变异测试优化技术综述汇报

南京大学软件学院 刘承杰

变异测试优化技术综述

刘承杰　赵凝晖　刘晓旭　王骁

(南京大学软件学院，南京，210093)

摘　要　变异测试是一种基于故障的软件测试技术，广泛用来评估测试用例集的充分性与软件测试技术的有效性。尽管变异测试具有较强的故障检测能力，但由于数量庞大的变异体导致了计算开销大的问题，阻碍了变异测试在实践中的广泛应用。针对于变异测试开销过大的这个问题，国内外的很多学者对于变异测试优化这一领域进行了一系列的研究并取得了一些成果。通过对已有的工作进行总结，本文将变异测试优化技术分为变异体选择优化和变异体执行优化两个模块。在变异体选择优化模块中，将现有的优化方法分成随机选择法、聚类选择法、变异算子选择法、高阶变异优化法以及程序分析法五个方面，并进行分类总结；在变异体执行优化模块中，从变异体检测优化、变异体编译优化以及并行执行变异体三个角度总结分类现有研究成果。最后对变异测试优化的未来研究方向进行展望。

关键词　变异测试；变异测试优化；变异体选择；变异体执行；综述

A Survey on Optimizing Mutation Testing Methods

Liu Chengjie　Zhao Ninghui　Liu Xiaoxu　Wang Xiao

(Software Institute, Nanjing University, Nanjing, 210093)

Abstract Mutation testing is a fault-based software testing technique，which is widely used to evaluate the adequacy of a given test suite or the fault detection effectiveness of a given software testing technique. Although mutation testing has a strong fault detection capability, the high computation cost incurred by a huge number of mutants prevents mutation testing from being widely adopted in practice. Directed against the problem that mutation testing has occasion to excessive cost, a series of research have been conducted on mutation testing optimization by scholars globally. Summarizing the existing work and achievements, we can divide mutation testing optimization technology into two modules in this paper, which are called mutation selection optimization and mutation execution optimization. In the module “mutation selection optimization”, the existing methods are divided into five aspects: random selection method, cluster selection method, mutation operator method, high-order mutation optimization method and program analysis method. On this basis, we will summarize by the classification. In the module “mutation execution optimization”, we sort and summarize existing research results from the perspective of mutation detection optimization, mutation compilation optimization and parallel execution mutation. At last, we will foresee and outlook the future research direction to mutation testing optimization.

Key words　mutation testing; optimize mutation testing; mutant selection; mutant runtime; survey

