

图中, X 轴表示不同的测试用例约减方法, Y 轴表示对应方法的 EXAM Score, 子图中每一个类似小提琴的部分都表示一种测试用例约减方法在上述实验程序上的错误定位精度分布, 其中, 外部形状为核密度估计, 内部白色的点是中位数, 黑色盒型的范围是下四分位点到上四分位点. 对应方法的错误定位精度分布在 Y 轴较低位置的宽度越宽, 较高位置的宽度越窄, 表示对应的方法定位错误需要检查的语句越少, 错误定位精度越高.

在图 4 的全部 5 个怀疑度公式中, 本文提出的方法 IETCR 几乎与原始 MBFL 具有相同的数据分布, 与其他方法相比, IETCR 方法 EXAM Score 的分布更接近原始 MBFL. 因此, 与原始 MBFL 相比, IETCR 没有明显降低错误定位精度, 并且明显优于其他测试用例约减方法 SAMP 30% 和 FTMES. 对于 SAMP 30%, FTMES 方法而言, 错误定位效果较好的是 SAMP 30%, FTMES 次之.

为了更加精确地比较 IETCR 方法的定位效果, 表 6 分别列出了原始 MBFL, IETCR, SAMP(30%) 方法的 EXAM Score 值在不同检查比例下所占的百分比, 即 TOP-N% 值. 从表 6 中的数据可观察到, 使用 IETCR 方法的错误定位精度几乎等同于原始 MBFL 技术, 优于现有方法 FTMES 和 SAMP 30%. 具体而言, 以第 3 行前两列 $<1\%, 0.31>$ 为例, 表示使用怀疑度公式 Jaccard(JA) 对所有版本检查 1% 的代码时, 原始 MBFL 方法可以定位其中 31% 的错误, 加粗部分表示使用对应怀疑度公式和检查比例, 原始 MBFL, IETCR, SAMP 30% 方法定位错误百分比的最大值. 使用全部怀疑度公式, 对所有版本检查 1% 的代码, IETCR 方法定位到的错误所占比例为: 0.29(JA), 0.29(OC), 0.27(OP), 0.02(TA), 0.28(DS), 原始 MBFL 能够定位到的错误比例为: 0.31(JA), 0.31(OC), 0.27(OP), 0.05(TA), 0.30(DS), 与原始 MBFL 方法相比, IETCR 定位错误的百分比减少程度不超过 3%, 并且明显优于 FTMES 和 SAMP 30% 方法, 进一步说明 IETCR 方法能够精准定位被测程序中的错误, 具有一定的实用价值, 在错误定位领域, 能精确定位少量错误, 比模糊定位大量错误更具有实用意义. 表中使用 Jaccard(JA) 公式检查 10%, 15%, 50%, 60% 的代码时, IETCR 方法定位错误的比例高于原始 MBFL, 在错误定位精度方面有了一定提升.

