法 VS-PSTG 生成了相应的覆盖表.2012 年,王子元等人[31]提出一种新的可变力度组合测试模型,并给出了基于 one-test-at-a-time 策略的可变力度覆盖表生成算法.在新的变力度测试模型中,测试用例集不再需要覆盖任意 $t$ 个参数间取值组合,仅需要满足给定的覆盖需求即可,因此,该模型可以更为准确地描述系统中参数间的交互作用.在覆盖表规模方面,该文中的算法相对于某些可变力度覆盖表生成工具如 PICT[32]具有一定的优势,但仍不如 Cohen 等人提出的 SA.

在并行计算方面,2011 年,Himer 等人[33]利用高性能计算机并行地统计各参数取值组合被覆盖情况,依此来验证一个矩阵是否满足全覆盖,但该文献并不进行覆盖表的生成.2014 年,Lopez-Herrejon 等人[34]利用并行遗传算法为软件生产线生成具有优先级的 2 维覆盖表.近年来,一些研究者开始利用分布式系统进行并行化.例如,Geronimo 等人[35]利用 Hadoop MapReduce 实现遗传算法并行化,提高覆盖表生成速度.最近,戚荣志等人[29]还基于另外一种分布式并行计算框架 Spark 提出了一种岛模型并行化遗传算法,该方法支持大规模种群下的快速寻优,能大大加快覆盖表的生成速度.目前,这些分布式并行化研究主要还是针对遗传算法.

## 7 结论及未来工作

覆盖表生成作为组合测试的关键问题之一,一直受到工业界和学术界的重视.覆盖表规模、种类以及生成时间是衡量覆盖表生成算法的 3 个重要指标.作为一种有效的覆盖表生成算法,禁忌搜索在覆盖表规模方面表现出一定优势,但解的质量和运算速度仍然有提升的空间.特别是目前已有的禁忌搜索算法只关注于相对简单的固定力度无约束的组合测试模型,而事实上,在很多软件系统中,参数间的交互力度往往是不同的,参数间的约束也普遍存在.如果不考虑这些应用场景,组合测试将难以发挥其作用.

在已有的研究工作基础上,为了进一步提升 TS 在覆盖表生成上的应用潜力,本文提出了一种新的 TS 算法 PTS_CVS,所做的主要工作如下.

- 首先,通过 pair-wise 实验和爬山实验构成的参数配置调优,获得了 TS 最佳参数配置,并使用 Friendman 和 Wilcoxon 检验法对该最优配置进行统计分析;
- 其次,将 TS 与禁止元组的约束处理方法相结合,使 TS 能够生成带约束的覆盖表,并进一步扩展 TS 使其支持可变力度覆盖表的生成;
- 最后,为了提高覆盖表生成速度,使用 ForkJoin 框架实现 TS 并行化.

实验结果表明:在覆盖表规模方面,PTS_CVS 在多种测试模型下都更新了一些覆盖表的上界;在生成速度上,虽然不同测试模型会导致并行化提速的不同,但是平均上仍有 2.6 倍的提速.

在下一步工作中,我们将继续探索新的约束处理方法,进一步提高 PTS_CVS 的约束处理能力.基于禁止元组约束处理,在约束量不大情况下效果较好;但是当约束个数较多时,效果并不理想.例如,SPIN-V 实例中的禁止元组数高达 66 个,并且 60%的参数至少参与一个禁止元组,此时 PTS_CVS 生成的覆盖表是所有方法中最大的(见表 23).其次,利用 ForkJoin 框架进行算法并行化,虽然有一定的提速,但由于本地并行化受单机硬件资源限制,效果有限.因此,我们计划将实验迁移到分布式或云平台环境中,利用这些平台超强的计算能力更快更好地实施实验,进一步提升覆盖表生成效率.

**References**:

[1] Nie CH, Leung H. A survey of combinatorial testing. ACM Computing Surveys, 2011,43(2):1−29. [doi: 10.1145/1883612.1883618]

[2] Nie CH, Wu HY, Niu XT, Kuo FC, Leung H, Colbourn CJ. Combinatorial testing, random testing, and adaptive random testing for detecting interaction triggered failures. Information and Software Technology, 2015,62:198−213. [doi: 10.1016/j.infsof.2015.02.008]

[3] Kuhn DR, Reilly MJ. An investigation of the applicability of design of experiments to software testing. In: Proc. of the 27th Annual NASA Goddard Software Engineering Workshop. IEEE, 2002. 91−95. [doi: 10.1109/SEW.2002.1199454]

[4] Bryce RC, Colbourn CJ, Cohen MB. A framework of greedy methods for constructing interaction test suites. In: Proc. of the 27th Int'l Conf. on Software Engineering. IEEE, 2005. 146−155.[doi: 10.1109/ICSE.2005.1553557]

[5] Colbourn CJ, Cohen MB, Turban R. A deterministic density algorithm for pairwise interaction coverage. In: Proc. of the IASTED Conf. on Software Engineering. 2004. 345–352.

