文章编号:1671-4598(2018)07-0026-06 DOI:10.16526/j.cnki.11—4762/tp.2018.07.006 中图分类号:TP311 文献标识码:A

基于 OTT 策略的可变力度组合测试用例优先级排序方法

张 娜[1],林青霞[1],吴 彪[2],金瑜婷[1],史佳炳[1]
(1. 浙江理工大学 信息学院,杭州 310018;2. 山口大学 东亚研究科,日本 山口 753—8513)

摘要:如何选取组合力度用于测试是可变力度组合测试用例优先级排序方法中迫于解决的关键问题;采用组合覆盖率为排序标准能够满足组合覆盖率,但其排序因素单一,测试用例优先级排序结果差异性较大,且无法根据测试结果反馈信息及时调整组合测试用例优先级;针对上述问题,该文结合 One—test—at—a—time (OTT) 策略、利用局部组合覆盖率、测试用例失效率和测试用例重要程度对组合测试用例优先级排序方法进行了研究;在测试过程中,实时关注测试用例的执行结果用于在线调整测试用例排序因素的取值情况,以达到实时更新组合测试用例优先级的目的;实验结果表明:相较于 Random、ICBP、GISVSP 和 LISVSP 方法,该方法使得组合测试用例优先级排序结果相对稳定,且在缺陷检测率上具有竞争力。

关键词:可变力度组合测试;测试用例;优先级

A Method Based OTT Strategy of Variable Strength Combinatorial Test Case Prioritization

Zhang Na[1], Lin Qingxia[1], Wu Biao[2], Jin Yuting[1], Shi Jiabing[1]
(1. School of Information, Zhejiang Sci—Tech University, Hangzhou 310018, China;
2. Graduate School of East Asian Studies, Yamaguchi University 753—8513, Japan)

Abstract: How to choose combinatorial strength for testing is a key problem that is urgently solved in the prioritized methodologies of variable strength combinatorial test cases. The combined coverage ratio can meet the combination coverage criteria, but the ranking factor is single, the test case prioritization results are quite different, and the priority of the combined test cases can't be adjusted timely according to the feedback information of the test results. In order to solve the above problems, this paper studies the prioritization method of combinatorial test cases based on One—test—at—a—time (OTT) strategy, using locally—interaction—coverage rate, test case failure rate and test case importance. In the testing process, the execution result is used to adjust the value of the sequencing factor achieve the purpose of updating the priority of the combined test case in real time. Experimental results show that compared with Random, ICBP, GISVSP and LISVSP methods, this method makes the prioritization results more stable, and the defect detection rate is competitive.

Keywords: variable strength combinatorial testing; test case; prioritization

0 引言

组合测试已广泛应用于软件测试中,该方法能够缩减测试用例的规模[1]。由于软件产品更新换代的频率逐渐上升,对组合测试用例进行完全测试的成本不断增加[2]。针对此问题,将优先级技术[3-4]引入到组合测试内,能够在软件测试过程中,提高测试效率。Kuhn 等人[5]发现组合测试中,两个参数相互组合所生成的用例可以检测出 70%的错误,90%以上的错误可由三个以内参数相互组合找出。围绕组合测试用例优先级排序问题已有相应研究:Bryce 等人[6]利用单一组合覆盖信息实现组合测试用例的排序问题;黄如兵等[7]从多重组合覆盖情况角度保证测试的有序进行;王子元等[8]提出了以组合权重和测试代价为标准的组合测试用例优先级排序的方法。

目前,针对可变力度的组合测试用例优先级排序方法的研究仍然较少,排序过程中组合力度选取困难,且当前固定力度组合测试用例的优先级排序方法无法满足复杂的交互关系,因此结合局部组合覆盖率、测试用例失效率和测试用例的重要程度,避免优先级排序因素单一;利用测试过程中的反馈信息,实现优先级在线排序;本文提出了一种基于 OTT 策略思想的可变力度组合测试用例优先级在线排序方法。

1 优先级排序因素

待测软件系统 (Software Under Test, SUT) 中,组合

收稿日期:2017-11-16; **修回日期**:2017-12-20。
基金项目:国家自然科学基金(61502430,61379036,61562015);浙江理工大学 521 人才培养计划(20150428)。
作者简介:张 娜(1977-),女,浙江宁波人,硕士,副教授,主要从事软件工程、软件测试方向的研究。