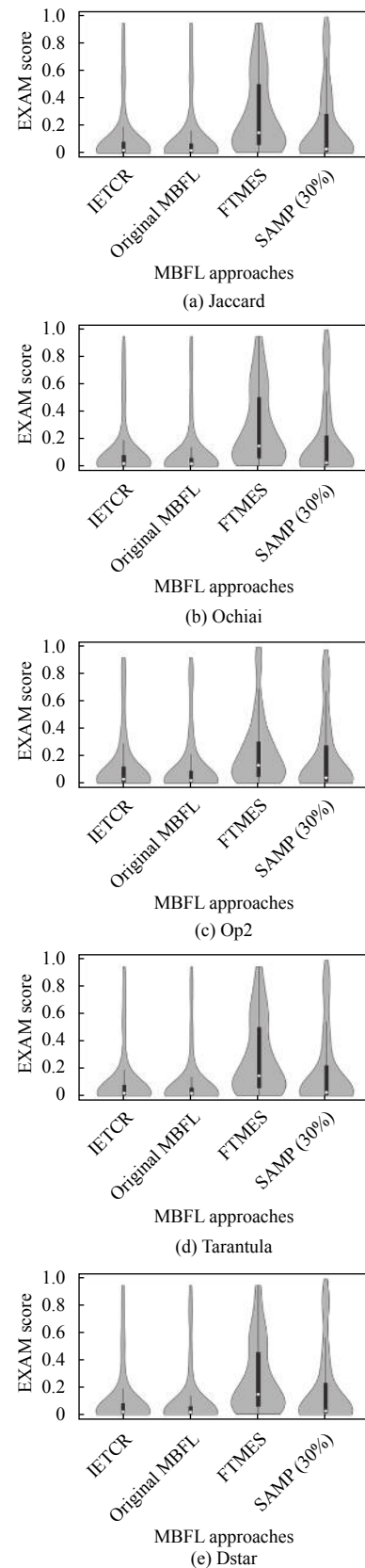


图 4 MBFL 约减方法错误定位精度比较

表6 MBFL, IETCR, FTMES, SAMP 30% 错误定位精度比较

EXAM threshold(%)	Original MBFL					IETCR					FTMES					SAMP 30%				
	JA	OC	OP	TA	DS	JA	OC	OP	TA	DS	JA	OC	OP	TA	DS	JA	OC	OP	TA	DS
1	0.31	0.31	0.27	0.05	0.30	0.29	0.29	0.27	0.02	0.28	0.03	0.03	0.03	0.01	0.03	0.22	0.22	0.20	0.05	0.22
5	0.72	0.74	0.70	0.37	0.74	0.70	0.70	0.64	0.21	0.69	0.23	0.23	0.27	0.19	0.24	0.55	0.57	0.56	0.28	0.58
10	0.79	0.81	0.76	0.55	0.80	0.80	0.80	0.73	0.33	0.77	0.36	0.36	0.39	0.27	0.36	0.63	0.65	0.63	0.42	0.66
15	0.80	0.82	0.79	0.61	0.82	0.82	0.82	0.78	0.41	0.81	0.50	0.50	0.57	0.34	0.50	0.68	0.68	0.69	0.48	0.69
20	0.84	0.84	0.81	0.69	0.84	0.82	0.82	0.78	0.42	0.81	0.52	0.52	0.58	0.35	0.55	0.71	0.74	0.71	0.55	0.73
30	0.88	0.89	0.86	0.80	0.88	0.88	0.88	0.86	0.58	0.88	0.64	0.64	0.74	0.49	0.66	0.75	0.78	0.77	0.62	0.78
40	0.88	0.89	0.88	0.81	0.89	0.88	0.88	0.88	0.78	0.88	0.73	0.73	0.83	0.62	0.74	0.81	0.83	0.82	0.67	0.83
50	0.89	0.92	0.90	0.83	0.91	0.90	0.90	0.90	0.90	0.81	0.90	0.75	0.85	0.67	0.78	0.87	0.86	0.85	0.81	0.86
60	0.94	0.94	0.92	0.90	0.93	0.95	0.95	0.93	0.90	0.95	0.85	0.86	0.92	0.80	0.89	0.89	0.89	0.99	0.88	0.89
70	0.95	0.95	0.93	0.94	0.94	0.95	0.95	0.93	0.90	0.95	0.90	0.90	0.92	0.86	0.92	0.90	0.89	0.99	0.89	0.89
80	0.97	0.97	0.96	0.97	0.97	0.97	0.97	0.96	0.96	0.97	0.96	0.96	0.94	0.94	0.96	0.93	0.93	0.93	0.92	0.92
90	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.97	0.97	0.96	0.97	0.97	0.98	0.98	0.98	0.98	0.98
100	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