[6] Tai KC, Lei Y. A test generation strategy for pairwise testing. IEEE Trans. on Software Engineering, 2002,28(1):109–111. [doi: 10.1109/32.979992]

[7] Lei Y, Kacker R, Kuhn DR, Okun V, Lawrence J. IPOG: A general strategy for *t*-way software testing. In: Proc. of the 14th Annual IEEE Int'l Conf. and Workshops on the Engineering of Computer-Based Systems. IEEE, 2007. 549–556. [doi: 10.1109/ECBS.2007.47]

[8] Nie CH, Wu HY, Liang YL. Search based combinatorial testing. In: Proc. of the 19th Asia-Pacific Software Engineering Conf. IEEE, 2012. 778–783. [doi: 10.1109/APSEC.2012.16]

[9] Willams AW. Determination of test configurations for pair-wise interaction coverage. In: Proc. of the IFIP TC6/WG6.1 13th Int'l Conf. on Testing Communicating Systems, Vol.48. Boston: Springer-Verlag, 2000. 59–74. [doi: 10.1007/978-0-387-35516-0_4]

[10] Kobayashi N. Design and evaluation of automatic test generation strategies for functional testing of software [Ph.D. Thesis]. Osaka: Osaka University, 2002.

[11] Gonzalez-Hernandez L. New bounds for mixed covering arrays of *t*-way testing with uniform strength. Information and Software Technology, 2015,59:17–32. [doi: 10.1016/j.infsof.2014.10.009]

[12] Liang YL, Nie CH. The optimization of configurable genetic algorithm for covering arrays generation. Chinese Journal of Computers, 2012,35(7):1522–1538 (in Chinese with English abstract).

[13] Cohen MB, Gibbons PB, Mugridge WB, Colbourn CJ, Collofello JS. Variable strength interaction testing of components. In: Proc. of the 27th Annual Int'l Conf. on Computer Software and Applications. IEEE, 2003: 413–418. [doi: 10.1109/CMPSAC.2003.1245373]

[14] Chen X, Gu Q, Li A, Chen D. Variable strength interaction testing with an ant colony system approach. In: Proc. of the 2009 16th Asia-Pacific Software Engineering Conf. IEEE, 2009. 160–167. [doi: 10.1109/APSEC.2009.18]

[15] Ahmed BS, Zamli KZ. A variable strength interaction test suites generation strategy using particle swarm optimization. Journal of Systems and Software, 2011,84(12):2171–2185. [doi: 10.1016/j.jss.2011.06.004]

[16] Lakhotia K, Harman M, Gross H. AUSTIN: An open source tool for search based software testing of C programs. Information and Software Technology, 2013,55(1):112–125. [doi: 10.1016/j.infsof.2012.03.009]

[17] Glover F. Tabu Search-Part 2. ORSA Journal on Computing, 1990,2(1):4–32. [doi: 10.1287/ijoc.2.1.4]

[18] Nurmela KJ. Upper bounds for covering arrays by tabu search. Discrete Applied Mathematics, 2004,138(1-2):143–152. [doi: 10.1016/S0166-218X(03)00291-9]

[19] Jia Y, Cohen MB, Harman M, Petke J. Learning combinatorial interaction testing strategies using hyperheuristic search. In: Proc. of the 2015 IEEE/ACM 37th IEEE Int'l Conf. on Software Engineering. IEEE, 2015. 540–550. [doi: 10.1109/ICSE.2015.71]

[20] Yu L, Duan F, Lei Y, Kacher RN, Kuhn DR. Constraint handling in combinatorial test generation using forbidden tuples. In: Proc. of the IEEE 8th Int'l Conf. on Software Testing. IEEE, 2015. 1–9. [doi: 10.1109/ICSTW.2015.7107441]

[21] Cohen MB, Dwyer MB, Shi J. Interaction testing of highly-configurable systems in the presence of constraints. In: Proc. of the 5th Int'l Symp. on Software Testing and Analysis. London: ACM Press, 2007. 129–139. [doi: 10.1145/1273463.1273482]

[22] Cohen MB, Dwyer MB, Shi J. Constructing interaction test suites for highly-configurable systems in the presence of constraints: A greedy approach. IEEE Trans. on Software Engineering, 2008,34(5):633–650. [doi: 10.1109/TSE.2008.50]

