

组合的测试用例设计与评价方法

浦云明

(集美大学 计算机工程学院,福建 厦门 361021)

摘要:在软件开发过程中,软件测试成为唯一的选择,在保证软件质量前提下,设计与工程人员的首要工作就是降低测试用例数量,优化测试用例设计,减轻测试工作量。提出组合测试技术和正交实验方法在软件测试中的应用方法,分析了成对组合的测试用例方法,设计了测试用例的生成方法,设计测试评价方法。在一应用系统的负载测试实验中,使用 R 法和方差分析方法,分析了网络带宽、处理器速度、并发用户等负载因素对系统响应时间的影响,确定最优的性能指标和测试组合。

关键词:组合测试;评价方法;性能指标;测试用例;正交实验

中图分类号:TP311

文献标识码:A

文章编号:1673-629X(2014)03-0117-05

doi:10.3969/j.issn.1673-629X.2014.03.030

Method of Combinatorial Test Case Design and Assessment

PU Yun-ming

(School of Computer Engineering, Jimei University, Xiamen 361021, China)

Abstract: The software testing becomes the necessary choice in software development. For the assurance of software quality, the mainly work for the software designer and engineer is to decrease the quantity of test case, optimize the design of test case, and low the testing work load. The combinatorial testing technology and the orthogonal experiment method were provided. The pair-wise of test case design method was analyzed, designing the method for the test case and the assessment criteria of test case. In a load testing solution for one application system, using R and variance analysis method, the load testing factor to one system response time was analyzed, including network bandwidth, processor speed and concurrency users etc., the optimization performance criteria to the system was determined, and the combinatorial testing factors were selected.

Key words: combinatorial testing; assessment method; performance criteria; test case; orthogonal experiment

0 引言

可靠性的软件设计应避免缺陷的产生和消除已经产生的缺陷,使程序中的缺陷密度达到业界和用户可以承受的程度。20 世纪的软件危机以来,软件业界做了各种尝试,包括程序语言的更新换代,结构化设计方法,面向对象设计方法,形式化描述语言,可视化设计工具等^[1]。理论界试图用形式化证明方法来证明程序的正确性,数学家将程序作为对象,采用数学方法证明程序的正确性,并发表了大量相关的论文,但实际作用有限,在形式化需求描述语言不完善的情况下,形式化证明方法只有在代码完成后才能使用,对于一些大的程序也无法证明其正确性^[2]。因此,在软件开发过程中,软件测试成为唯一的选择,工程项目也证明,测试能发现缺陷、定位缺陷,提高软件质量,软件测试必须

贯穿于整个开发过程中。

1992 年, Brownlie 使用正交表对 PMX/StarMail 系统进行测试研究,检测到了以前没有发现的软件缺陷。1998 年,北卡罗来纳大学计算机系的 Kuo-Chung Tai 等人提出了启发式算法(IPO),根据该算法实现了相应测试用例自动生成系统 Pairwise Test^[3]。陈皓等人也提出了成对用于组合测试数据的生成方法,以及相关优化处理技术^[4]。

2001 年, Salem 在其博士论文中系统研究了如何将实验设计方法应用于软件测试。2002 年, Kuhn 和 Reilly 研究了组合测试的可用性,发现大约 70 % 的故障是由两个以下参数的相互作用引起的,由此说明了两两组合测试的应用价值^[5]。

2003 年, Cohen M. B 等人将模拟退火算法应用于

收稿日期:2013-05-24

修回日期:2013-08-28

网络出版时间:2014-01-07

基金项目:福建省基金项目(2013J01243, 2013H0032)

作者简介:浦云明(1966-),男,CCF 会员,研究方向为软件测试技术、嵌入式系统。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140107.1723.046.html>