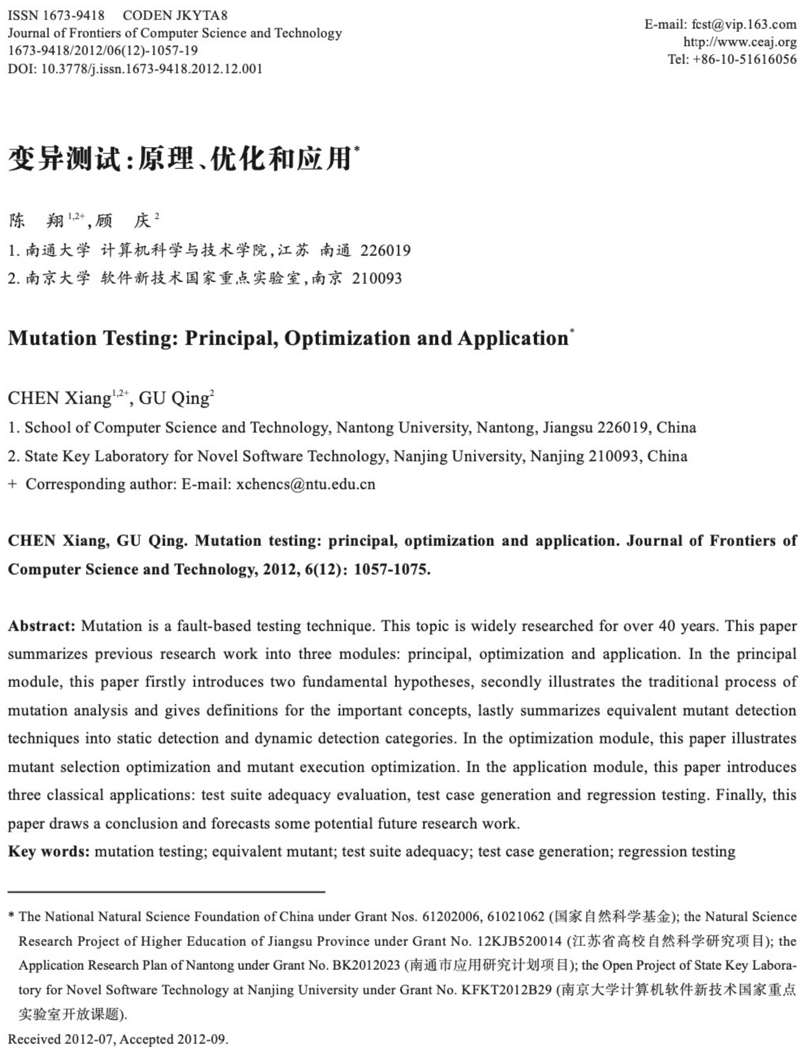
1　引言

变异测试是一种基于故障的软件测试方法[1]，在最近几十年的时间里得到国内外学者的广泛关注。作为一种软件测试技术，变异测试与传统的借助数据流或者控制流分析来检验测试充分性的技术不同，该测试技术从评估和改进测试用例集的角

综述研究流程

- 三个环节，五个步骤
  - 三个环节：规划（Plan）、实施（Conduct）、报告（Report）
  - 五个步骤
    - **框架搭建**：制定研究框架、设计研究问题
    - **文献检索**：检索目标领域近5~10年的工作
    - **文献阅读**：阅读文献，了解文章内容
    - **文献分类**：按照一定的规则，将收集到的文献划分成3~6个正交（Orthogonal）类别
    - **文献分析**：提取文献共性、甄别文献特点、得出研究方向

01 框架搭建



ISSN 1673-9418　CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2012/06(12)-1057-19
DOI: 10.3778/j.issn.1673-9418.2012.12.001

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-51616056

变异测试:原理、优化和应用*

陈　翔[1,2+],顾　庆[2]
1. 南通大学 计算机科学与技术学院,江苏 南通 226019
2. 南京大学 软件新技术国家重点实验室,南京 210093

Mutation Testing: Principal, Optimization and Application*

CHEN Xiang[1,2+], GU Qing[2]
1. School of Computer Science and Technology, Nantong University, Nantong, Jiangsu 226019, China
2. State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China
+ Corresponding author: E-mail: xchencs@ntu.edu.cn

CHEN Xiang, GU Qing. Mutation testing: principal, optimization and application. Journal of Frontiers of Computer Science and Technology, 2012, 6(12): 1057-1075.

Abstract: Mutation is a fault-based testing technique. This topic is widely researched for over 40 years. This paper summarizes previous research work into three modules: principal, optimization and application. In the principal module, this paper firstly introduces two fundamental hypotheses, secondly illustrates the traditional process of mutation analysis and gives definitions for the important concepts, lastly summarizes equivalent mutant detection techniques into static detection and dynamic detection categories. In the optimization module, this paper illustrates mutant selection optimization and mutant execution optimization. In the application module, this paper introduces three classical applications: test suite adequacy evaluation, test case generation and regression testing. Finally, this paper draws a conclusion and forecasts some potential future research work.

Key words: mutation testing; equivalent mutant; test suite adequacy; test case generation; regression testing

* The National Natural Science Foundation of China under Grant Nos. 61202006, 61021062 (国家自然科学基金); the Natural Science Research Project of Higher Education of Jiangsu Province under Grant No. 12KJB520014 (江苏省高校自然科学研究项目); the Application Research Plan of Nantong under Grant No. BK2012023 (南通市应用研究计划项目); the Open Project of State Key Laboratory for Novel Software Technology at Nanjing University under Grant No. KFKT2012B29 (南京大学计算机软件新技术国家重点实验室开放课题).

Received 2012-07, Accepted 2012-09.