测试内部影响因子的相互关系并非完全一致，部分影响因子之间的相互作用可能更加紧密[9]，仅依靠固定力度的组合测试优先级排序方法无法满足影响因子之间这种复杂的交互关系。假设存在 n 个影响因素，这些影响因子构成一个有限集合 $F=\{f_1,f_2,\cdots f_n\}$，其中每个影响因素 f_i 的取值为 $v_i=\{p_1,p_2,\cdots p_j\}$。那么，SUT 的一条测试用例 $tc=\{x_1,x_2,\cdots x_n\}(x_1\in v_1,x_2\in v_2,\cdots x_n\in v_n)$。为了更好地描述可变力度组合测试用例优先级排序方法，对本文出现的相关概念进行如下描述：初始测试用例序列 S_{t_0} 表示组合测试用例排序之前，测试用例序列的集合，一般为 ϕ。初始测试用例集 $T_{t_0}=\{tc_1,tc_2,\cdots tc_N\}$ 是针对 SUT，根据组合测试方法及可变强度覆盖表 $VSCA(N,\lambda_m,n,F,CA(N',\lambda_s,n',F'))$ 生成的测试用例集合。显然，$1\leqslant\lambda_m<\lambda_s\leqslant|F'|<n$ 且 $F'\subseteq F$。测试用例序列 S_{t_i} 表示测试过程中，t_i 时刻已测试完毕的组合测试用例的有序集合。测试用例集 T_{t_i} 表示测试过程中，t_i 时刻尚未测试的组合测试用例的集合，或称为 t_i 时刻候选测试用例集。

可变力度的组合测试用例优先级中，需考虑组合覆盖率的影响。然而，如何在影响因子集下选取一个合适的组合力度作为衡量组合覆盖率的标准成为了难题。研究表明[7]：若选取 λ_m 为组合覆盖力度，当覆盖了所有 λ_m 组合后，优先级的选取演变成了随机排序方式；若选取 λ_s 为组合覆盖力度，则忽略了影响因子集 F' 中 λ_m 的组合覆盖情况；若选取的组合覆盖力度介于 λ_m 与 λ_s 之间，则面临了上述两种问题。故根据不同的组合覆盖力度，需考虑不同局部影响因子集下组合测试用例的组合覆盖率情况。测试用例的优先级排序中，用例的缺陷检测能力往往是一个衡量测试执行效率的重要标准，大量研究[10-11]中也通过观察测试执行的失效情况来评判测试用例的缺陷检测能力。而实际测试过程中，测试人员通常需要根据需求分析或概要设计来分辨测试用例在此次测试执行过程中的重要程度。为了尽快满足可变力度的组合覆盖要求和缺陷的检测能力，实现 SUT 设计文档的需求，本文引入局部组合覆盖率、测试用例失效率和测试用例重要程度这 3 个排序因素确定可变力度的组合测试用例的优先级问题。

1）局部组合覆盖率（Locally－Interaction－Coverage Rate，LICR），局部组合覆盖率是当前时刻下测试用例在局部影响因子集中覆盖所有 λ 元组合，且这些 λ 元组合尚未被测试用例序列覆盖的概率。

$$LICR_{(\lambda,t_i,F')}(tc,S_{t_{i-1}})=\frac{|CombSet_{(\lambda,F')}(tc)\cap UncovCombSet_{(\lambda,F')}(S_{t_{i-1}})|}{C_n^{\lambda}} \quad (1)$$

其中：$CombSet_{(\lambda,F')}(tc)$ 是测试用例 tc 在 F' 中覆盖所有 λ 元组合的集合，$UncovCombSet_{(\lambda,F')}(S_{t_{i-1}})$ 是测试用例序列 $S_{t_{i-1}}$ 在 F' 中未覆盖所有 λ 元组合的集合，n 为 F' 中影响因子的个数。当 $F'=F$ 时，局部组合覆盖率 $LICR_{(\lambda,t_i,F')}(tc,S_{t_{i-1}})$ 即为测试用例 tc 在 t_i 时刻的 λ 元组合覆盖率。

为了便于对局部组合覆盖率的理解，给出图 1 所示的计算实例。当前存在 5 个影响因子 $F=\{f_1,f_2,f_3,f_4,f_5\}$，每个因素分别有两种取值。$S_{t_3}$ 中已经执行了三个测试用例，T_{t_3} 中余下测试用例 tc_1 和 tc_2。t_4 时刻，$\lambda=2$ 时，局部影响因子集 $F'=\{f_1,f_3,f_4,f_5\}$ 分别计算 $LICR_{(2,t_4,F')}(tc_1,S_{t_3})$ 和 $LICR_{(2,t_4,F')}$（tc_2，S_{t_3}）。