多维组合测试用例的生成^[6]。2004 年, Kuhn 等人对 Mozilla 浏览器的测试中, Kuhn 和 Reilly 发现 70% 的错误是由两个参数相互作用引起的, 90% 的错误是由 3 个以内参数造成的, 因此, 选择测试用例, 使得任意 $t(t \leq 3)$ 个参数, 其所有可能取值的组合至少被一个测试用例覆盖。在大规模分布式系统中组合测试可用性的研究中, 发现此类系统中的故障一般最多由 4~6 个参数的相互作用引发。同年, Kobayashi 等人研究了遗传算法和蚁群算法在组合测试数据自动生成中的应用^[7]。2005 年, Colbourn 和 Sherwood 等人研究了两种组合覆盖表的生成方法, 分别应用于具有不同参数取值的两两组合覆盖表和高维组合覆盖表的生成^[8]。

测试能发现引入的错误, 也能估计软件错误密度, 确保软件满足应用需求及相应的开发标准。测试用例是软件测试执行的最小单元, 是为特定目的而设计的带有唯一标识的一组输入、执行条件和预期结果。测试用例构成了设计和制订测试过程的基础, 测试用例的质量决定了测试工作的效率。在实际工作中, 测试工程师面对大量甚至爆炸的测试用例, 有时无所适从, 有时痛苦难忍, 因此, 探寻优化方法、优化测试用例, 减轻负担, 提高测试效率, 是测试工程师迫切需要面对的技术问题。因此, 针对测试用例的设计和优化工作, 提出正交实验设计法、均匀实验设计法和成对组合覆盖测试技术等。

1 组合测试方法

在测试过程中, 采用不同的软件测试技术, 无限的输入空间可以用有限的输入空间来表示, 从而在保证测试效果的前提下, 在一定程度上解决测试用例爆炸问题。输入的不同组合会直接影响软件的执行效率, 软件测试用例的设计就必须先解决输入组合的问题。

20 世纪 90 年代以来, 出现了许多组合测试的研究成果, 可分为两大类, 一类是非确定性组合方法, 包括随机选择测试用例、启发式算法、约束数组测试系统等; 另一类是确定性组合测试方法, 包括单因素组合法、部分成对组合法、全组合法、最大距离组合法等^[9]。

(1) 单因素组合法, 选择测试用例, 每个因素的水平至少覆盖 1 次, 满足单因素组合覆盖标准, 每次选择没有使用过的水平值作为该因素的取值。

(2) 部分成对组合法, 确定 2 个最重要的因素, 产生覆盖这 2 个因素所有水平的组合, 并确保覆盖其他因素的每个水平至少 1 次的测试用例。

(3) 全组合法, 该方法产生的测试用例覆盖全部因素水平组合。

(4) 最大距离组合法, 使用二进制向量对因素和水平进行编码, 选择与已经选择的测试用例的汉明码距离最远或笛卡尔长度最大的测试用例。

成对组合 (pair-wise), 又称为两两组合, 是将所有因素的水平按照两两组合的原则产生测试用例的技术, 是一种经济、有效的测试技术。R. Mandl 于 1985 年在测试 Ada 编译程序时提出并发表了第一篇组合测试论文, Cohen 等人应用成对组合测试技术对 UNIX 的 Sort 命令进行了测试, 测试结果表明, 模块覆盖率为 93.5%, 判断覆盖率为 83%, 计算引用 (C-Use) 覆盖率为 76%, 判断引用 (P-Use) 覆盖率为 73.5%。

1.1 用例的成对组合表示

在组合形式化的描述中, k 表示参数个数, 各个参数相互独立 (参数的取值不影响其他参数的取值), 第 i 个参数有 n_i 个可能值, 不失一般性, 假设 $n_0 \geq n_1 \geq \dots \geq n_{k-1}$, v_i 表示第 i 个参数的某个取值 (水平), 测试用例可以用 t 表示, $t = (v_0, v_1, \dots, v_{k-1})$, 那么所有可能的测试用例 $T = \{t = (v_0, v_1, \dots, v_{k-1}) \mid \}$, S 表示选定的某个测试用例集, $S \subseteq T$ 。因此, 在理想情况下, 测试用例个数为 $|T| = n_0 * n_1 * \dots * n_{k-1}$, 当参数个数 k 或参数的取值 n_i 较大时, 测试的数量就非常庞大, 有时也无法实现全组合测试, 如何选择其中的部分进行测试就成为关键^[10]。

