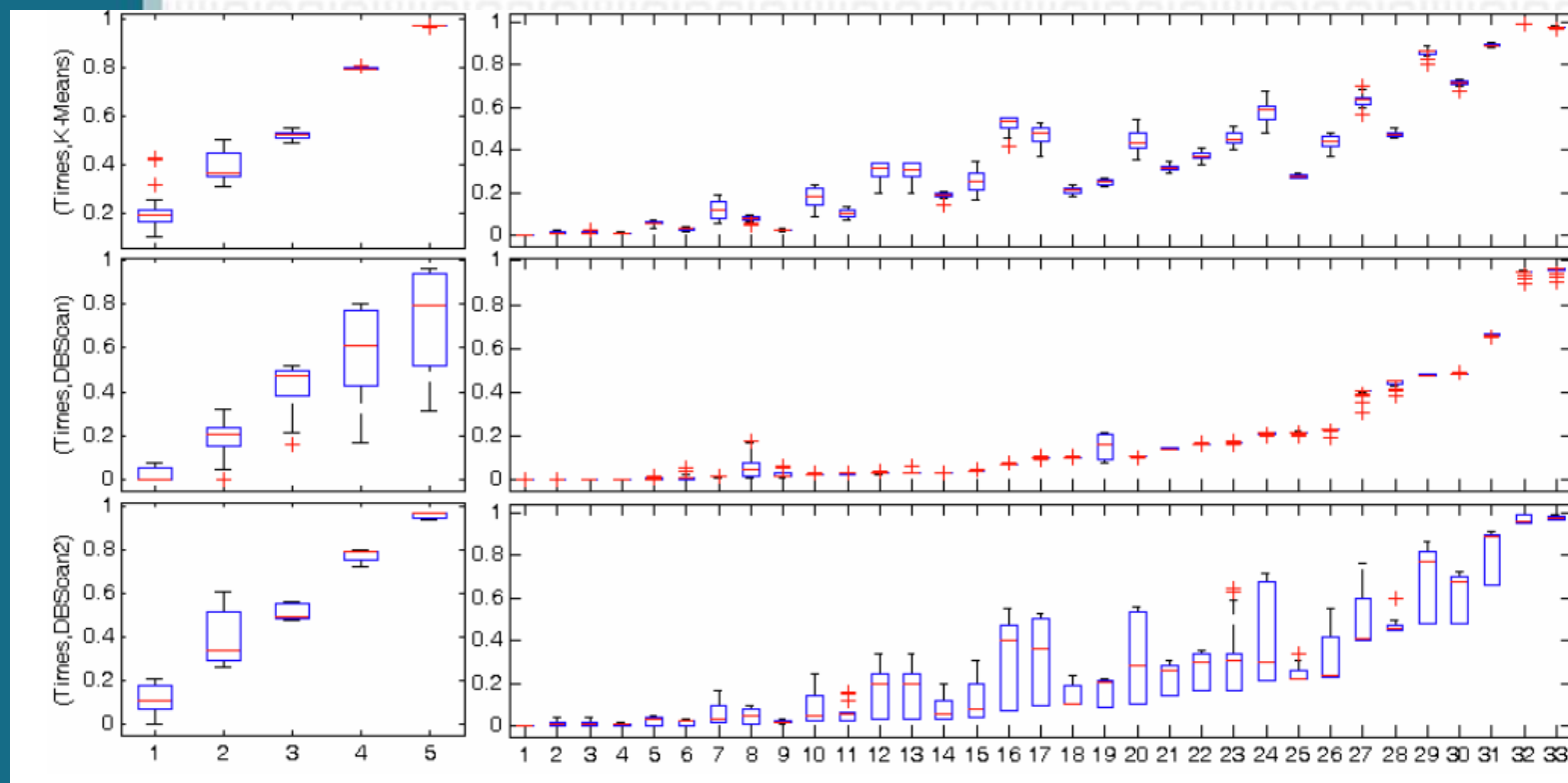
-海明距离





实证结果

--欧式距离





实证结果-距离影响分析

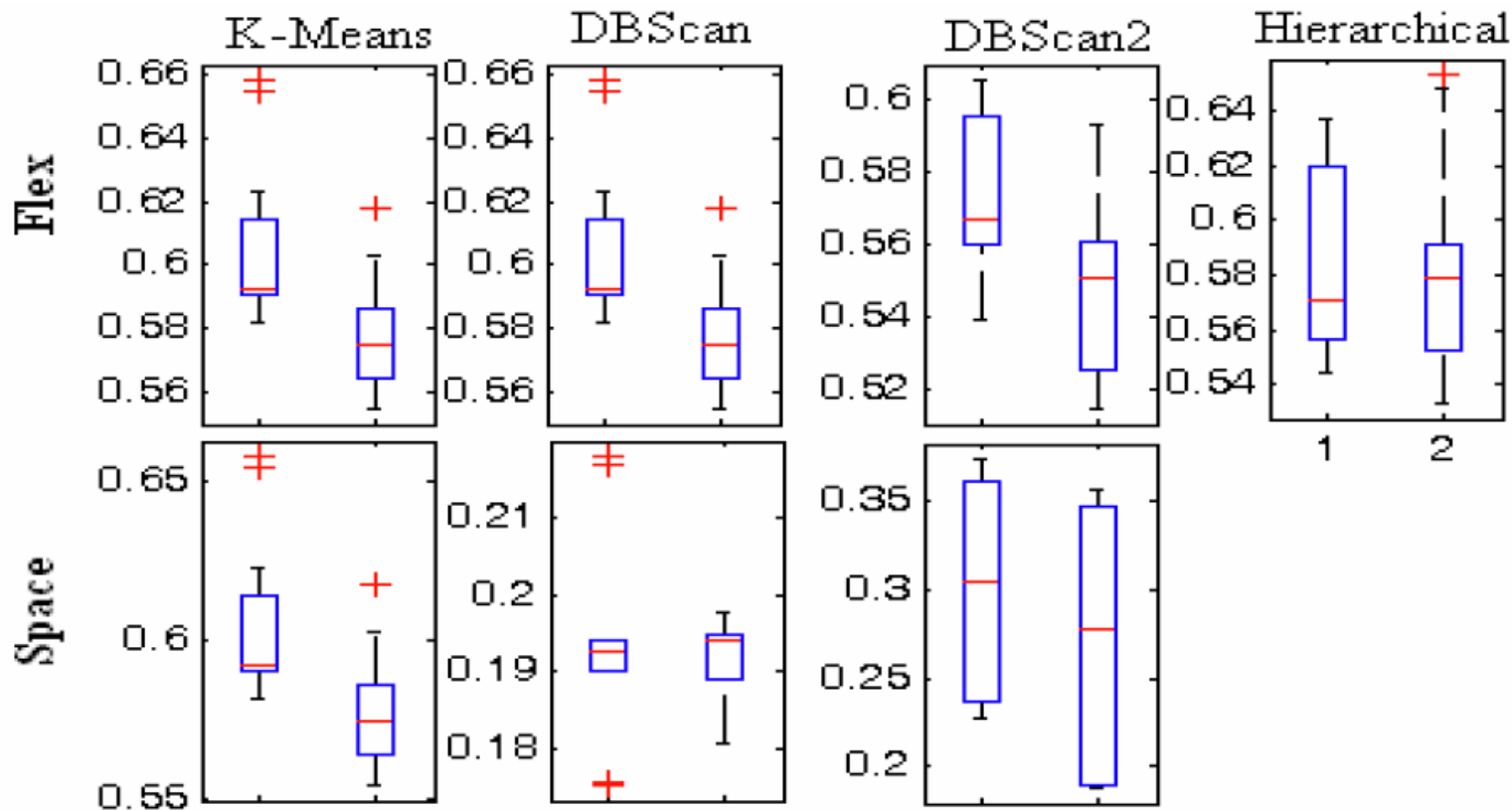


图 3：执行剖面的影响分析

程序 \ 聚类方法		K-means	DBScan	DBScan2	Hierarchical
Flex	h	1	1	1	0
	p-value	0.000014	0.018098	0.00037	0.404789
Space	h	1	0	1	
	p-value	0.002895	0.933260	0.0093896	