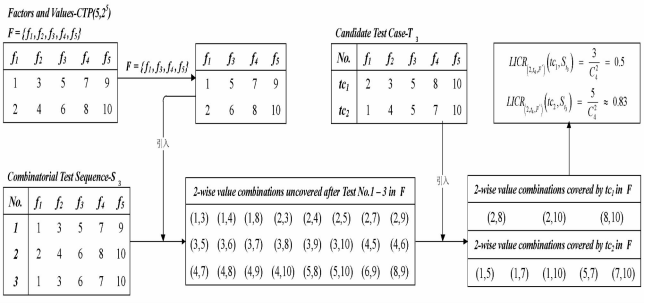

图 1　局部组合覆盖率计算实例

2）测试用例失效率（test case failure rate，FR），测试用例失效率是当前时刻下测试用例局部影响因子集 F' 中所有影响因子对应参数取值失效率的平均值。参数取值失效率 $FR_{(tc,t_i,F')}(f_k,p)$ 是 t_i 时刻测试用例 tc 在影响因子集 F' 中，f_k 对应取值为 p 的失效率。用 $m_{(tc,t_i,F')}$ 表示取值失效个数，即测试用例 tc 在局部影响因子集 F' 中，t_i 时刻参数取值失效率不为 0 的参数取值个数，即 $FR_{(tc,t_i,F')}(f_k,p)\neq 0$ 的参数取值个数。那么 t_i 时刻下，测试用例失效率 $FR_{(tc,t_i,F')}$ 计算公式如下：

$$FR_{(tc,t_i,F')}=\frac{\sum FR_{(tc,t_i,F')}(f_k,p)}{m_{(tc,t_i,F')}} \quad (2)$$

其中：$f_k\in F'$，$\sum FR_{(tc,t_i,F')}(f_k,p)$ 是计算局部影响因子集 F' 中所有 f_k 对应取值为 p 失效率之和。

为了清晰说明测试用例失效率的情况，给出如下示例。假设当前存在某个组合配置空间的影响因子集 $F=\{f_1,f_2,f_3,f_4,f_5\}$，其参数取值失效率情况如表 1。假设计算 t_{i-1} 时刻测试用例 $tc=\{5,1,2,3,4\}$ 在局部影响因子集 $F'=\{f_1,f_3,f_4,f_5\}$ 下的测试用例失效率 $FR_{(tc,t_{i-1},F')}=\frac{0.3+0.5+0.0+0.8}{3}\approx 0.533$。

表 1　某组合空间配置在 t_{i-1} 时刻参数取值失效率的情况

参数	f_1	f_2	f_3	f_4	f_5
p_0	0(0.1)	1(0.4)	2(0.5)	3(0.0)	4(0.8)
p_1	5(0.3)	6(0.2)	7(0.3)	8(0.1)	
p_2	9(0.6)				

2　优先级排序因素在线调整策略

2.1　局部组合覆盖率调整方法设计

测试过程中，测试用例序列和测试用例集中测试用例的个数不断发生变化，使得局部组合覆盖率也不断发生变化。优先执行局部组合覆盖率较大的测试用例，能够保证尽快满

足组合覆盖率的要求。若 $t_{i-1}(i \geqslant 1)$ 时刻所选取的测试用例为 $A_{t_{i-1}}$，那么在该测试用例执行测试后，测试用例序列 $S_{t_{i-1}}$ 和测试用例集 $T_{t_{i-1}}$ 都发生了改变，进而得到每个测试用例在当前时刻下的覆盖组合集 CombSet 和未覆盖组合集 UncovCombSet，并为下一次求解局部组合覆盖率做准备。

其中，t_i 时刻的测试用例序列 S_{t_i} 以 t_{i-1} 时刻执行的测试用例 $A_{t_{i-1}}$ 顺序插入到 t_{i-1} 时刻的测试序列 $S_{t_{i-1}}$ 的形式表示，如公式（3）所示。

$$S_{t_i} = S_{t_{i-1}} \succ A_{t_{i-1}}, A_{t_{i-1}} \in T_{t_{i-1}} \tag{3}$$

t_i 时刻的测试用例集 T_{t_i} 通过测试用例 $A_{t_{i-1}}$ 从 t_{i-1} 时刻的测试用例集 $T_{t_{i-1}}$ 中移除得到，如公式（4）所示。

$$T_{t_i} = T_{t_{i-1}} - A_{t_{i-1}}, A_{t_{i-1}} \in T_{t_{i-1}} \tag{4}$$

t_i 时刻，未覆盖 λ 元组合集 $UncovCombSet_{(\lambda,F')}(S_{t_i})$ 是通过移除测试用例 $A_{t_{i-1}}$ 在 t_{i-1} 时刻覆盖了测试用例序列 S_{t_i} 尚未覆盖的 λ 元组合集而得，如公式（5）所示。

$$UncovCombSet_{(\lambda,F')}(S_{t_i}) = UncovCombSet_{(\lambda,F')}(S_{t_{i-1}}) - (UncovCombSet_{(\lambda,F')}(S_{t_{i-1}}) \cap CombSet_{(\lambda,F')}(A_{t_{i-1}})), \quad A_{t_{i-1}} \in T_{t_{i-1}} \tag{5}$$