定义 1: 集合 $S = \{(v_{i1}, v_{j1}) \mid 1 \leq i < k, 2 \leq j < k\}$, 二元组合集合 S 为测试用例 $T = (v_{i1}, v_{21}, \dots, v_{k1})$ 的成对组合表示。

例 1, 假设系统有 3 个因素 A、B 和 C, 每个因素有 3 个水平, 分别为 $x_1, x_2, x_3, y_1, y_2, y_3, z_1, z_2, z_3$, 对应测试用例 $t = (x_1, y_1, z_1)$, 该测试用例的成对组合表示是: $S = \{(x_1, y_1), (x_1, z_1), (y_1, z_1)\}$, 对应测试用例 $t = (x_2, y_2, z_2)$, 该测试用例的成对组合表示是: $S = \{(x_2, y_2), (x_2, z_2), (y_2, z_2)\}$ 。

定义 2: 系统有 k 个因素, 每个因素的水平为 n , 在等水平假设下, 则全组合数 $T = nk$, 系统全覆盖的最小测试用例数量为 $n2$ 。

定义 3: 覆盖数组 $CA(N; t, k, v)$ 是一个值域大小为 v 的 $N \times k$ 矩阵, 任意的 $N \times t$ 子矩阵包含了在 v 值域上所有大小为 t 的排列。 t 称为强度或水平, k 被称为阶数, v 称为序。一个覆盖数组如果具有最小的行数, 则被称为最优的, 这个最小的行数称为覆盖数, 记为 $CAN(t, k, v)$ 。

通常, 强度为 t 的覆盖数组被称为 t 覆盖数组。当 $t = 2$, 覆盖数组称为成对覆盖数组, 覆盖数组要求矩阵的每一列具有相同大小的值域^[11]。当错误被检出后, 如何利用运行失败的测试用例进行错误定位是软件测试过程中必不可少的一个环节。相对于程序所有参数

取值的 t 组合,组合测试用例的数量要小很多,Kuhn 等人的研究表明,采用覆盖数组的组合测试的错误定位能力,明显优于分类树等其他分析方法^[12]。

1.2 组合算法

选择一个空测试用例集合,根据贪心算法产生一组候选用例,从中选择能够覆盖最多未被覆盖的成对组合的用例,每次给测试用例集加入一个测试用例。

假设一应用系统有 k 个因素 m_1, m_2, \dots, m_k , 第 i 个因素有 v_i 个不同水平。设计算法如下:

(1) 成对组合集合 $\pi \neq \emptyset$, 选择因素 m , 使其水平 v 在未覆盖的成对组合中出现的次数最多;

(2) 令 $m_1 = m$, 对剩余因素进行随机排序, 得到 k 个因素的新排序 m_1, m_2, \dots, m_k ; 假设因素 m_1, m_2, \dots, m_j 的水平已选定, $1 \leq i \leq j$, 因素 m_i 选定的水平为 v_i , 按照如下方法选定因素 m_{j+1} 及其水平 v_{j+1} , 对 m_j 的每个水平 v , 查找所有成对组合 $\{(m_{j+1} = v, m_i = v_i), 1 \leq i \leq j\}$, 这些组合中出现次数最多的 v_i 作为 v_{j+1} , 并且每个因素的水平只在候选测试用例中出现一次, 选择 m_{j+1} 中的水平, 只考虑已选取因素 m_1, m_2, \dots, m_j 的 j 个水平进行比较。从 π 中删除找出的已覆盖的成对组合。

(3) 重复步骤(2), 直到 π 为空, 结束。

1.3 组合算法设计测试用例

例2, 设某个被测系统有3个因素(A,B,C), 因素A有水平 x_1 和 x_2 , 因素B有水平 y_1 和 y_2 , 因素C有水平 z_1, z_2 和 z_3 。应用 AETG 算法计算满足成对组合覆盖的最小测试用例集。计算测试用例集步骤如下:

(1) 求出未被覆盖的成对组合集合。

$\pi = \{(x_1, y_1), (x_1, y_2), (x_1, z_1), (x_1, z_2), (x_1, z_3), (x_2, y_1), (x_2, y_2), (x_2, z_1), (x_2, z_2), (x_2, z_3), (y_1, z_1), (y_1, z_2), (y_1, z_3), (y_2, z_1), (y_2, z_2), (y_2, z_3)\}$;

(2) 选择测试用例 $T_1 = (x_1, y_1, z_1)$, T_1 能够覆盖的成对组合有 $\{(x_1, y_1), (x_1, z_1), (y_1, z_1)\}$, 并将该组合从集合 π 中删除;

(3) 此时, $\pi = \{(x_1, y_2), (x_1, z_2), (x_1, z_3), (x_2, y_1), (x_2, y_2), (x_2, z_1), (x_2, z_2), (x_2, z_3), (y_1, z_2), (y_1, z_3), (y_2, z_1), (y_2, z_2), (y_2, z_3)\}$, 其中 x_1 出现3次, x_2 出现5次, y_1 出现3次, y_2 出现5次, z_1 出现2次, z_2 出现4次, z_3 出现4次, 因此, 选择出现次数最多的 y_2 , 测试用例 $T_2 = (-, y_2, -)$; 下面要选择 x_1 或 x_2 之一作为因素A的水平值, (x_1, y_2) 或 (x_2, y_2) , 在集合 π 中的覆盖次数相同, 任选 x_1 , 则测试用例 $T_2 = (x_1, y_2, -)$; 然后选择C因素的水平, 若选择 z_1 , $T_2 = (x_1, y_2, z_1)$, 覆盖 π 中的 (x_1, y_2) 和 (y_2, z_1) 共2个组合; 若选择 z_2 , $T_2 = (x_1, y_2, z_2)$, 覆盖 π 中的 (x_1, y_2) 、 (x_1, z_2) 和

(y_2, z_2) 共3个组合; 若选择 z_3 , $T_2 = (x_1, y_2, z_3)$, 覆盖 π 中的 (x_1, y_2) 、 (x_1, z_3) 和 (y_2, z_3) 共3个组合。由于选择 z_2 和 z_3 覆盖 π 中组合个数相同, 任选 z_2 , 因此, 测试用例 $T_2 = (x_1, y_2, z_2)$, 将 T_2 的3个覆盖组合从 π 中删除;

(4) $\pi = \{(x_1, z_3), (x_2, y_1), (x_2, y_2), (x_2, z_1), (x_2, z_2), (x_2, z_3), (y_1, z_2), (y_1, z_3), (y_2, z_1), (y_2, z_3)\}$, 按以上方法选择, 测试用例 $T_3 = (x_2, y_1, z_2)$, 将 T_3 的3个组合从 π 中删除;

(5) $\pi = \{(x_1, z_3), (x_2, y_2), (x_2, z_1), (x_2, z_3), (y_1, z_3), (y_2, z_1), (y_2, z_3)\}$; x_2 出现次数最多, y_1 已经覆盖, 选择 y_2 , $T_4 = (x_2, y_2, -)$, 按以上方法选择, 测试用例 $T_4 = (x_2, y_2, z_1)$, 将 T_4 的3个组合从 π 中删除;

(6) $\pi = \{(x_1, z_3), (x_2, z_3), (y_1, z_3), (y_2, z_3)\}$, $T_5 = (x_1, y_1, z_3)$; 将 T_5 的2个组合从 π 中删除;

(7) $\pi = \{(x_2, z_3), (y_2, z_3)\}$, $T_6 = (x_2, y_2, z_3)$; π 中删除2个组合覆盖, π 为空, 结束。

表1 为 AETG 测试用例集合。

表1 AETG 测试用例集合

测试用例	A	B	C
T_1	x_1	y_1	z_1
T_2	x_1	y_2	z_2
T_3	x_2	y_1	z_2
T_4	x_2	y_2	z_1
T_5	x_1	y_1	z_3
T_6	x_2	y_2	z_3

2 正交实验法

正交实验设计是利用正交表来安排与分析多因素实验的一种设计方法。在实验因素的全部水平组合中, 挑选部分有代表性的水平组合进行实验, 通过对这部分实验结果的分析了解全面实验的情况, 找出最优的水平组合。全面实验可以分析各因素的效应和交互作用, 也可选出最优水平组合。但全面实验包含的水平组合数较多, 在有些情况下无法完成, 需要在设计上应用正交等相关方法。