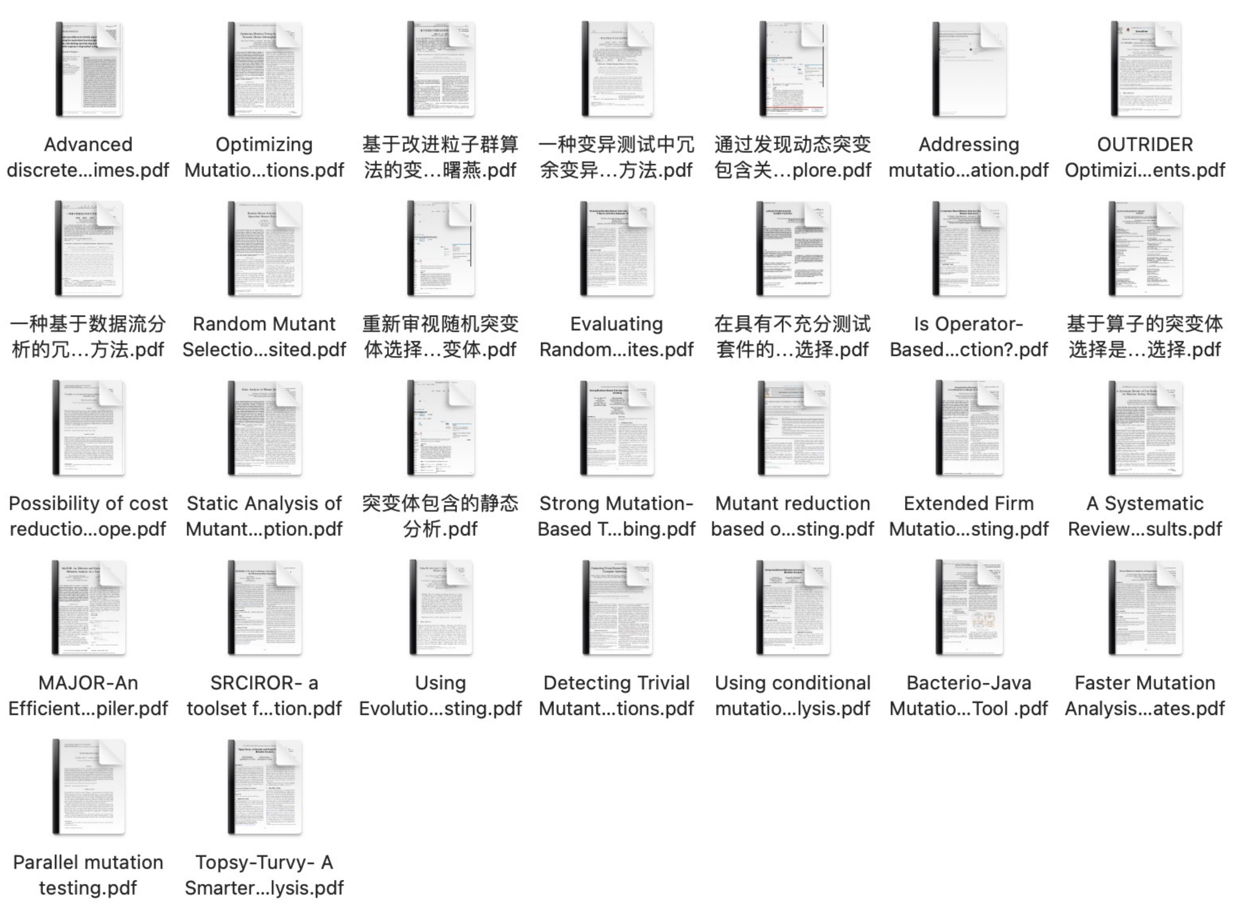
陈翔, 顾庆. 变异测试: 原理, 优化和应用[J]. 计算机科学与探索, 2012, 6(12): 1057-1075.

核心：确定研究问题

1	变异测试是什么
2	变异测试的在工业界的应用现状
3	为什么要对变异测试进行优化
4	对变异测试进行优化的角度有哪些
5	优化的成果如何
6	对未来应用的展望