实证结果-聚类算法影响分析

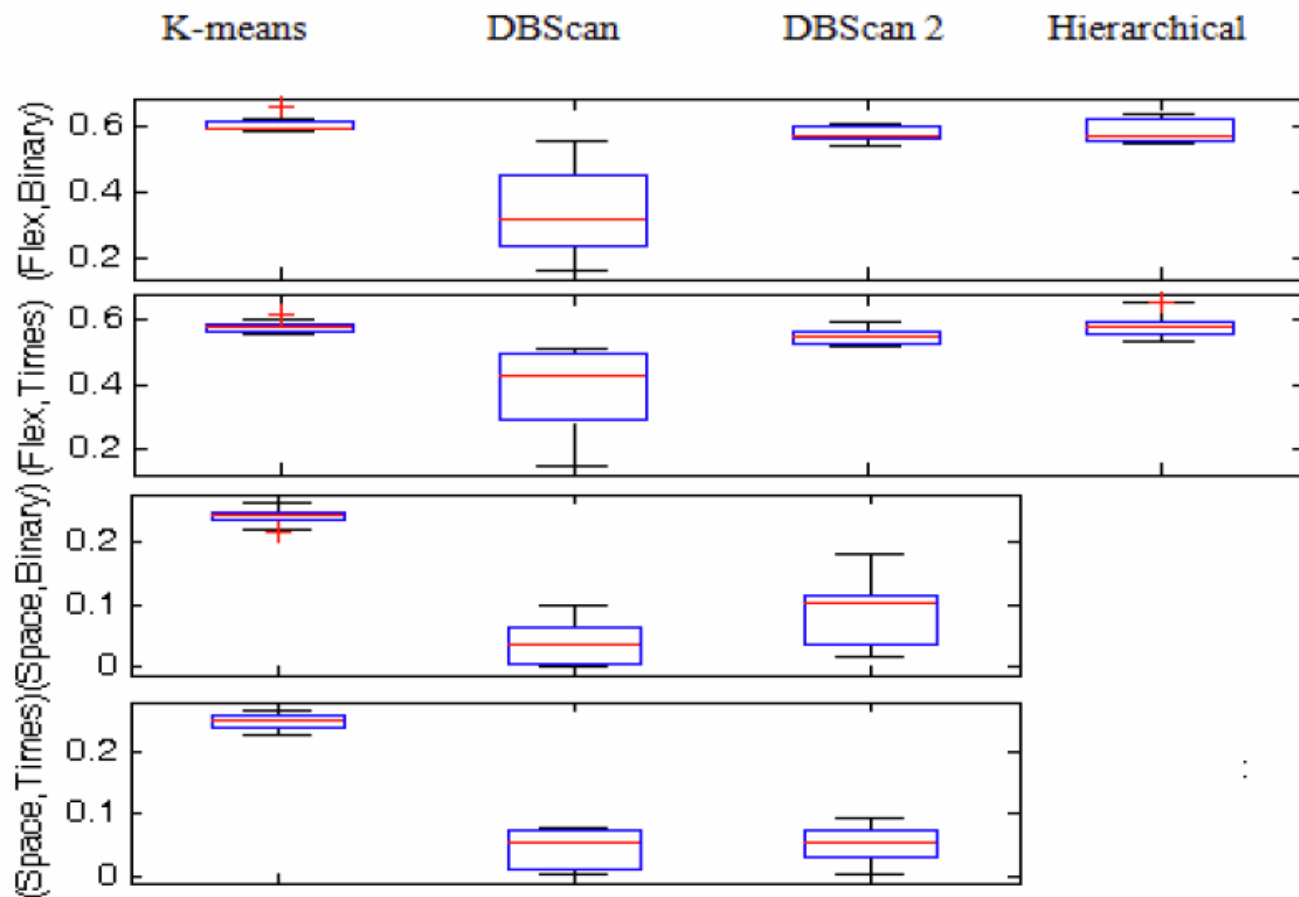


图 4: 聚类算法的影响分析



实证结果-聚类算法影响分析

表 4: Flex-函数调用与否剖面的方差分析结果($F_{0.01}(3,76)=4.0502$)

	V1	V2	V3	V4	V5
F 值	18.435	39.885	31.478	41.756	17.604

表 5: Flex-函数调用次数剖面的方差分析结果($F_{0.01}(3,76)=4.0502$)

	V1	V2	V3	V4	V5
F 值	12.341	17.292	27.616	17.787	31.354

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
F 值	18.598	32.520	23.941	24.542	57.474	18.035	17.429	1.975	8.932	24.987	99.808
	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
F 值	59.087	58.654	63.692	29.735	72.203	75.981	35.311	36.092	36.344	55.325	101.827
	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32	V33
F 值	146.463	19.403	53.400	22.766	22.349	23.763	244.738	289.358	597.559	26.729	13.455

表 7 : Space 函数调用次数剖面的方差分析结果($F_{0.01}(2,57)=4.9981$)

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11
F 值	43.285	23.371	22.532	46.308	115.998	17.443	38.459	9.263	2.452	44.046	36.456
	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22
F 值	78.534	77.249	123.392	62.044	84.329	59.118	56.713	27.161	38.775	101.340	100.728
	V23	V24	V25	V26	V27	V28	V29	V30	V31	V32	V33
F 值	48.914	37.683	40.257	35.928	36.984	13.058	86.800	71.730	60.977	49.566	15.355



Slice Filtering

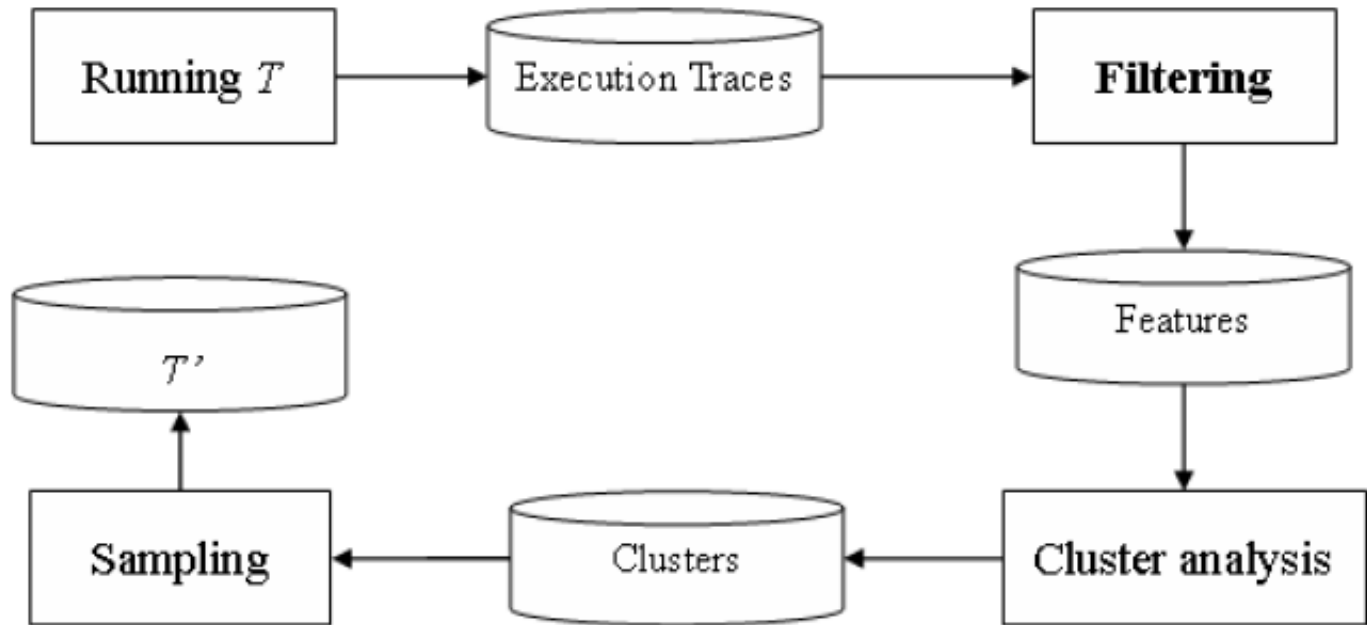


Fig. 1. A Framework of Cluster Test Selection

Zhenyu Chen, Yongwei Duan, Zhihong Zhao, Baowen Xu, Ju Qian:
Using Program Slicing to Improve the Efficiency and Effectiveness of
Cluster Test Selection. IJSEKE 21(6): 759-777 (2011)



切片过滤示例

Table 1. A Simple Example of Slice Filtering

No.	Statement	Slice	t_1 (1,6)	t_2 (2,0)	t_3 (7,6)
	<code>viod f (int m, int n){</code>				
1	<code>int a=0, b=0, c=0, d=0;</code>	★	●	●	●
2	<code>if (m<n)</code>	★	●	●	●
3	<code> a=1;</code>	★	●		
	<code> else</code>				
4	<code> b=1;</code>			○	○
5	<code>while(a>0){ /* (a>=0)*/</code>	★	●	●	●
6	<code> c=c+a;</code>	★	●		
7	<code> a=a-1;}</code>	★	●		
8	<code>printf("%s",c);</code>	★	●	●	●
9	<code>if(n>0){</code>		○	○	○
10	<code> b=n-5;</code>		○		○
11	<code> while(b>0){</code>		○		○
12	<code> d=d+b;</code>		○		○
13	<code> b=b-1;}</code>		○		○
14	<code> printf("%s",d);}</code>		○		○
	<code>}</code>				

Note: The modification is in the statement s_5 . ★ denotes the statement in the program slice. ● denotes the statement remained by slice filtering. ○ denotes the statement removed by slice filtering.