2.1 正交设计原理

在实验安排中, 每个因素在研究的范围内选几个水平, 如3因素3水平的实验, 可以用一个立方体表示, 立方体划分成27个格点, 若27个格点都实验, 就是全面实验。其全面实验水平组合数为 $3^3 = 27$, 4因素3水平的全面实验水平组合数为 $3^4 = 81$, 随着数据的增大, 科学实验有可能做不到。正交设计就是从全面实验点(水平组合)中挑选出有代表性的部分实验点(水平组合)来进行实验。3因素3水平的实验可利

用正交表 $L_9(3^4)$ 从 27 个实验点中挑选出来的 9 个实验点。

表 2 是一张 4 因素 3 水平的正交表,记号为 $L_9(3^4)$,其中 L 代表正交表;L 右下角的数字 9 表示有 9 行,用这张正交表安排 9 个实验,括号内的底数 3 表示因素的水平数,括号内 3 的指数 4 表示有 4 列。

表 2 正交表 $L_9(3^4)$

实验	因素			
	A	B	C	D
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

(1)任一列中,各个水平都出现,且出现的次数相等。例如 $L_8(2^7)$ 中不同数字只有 1 和 2,它们各出现 4 次; $L_9(3^4)$ 中不同数字有 1、2 和 3,它们各出现 3 次。

(2)任两列之间各种不同水平的所有可能组合都出现,且出现的次数相等。例如 $L_8(2^7)$ 中(1,1), (1,2), (2,1), (2,2)各出现两次; $L_9(3^4)$ 中(1,1), (1,2), (1,3), (2,1), (2,2), (2,3), (3,1), (3,2), (3,3)各出现 1 次。任意两列各个数字之间的搭配是均匀的。

(3)正交表可分为水平正交表和混合水平正交表。各列水平数相同的正交表称为等水平正交表。如 $L_4(2^3)$ 、 $L_8(2^7)$ 、 $L_{12}(2^{11})$ 等各列中的水平为 2,称为 2 水平正交表; $L_9(3^4)$ 、 $L_{27}(3^{13})$ 等各列水平为 3,称为 3 水平正交表。各列水平数不完全相同的正交表称为混合水平正交表。如 $L_8(4 \times 2^4)$ 表中有一列的水平数为 4,有 4 列水平数为 2。可以安排一个 4 水平因素和 4 个 2 水平因素。

2.2 正交法测试用例设计

对于多因素实验,正交实验设计是简单常用的一种实验设计方法。正交实验设计的基本程序包括实验方案设计及实验结果分析两部分。实验方案设计步骤:

- (1)明确实验目的,确定实验指标。
- (2)选因素、定水平,列出因素水平表。
- (3)选择合适的正交表,并设计表头。
- (4)编制实验方案,按方案进行实验,记录实验结果。
- (5)分析实验结果,可以采用极差分析法-R 法或

方差分析。

例 3,在某应用系统响应时间的测试研究中,采用正交实验方法设计测试用例,考虑 3 因素 3 水平情况,采用正交表 $L_9(3^4)$,因素 A 为网络带宽,因素 B 为处理器速度,因素 C 为并发用户数,为满足正交表格式需要增加一空列,所要检测的结果是系统响应时间 y 。

首先,分析 A 因素各水平对实验指标的影响。由表 3 可以看出,A₁ 的影响反映在第 1、2、3 号实验中,A₂ 的影响反映在第 4、5、6 号实验中,A₃ 的影响反映在第 7、8、9 号实验中。

- (1)A 因素的 1 水平所对应的实验指标之和为
 $K_1 = y_1 + y_2 + y_3 = 15.67$;
- (2)A 因素的 2 水平所对应的实验指标之和为
 $K_2 = y_4 + y_5 + y_6 = 18.57$;
- (3)A 因素的 3 水平所对应的实验指标之和为
 $K_3 = y_7 + y_8 + y_9 = 31.25$ 。