02 文献检索

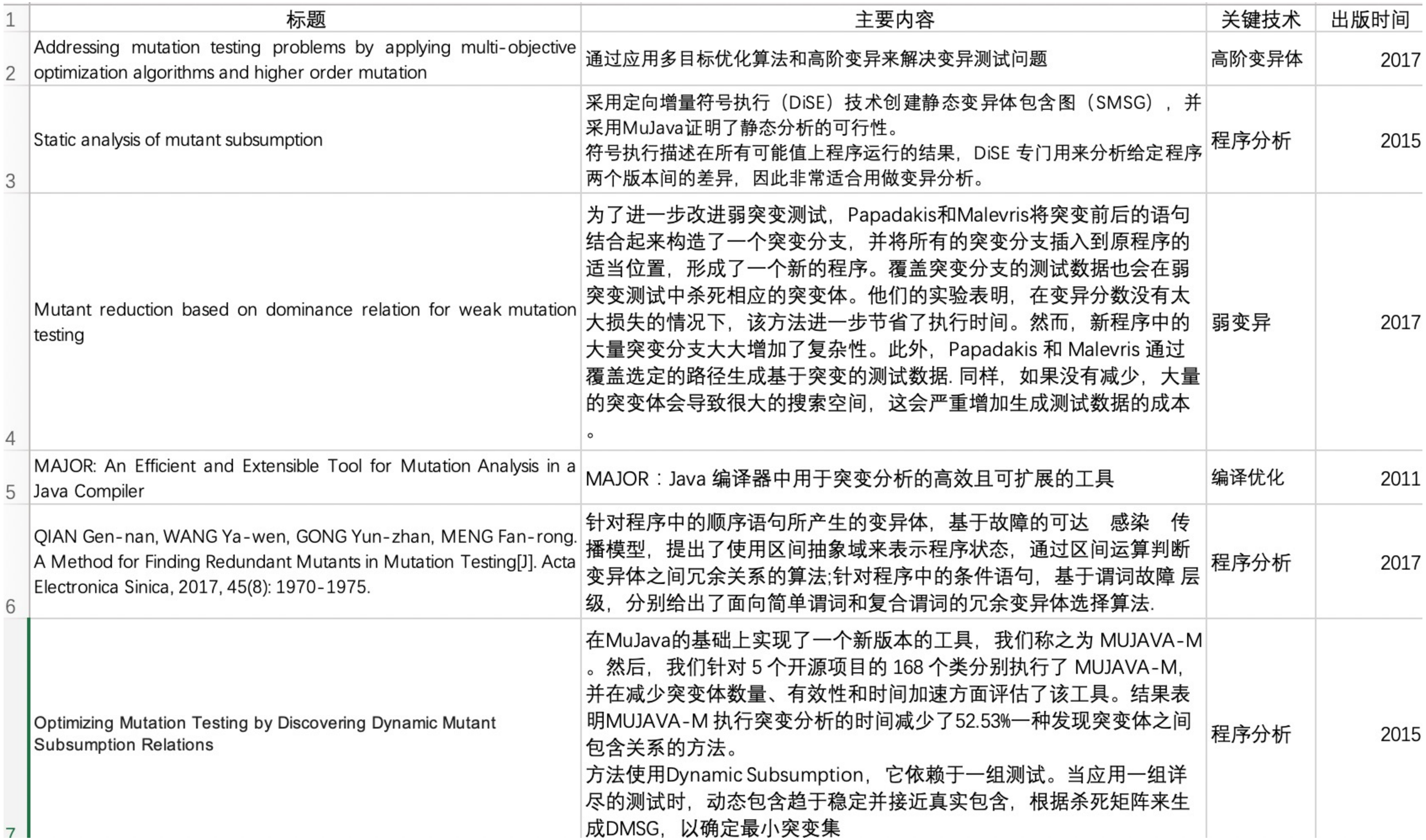
- 检索范围
  - 近10年
  - CCF推荐国际学术会议和期刊目录
- 检索方法
  - Google Schola
  - 检索综述的参考文献
    - Mutation testing advances: an analysis and survey
    - Predictive mutation testing
    - An Analysis and Survey of the Development of Mutation Testing
- 细分关键词搜索
  - Optimization
  - Mutant Reduction
  - Execution Cost Reduction



Advanced discrete...imes.pdf
Optimizing Mutatio...tions.pdf
基于改进粒子群算法的变…曙燕.pdf
一种变异测试中冗余变异…方法.pdf
通过发现动态突变包含关…plore.pdf
Addressing mutatio...ation.pdf
OUTRIDER Optimizi...ents.pdf
一种基于数据流分析的冗…方法.pdf
Random Mutant Selectio...sited.pdf
重新审视随机突变体选择…变体.pdf
Evaluating Random...ites.pdf
在具有不充分测试套件的…选择.pdf
Is Operator-Based...ction?.pdf
基于算子的突变体选择是…选择.pdf
Possibility of cost reductio...ope.pdf
Static Analysis of Mutant...ption.pdf
突变体包含的静态分析.pdf
Strong Mutation-Based T...bing.pdf
Mutant reduction based o...sting.pdf
Extended Firm Mutatio...sting.pdf
A Systematic Review...sults.pdf
MAJOR-An Efficient...piler.pdf
SRCIROR- a toolset f...tion.pdf
Using Evolutio...sting.pdf
Detecting Trivial Mutant...tions.pdf
Using conditional mutatio...lysis.pdf
Bacterio-Java Mutatio...Tool .pdf
Faster Mutation Analysis...ates.pdf
Parallel mutation testing.pdf
Topsy-Turvy- A Smarter...lysis.pdf

