Test Set Diameter: Quantifying the Diversity of Sets of Test Cases

2016 IEEE

作者: Robert Feldt, Simon Poulding,David Clark, Shin Yoo

主讲人：成子谦、陈逸超

Abstract

写作出发点

- 许多已有的研究正在探讨测试用例的输入/输出有着怎样的不同。
- 共同缺陷：它们对特定数据类型或成对的输入/输出进行研究。
- 难以统一度量所有的测试集。

新标准：测试集直径(TSDm)

- 特点：基于信息理论中有关多集归一化压缩距离(NCD)计算的最新进展而提出。

- 优点：TSDm是通用的多样性度量，可应用于任何测试集，而与测试输入的数据类型无关。

- 代价：更大的计算量。

Introduction

基本问题

- 如何选择一小组最有效的能用于测试软件系统的测试用例。

已有方法

- 自适应随机测试(传统做法)。
- 基于Kolmogorov复杂度的信息距离度量法。
- 例子：Cilibrasi和Vitanyi的归一化压缩距离(NCD)。

新观点：扩展到multiset

- 把NCD扩展到多重集合(multiset)，提供了一种通用方法来解决测试用例选择问题。
- 优势：这种基于理论动机的、基于集合多样性的测试选择方法可以创建比启发式方法更有效的测试集。

Basic theory

Kolmogorov复杂度

- 定义1——Kolmogorov复杂度：对于符号串S，该串的Kolmogorov复杂度指的是能输出符号串S的最短程序的长度(以bit为单位)，表示为K(S)。

- 定义2——条件Kolmogorov复杂度：对于给定输入符号串y，能输出符号串x的最短程序长度，记作K(x | y)。

信息距离(ID)

- Bennett等人使用条件Kolmogorov复杂度来衡量信息距离：对于x和y两个符号串，信息距离ID的计算方式如下：

$$\mathrm{ID}(x,y) = \max\{K(x|y), K(y|x)\}$$

归一化信息距离(NID)

- 为了能够比较不同长度符号串的相似性，Li等人提出了归一化信息距离NID的计算方式如下：

$$\mathbf{NID}(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}}$$

- NID取值范围为[0,1]。

NID的优势

- 主要优势：保留了度量的特性。
- NID(x, x) = 0
- NID(x, y) + NID(y, z) >= NID(x, z)
- NID(x, y) = NID(y, x)

归一化压缩距离(NCD)

- 基于如下事实：当使用实际压缩程序压缩符号串时，输出文件的大小很好地近似了该符号串的Kolmogorov复杂度。

NCD定义

- 定义3——压缩符号串S后所得到的长度，记为C(S)。
- 定义4——归一化压缩距离NCD的计算方式如下：

$$\mathbf{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}}$$

- NCD取值范围[0, 1+ε]，其中ε用于刻画压缩程序对Kolmogorov复杂度的近似能力。

将NCD扩展到multiset

- 本文研究一组测试用例内部元素的相似性。Cohen和Vitányi把NCD的概念扩展到multiset。

- 定义5——对于一个多重集合X，NCD的计算方法为：

$$\mathrm{NCD}_1(X) = \frac{C(X) - \min_{x \in X}\{C(x)\}}{\max_{x \in X}\{C(X \setminus \{x\})\}}$$

$$\mathrm{NCD}(X) = \max\left\{\mathrm{NCD}_1(X), \max_{Y \subset X}\{\mathrm{NCD}(Y)\}\right\}$$

将NCD扩展到multiset

- 通过将一个字符串的集合的NCD定义为0，可以使得两个多重集合的NCD与任意一对字符串的NCD相同。
- 该NCD的计算方法时间复杂度为O(2^n)，难以实现。

改进算法

- Cohen和Vitányi从多重集合X={x1,x2,...,xn}开始并进行如下操作：

1. 查找使得$C(Y_k \backslash \{x_i\})$最大的下标$i$
2. 令$Y_{k+1} = Y_k \backslash x_i$
3. 重复第一步直到子集中只包含两个元素
4. 计算NCD(X)=max{NCD1(Yk)} $0 \leq k \leq n-2$