Dimensionality Reduction *--statement*

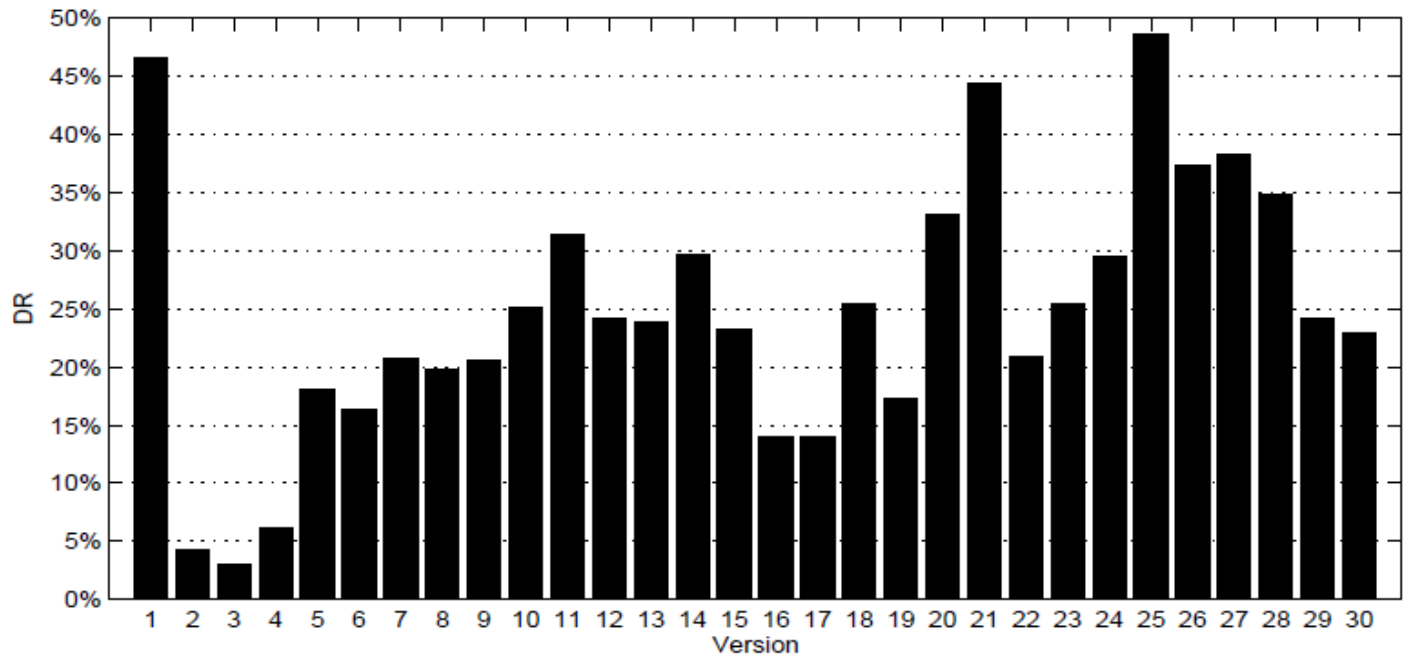


Fig. 2. DR of Statement: Mean 24.77%, Max 48.61%, Min 2.93%



Matrix Reduction *--statement*

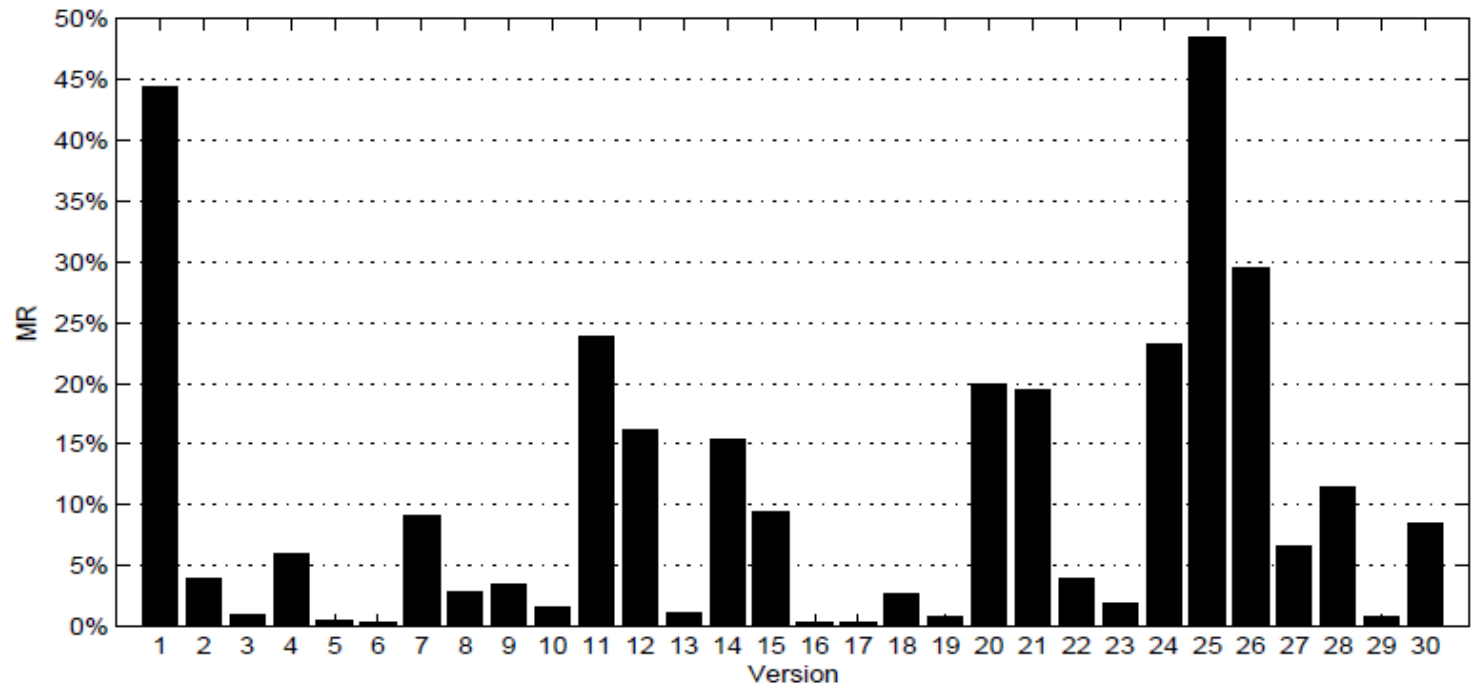


Fig. 3. MR of Statement: Mean 10.55%, Max 48.50%, Min 0.33%



Dimensionality Reduction *--block*

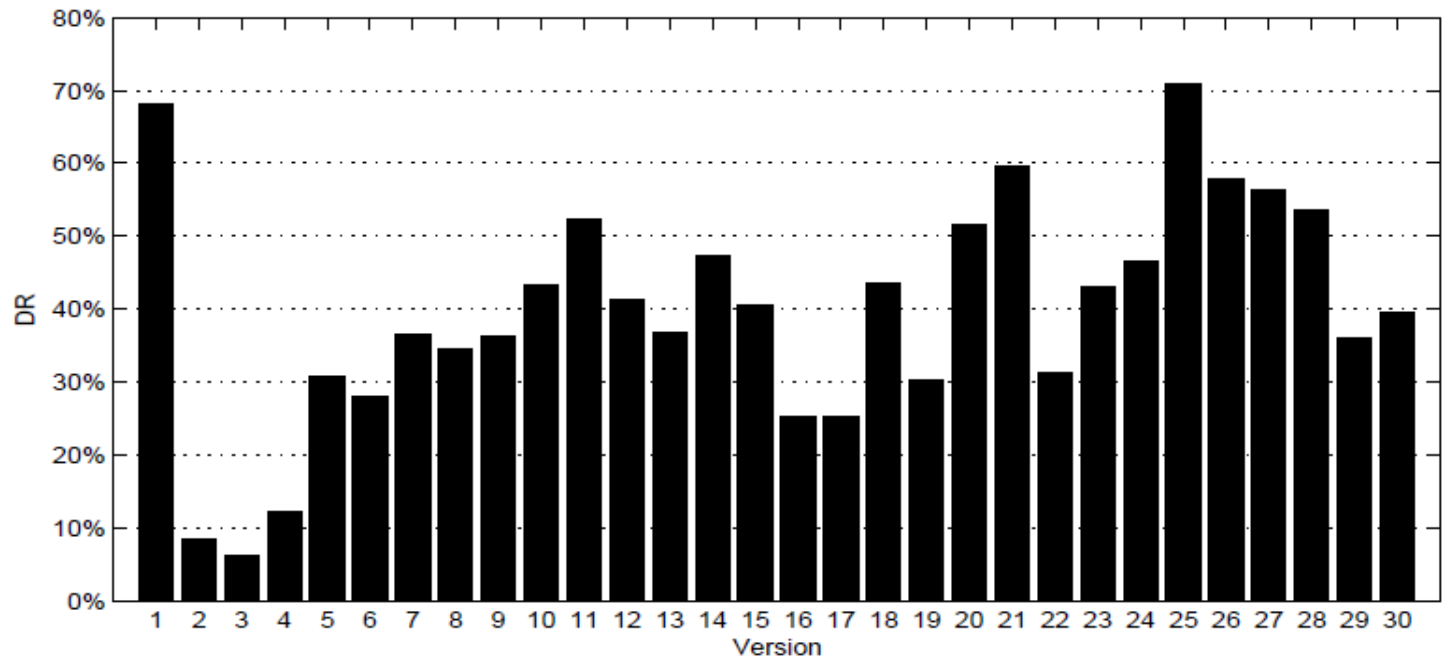


Fig. 4. DR of Block: Mean 39.75%, Max 70.97%, Min:6.07%



Matrix Reduction *--block*

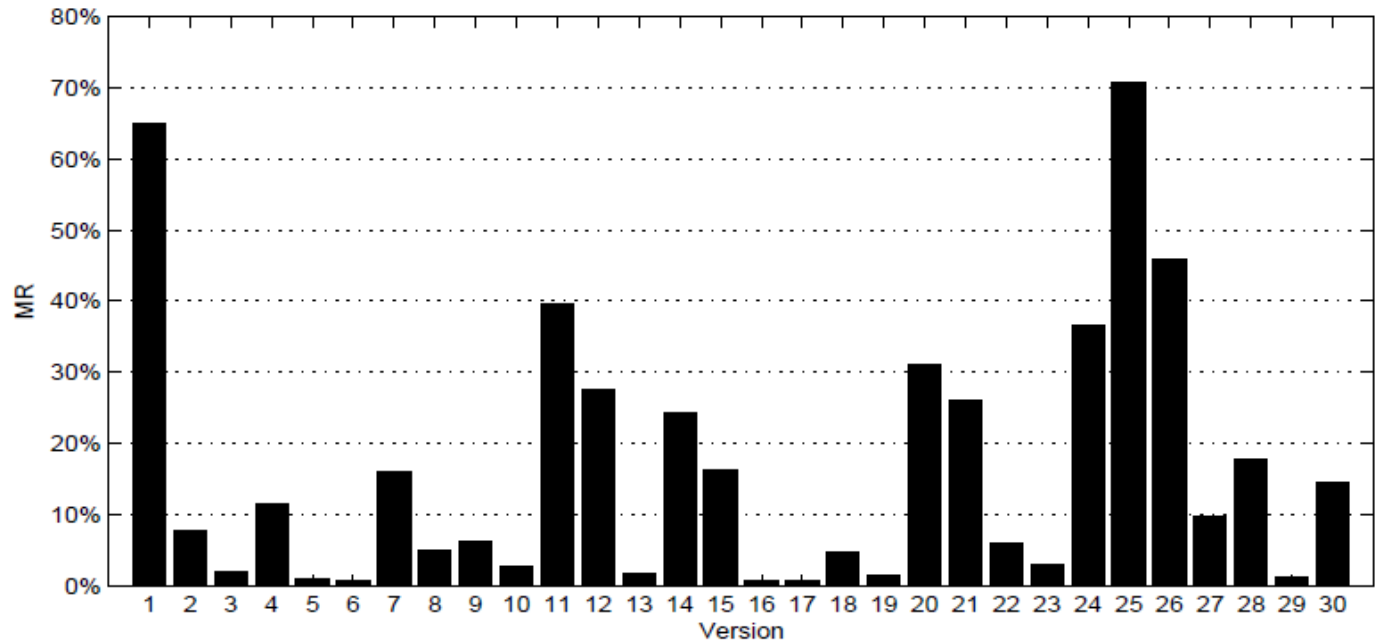


Fig. 5. MR of Block: Mean 16.56%, Max 70.82%, Min 0.57%



Effectiveness Evaluation

F-measure

$$F = \frac{2 \times recall \times precision}{recall + precision}$$



半监督学习动机

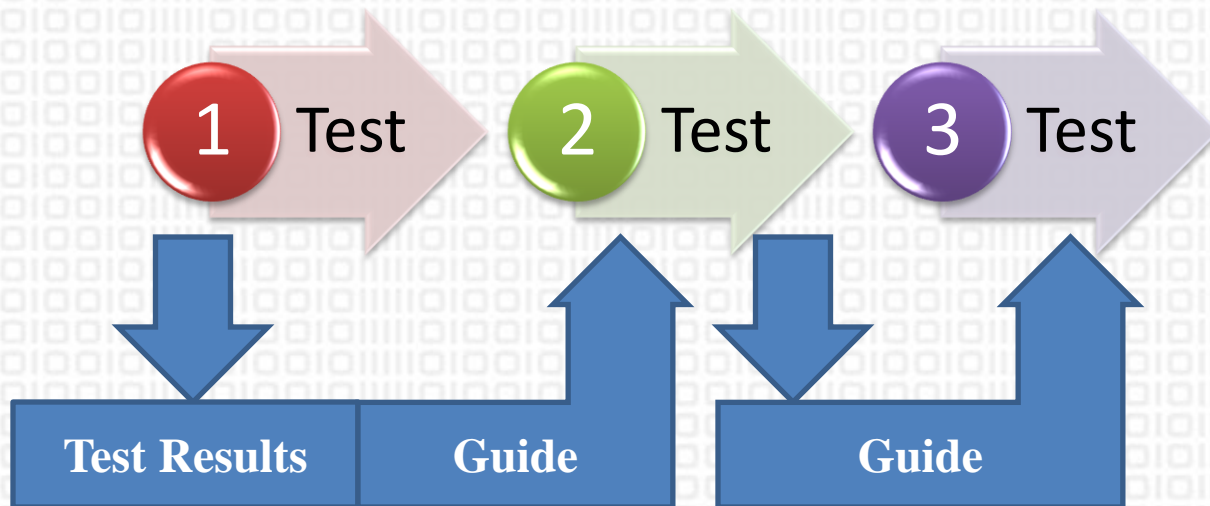
- Distance
 - $D(t1, t2)=4$
 - $D(t1, t3)=3$
 - $D(t1, t2) > D(t1, t3)$
- But, actually:
 - $t1$ and $t2$ triggered by a same fault.
 - $t3$ is a passing test.
 - $D(t1, t2) < D(t1, t3)$
- What should we do?
 - Transformation

	t1	t2	t3	t4
f1	1	0	1	0
f2	1	1	0	1
f3	0	1	0	1
f4	1	1	1	0
f5	1	0	1	1
f6	0	0	1	1
f7	1	1	0	1



半监督学习动机

- Regression testing is always a step by step process.



- How to use the information in previous step to improve the cluster and selection in next step?



聚类

- Clustering is a grouping process to achieve
 - maximize similarity in the same clusters (Internal)