3&4 文献阅读&分类

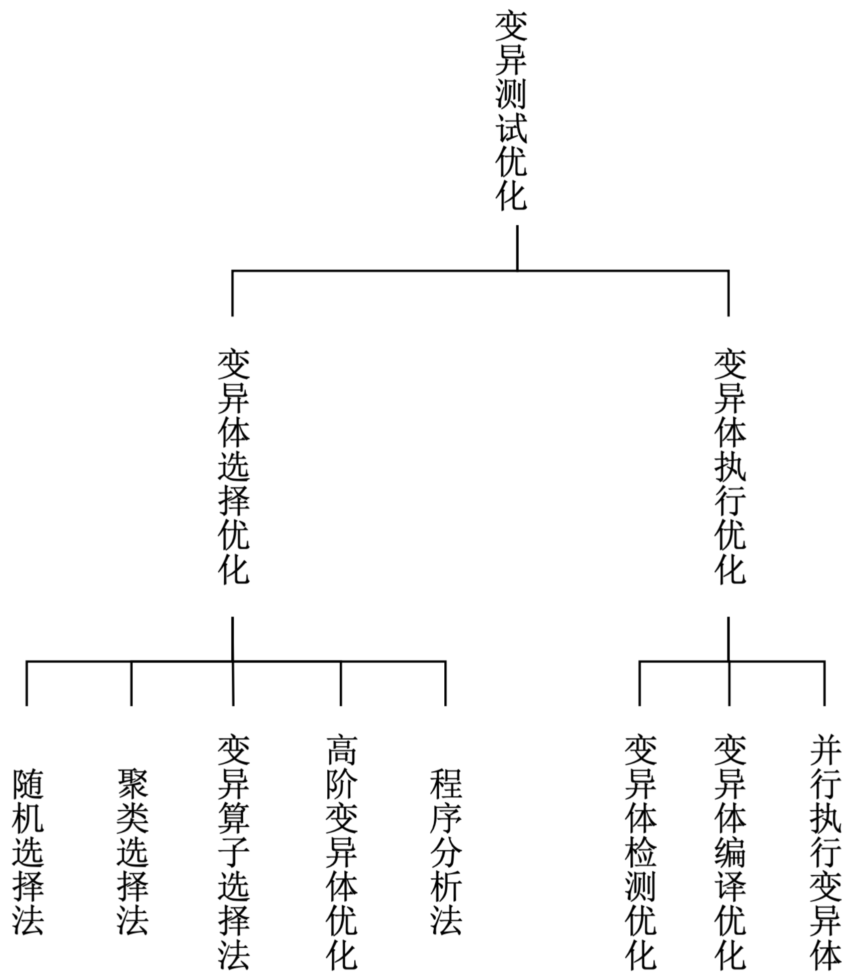
NANJING UNIVERSITY



1	标题	主要内容	关键技术	出版时间
2	Addressing mutation testing problems by applying multi-objective optimization algorithms and higher order mutation	通过应用多目标优化算法和高阶变异来解决变异测试问题	高阶变异体	2017
3	Static analysis of mutant subsumption	采用定向增量符号执行（DiSE）技术创建静态变异体包含图（SMSG），并采用MuJava证明了静态分析的可行性。 符号执行描述在所有可能值上程序运行的结果，DiSE 专门用来分析给定程序两个版本间的差异，因此非常适合用做变异分析。	程序分析	2015
4	Mutant reduction based on dominance relation for weak mutation testing	为了进一步改进弱突变测试，Papadakis和Malevris将突变前后的语句结合起来构造了一个突变分支，并将所有的突变分支插入到原程序的适当位置，形成了一个新的程序。覆盖突变分支的测试数据也会在弱突变测试中杀死相应的突变体。他们的实验表明，在变异分数没有太大损失的情况下，该方法进一步节省了执行时间。然而，新程序中的大量突变分支大大增加了复杂性。此外，Papadakis 和 Malevris 通过覆盖选定的路径生成基于突变的测试数据. 同样，如果没有减少，大量的突变体会导致很大的搜索空间，这会严重增加生成测试数据的成本 。	弱变异	2017
5	MAJOR: An Efficient and Extensible Tool for Mutation Analysis in a Java Compiler	MAJOR：Java 编译器中用于突变分析的高效且可扩展的工具	编译优化	2011
6	QIAN Gen-nan, WANG Ya-wen, GONG Yun-zhan, MENG Fan-rong. A Method for Finding Redundant Mutants in Mutation Testing[J]. Acta Electronica Sinica, 2017, 45(8): 1970-1975.	针对程序中的顺序语句所产生的变异体，基于故障的可达　感染　传播模型，提出了使用区间抽象域来表示程序状态，通过区间运算判断变异体之间冗余关系的算法;针对程序中的条件语句，基于谓词故障 层级，分别给出了面向简单谓词和复合谓词的冗余变异体选择算法.	程序分析	2017
7	Optimizing Mutation Testing by Discovering Dynamic Mutant Subsumption Relations	在MuJava的基础上实现了一个新版本的工具，我们称之为 MUJAVA-M。然后，我们针对 5 个开源项目的 168 个类分别执行了 MUJAVA-M，并在减少突变体数量、有效性和时间加速方面评估了该工具。结果表明MUJAVA-M 执行突变分析的时间减少了52.53%一种发现突变体之间包含关系的方法。 方法使用Dynamic Subsumption，它依赖于一组测试。当应用一组详尽的测试时，动态包含趋于稳定并接近真实包含，根据杀死矩阵来生成DMSG，以确定最小突变集	程序分析	2015

05 文献分析

确定研究框架



变异测试优化

变异体选择优化

随机选择法

聚类选择法

变异算子选择法

高阶变异体优化

程序分析法

变异体执行优化

变异体检测优化

变异体编译优化

并行执行变异体

文章写作结构

- 引言
- 变异测试相关知识
  - 基本假设
  - 传统变异测试流程
  - 相关术语
- 变异体选择优化技术
- 变异体执行优化技术
- 结束语

附 参考文献

[1] DeMilo R A，Lipton R J，Sayward F G.Hints on test data selection:Help for the practicing programmer.IEEE Computer， 1978，11(4):34-41