每执行一个测试用例，都需要对当前测试用例序列、测试用例集和每个测试用例的局部组合覆盖率了进行更新，执行完测试用例 $A_{t_{i-1}}$ 后，每个测试用例的局部组合覆盖率的计算公式如（6）所示。

$$LICR_{(\lambda,t_i,F')}(tc, S_{t_i}) = \frac{|CombSet_{(\lambda,F')}(tc) \cap UncovCombSet_{(\lambda,F')}(S_{t_i})|}{C_n^{\lambda}} \tag{6}$$

2.2 测试用例失效率调整方法设计

实际测试中，当前测试用例的执行结果能够反馈出 SUT 存在的问题。研究表明，相同组合影响因子的参数取值引起的失效，可能隐藏了更多的错误，并且可以会检测出相同或者类似的错误[12-13]。测试用例的执行能够发现存在的错误与缺陷，那么测试用例在当前选取的局部影响因子集 F' 中所覆盖的参数取值的失效率需要做出相应的调整，以保证测试用例失效率能够实时计算，确保最终优先级排序的准确性。若 $t_{i-1}(i \geqslant 1)$ 时刻，测试用例 $C_{t_{i-1}}$ 检测出 SUT 中存在缺陷，测试结果只能反应软件失效，但无法判断究竟是由哪些参数相互作用引发的失效。因此，只能对 $C_{t_{i-1}}$ 覆盖部分影响因子集 F' 中所有参数取值的失效率相应增加，其他参数取值的失效率保持不变。则 t_i 时刻，各参数取值失效率可用以下公式进行调整：

$$FR_{(tc,t_i,F')}(f_k,p) = \begin{cases} FR_{(tc,t_{i-1},F')}(f_k,p) + \Delta c & p \in C_{t_{i-1}} \\ FR_{(tc,t_{i-1},F')}(f_k,p) & otherwise \end{cases} \tag{7}$$

其中：Δc 是一个较小的常量。

若 t_{i-1} 时刻，测试用例 $C_{t_{i-1}}$ 未检测出 SUT 中存在缺陷，测试结果能够反应出当前测试用例中所有参数取值不会对 SUT 造成缺陷，则该测试用例覆盖局部影响因子集 F' 中所有参数取值的失效率变为 0，其他参数取值的失效率保持不变。则 t_i 时刻，各参数取值失效率可用以下公式进行调整：

$$FR_{(tc,t_i,F')}(f_k,p) = \begin{cases} 0 & p \in C_{t_{i-1}} \\ FR_{(tc,t_{i-1},F')}(f_k,p) & otherwise \end{cases} \tag{8}$$

为了便于对局部影响因子集 F' 中参数取值失效率在线调整的过程，通过以下示例对其做出解释。参数取值失效率情况见表 1。若当前测试用例 $tc = \{5,1,2,8,4\}$ 在 t_{i-1} 时刻检测出 SUT 中存在缺陷，测试时对应的局部影响因子集 $F' = \{f_1, f_3, f_4, f_5\}$，此时需要对测试用例 tc 覆盖部分影响因子集 F' 中所有参数取值的失效率相应增加 Δc，故这个组合空间配置参加取值失效率发生变化，则 t_i 时刻参数取值失效率的情况如表 2；同理，若 tc 未检测出缺陷，相应参数取值失效率变为 0 即可。

表 2 某组合空间配置在 t_i 时刻的参数取值失效率情况

参数	f_1	f_2	f_3	f_4	f_5
p_0	0(0.1)	1(0.4)	2(0.5+Δc)	3(0.0)	4(0.8+Δc)
p_1	5(0.3+Δc)	6(0.2)	7(0.3)	8(0.1+Δc)	
p_2	9(0.6)				

3 组合测试用例优先级排序方法分析

3.1 OTT 基本算法框架

在组合测试过程中，由于 OTT 策略高效、简单、便于扩展等特点[14]，使得该策略在组合测试中得到了广泛的应用，其在组合测试用例优先级排序中的作用也不容忽视。张娜等人[15]在固定力度组合测试用例优先级排序算法中，结合了 OTT 策略，使得排序后的测试用例具有更强的缺陷检测能力。为此，有必要进一步对 OTT 策略在可变力度的组合测试用例优先级排序中的应用进行研究。Cohen 等人[16]在组合测试的研究中，利用 OTT 策略构建了 Greedy 算法的框架。该策略为一维扩展机制，本文结合该策略构建了可变力度组合测试用例优先级排序的算法框架，即每次选取当前优先级最高的测试用例用于执行。本文 OTT 策略的可变力度组合测试用例优先级排序算法框架流程图见图 2。