表 3 某系统研究指标与响应时间分析

序号	因素 A	因素 B	因素 C	空列	响应时间 y/ms
1	1	1	1	1	6.25
2	1	2	2	2	4.97
3	1	3	3	3	4.54
4	2	1	2	3	7.53
5	2	2	3	1	5.54
6	2	3	1	2	5.5
7	3	1	3	2	11.4
8	3	2	1	3	10.9
9	3	3	2	1	8.95
K_{1j}	15.76	25.18	22.65	20.74	$T = 65.58$
K_{2j}	18.57	21.41	21.45	21.87	
K_{3j}	31.25	18.99	21.48	22.97	
K_{1j}^2	248.38	634.03	513.02	430.15	
K_{2j}^2	344.84	458.39	460.10	478.30	
K_{3j}^2	976.56	360.62	461.39	527.62	

在因素 A 均值计算的基础上,计算极差 R,确定优水平。同理分析 B、C 和 D 因素,最终可以确定系统影响因素的优组合。

3 测试用例评价方法

3.1 R 评价法

- 依据表 3,计算各影响因素的极差如下:
- A 因素极差 $R = 31.25 - 15.76 = 15.49$;
- B 因素极差 $R = 25.18 - 18.99 = 6.19$;
- C 因素极差 $R = 22.65 - 21.45 = 1.20$;
- D 因素极差 $R = 22.97 - 20.74 = 2.23$ 。

确定各因素的主次顺序为 $A > B > D > C$,响应时间小的为优,4 个因素的优水平分别为 A₁、B₃、C₂ 和 D₁,

优组合为 $A_1B_3C_2D_1$ 。

由于优组合 $A_1B_3C_2D_1$ 并没有出现在正交实验的 9 个方案中,其中方案 3 ($A_1B_3C_3D_3$) 的响应时间最短为 4.54 ms,因此需要对优组合进行进一步的分析,并进行实验验证。当实验条件允许的情况下,需要对优组合进行反复实验和数据分析,以保证数据的可信性。因此,安排了进一步实验,按照优组合 $A_1B_3C_2D_1$ 测试,得到系统的响应时间为 4.55 ms,该指标与方案 3 的响应时间 4.54 ms 相当,这从另一侧面证明了影响因素主要是因素 A 和 B,因素 C 和 D 影响不大。

3.2 方差评价法

R 法简单,但不能区分因素各水平间对应的实验结果的差异究竟是由于因素水平不同引起的,还是由于实验误差引起的,无法估计实验误差的大小。此外,各因素对实验结果的影响大小无法以精确的数量估计,不能提出一个标准来判断所考察因素作用是否显著。为了弥补极差分析的缺陷,可采用方差分析。方差分析考察因素对指标的影响是否显著。