[2] Chen Xiang, Gu Qing. Mutation Testing: Principal, Optimization and Application, Journal of Frontiers of Computer Science and Technology, 2012, 6(12): 1057-1075 (in Chinese) (陈翔, 顾庆. 变异测试: 原理, 优化和应用[J]. 计算机科学与探索，2012，6(12): 1057-1075.)

[3] Offutt J, Pan Jie. Automatically detecting equivalent mutants and infeasible paths[J]. Software Testing, Verification and Reliability, 1997, 7(3): 165-192.

[4] Mathur A P, Wong W E. An empirical comparison of data flow and mutation-based test adequacy criteria[J]. Software Testing, Verification and Reliability, 1994, 4(1): 9-31.

[5] Zhang Lu, Hou Shanshan, Hu Junjue, et al. Is operator-based mutant selection superior to random mutant selection? [C]// Proceedings of the 32nd International Conference on Soft- ware Engineering (ICSE 10). New York, NY, USA: ACM, 2010: 435-444.

[6] Parsai A, Murgia A, Demeyer S. Evaluating random mutant selection at class-level in projects with non-adequate test suites[C]//Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering. 2016: 1-10.

[7] R. Pitts, "Random Mutant Selection and Equivalent Mutants Revisited," 2022 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2022, pp. 170-178, doi: 10.1109/ICSTW55395.2022.00040.

[8] Wang Shuyan, Yang Yue, Sun Jiaze. Mutants selection based on improved particle swarm optimization algorithm[J]. Application Research of Computers, 2017, 34(3): 752-755

[9] Yu M, Ma Y S. Possibility of cost reduction by mutant clustering according to the clustering scope[J]. Software Testing, Verification and Reliability, 2019, 29(1-2): e1692.

[10] L. Madeyski, W. Orzeszyna, R. Torkar, M. Józala, Overcoming the equivalent mutant problem: A systematic literature review and a comparative experi- ment of second order mutation, IEEE Trans. Softw. Eng.. 40 (1) (2014) 23–44.

[11] E. Omar, S. Ghosh, D. Whitely, Constructing subtle higher order mutants for Java and AspectJ programs, in: Proceedings of the 24th IEEE International Symposium on Software Reliability Engineering, ISSRE, 2013, pp. 340–349.