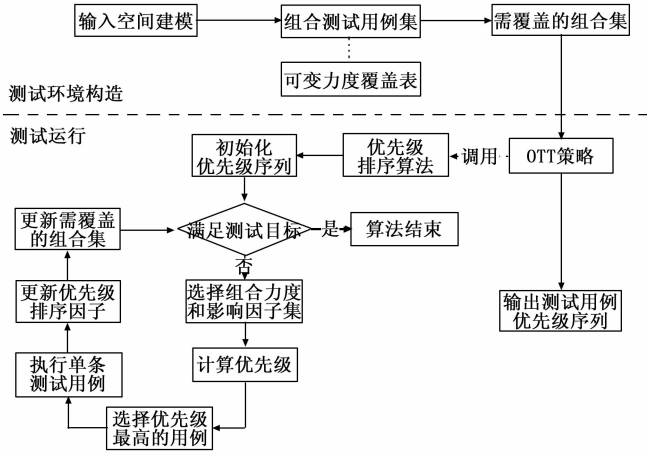

图 2 OTT 策略基本算法框架流程图

3.2 优先级排序方法设计

随着软件版本更新升级速度的提升，对测试效率的要

求也随之增高。由于资源时间等的限制，目前的情形无法达到完全测试的目标，那么如何选取测试用例用于执行、测试用例的执行顺序显得尤为重要。本文结合 OTT 策略将可变力度的组合测试用例优先级排序方法转换成综合考虑多个优先级排序影响因素共同计算优先级、每次选取当前优先级最高的用例用于执行的问题。为了尽快满足可变力度组合测试在不同影响因子集下的组合覆盖率，尽可能选取局部组合覆盖率较高的测试用例；同时，提高缺陷检测的能力，要求优先考虑测试用例失效率较高的测试用例；并且，测试用例的设计和生成过程中，本身存在一定的优先级，测试人员可结合需求设计文档或凭借测试经验，给组合测试用例赋予一定的重要程度。那么，不同参数取值的重要程度本身存在差异，使用参数取值重要程度的加权平均值作为测试用例重要程度的计算。根据 OTT 策略基本算法框架扩展组合测试用例优先级排序算法，每次选择出当前优先级最高的测试用例。测试用例重要程度 $I_{(tc,F')}$ 是测试用例 tc 的重要程度，即测试用例在局部影响因子集 F' 所有影响因子对应取值权重的平均值。参数取值重要程度 $\omega(f_k,p)$ 是影响因子 f_k 中，取值为 p 的重要程度。

那么，测试用例 tc 在 t_i 时刻的优先级 Pr 由测试用例的局部组合覆盖率、测试用例失效率和测试用例重要程度这三个排序因素共同决定，并利用权重因子，以便不同测试环境下对这三个排序因素比重做出相应调整，保证该方法能够广泛的用于软件测试中。优先级计算方法如下所示：

$$Pr_{(tc,t_i)} = \alpha \cdot LICR_{(\lambda,t_i,F')}(tc,S_{t_{i-1}}) + \beta \cdot FR_{(tc,t_i,F')} + \gamma \cdot I_{(tc,F')} \tag{9}$$

其中：α、β、γ 分别表示局部组合覆盖率、测试用例失效率和测试用例重要程度的权重因子，实际测试过程中可根据具体情况相应调整，只有保证 $\alpha+\beta+\gamma=1$ 即可。测试用例的重要程度使用每个参数取值重要程度的平均值进行计算。通过上述公式的描述，测试用例 tc 在 t_i 时刻的优先级 $Pr_{(tc,t_i)}$ 实际可以表示成如下形式：

$$Pr_{(tc,t_i)} = \alpha \cdot \frac{|\, CombSet_{(\lambda,F')}(tc) \cap UncovCombSet_{(\lambda,F')}(S_{t_{i-1}}) \,|}{C_n^{\lambda}} + \beta \cdot \frac{\sum FR_{(tc,t_i,F')}(f_k,p)}{m_{(tc,t_i,F')}} + \gamma \cdot \frac{\sum \omega(f_k,p)}{n} \tag{10}$$

其中：$f_k \in F'$，n 为 F' 中影响因子的个数，$\sum \omega(f_k,p)$ 是计算计算局部影响因子集 F' 中所有 f_k 对应取值为 p 重要程度之和。

测试用例 tc 在 t_i 时刻优先级 Pr 计算过程中，还应当考虑当前影响因子集如何选取的问题，此问题是可变力度的组合测试用例优先级计算时必须着重讨论的。t_i 时刻，当所有 λ_m 元组合还未被测试用例序列 $S_{t_{i-1}}$ 所覆盖时，考虑影响因子集 F 上 λ_m 元的组合覆盖率；若此时所有 λ_m 元组合均被测试用例序列 $S_{t_{i-1}}$ 所覆盖，则不再考虑影响因子集 F 上 λ_s 元组合覆盖情况，而是仅考虑影响因子集 F' 上 λ_s 元的组合覆盖率。

基于 OTT 策略的可变力度组合测试用例优先级排序算法（Variable Combinatorial Test Case Prioritization Based Strategy of One－test－at－a－time，VCPO）如下所示。