[23] Garvin BJ, Cohen MB, Dwyer MB. An improved meta-heuristic search for constrained interaction testing. In: Proc. of the 1st Int'l Symp. on Search Based Software Engineering. IEEE, 2009. 13–22. [doi: 10.1109/SSBSE.2009.25]

[24] Gonzalez-Hernandez L, Torres-Jimenez J, Rangel-Valdez N. MiTS in depth: An analysis of distinct tabu search configurations for constructing mixed covering arrays. Artificial Intelligence Evolutionary Computing and Metaheuristics, 2013,427:371–402. [doi: 10.1007/978-3-642-29694-9_15]

[25] Walker II RA, Colbourn CJ. Tabu search for covering arrays using permutation vectors. Journal of Statistical Planning and Inference, 2009,139(1):69–80. [doi: 10.1007/978-3-642-29694-9_15]

[26] Yamada A, Kitamura T, Artho C, Choi EH, Oiwa Y. Optimization of combinatorial testing by incremental SAT solving. In: Proc. of the 2015 IEEE 8th Int'l Conf. on Software Testing, Verification and Validation. 2015. 1–10. [doi: 10.1109/ICST.2015.7102599]

[27] Garvin BJ, Cohen MB, Dwyer MB. Evaluating improvements to a meta-heuristic search for constrained interaction testing. Empirical Software Engineering, 2011,16(1):61–102. [doi: 10.1007/s10664-010-9135-7]

[28] Cohen MB, Gibbons PB, Mugridge WB, Colbourn CJ. Constructing test suites for interaction testing. In: Proc. of the 25th Int’l Conf. on Software Engineering. IEEE, 2003. 38–48. [doi: 10.1109/ICSE.2003.1201186]

[29] Qi RZ, Wang ZJ, Huang YH, Li SY. Generating combinatorial test suite with spark based parallel approach. Chinese Journal of Computers, 2018,41(6):1284–1299 (in Chinese with English abstract).

[30] Zamli KZ, Alkazem BY, Kendall GA. Tabu search hyper-heuristic strategy for t-way test suite generation. Applied Soft Computing, 2016,44:57–74. [doi: 10.1016/j.asoc.2016.03.021]

[31] Wang Z, Qian J, Chen L, Xu BW. Generating variable strength combinatorial test suite with one-test-at-a-time strategy. Chinese Journal of Computers, 2012,35(12):2541–2552 (in Chinese with English abstract).

[32] Czerwonka J. Pairwise testing in real world: Practical extensions to test case scenarios. In: Proc. of the 24th Pacific Northwest Software Quality Conf. 2006. 419–430.

[33] Himer AG, Jose TJ, Hernandez V, Nelson RV. A parallel algorithm for the verification of covering arrays. In: Proc. of the Int’l Conf. on Parallel and Distributed Processing Techniques and Applications (PDPTA). 2011. 879–885.

[34] Lopez-Herrejon RE, Ferrer J, Chicano F. A parallel evolutionary algorithm for prioritized pairwise testing of software product lines. In: Proc. of the 16th Genetic and Evolutionary Computation Conf. Vancouver: ACM Press, 2014. 1255–1262. [doi: 10.1145/2576768.2598305]

[35] Geronimo LD, Ferrucci F, Murolo A. A parallel genetic algorithm based on hadoop mapReduce for the automatic generation of JUnit test Suites. In: Proc. of the 2012 IEEE 5th Int’l Conf. on Software Testing, Verification and Validation. IEEE, 2012. 785–793. [doi: 10.1109/ICST.2012.177]

**附中文参考文献:**

[12] 梁亚澜,聂长海.覆盖表生成的遗传算法配置参数优化.计算机学报,2012,35(7):1522–1538.

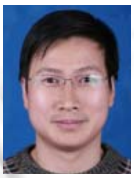
[29] 戚荣志,王志坚,黄宜华,李水艳.基于 Spark 的并行化组合测试用例集生成方法.计算机学报,2018,41(6):1284–1299.

[31] 王子元,钱巨,陈林,徐宝文.基于 One-test-at-a-time 策略的可变力度组合测试用例生成方法.计算机学报,2012,35(12):2541–2552.


**王燕**(1978－),女,湖南麻阳人,讲师,CCF 专业会员,主要研究领域为软件测试.

**吴化尧**(1989－),男,硕士,主要研究领域为组合测试,基于搜索的软件测试.

**聂长海**(1971－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件测试.



**徐家喜**(1972－),男,高级工程师,CCF 专业会员,主要研究领域为软件测试.

**钮鑫涛**(1988－),男,硕士,CCF 学生会员,主要研究领域为组合测试,软件测试,故障定位.