 - maximize dissimilarity between clusters (External)
- K-means (Weka)

$$J_{kmeans} = \sum_{j=1}^k \sum_{x_i \in C_j} || x_i - \mu_j ||^2$$



基于约束的半监督聚类

- Use pair-wise constraints to label partial data.
 - Must-Link:
 - Cannot –Link:





基于约束的半监督聚类

- Use pair-wise constraints to label partial data.
 - Must-Link: two data items must be in a same cluster.
 - Tests triggered by some same faults.
 - Tests triggered by some similar faults.
 - Cannot –Link: two data items cannot be in a same cluster.
 - Passing tests and failing tests.
 - Tests triggered by different faults.



半监督 K-means

- x_i is a test, represented by feature vector.
 - For example, $x_i = (0, 1, 1, 0, 0, 1)$
- w is a weight matrix for transformation.
- y_i is a test transformed from x_i by w .
 - $y_i = w^T x_i$
- Find a w to max the objective function

$$J(w) = \frac{1}{2n_C} \sum_{(x_i, x_j) \in C} (y_i - y_j)^2 - \frac{\beta}{2n_M} \sum_{(x_i, x_j) \in M} (y_i - y_j)^2$$



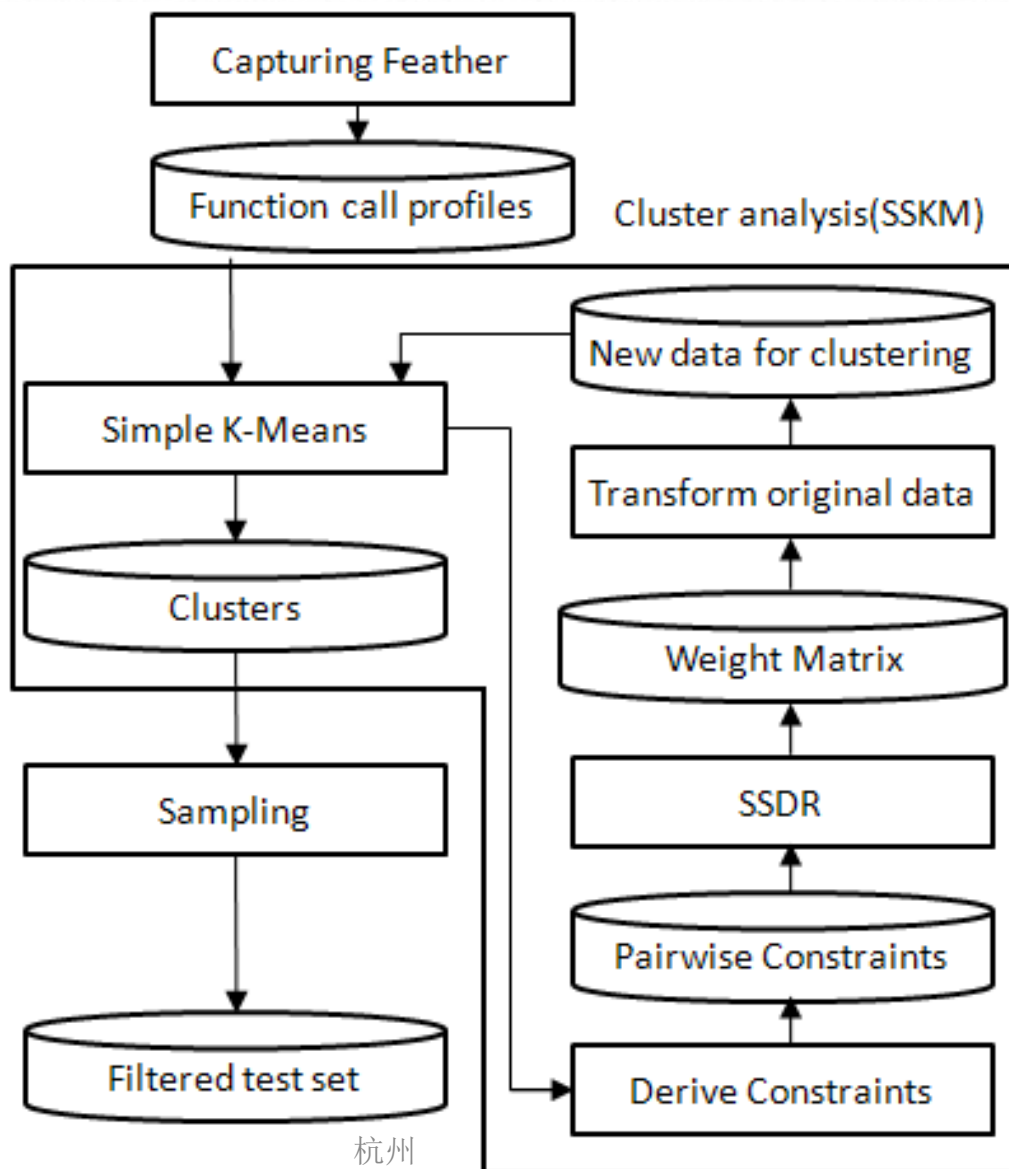
半监督 K-means

- To use the information of many unlabeled data, we use the following extended objective function

$$\begin{aligned} J(w) = & \frac{1}{2n^2} \sum_{i,j} (w^T x_i - w^T x_j)^2 \\ & + \frac{\alpha}{2n_C} \sum_{(x_i, x_j) \in C} (w^T x_i - w^T x_j)^2 \\ & - \frac{\beta}{2n_M} \sum_{(x_i, x_j) \in M} (w^T x_i - w^T x_j)^2 \end{aligned}$$