- 时间复杂度：O(n^2)

Test set diameter

- 引入术语“测试集直径”(TSDm)表示上一部分中定义的NCD进行多重集合度量所测得的测试集的多样性。
- 定义这样的度量标准：输入集合的TSDm；输出集合的TSDm；跟踪TSDm。

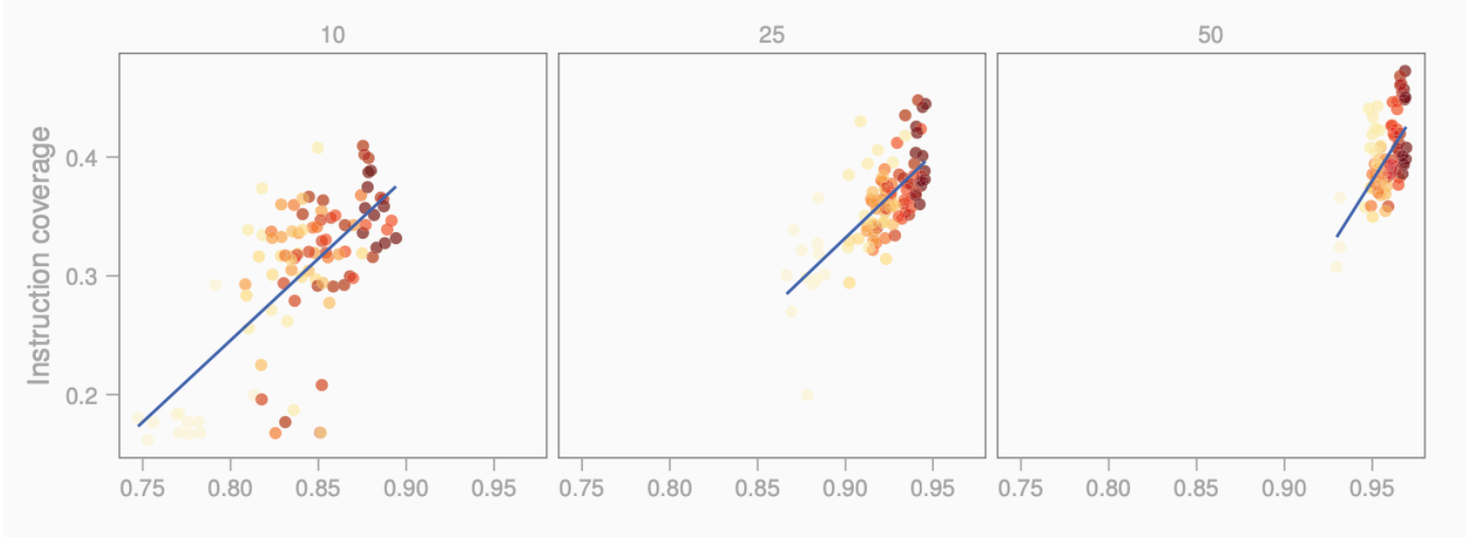
TSDm1

- 本文研究了用TSDm从更大的潜在测试用例池中选择指定大小的多样化测试集。
- 本文利用了Cohen和Vitányi提出的NCD1改进算法来对多重集合的NCD进行近似。
- 由于此算法使用NCD1度量来近似NCD，因此该过程称为TSDm1。