算法 1：VCPO 算法。

输入：初始测试用例集 T_{t_0}，组合测试覆盖表 $VSCA(N,\lambda_m,n,F,CA(N',\lambda_s,n',F'))$。

输出：测试用例序列 S_{t_i}。

1) $S_{t_0}=\phi$; $i=0$;

2)while $|S_{t_i}| \neq N$ do//未达到测试目标

3) $highest=0$;

4) $TestCase=\phi$; //当前优先级最高的测试用例集

5) $R=F$; $\lambda=\lambda_m$;

6)for each element $e \in (T_{t_i} \backslash\backslash S_{t_i})$ do

// e 在(初始)候选测试用例集 T_{t_i} 中，但不在测试用例序列 S_{t_i} 中

7)if $R==F$ && $\lambda==\lambda_m$ && $UncovCombSet_{(\lambda,R)}(S_t)\neq\phi$

//根据测试用例序列 S_{t_i} 中覆盖组合情况，选择影响因子集和测试力度

8) $R=F$; $\lambda=\lambda_m$;

9)else

10) $R=F'$; $\lambda=\lambda_s$;

11) $priority=\alpha \cdot LICR_{(\lambda,t_{i+1},R)}(e,S_{t_i})+\beta \cdot FR_{(e,t_{i+1},R)}+\gamma \cdot I_{(e,R)}$;

12)if $priority \geqslant highest$ then

13) $TestCase=\phi$;

14) $highest=priority$;

15) $TestCase=TestCase \cup \{e\}$;

16)end if

17)end for

18) tc = random($TestCase$); //在 $TestCase$ 随机选取一个测试用例

19)执行测试用例 tc;

20)if tc is false then //测试用例检测出缺陷，按照公式(7)调整失效率

21)for each element $f_k \in R$ do

22)for each element $p \in f_k$ do

23)if $p \in tc$ then

$FR_{(tc,t_{i+1},R)}(f_k,p)=FR_{(tc,t_i,R)}(f_k,p)+\Delta c$;

24)else $FR_{(tc,t_{i+1},R)}(f_k,p)=FR_{(tc,t_i,R)}(f_k,p)$;

25)end for

26)end for

27)else //测试用例未检测出缺陷，按照公式(8)调整失效率

28)for each element $f_k \in R$ do

29)for each element $p \in f_k$ do

30)if $p \in tc$ then $FR_{(tc,t_{i+1},R)}(f_k,p)=0$;

31)else $FR_{(tc,t_{i+1},R)}(f_k,p)=FR_{(tc,t_i,R)}(f_k,p)$;

32)end for

33)end for

34) i++;

35) $S_{t_i} = S_{t_{i-1}} \succ \{tc\}$; //将测试用例 tc 有序插入到测试用例序列 S 中

36) $T_{t_i} = T_{t_{i-1}} - \{tc\}$; //将测试用例 tc 从测试用例集 T 中删除

37) $UncovCombSet_{(\lambda,R)}(S_{t_i}) = UncovCombSet_{(\lambda,R)}(S_{t_{i-1}}) - (UncovCombSet_{(\lambda,R)}(S_{t_{i-1}}) \cap CombSet_{(\lambda,R)}(tc))$;

38)end while

39)return S_{t_i};

4 实验与总计

4.1 实验数据与结果分析

为了验证所述方法的有效性，选取以下两个组合空间配置 $VSCA1(N,\lambda_m,8,4^3 5^3 6^2,CA(N',\lambda_s,7,4^3 5^3 6^1))$ 和 $VSCA2(N,\lambda_m,9,10^1 9^1 8^1 7^1 6^1 5^1 4^1 3^1 2^1,CA(N',\lambda_s,5,6^1 5^1 4^1 3^1 2^1))$，分别采用 Random、ICBP、GISVSP、LISVSP 和本文所述方法对其缺陷检测能力进行比较。由于实际测试过程中，无法执行组合测试用例集里所有的测试用例，且组合测试用例集不能检测出 SUT 中所有的缺陷，故缺陷检测能力采用文献［17］提出的标准化的测试用例序列度量标准（NAPFD）。$VSCA1$ 和 $VSCA2$ 在 ACTS 工具下生成的测试用例情况如表 3。

表 3 ACTS 生成的可变力度组合测试用例集大小

VSCA	缩写	λ_s	用例个数
VSCA1 ($\lambda_m = 2$)	VSCA1－A	3	207
	VSCA1－B	4	1062
VSCA2 ($\lambda_m = 2$)	VSCA2－A	3	128
	VSCA2－B	4	361