方法框架



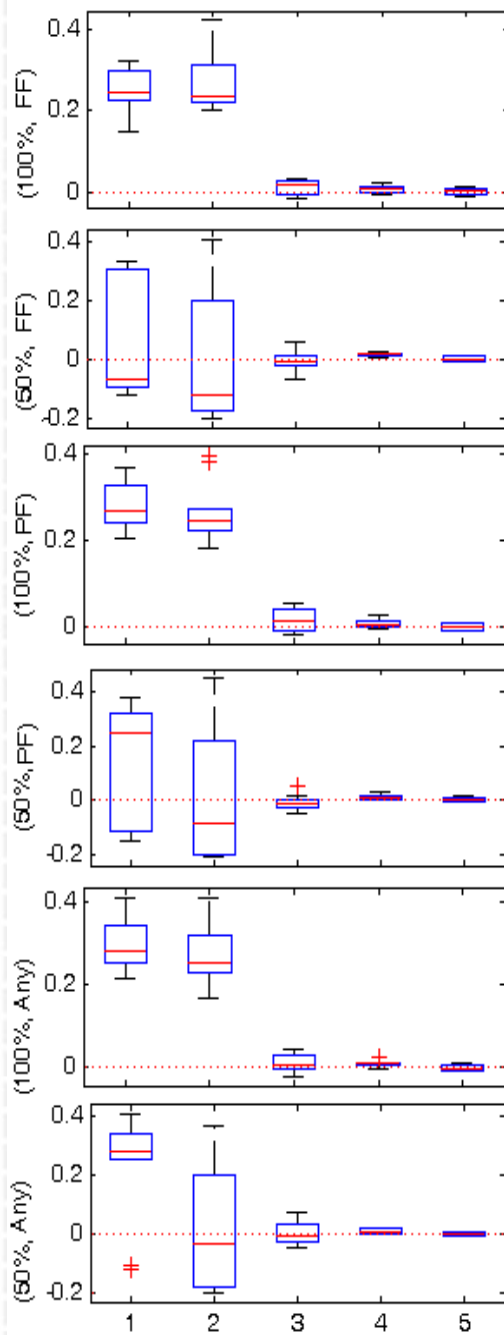


实验设计

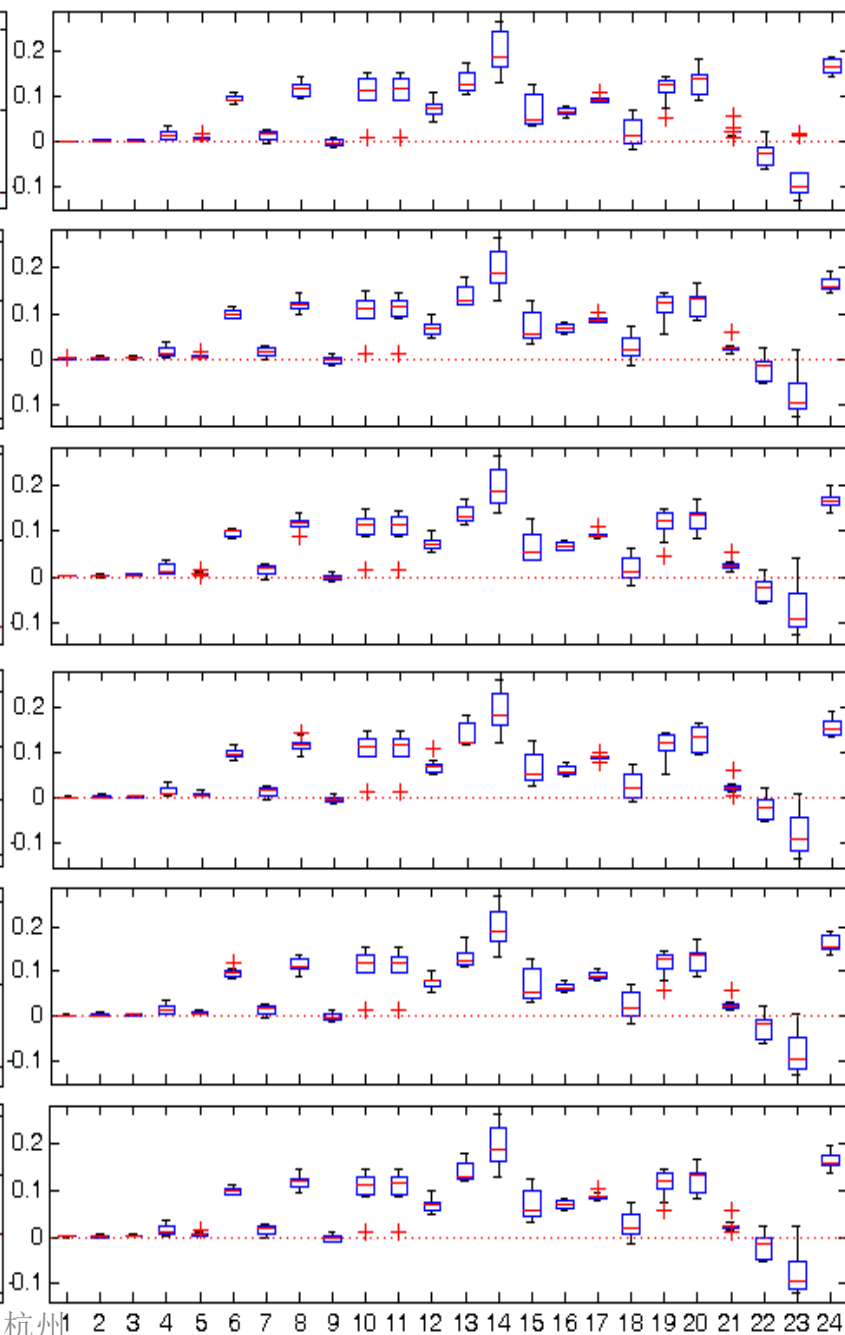
- K-means clustering
- Select randomly tests from each cluster until 10 tests are fault-revealing tests.
- Get pair-wise constraints
 - Must-link
 - Cannot-Link
- Transformation w (*D. Zhang SDM'05*)
- Loop: next K-means clustering.



Flex



Space





Multi-label Learning

- Why do we need multi-label learning?

Disadvantage of Single Label Learning:

- 1、 may produce incomplete learning results
- 2、 single label learning in a multi-label task may damage the training process

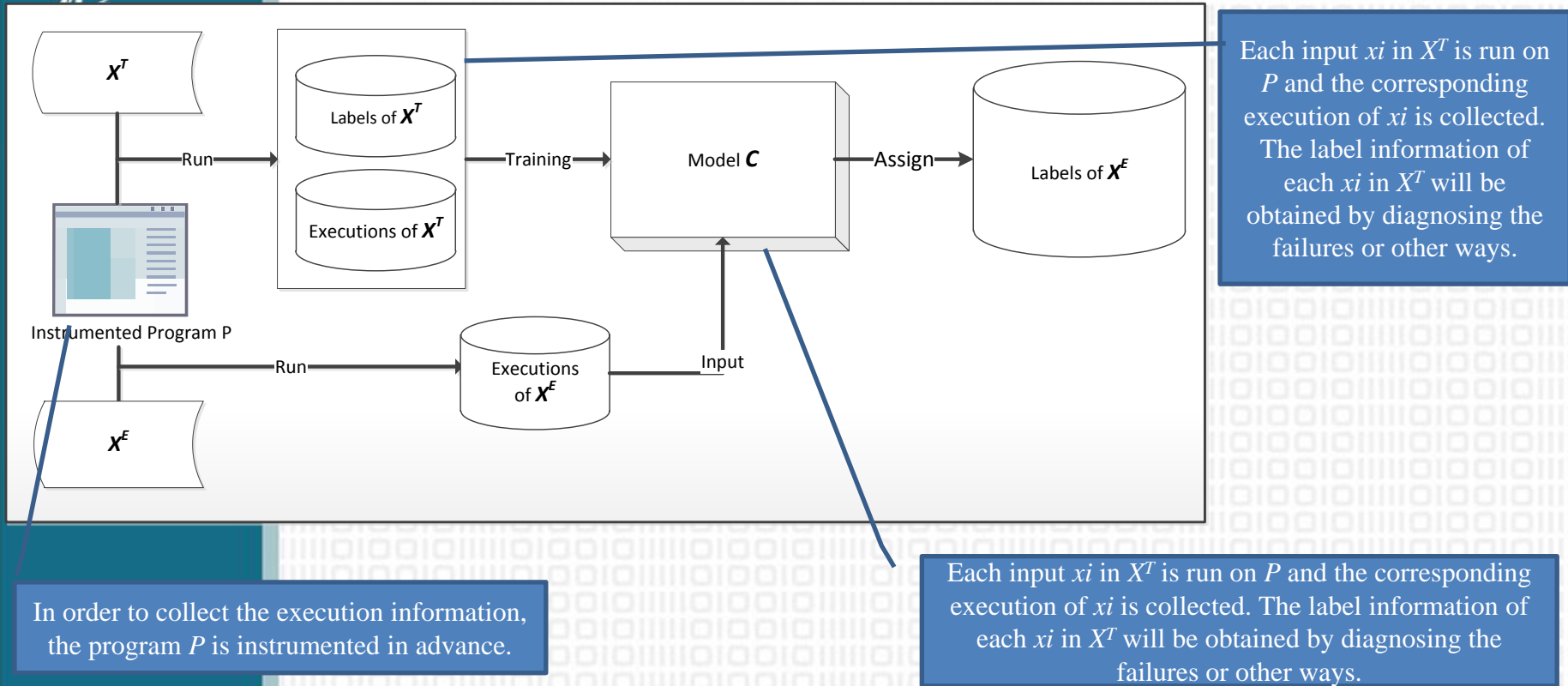
Yang Feng and Zhenyu Chen. Multi-label Software Behavior Learning. ICSE 2012 NIER track. (17.7%)





NAN

多标签行为学习框架



In single-label software behavior learning, X^T with its single-label information will be used to train C_S . multi-label data are omnipresent in real applications. Hence, it usually randomly select one label to train C_S . For a new input x , C_S outputs one label for x . Multi-label learning is a generalization of single-label learning. It uses multi-label data to train C_M , which can output one or more labels for new x .



实验结果1

ID	F1	F2	F3	F4	G1	G2	G3
Ver	2.5.1	2.5.2	2.5.3	2.5.4	2.2	2.3	2.4
L	14	9	11	5	7	4	9
X	552	237	542	560	781	158	784
NM	544	113	542	559	324	29	675
AL	6.68	1.98	5.22	4.89	1.42	1.18	3.54

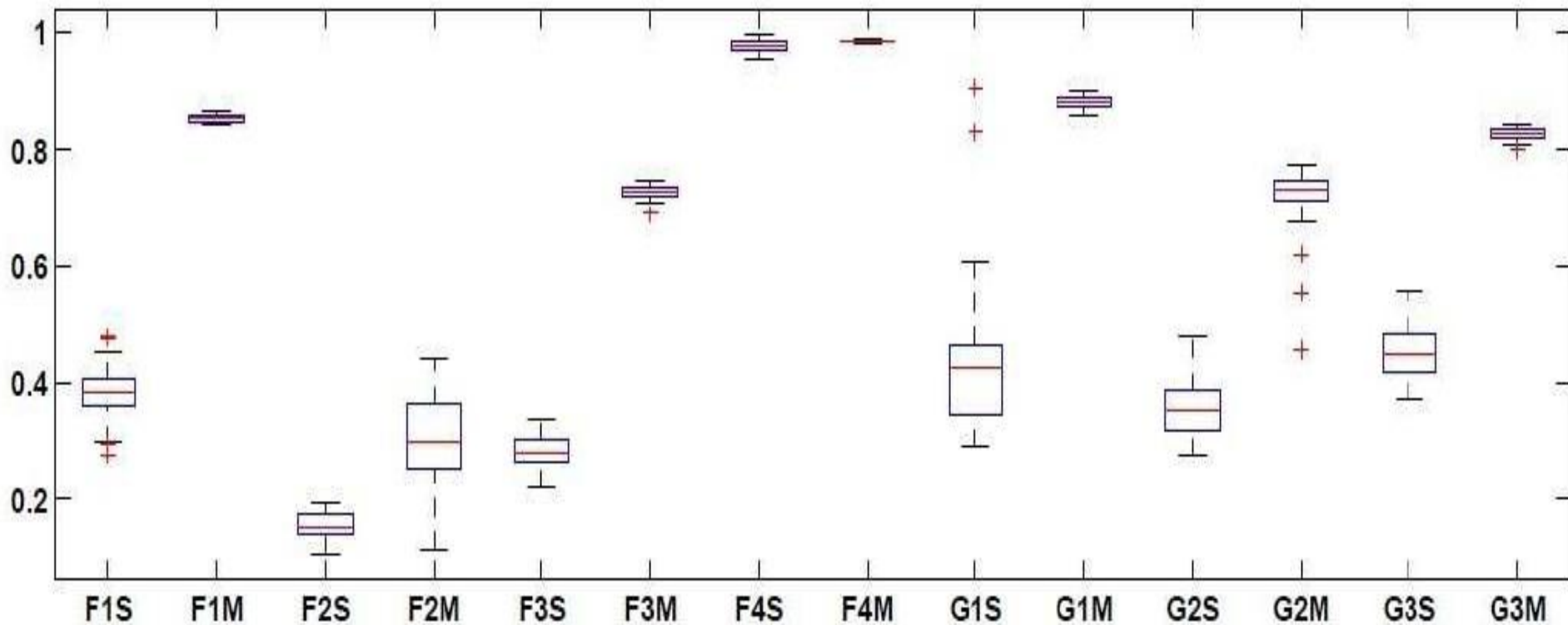
Note:

- 1、“F*” and “G*” denote the version of Flex and Grep
- 2、“Ver” is the version number
- 3、L and X denote the numbers of faults and the number of all failing inputs, respectively
- 4、“NM” denotes the number of multi-label failing inputs
- 5、“AL” denotes the average of L



NANJING UN
南京

实验结果2





工业应用？

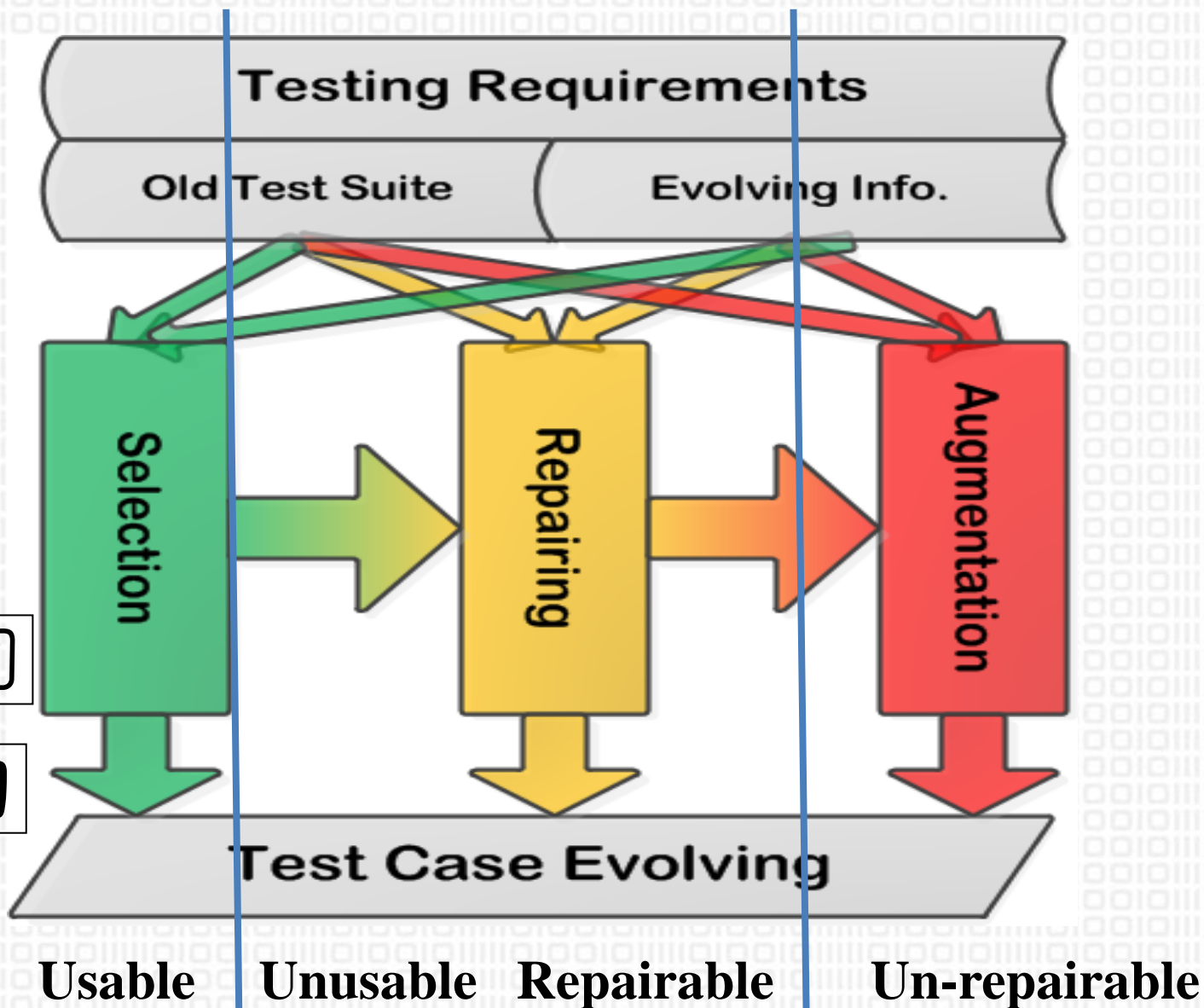
- 基于软件行为聚类分析的测试用例选择技术
 - 获得了基金支持
 - 培养了一批学生
 - 初步的工业应用
 - 建立了研究声誉
- 逻辑测试/测试用例选择
 - 博士后工作的延续
- 电子商务精准营销分析
 - 工业应用尝试



2010年启动

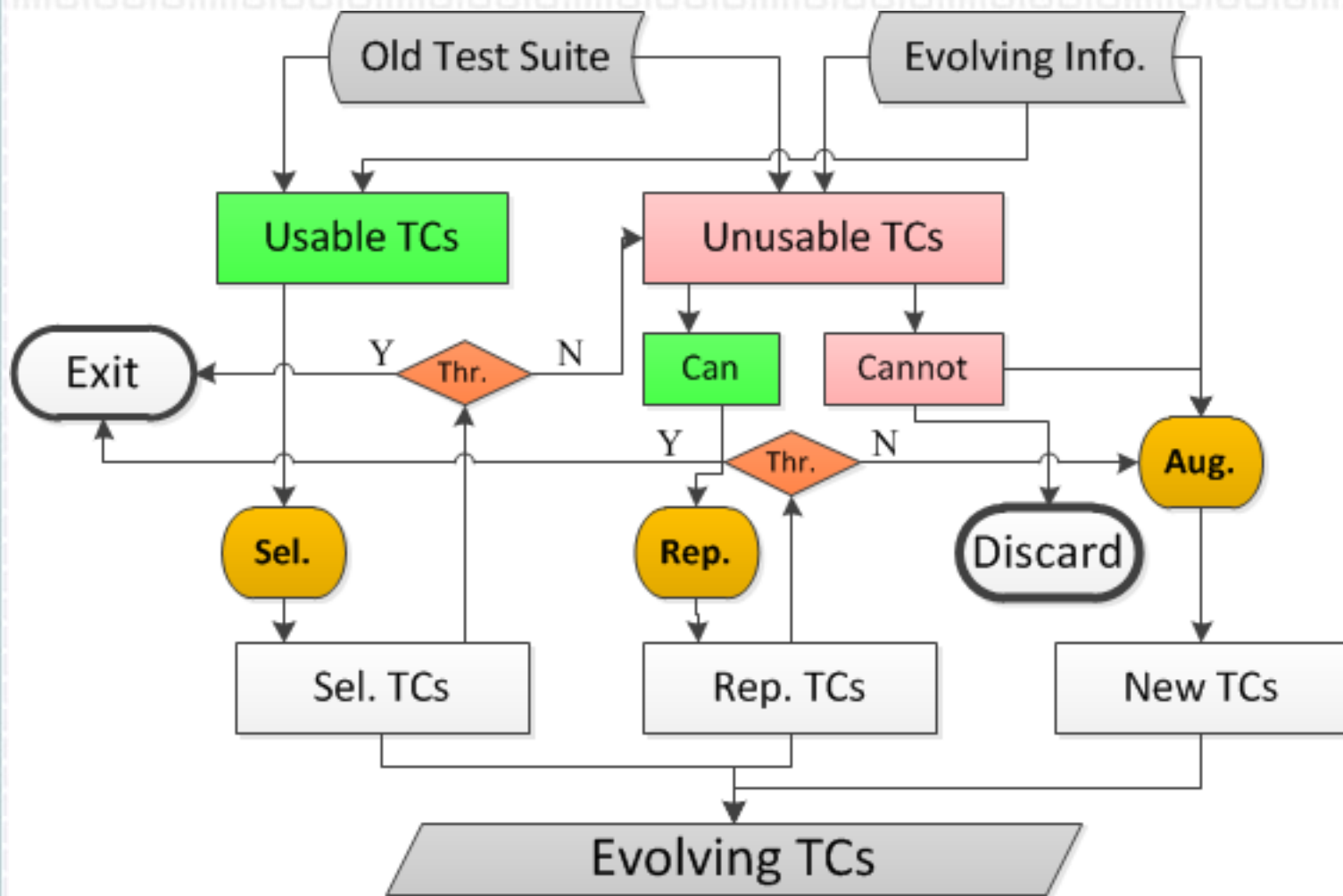
2011年资助

测试用例演化





测试用例演化框架





在线测试系统

- 自动化+半自动化
- 全流程测试服务
 - 测试生成（半自动化）
 - 测试执行（自动化）
 - 测试选择（自动化）
 - 测试修复（半自动化）
 - 测试扩增（半自动化）
- Web应用+移动应用