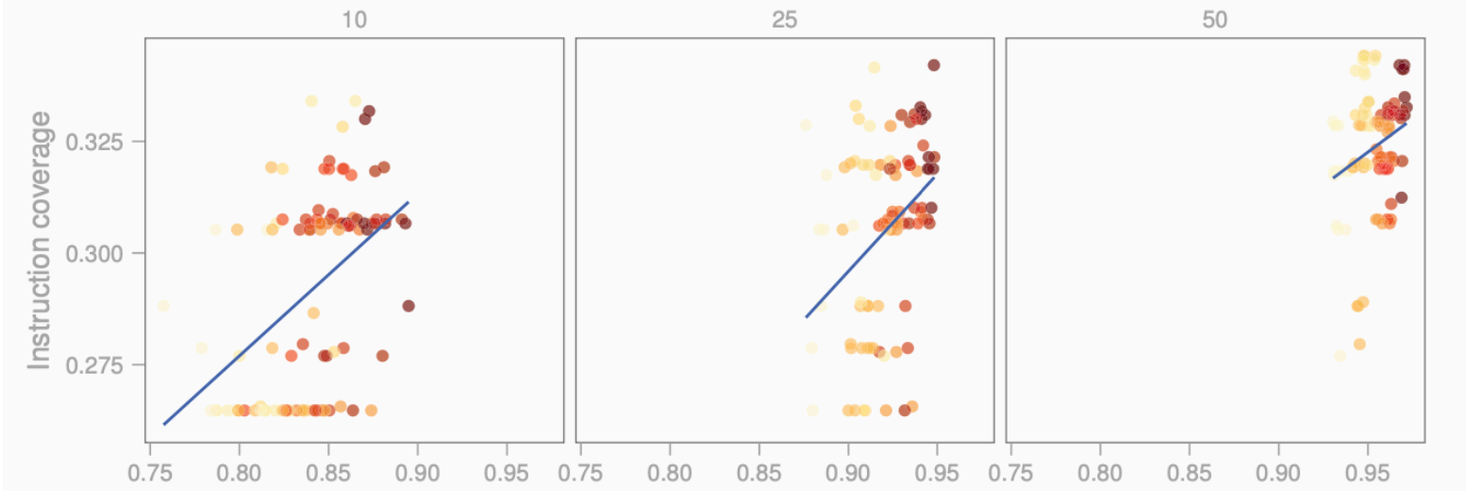
Empirical study

研究问题

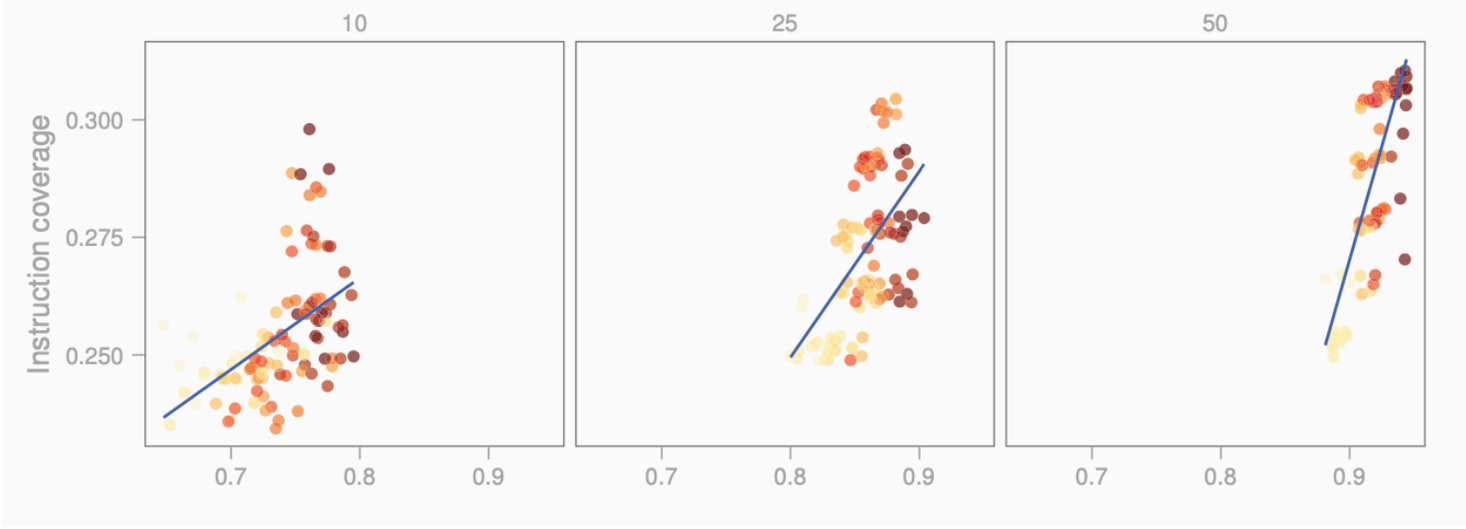
- 与代码覆盖率的关系
- 结构覆盖率
- 长度限制下的结构覆盖率
- 故障发现能力
- 测试集选择时间