对于上述方法及组合空间配置的情况，为取得适合本文组合空间配置的权重因子 α、β、γ，多次实验以调整优先级。并且在测试过程中，每组权重因子下，分别多次模拟测试，记录每次测试的运行结果，用于计算上述方法的 NAPFD 值。实验对比后，本文对 α、β、γ 的取值分别为 0.41、0.35、0.24，且对比每种方法在实验中 NAPFD 的最大值和平均值。测试过程中某个测试用例发现缺陷时，则该测试用例覆盖影响因子集中的所有参数取值失效率需相应增加 Δc，实验中 Δc 的取值为 0.1。ICBP 方法是固定力度组合测试用例优先级排序方法，则选取不同组合力度，以观察力度选取的影响。实验结果如表 4。

上述表中，ICBP－$\lambda = n$ 为 ICBP 方法在测试时，选择组合力度为 n 的情况；下划线、加粗为每行中最大值；下划线、斜体为每行中次最大值；“－”为不存在的情况，无实验数据。

从表中描述的所有实验结果的 Max 和 Avg 可以看到，Random、ICBP 方法，在可变力度组合测试的缺陷检测能力上几乎无任何优势。可变力度的组合测试用例优先级排序方法中，GISVSP 方法相较于 LISVSP 方法有更高的最大值；VCPO 方法则在最大值和平均值上都占有一定优势，且在每行的最大值和次最大值中都能够有较高的 NAPFD 值，因此缺陷检测能力相对较高。

因此可得到如下结论：

1）Random 和固定力度组合测试优先级排序方法 ICBP 在可变力度组合测试优先级排序中不适用，原因在于无法选取合适的组合力度用于测试；

2）可变力度的组合测试用例优先级排序方法中，GISVSP 方法和 LISVSP 方法不会面临组合力度选取困难的问题，能够适用于可变力度的组合测试优先级的排序，但其仅考虑组合覆盖率的情况，排序因素单一，故缺陷检测能力没有表现为最优。

3）可变力度的组合测试用例优先级排序方法中，VCPO 方法能够解决组合力度选取的难题和排序因素单一的问题，使得排序结果相对稳定。实验结果表明，该方法在大多数情况下缺陷检测能力比其它排序方法更优。

4.2 总结

目前组合测试领域中，因实际测试中各影响因子间相对复杂的交互关系，可变力度的组合测试备受青睐，众多研究者关注于可变力度的组合测试用例优先级排序的问题。本文提出了一种基于 OTT 策略的可变力度组合测试用例优先级排序方法，该方法能够更为广泛的适用于可变力度的组合测试中，且结合了多个排序因素，利用在线的测试反馈信息，在符合实际的测试过程的前提下，进一步提升了缺陷检测能力。

由于测试过程中，本文提出的方法存在不够理想的实验结果，且 α、β、γ 的权重比值需人为分配，实验中 Δc 的取

表 4 五种优先级排序算法在不同组合空间配置下的 NAPFD(%)

VSCA	准则	Random	ICBP－$\lambda=2$	ICBP－$\lambda=3$	ICBP－$\lambda=4$	GISVSP	LISVSP	VCPO
VSCA1－A	Max	92.50	92.76	93.16	—	93.14	93.12	93.27
	Avg	85.83	88.08	88.54	—	88.59	88.52	88.80
VSCA1－B	Max	97.12	97.56	97.89	97.69	98.17	97.90	98.08
	Avg	94.43	94.78	95.88	95.29	95.83	95.97	96.16
VSCA2－A	Max	86.83	89.31	88.27	—	89.34	89.35	89.30
	Avg	79.72	82.30	82.21	—	83.63	82.91	84.14
VSCA2－B	Max	91.69	93.27	94.15	94.05	94.19	93.84	93.95
	Avg	85.70	89.19	89.36	88.41	89.55	89.77	90.05

值相对固定，后期希望能够选取更合适的排序因素且结合自适应的方法进行在线调整权重因子和 Δc 的取值。

参考文献：

[1] 毛澄映，喻新欣，薛云志，等. 基于粒子群优化的测试数据生成及其实证分析 [J]. 计算机研究与发展，2014，51 (4)：824－837.

[2] 张　娜，姚　澜，包晓安，等. 多目标的测试用例优先级在线调整策略 [J]. 软件学报，2015，26 (10)：2451－2464.

[3] Yoo S，Harman M. Regression testing minimization，selection and prioritization：A survey [J]. Software Testing，Verification and Reliability，2012，22 (2)：67－120.

[4] Bryce，Ren& C E，Colbourn C J. Constructing interaction test suites with greedy algorithms [A]. IEEE/ACM International Conference on Automated Software Engineering. DBLP [C]. 2005：440－443.

[5] Kuhn D R，Reilly M J. An Investigation of the Applicability of Design of Experiments to Software Testing [A]. Software Engineering Workshop，2002. Proceedings. NASA Goddard/IEEE [C]. IEEE，2002：91－95.

[6] Bryce R C，Sampath S，Memon A M. Developing a Single Model and Test Prioritization Strategies for Event－Driven Software [J]. IEEE Transactions on Software Engineering，2010，37 (1)：48－64.