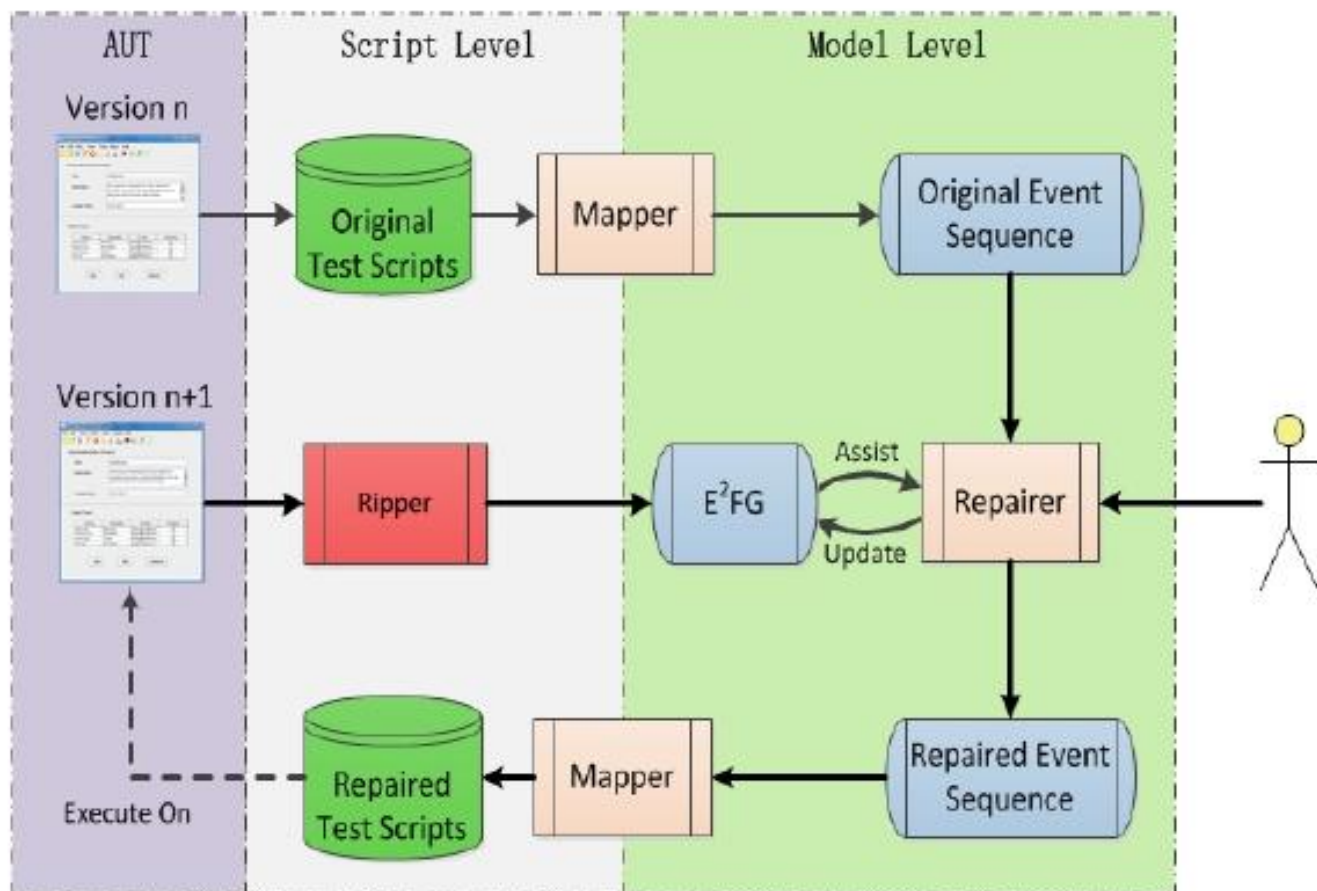


测试用例修复的挑战

- 挑战1： 不完全模型
 - 代码常常不能获取，黑盒测试。
 - 由于逆向工程技术的限制，我们难以取得所有的对象及其关系。
- 挑战2： 对象异构表示
 - 测试脚本/测试模型中相同对象的表示方法不同。



CAR: 测试用例修复框架



CAR: Computer-Aided Repairing of GUI Test Scripts



测试脚本失效

Project Management System

File Edit View Team Tools About Help

General Information of Project:

Title: SwiftBuilder

Instruction: The project is developed for quick data form building. It provides convenient facilities such as drag-and-drop and auto-data-binding.

Created Time: 03.01.2012

Project Team:

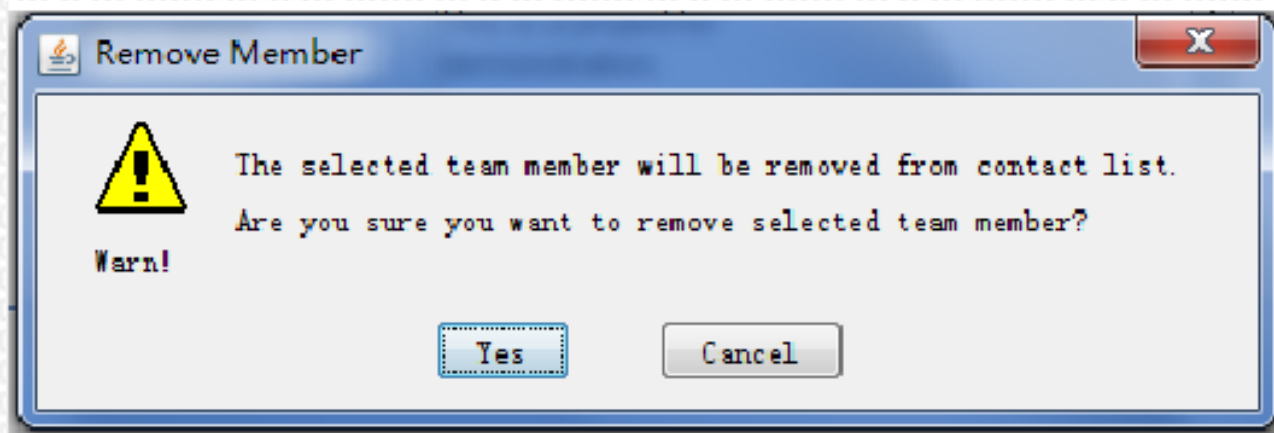
Name	Character	E-mail	Full-time
Salo Ding	Manager	qlding@software...	<input checked="" type="checkbox"/>
Daniel Gao	Developer	zbgao@software....	<input type="checkbox"/>
Jonh Ding	Tester	xtding@software....	<input checked="" type="checkbox"/>
Eric Gu	Developer	jgu@software.nj...	<input checked="" type="checkbox"/>

Add Edit Remove



自动修复

- | | |
|---|---|
| 1 | <code>Window("PMS").Button("Remove").Click</code> |
| 2 | <code>Window("PMS").Menu("File").Menu("Save project").Select</code> |



`Window("PMS").Dialog("Remove Member?").Button("OK").Click`



人工确认

Window("PMS").Field("Instruction").Set "A project for test."

Project Management System

File Edit View Team Tools About Help

General Information of Project:

Title: SwiftBuilder

Instruction: The project is developed for quick data form building. It provides convenient facilities such as drag-and-drop and auto-data-binding.

Created Time: 03.01.2012

Project Team:

Name	Character	E-mail	Full-time
Salo Ding	Manager	qlding@software...	<input checked="" type="checkbox"/>
Daniel Gao	Developer	zbgao@software....	<input type="checkbox"/>
Jonh Ding	Tester	xtding@software....	<input checked="" type="checkbox"/>
Eric Gu	Developer	jgu@software.nj...	<input checked="" type="checkbox"/>

Add Edit Remove

from: *Window("PMS").Field("Instruction")*

to: *Window("PMS").Field("Description")*.



人工添加

1	<code>Window("PMS").Button("Add").Click</code>
2	<code>Window("PMS").Dialog("Add Member").Field("Name").Set "N"</code>
3	<code>Window("PMS").Dialog("Add Member").Field("Role").Set "Tester"</code>
4	<code>Window("PMS").Dialog("Add Member").Button("OK").Click</code>

- 新版本中添加成员类型
- Member Type

`Window("PMS").Dialog("Member Type").Field("Type").Set "1"`



知识积累

1	<i>Window("PMS").Menu("File").Menu("Create project").Select</i>
2	<i>Window("PMS").Field("Title").Set "ProjTest"</i>
3	<i>Window("PMS").Field("Instruction").Set "To test creating project."</i>
4	<i>Window("PMS").Button("Add").Click</i>
5	<i>Window("PMS").Dialog("Add Member").Field("Name").Set "M"</i>
6	<i>Window("PMS").Dialog("Add Member").Field("Role").Set "Manager"</i>
7	<i>Window("PMS").Dialog("Add Member").Button("OK").Click</i>

Thanks to the knowledge of "Field('Description')", CAR could automatically change line 3 in Table 3 to a correct statement.

Window("PMS").Field("Description").Set "To test creating project."

Thanks to the knowledge of "Dialog('Member Type')", CAR could automatically add the following statement between line 4 and 5 in Table 3.

Window("PMS").Dialog("Member Type").Field("Type").Set "1"



实验设计

- Q1: 测试脚本失效比例
- Q2: 自动化修复的成功率
- Q3: CAR的修复效果
- Q4: CAR的修复成本

Table 4: Basic Information

Applications	<i>TSs</i> in Pool	Selected <i>TSs</i>	All Widgets	Modified Widgets		Unusable <i>TSs</i>	
				Changed	Deleted	#	%
Crossword Sage	1070	105	80	5	4	88	83.8
PDFsam	1025	148	184	2	0	148	100
OmegaT	1297	129	519	21	0	129	100



事件模型是否完整

Table 5: Completeness Degree of Initial Model

Application	Total Events #	Recognized Events #	Percent
Crossword Sage	3931	475	12%
PDFsam	4456	2908	65%
OmegaT	4656	1619	35%



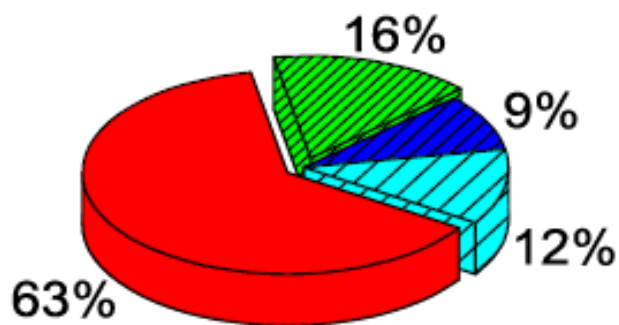
CAR的修复成功率

Table 6: Repairing Results

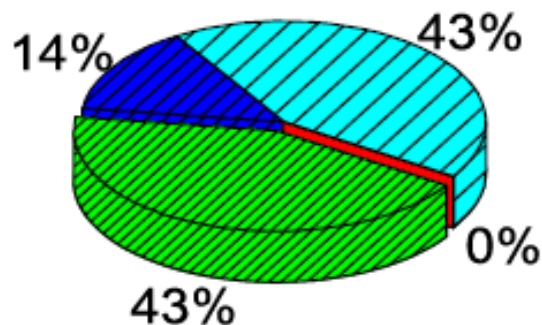
Application	Unusable <i>TSs</i> #	Repaired <i>TSs</i> #	Ratio
Crossword Sage	88	45	51%
PDFsam	148	133	90%
OmegaT	129	106	82%