(a) JEuclid



(b) NanoXML



(c) ROME

Fig. 1. Scatterplots showing the correlation between Input-TSDm (x axes) with instruction coverage (y axes) for test sets of three different sizes (10, 25, and 50 test inputs respectively) randomly sampled from 10 different strata. Test sets with darker colors are sampled from strata with test inputs that are selected by the I-TSDm selection procedure when the subset is the smallest.

与代码覆盖率的相关性

- 结论：平均而言，更高输入直径的测试集将具有更高的代码覆盖率。

结构覆盖率

- 与贪心算法比较。

$$\mathrm{NCD}_1(X) = \frac{C(X) - \min_{x \in X}\{C(x)\}}{\max_{x \in X}\{C(X \setminus \{x\})\}}$$

$$\mathrm{NCD}(X) = \max\left\{\mathrm{NCD}_1(X), \max_{Y \subset X}\{\mathrm{NCD}(Y)\}\right\}$$

- 与随机算法比较。

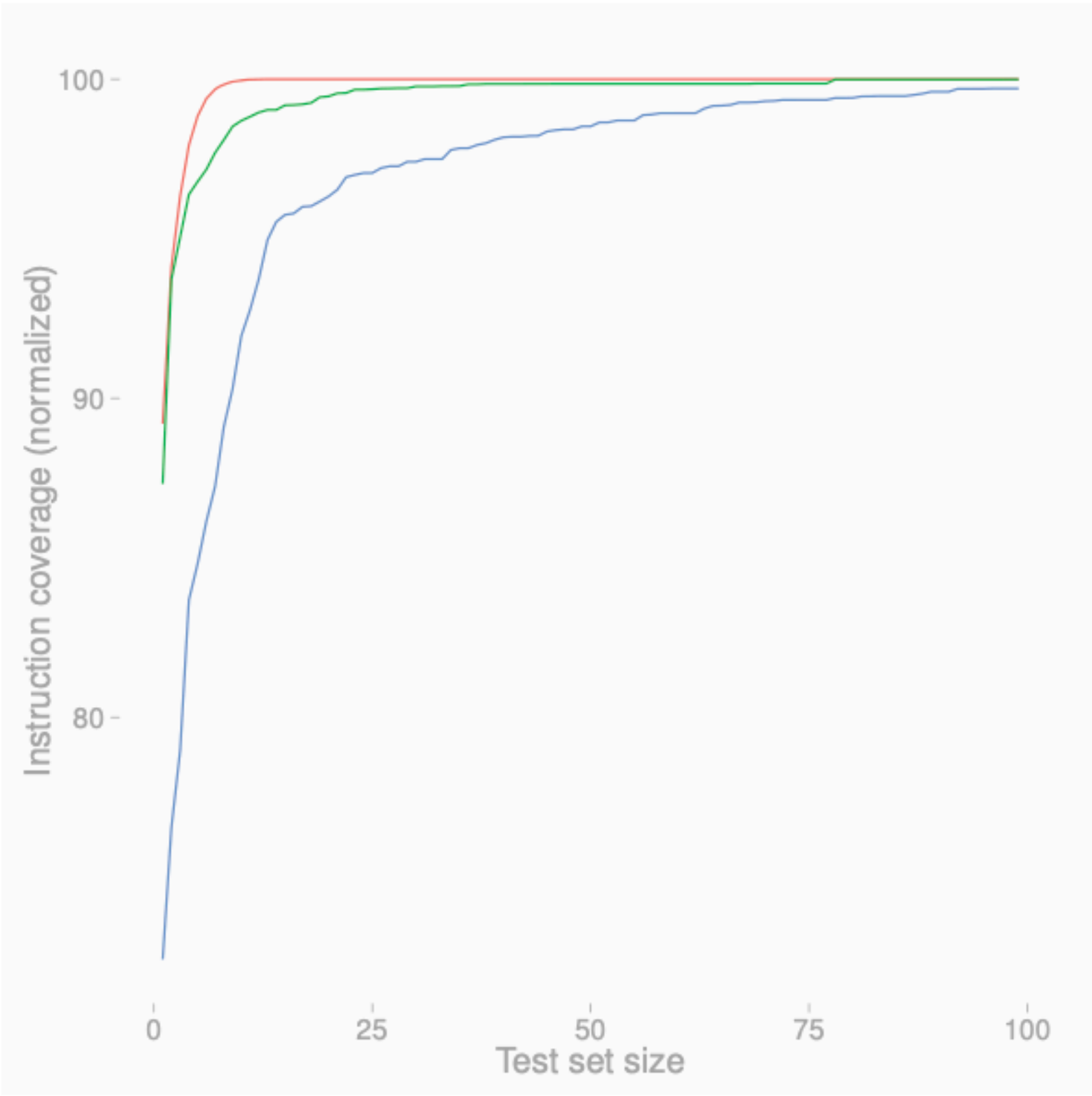


Fig. 2. Instruction coverage (normalized) of the ROME library against size for test sets selected using the greedy algorithm (red), I-TSDm$_1$ procedure (green), and random algorithm (blue) from an initial pool of 250 randomly-generated MathML inputs. The plots show the average values over 10 runs.

结构覆盖率

- 结论：由最大测试输入直径选择的测试集比随机选择的测试集具有更高的结构覆盖率。
- 风险：调查结构覆盖率时看到的变化仅是由于输入大小而不是给定大小的输入之间的差异所致。