[7] 黄如兵. 组合测试用例的自适应随机生成与优先级排序方法研究 [D]. 武汉：华中科技大学，2013.

[8] Wang Ziyuan，Chen Lin，Xu Baowen，et al. Cost－Cognizant Combinatorial Test Case Prioritization [J]. International Journal of Software Engineering & Knowledge Engineering，2011，21 (6)：829－854.

[9] 王子元，钱　巨，陈　林，等. 基于 One－test－at－a－time 策略的可变力度组合测试用例生成方法 [J]. 计算机学报，2012，35 (12)：2541－2552.

[10] 董　萌，包晓安，张　娜，等. 基于缺陷传递的缺陷关联系数调整策略研究 [J]. 浙江理工大学学报，2017，37 (2)：232－236.

[11] 雷　晏. 基于关联性分析的缺陷定位技术研究 [D]. 长沙：国防科学技术大学，2014.

[12] Chen T Y，Huang D H，Zhou Z Q. Adaptive Random Testing Through Iterative Partitioning [J]. Journal of Information Science & Engineering，2006，27 (4)：155－166.

[13] 包晓安，谢晓鸣，张　娜，等. 基于缺陷关联度的 Markov 模型软件优化测试策略 [J]. 软件学报，2015，26 (1)：14－25.

[14] Bryce，Ren& C E，Colbourn C J. Constructing interaction test suites with greedy algorithms [A]. IEEE/ACM International Conference on Automated Software Engineering. DBLP [C]. 2005：440－443.

[15] 张　娜，林青霞，包晓安，等. 基于 OTT 策略的组合测试用例优先级排序方法 [J]. 计算机工程与应用，2017.

[16] Cohen M B，Colbourn C J，Bryce R C. A framework of greedy methods for constructing interaction test suites [A]. International Conference on Software Engineering [C]. 2005. ICSE 2005. Proceedings. 2005：146－155.

[17] Qu X，Cohen M B，Woolf K M. Combinatorial Interaction Regression Testing：A Study of Test Case Generation and Prioritization [C]. IEEE International Conference on Software Maintenance. IEEE，2007：255－264.

(上接第 15 页)

在装备维修决策方面，通过流程再造和建立维修决策模型，实现了多个指标自主优化决策。装备工作流程再造后，我们将每次的维修情况作为历史数据纳入数据库，并将这些数据库作为下次维修决策的主要依据，根据维修情况重新修正其物理模型，从而得到更加准确、真实的预测寿命及发展衰退趋势。

3）实现了风洞装备日常管理由人工模式向自动化信息化的转变。

通过 2.4 米跨声速风洞自主式维修保障系统，实现了风洞状态自动巡检，实现了由分散式、人工的检查向自动化的智能巡检转变，大大提高了试验运行效率，节省了试验准备时间。为进一步提高信息化水平，本系统建设时，还重点开展了自主式维修保障业务管理平台建设工作，系统平台遵循一体化设计的思路，实现了试验装备的三化管理、实力管理、日常管理、动力运行、电子档案、合同管理以及各种法规制度的统一管理等功能。在信息录入方面，采用了多种简化操作、方便使用的措施，实现了信息自动入库，大大提高了自动化、信息化水平。

5　结束语

自主式维修保障系统是为解决大型风洞繁重的试验任务与设备维修导致停机之间的迫切矛盾而设计和搭建的，目前系统已经投入运行使用，运行状态稳定可靠。系统满足了大型风洞在状态监测、故障诊断、故障和剩余寿命预测功能上的需求，显著提高了风洞的试验效率，保障了各科研型号在风洞中的试验任务，也为自主式维修保障系统在其他大型风洞中的应用打下了良好的基础。

参考文献：

[1] 张宝珍. 自主式保障—信息时代武器装备保障新模式 [A]. 论文集 [C]. 北京：武器装备综合保障信息化技术研讨会，2009.

[2] 曹艳华，等. 装甲装备自主式保障关键要素分析 [D]. 北京：装甲兵工程学院学报，2010.

[3] Dreyer SL. Autonomic Logistics－Developing an Implementation Approach for an Existing Military Weapon System [J]. IEEE Instrumentation & Measurement Magazine，2006，9 (4)：16－21.

[4] 刘龙兵，郁文山，杨兴锐，等. 基于 LabVIEW 的状态监测系统在 2.4 m 风洞的设计与实现 [J]. 自动化与仪器仪表，2016 (210)：62－64.

[5] 郁文山，易　凡，蔺元臣，等. 基于 RBR 和 PCA 的 2.4 米风洞故障诊断系统应用研究 [J]. 计算机测量与控制，2016 (7)：31－34.

[6] 张金玉，张　炜. 装备智能故障诊断与预测 [M]. 北京：国防工业出版社，2013.