为了分析 IETCR 方法与原始 MBFL, FTMES, SAMP 30% 方法之间是否存在显著性差异, 本文在置信度为 95% 的水平下对上述方法进行了秩和检验 (Wilcoxon signed-rank test), 试验结果如表 7 所示. 表中包括 5 个怀疑度公式, 每个对应的值包括两部分, 第一部分为对应方法在所有被测程序中 EXAM Score 的平均值, 括号中的值为对应方法与 IETCR 方法秩和检验的 P-Values 值. 加粗部分对应的 P-Values 值大于 0.05, 表示对应方法与原始 MBFL 之间没有存在显著差异. 依据表中数据, IETCR 方法与原始 MBFL 之间不存在显著性差异, 与 FTMES, SAMP 30% 方法 (Tarantula 公式除外) 之间存在显著性差异

表7 显著性分析

	MBFL	IETCR	MBFL	FTMES	SAMP 30%
Jaccard	0.10(-)	0.10(0.47)	0.27(0.00)	0.28(0.00)	
Ochiai	0.10(-)	0.09(0.36)	0.27(0.00)	0.27(0.00)	
Op2	0.12(-)	0.11(0.36)	0.22(0.00)	0.28(0.00)	
Tarantula	0.28(-)	0.19(0.09)	0.35(0.04)	0.33(0.70)	
Dstar	0.10(-)	0.10(0.29)	0.25(0.00)	0.28(0.00)	

4.4.3 方法的影响因素

为探讨 RQ3, 本文从计算测试用例信息熵的怀疑度公式和信息熵的计算过程出发, 讨论 IETCR 的影响因素和额外开销.

本节首先研究不同 SBFL 公式的定位效果, 并选择性能最优的怀疑度公式用于 IETCR 方法中测试用例信息熵的计算. 实验使用 5 种不同的 SBFL 公式, 对上述实验程序中的错误进行定位, 实验结果如表 8 所示. 表 8 列出了不同 SBFL 公式的 EXAM Score 值在不同检查比例下所占的百分比, 其中加粗部分表示使用对应怀疑度公式和检查比例, SBFL 公式定位错误百

分比的最大值. 从表中数据可知, 所有检查比例中, 使用 Dstar(DS) 公式均能定位最多的错误, 说明在上述 SBFL 公式中, Dstar(DS) 的性能最优. 因此, 本文使用 Dstar 公式计算测试用例的信息熵.

表8 SBFL 方法错误定位精度比较

Score ≤ (%)	JA	OC	OP	TA	DS
1	0.27	0.29	0.3	0.06	0.33
5	0.46	0.5	0.55	0.35	0.57
10	0.72	0.73	0.74	0.57	0.81
15	0.77	0.77	0.76	0.73	0.84
20	0.77	0.8	0.84	0.74	0.86
30	0.9	0.89	0.89	0.91	0.95
40	0.91	0.91	0.91	0.91	0.95
50	0.91	0.91	0.91	0.91	0.95
60	0.96	0.96	0.96	0.96	0.96
70	0.99	0.99	1.00	1.00	1.00
80	0.99	0.99	1.00	1.00	1.00
90	1.00	1.00	1.00	1.00	1.00
100	1.00	1.00	1.00	1.00	1.00

本节其次讨论 IETCR 的额外时间开销. 本文统计了该方法在实验程序上计算信息熵的时间开销, 如图 5 所示. 在图 5 中, 横坐标表示不同的实验程序, 纵坐标表示执行时间, 单位是秒, 图中每一部分表示 IETCR 方法在对应实验程序上计算全部通过测试用例的信息熵所花费时间的分布情况, 其中包括 5 个数值点, 从下到上依次为最小值, 下四分位数, 中位数, 上四分位数, 最大值.

由图 5 可知, IETCR 方法在 6 个程序上测试用例的信息熵计算时间范围为 20~800 s, 其中执行时间最少的程序是 totinfo, 最大的程序是 printtokens, 有 3 个程序的平均执行时间少于 200 s, 相比 IETCR 方法约减的变异体执行开销, 这些额外执行时间是可以忽略不

计的. 因此, IETCR 有更好的约减效果, 同时伴随着少量测试用例信息熵的计算开销.