方差分析基本思想是将数据的总变异分解成因素引起的变异和误差引起的变异两部分,构造 F 统计量,作 F 检验,即可判断因素作用是否显著。

(1) 计算 K 值、各列离差平方和及自由度。

计算各列及各水平对应数据之和 K_{1j} 、 K_{2j} 、 K_{3j} 及其平方 K_{1j}^2 、 K_{2j}^2 、 K_{3j}^2 。

$$SS_j = \frac{1}{r} \sum_{i=1}^m K_{ij}^2 - CT$$

$$CT = \frac{T^2}{n} = \frac{65.58^2}{9} = 477.86$$

所以,因素 A 的离差平方和 SS_A 为

$$SS_A = (K_{11}^2 + K_{21}^2 + K_{31}^2)/3 - CT = (248.38 + 344.84 + 976.56)/3 - 477.86 = 45.4$$

同理, $SS_B=6.49$, $SS_C=0.31$, $SS_e=0.83$ (空列)。

自由度计算: $df_A=df_B=df_C=df_e=3-1=2$ 。

(2) 计算方差。

$$V_A = \frac{SS_A}{df_A} = \frac{45.4}{2} = 22.7$$

$$V_B = \frac{SS_B}{df_B} = \frac{6.49}{2} = 3.23$$

$$V_C = \frac{SS_C}{df_C} = \frac{0.31}{2} = 0.155$$

$$V_e = \frac{SS_e}{df_e} = \frac{0.83}{2} = 0.415$$

(3) 显著性检验。

列出方差分析表,采用 $F_{0.05}(2,4) = 6.94$,构造 F 统计量, F 因素 = M 因素 / M 误差,进行检验。为了提

高 F 检验的灵敏度,在进行显著性检验之前,先将各因素方差与误差方差比较,若因素方差小于 2 倍误差方差,应将这些因素方差、自由度并入误差列,这样使误差的方差和自由度增大,提高了 F 检验的灵敏度,参见表 4。

表 4 方差分析评价

变异来源	方差	自由度	均方	F 值	显著性
A	45.40	2	22.70	79.6	* *
B	6.49	2	3.23	11.3	*
C△	0.31	2	0.155		
误差 e	0.83	2	0.415		
误差 e△	1.14	4	0.285		

经计算,A 因素 F 值 79.6 远大于 6.94,因素 A 高度显著,因素 B 显著,因素 C 不显著。因素主次顺序 A-B-C。

4 结束语

在软件测试用例的设计中,面临用例个数庞大甚至爆炸的可能,除了等价类测试和边界值测试技术外,设计人员一直在探寻优化方法,减少测试用例数量,以合理的方法对大规模软件进行充分的测试^[13]。研究表明,正交实验方法和组合测试技术可以用来减轻工作量,尽可能少地测试用例,达到测试需求及相应的覆盖指标。组合测试的测试用例集合构造是一个 NP-C 问题,能够在不降低其检错能力前提下,降低测试用例数量,优化测试用例的设计。正交实验是用部分实验来代替全面实验。今后的工作是进一步研究高性能组合测试用例的生成方法,研究组合测试的故障诊断技术和错误定位技术;进一步研究因素之间的交互作用,分析可能出现的交互作用及各因素间的相互制约。

参考文献:

[1] 浦云明. 一种平衡结构性测试和功能性测试的方法[J]. 计算机应用,2008,28(4):1023-1025.
[2] 单锦辉,姜 瑛,孙 萍. 软件测试研究进展[J]. 北京大学学报(自然科学版),2005,41(1):134-145.
[3] Tai K C,Lei Y. A test generation strategy for pairwise testing [J]. IEEE transactions on software engineering,2002,28(1):109-111.
[4] 陈 皓,王曙燕,潘晓英. 成对组合测试数据的整体优化和生成方法[J]. 计算机工程与应用,2012,48(11):32-36.
[5] Kuhn D R,Reilly M J. An investigation of the applicability of design of experiments to software testing[C]//Proceedings of the 27th NASA/IEEE software engineering workshop. [s. l.]: NASA Goddard Space Flight Center,2002:91-95.
[6] Cohen M B,Colbouns C J,Gibbons P B,et al. Constructing

的次数再乘以节点的度数。因此,从图3中也可以看出,度数增加越多的算法其信令开销就越大,意味着该算法的性能就越低,与前面的理论结果一致。

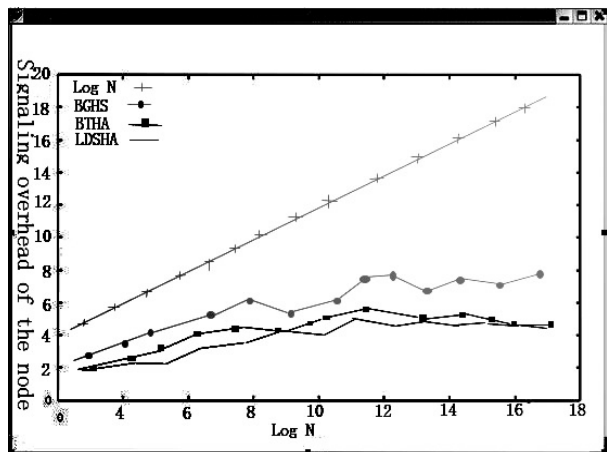


图3 不同修复策略的信令开销

4 结束语

文中提出了一种在可重构网络中的自适应入侵检测修复算法,与其他算法相比,该算法不仅能够主动检测网络中的路径中断,还能够修复网络以保持连通性。经过实验,表明该算法不仅在检测延迟方面比传统的被动感应式方法少,而且在修复过程中节点度数增加的较少,节点之间的信令开销较小。