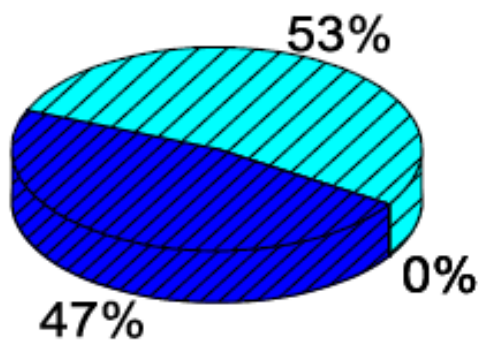
失败原因分析



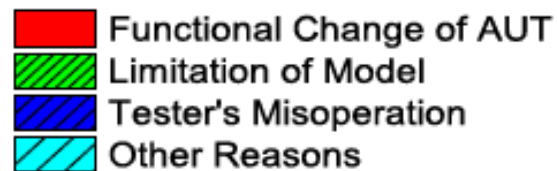
CrosswordSage



OmegaT



PDFsam





CAR的修复成本

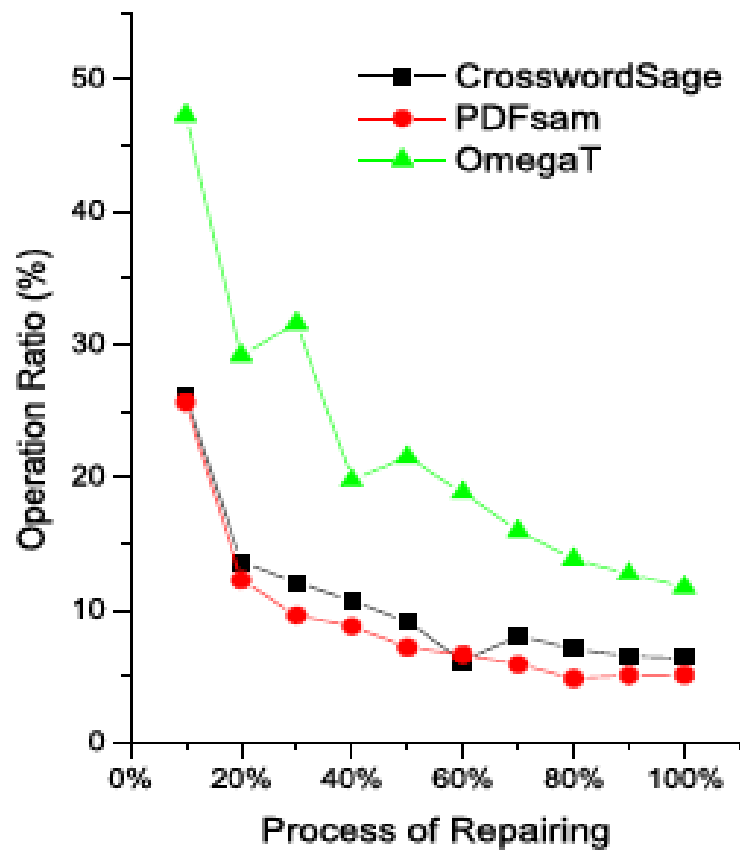
Table 7: Time Cost Effectiveness (minutes)

Application	Repairing	Creation	Ratio
CrossowrdSage	29.4	1317	2.2%
PDFsam	29.0	1342	2.2%
OmegaT	71.7	1415	5.1%

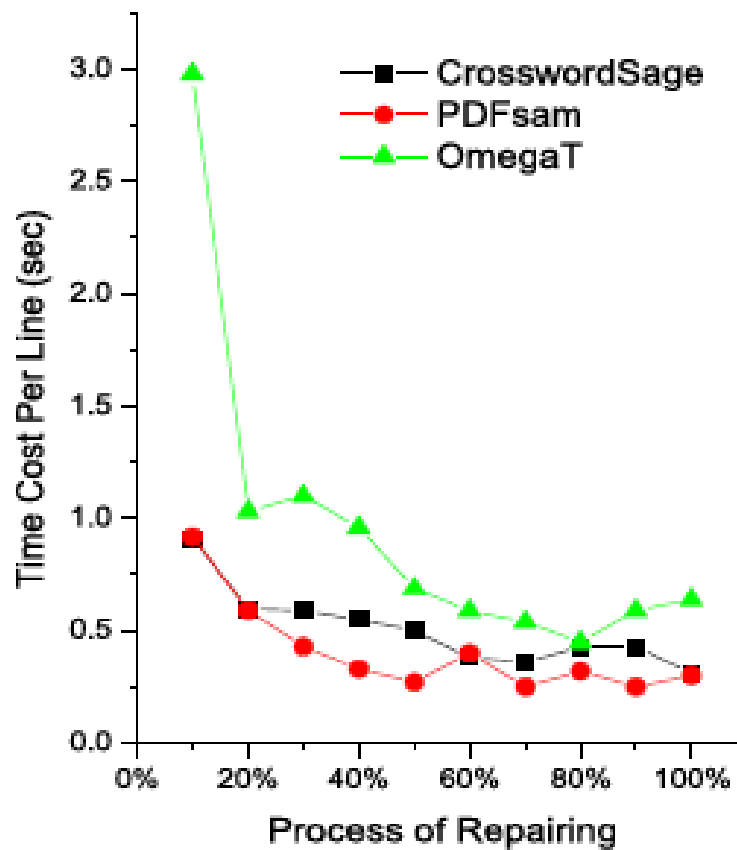


NANJING
UNIVERSITY OF AERONAUTICS
AND ASTRONAUTICS

知识积累效果



(a) Manual Operation Cost



(b) Time Cost Per Line



Crowdsourcing (众包)

- The act of outsourcing tasks, traditionally performed by an employee or contractor, to an undefined, large group of people or community (a "crowd"), through an open call.
- Problem solving, innovation and design
(e.g. innocentive.com, yourencore.com)
- Content and image contributing
(e.g. Wikipedia, iStockphoto)
- Remote freelance (general) staffing
(e.g. oDesk.com, mturk.com)



众测

- 传统测试
- 人工测试
- 自动化测试
- 适合众测的类型？



Crowdsourced Testing

- Access to diverse platforms, languages, and people (demographically and geographically)
- Real insights from the real world, not just made up test case results
- Testing done by hundreds of people at the same time
- Rapid feedback right away
- Beta testing
- Usability testing
- Small companies, start-ups that don't have and can't afford their own testing resources



众测任务

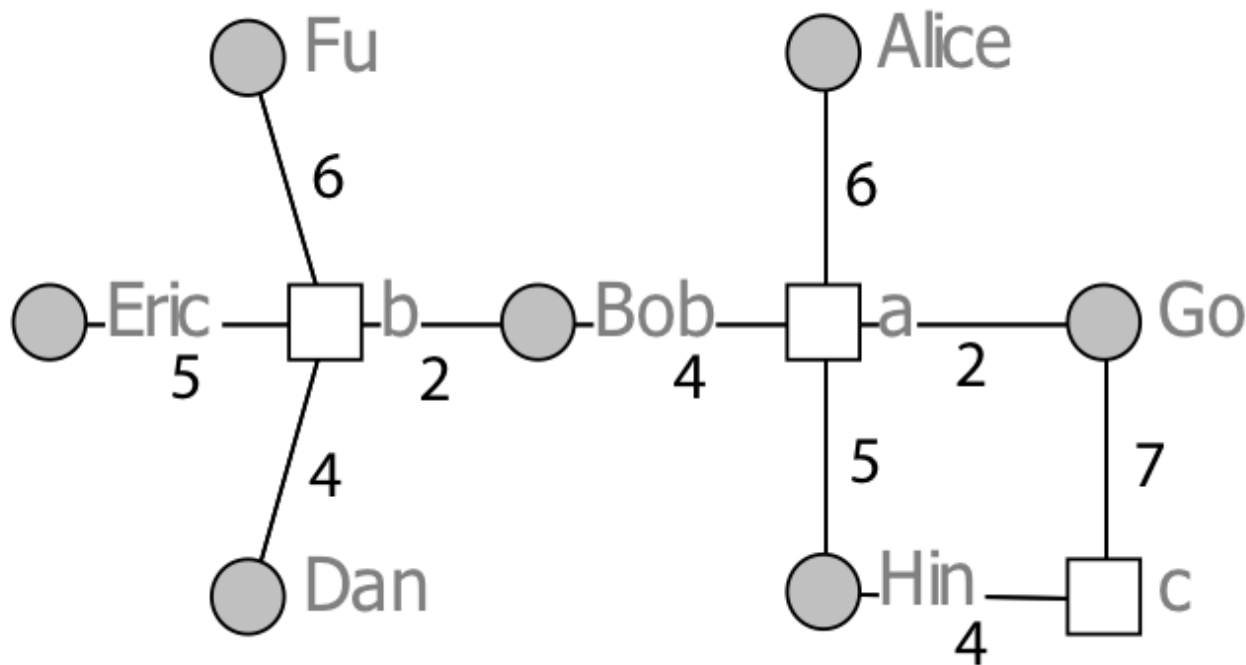
- Android端拼音输入法
 - 功能测试
- 百度浏览器
 - 功能测试与易用性测试
- 百度影音播放器
 - 性能测试与易用性测试



开发者社交网络分析

开发者的关系和软件缺陷/质量是否存在某种联系？

□ Windows binary (*.dll)
○ Developer





开发者社交网络分析

- 如何建立开发者社交网络？
- 如何分析开发者社交网络的演化？
- 开发者社交网络的特性？
- 开发者社交网络对软件维护的应用
 - 软件维护组织
 - 缺陷预测
 - Bug报告分配
 - 测试用例选择



当前研究课题

- 测试用例演化
 - 在线测试系统
 - 学术研究+工业应用
- 开发者社交网络分析
 - 学术研究
- 面向行业的数据分析
 - 工业应用



团队成员

- 博士生
 - 张智轶(11)、房春荣(12)、何铁科(12)、汪亚斌(13)、冯洋(13)
 - 时清凯(14)、张伟强(14)
 - 苏超(15)
- 硕士生
 - 高则宝、邹云潇、都兴中、惠成峰、魏帅、沈毅、孙一、豆梦宇
 - 冯奕彬、卢依宁、钱程、
- 本科生



NANJING UNIVERSITY · SOFTWARE INSTITUTE
南京大学软件学院



Thanks