长度限制下的结构覆盖率

- 生成大量输入，并且仅选择接近目标尺寸的那些。

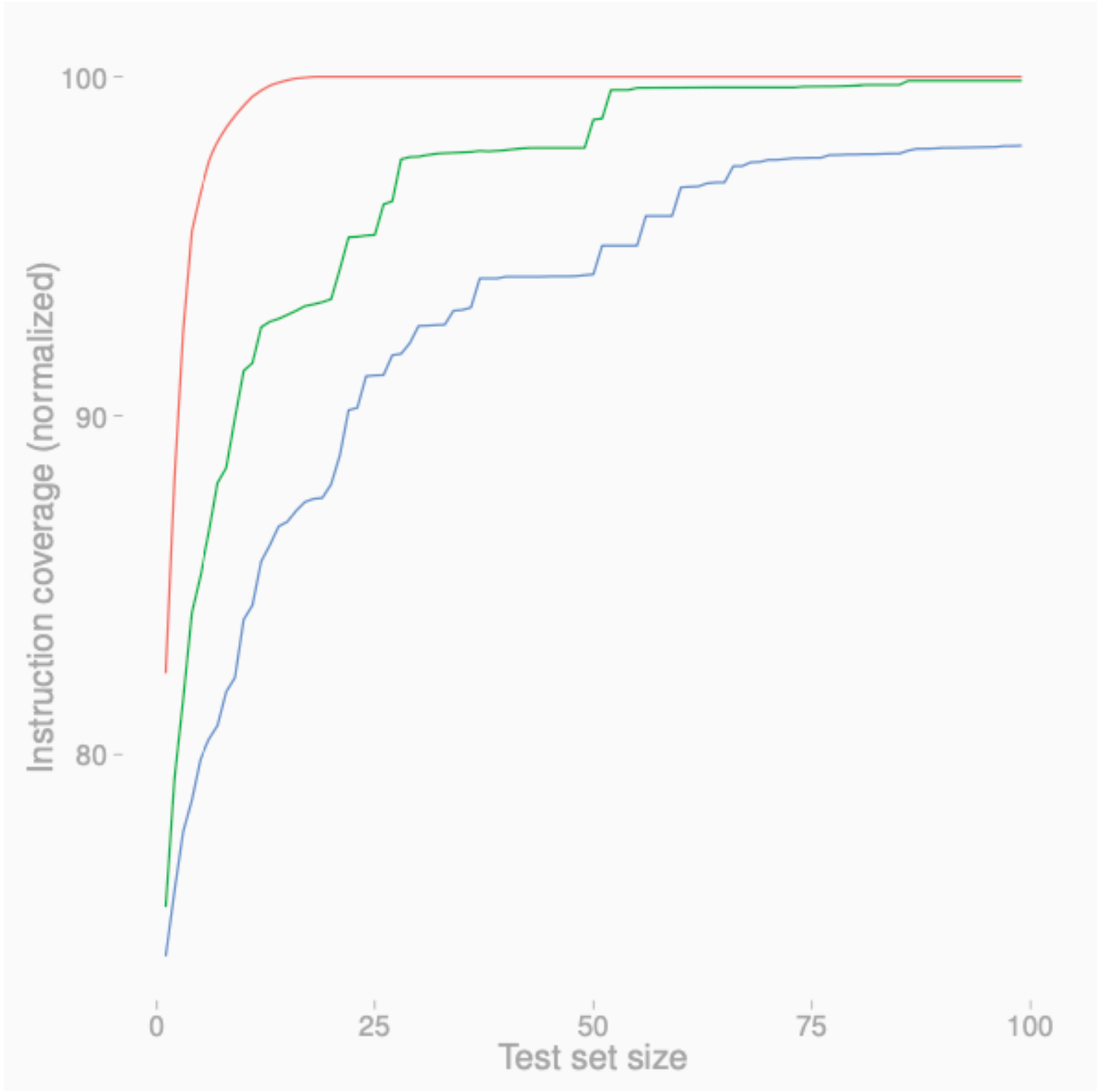


Fig. 4. Instruction coverage (normalized) of the ROME library against size for test sets selected using the greedy algorithm (red), I-TSDm$_1$ procedure (green), and random algorithm (blue) from an initial pool of 250 randomly-generated MathML inputs with lengths between 90 and 110 bytes. The plots show the average values over 10 runs.