但是,该算法适用于某一特定时刻只有一个节点被删除,入侵者在删除下一个节点之前该算法已经将网络修复,如果入侵者同时删除多个节点,使多条路径中断,这种情况下的检测以及修复算法还需要进一步研究。

参考文献:

- [1] 齐宁,汪斌强,王志明.可重构服务承载网容错构建算法研究[J].电子与信息学报,2012,34(2):468-473.
- [2] 李黎,管晓宏,蔡忠闽,等.可重构网络系统的模型及体系结构[J].小型微型计算机系统,2009,30(4):637-641.
- [3] 赵靓,汪斌强,张鹏.可重构柔性网络体系研究[J].电信科学,2012,28(2):133-137.
- [4] 刘衍珩,田大新,余雪岗,等.基于分布式学习的大规模网络入侵检测算法[J].软件学报,2008,19(4):993-1003.
- [5] Medard M, Finn S G, Barry R A. Redundant trees for pre-planned recovery in arbitrary vertex-redundant or edge-redundant graphs[J]. IEEE/ACM transactions on networking, 1999,7(5):641-652.
- [6] Andersen D, Balakrishnan H, Kaashoek F, et al. Resilient overlay networks[J]. SIGOPS oper syst rev, 2001,35(5):131-145.
- [7] 袁博,汪斌强,孔维功,等.可重构柔性网络中网络资源控制的冲突检测机制[EB/OL]. 2012. <http://www.paper.edu.cn>.
- [8] Boman I, Saia J, Abdallah C T, et al. Self-healing algorithms for reconfigurable networks[J]. Lecture notes in computer science, 2006,4280:563-565.
- [9] Trehan A. Algorithms for self-healing networks[D]. USA: The University of New Mexico, 2010.
- [10] The Network Simulator-ns2[EB/OL]. 2011. <http://www.isi.edu/nsnam/ns/>.
- [11] Yuan B, Wang B Q, Zhang B. A case study of green network-reconfigurable flexible network[J]. Telecommunication science, 2011,27(10):200-208.
- [12] Hitesh B, Paul F. CONMan: A step towards network manageability[C]//Proceedings of ACM SIGCOMM. Kyoto, Japan: [s. n.], 2007:154-162.

(上接第121页)

test suites for interaction testing[C]//Proceedings of the international conference on software engineering. Portland, OR: [s. n.], 2003:38-48.

- [7] Kobayashi N, Tsuchiya T, Kikuno T. A new method for constructing pairwise covering designs for software testing[J]. Information processing letters, 2002,81(2):85-91.
- [8] Colbourn C J, Martirosyan S S, Mullen G L, et al. Products of mixed covering arrays of strength two[J]. Journal of combinatorial designs, 2005,14(2):124-138.
- [9] 于秀山,于洪敏.软件测试新技术与实践[M].北京:电子

工业出版社,2006.

- [10] 严俊,张健.组合测试:原理与方法[J].软件学报, 2009,20(6):1393-1405.
- [11] 王子元,聂长海,徐宝文,等.相邻因素组合测试用例集的最优生成方法[J].计算机学报,2007,30(2):200-211.
- [12] Kuhn D R, Wallace D R. Software fault interaction and implication for software testing[J]. IEEE transactions on software engineering, 2004,30(6):1-4.
- [13] 范明红,浦云明,汪志华.等价类测试与划分研究[J].计算机技术与发展,2009,19(7):62-65.

组合的测试用例设计与评价方法

作者：

浦云明，[PU Yun-ming](#)

作者单位：

[集美大学 计算机工程学院, 福建 厦门, 361021](#)

刊名：

[计算机技术与发展](#)

英文刊名：

Computer Technology and Development

年, 卷(期):

2014(3)

本文链接：http://d.g.wanfangdata.com.cn/Periodical_wjfz201403030.aspx