[12] M. Kintis, M. Papadakis, N. Malevris, Isolating First Order Equivalent Mutants via Second Order Mutation, in: IEEE Fifth International Conference on Soft- ware Testing, Verification and Validation, 2012, pp. 701–710.

[13] M. Harman, Y. Jia, P. Reales Mateo, M. Polo, Angels and monsters: An empir- ical investigation of potential test effectiveness and efficiency improvement from strongly subsuming higher order mutation, in: Proceedings of the 29th ACM/IEEE International Conference on Automated Software Engineering, ASE'14, ACM, New York, NY, USA, 2014, pp. 397–408.

[14] Y. Jia, F. Wu, M. Harman, J. Krinke, Genetic Improvement using Higher Order Mutation, in: GECCO Companion'15: Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, 2015, pp. 803–804.

[15] Kurtz B, Ammann P, Offutt J. Static analysis of mutant subsumption[C]//2015 IEEE Eighth International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 2015: 1-10.

[16] Guimarães M A, Fernandes L, Ribeiro M, et al. Optimizing mutation testing by discovering dynamic mutant subsumption relations[C]//2020 IEEE 13th International Conference on Software Testing, Validation and Verification (ICST). IEEE, 2020: 198-208.

[17] Sun Changai et al. A Data Flow Analysis Based Redundant Mutants Identification Technique, CHINESE JOURNAL OF COMPUTERS[J], 2019, 1. (in Chinese) (孙昌爱, 郭新玲, 张翔宇, 等. 一种基于数据流分析的冗余变异体识别方法[J]. 计算机学报, 2019, 1.)

[18] QIAN Gen-nan, WANG Ya-wen, GONG Yun-zhan, MENG Fan-rong. A Method for Finding Redundant Mutants in Mutation Testing[J]. Acta Electronica Sinica, 2017, 45(8): 1970-1975. (in Chinese)

[19] M. Papadakis, N. Malevris, Automatically performing weak mutation with the aid of symbolic execution, concolic testing and search-based testing, Softw. Qual. J. 19 (4) (2011) 691–723.

[20] M. Papadakis, N. Malevris, Mutation based test case generation via a path se- lection strategy, Inform. Softw. Technol. 54 (9) (2012). 915–312

[21] Gong D, Zhang G, Yao X, et al. Mutant reduction based on dominance relation for weak mutation testing[J]. Information and Software Technology, 2017, 81: 82-96.

[22] Hariri F, Shi A. Srciror: A toolset for mutation testing of c source code and llvm intermediate representation[C]//Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering. 2018: 860-863.

[23] Just R, Schweiggert F, Kapfhammer G M. MAJOR: An efficient and extensible tool for mutation analysis in a Java compiler[C]//2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). IEEE, 2011: 612-615.

[24] Just R, Kapfhammer G M, Schweiggert F. Using conditional mutation to increase the efficiency of mutation analysis[C]//Proceedings of the 6th International Workshop on Automation of Software Test. 2011: 50-56.

[25] Kintis M, Papadakis M, Jia Y, et al. Detecting trivial mutant equivalences via compiler optimisations[J]. IEEE Transactions on Software Engineering, 2017, 44(4): 308-333.

[26] Delgado-Pérez P, Medina-Bulo I, Merayo M G. Using evolutionary computation to improve mutation testing[C]//International Work-Conference on Artificial Neural Networks. Springer, Cham, 2017: 381-391.

[27] Mathur A P, Krauser E W. Mutant unification for improved vectorization[R]. Purdue University, 1988.

[28] Mateo P R, Usaola M P. Bacterio: Java mutation testing tool: A framework to evaluate quality of tests cases[C]//2012 28th IEEE International Conference on Software Maintenance (ICSM). IEEE, 2012: 646-649.

[29] N. Li, M. West, A. Escalona, and V. H. S. Durelli, "Mutation Testing in Practice Using Ruby," in Mutation'15, 2015.

[30] Gopinath R, Jensen C, Groce A. Topsy-Turvy: a smarter and faster parallelization of mutation analysis[C]//Proceedings of the 38th International Conference on Software Engineering Companion. 2016: 740-743.

[31] Ezugwu A E. Advanced discrete firefly algorithm with adaptive mutation-based neighborhood search for scheduling unrelated parallel machines with sequence-dependent setup times[J]. International Journal of Intelligent Systems, 2022, 37(8): 4612-4653.