长度限制下的结构覆盖率

- 结论：即使我们控制测试输入的大小，测试集直径也会得到更高的代码覆盖率。

故障发现能力

- 仅覆盖程序的较大部分是不够的。我们最终想要的是找到故障，以便我们消除故障。

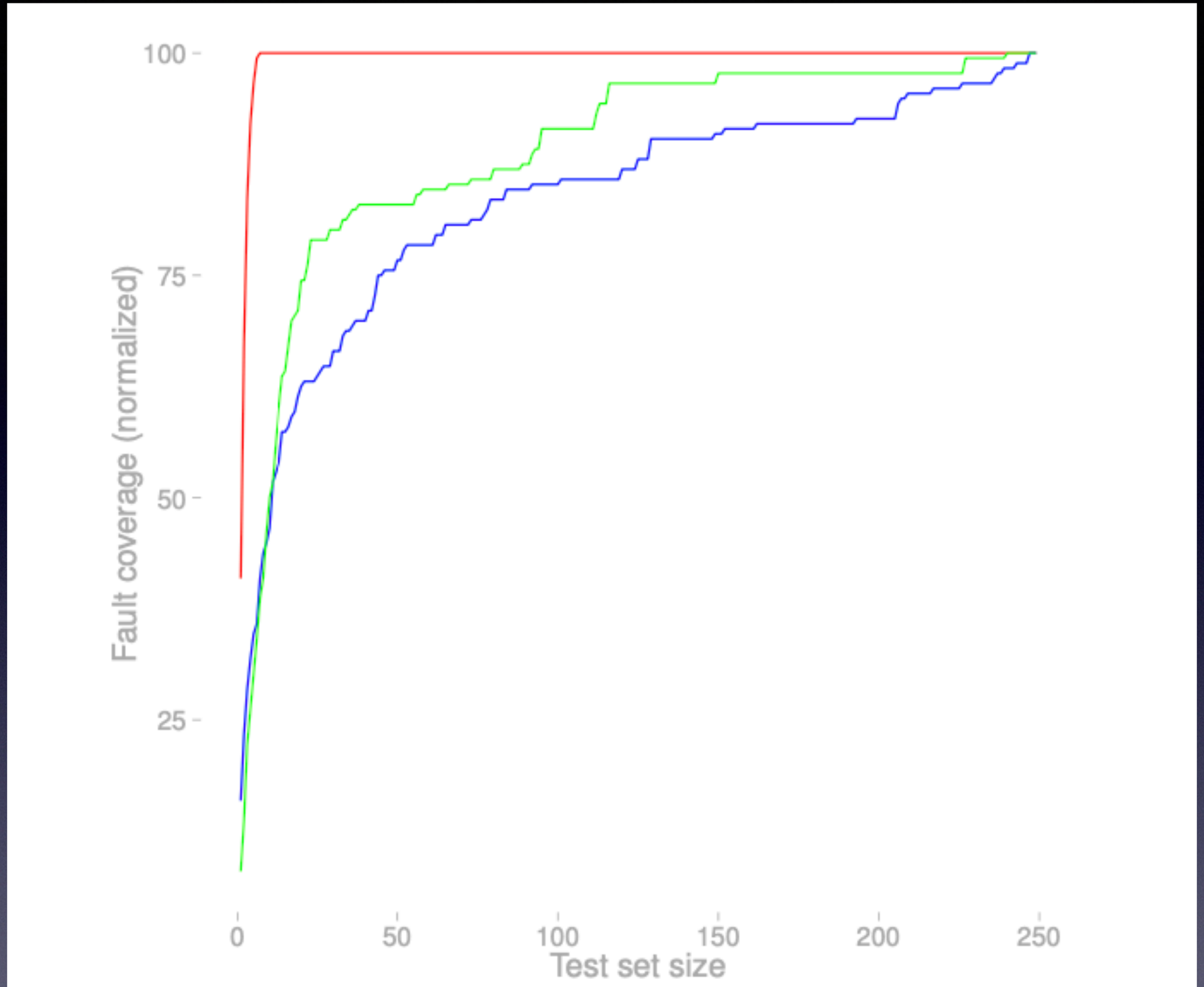


Fig. 5. Fault coverage (normalized) of the Replace SUT against size for test sets selected using the greedy algorithm (red), I-TSDm$_1$ procedure (green), and random algorithm (blue) from an initial pool of 250 randomly-generated inputs with the length of the regular expression between 9 and 11 bytes. The graphs are the average values over 10 repeated runs.

故障发现能力

- 结论：具有更大测试集直径（I-TSDm）的测试集可能具有更好的故障查找能力。

测试集选择时间