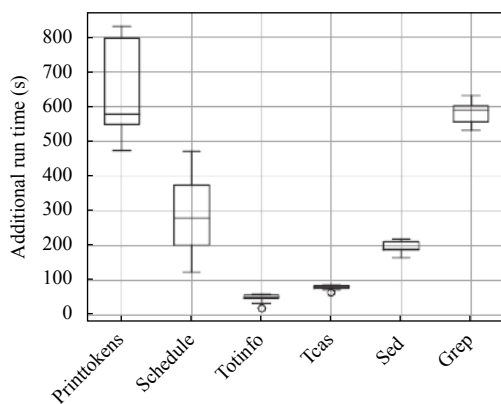


图 5 IETCR 方法额外执行开销

5 结论与展望

为了降低基于变异的错误定位技术的执行开销, 本文从测试用例角度, 提出了一种基于信息熵的测试用例约减策略 IETCR, 利用信息熵理论对通过测试用例进行排序, 选择执行部分检错能力强的通过测试用例和所有失败测试用例, 有效减少了测试用例的数量, 同时保证了测试用例集的质量. 实验结果表明, IETCR 方法能够约减 56.3%~88.6% 的执行开销, 而错误定位精度与原始 MBFL 技术没有显著性差异. 在后续的研究中, 作者将考虑扩大数据集来验证方法的有效性, 并结合基于机器学习的变异体执行结果预测方法, 在不执行变异体的前提下预测其被测试用例杀死与否的结果, 更进一步减少 MBFL 技术的执行开销.

参考文献

- Shin D, Bae DH. A theoretical framework for understanding mutation-based testing methods. 2016 IEEE International Conference on Software Testing, Verification and Validation (ICST). Chicago, IL, USA. 2016. 299–308.
- Howden WE. Theoretical and empirical studies of program testing. IEEE Transactions on Software Engineering, 1978, SE-4(4): 293–298.
- Wong WE, Gao RZ, Li YH, *et al.* A survey on software fault localization. IEEE Transactions on Software Engineering, 2016, 42(8): 707–740.
- Masri W. Fault localization based on information flow coverage. Software Testing, Verification and Reliability, 2010, 20(2): 121–147.
- Keller F, Grunske L, Heiden S, *et al.* A critical evaluation of spectrum-based fault localization techniques on a large-scale software system. 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). Prague, Czech. 2017. 114–125.
- Jones JA, Harrold MJ. Empirical evaluation of the Tarantula automatic fault-localization technique. Proceedings of the 20th IEEE/ACM International Conference on Automated Software Engineering. Long Beach, CA, USA. 2005. 273–282.
- Papadakis M, Le Traon Y. Using mutants to locate “Unknown” faults. 2012 IEEE Fifth International Conference on Software Testing, Verification and Validation. Montreal, QC, Canada. 2012. 691–700.
- Papadakis M, Le Traon Y. Metallaxis-FL: Mutation-based fault localization. Software Testing, Verification and Reliability, 2015, 25(5–7): 605–628.
- Moon S, Kim Y, Kim M, *et al.* Ask the mutants: Mutating faulty programs for fault localization. 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation. Cleveland, OH, USA. 2014. 153–162.
- Pearson S, Campos J, Just R, *et al.* Evaluating and Improving Fault Localization. 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE). Buenos Aires, Argentina. 2017. 609–620.
- Chekam TT, Papadakis M, Le Traon Y. Assessing and comparing mutation-based fault localization techniques. arXiv preprint arXiv: 1607.05512, 2016.
- Mathur AP. Performance, effectiveness, and reliability issues in software testing. Proceedings the Fifteenth Annual International Computer Software & Applications Conference. Tokyo, Japan. 1991. 604–605.
- De Oliveira AAL, Camilo-Junior CG, De Andrade Freitas EN, *et al.* FTMES: A failed-test-oriented mutant execution strategy for mutation-based fault localization. 2018 IEEE 29th International Symposium on Software Reliability Engineering. Memphis, TN, USA. 2018. 155–165.
- Liu Y, Li Z, Zhao RL, *et al.* An optimal mutation execution strategy for cost reduction of mutation-based fault localization. Information Sciences, 2018, 422: 572–596.
- Yoo S, Harman M, Clark D. Fault localization prioritization: Comparing information-theoretic and coverage-based approaches. ACM Transactions on Software Engineering and Methodology, 2013, 22(3): 19.
- Papadakis M, Le Traon Y. Effective fault localization via mutation analysis: A selective mutation approach.