- TSDm测试选择程序在要选择的初始测试库大小中按平方缩放，并且与平均测试长度成线性关系。

Conclusions

- 本文提出了一种称为测试集直径（TSDm）的度量，可衡量整个测试集的多样性。
- 具有Kolmogorov复杂性的正式基础，适用于任何数据类型和与测试相关的信息源。
- 通过使用现代压缩算法对其进行近似计算，可以将其实际用于测试选择和分析。

Future

- 显然，TSDm可用于选择比随机选择的测试集更有效的测试集。
- 即便测试人员将继续手动选择测试用例，他们也可以使用TSDm搜索其他测试用例，这些用例最大程度地增加了整个测试集的直径。
- 甚至可以使用TSDm分析现有的测试套件。

感想1

- 本文大胆地提出了测试集直径、量化测试用例集的多样性，并使用大量的实验证明它的可用性，使我在课堂外见识到了行业内更新的研究成果，增加了我对软件测试这个领域的认识。

感想2

- 这篇文章从软件测试的一个基本问题出发，基于已有的工作，对测试用例的异同提出了一个新的统一衡量指标，充分地描述了测试用例的多样性，开阔了我对测试用例异同的理解。个人认为这是软件测试的一个很好的研究方向，有助于建立行业标准，推进行业发展。但由于该研究目前尚处于初级阶段，距离实用化还有很长的一段空间。

Thanks for listening