- Proceedings of the 29th Annual ACM Symposium on Applied Computing. New York City, NY, USA. 2014. 1293–1300.
- 17 Demillo RA, Lipton RJ, Sayward FG. Hints on test data selection: Help for the practicing programmer. *Computer*, 1978, 11(4): 34–41.
 - 18 Offutt AJ, Untch RH. Mutation 2000: Uniting the Orthogonal. Wong WE. *Mutation Testing for the New Century*. Boston: Springer, 2001. 34–44.
 - 19 Voas JM. PIE: A dynamic failure-based technique. *IEEE Transactions on Software Engineering*, 1992, 18(8): 717–727.
 - 20 Chen MY, Kiciman E, Fratkin E, *et al.* Pinpoint: Problem determination in large, dynamic Internet services. *Proceedings International Conference on Dependable Systems and Networks*. Washington, DC, USA. 2002. 595–604.
 - 21 Abreu R, Zoetewij P, Van Gemund AJC. An evaluation of similarity coefficients for software fault localization. 2006 12th Pacific Rim International Symposium on Dependable Computing. Riverside, CA, USA. 2006. 39–46.
 - 22 Naish L, Lee HJ, Ramamohanarao K. A model for spectra-based software diagnosis. *ACM Transactions on Software Engineering and Methodology*, 2011, 20(3): 11.
 - 23 Jones JA, Harrold MJ, Stasko J. Visualization of test information to assist fault localization. *Proceedings of the 24th International Conference on Software Engineering*. Orlando, FL, USA. 2002. 467–477.
 - 24 Wong WE, Debroy V, Gao RZ, *et al.* The DStar method for effective software fault localization. *IEEE Transactions on Reliability*, 2014, 63(1): 290–308.
 - 25 Liu Y, Li Z, Wang LX, *et al.* Statement-oriented mutant reduction strategy for mutation based fault localization. 2017 IEEE International Conference on Software Quality, Reliability and Security (QRS). Prague, Czech. 2017. 126–137.
 - 26 Shannon CE. A mathematical theory of communication. *The Bell System Technical Journal*, 1948, 27(4): 623–656.
 - 27 Do H, Elbaum S, Rothermel G. Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact. *Empirical Software Engineering*, 2005, 10(4): 405–435.
 - 28 Yang Q, Li JJ, Weiss DM. A survey of coverage-based testing tools. *The Computer Journal*, 2007, 52(5): 589–597.
 - 29 Delamaro ME, Maldonado JC, Vincenzi AMR. Proteum/IM 2.0: An integrated mutation testing environment. In: Wong WE, ed. *Mutation Testing for the New Century*. Boston: Springer, 2001. 91–101.
 - 30 Le TDB, Lo D, Le Goues C, *et al.* A learning-to-rank based fault localization approach using likely invariants. *Proceedings of the 25th International Symposium on Software Testing and Analysis*. Saarbrücken, Germany. 2016. 177–188.
 - 31 Li X, Zhang LM. Transforming programs and tests in tandem for fault localization. *Proceedings of the ACM on Programming Languages*, 2017, 1(OOPSLA): 92.
 - 32 Wilcoxon F. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1945, 1(6): 80–83.
 - 33 Zhang LM, Marinov D, Khurshid S. Faster mutation testing inspired by test prioritization and reduction. *Proceedings of the 2013 International Symposium on Software Testing and Analysis*. Lugano, Switzerland. 2013. 235–245.