

# 正交表测试策略（OATS）技术

Jeremy M. Harrell  
Quality Assurance Manager  
Seilevel, Inc.  
本文由李正新翻译

## 1. 介绍

正交表测试策略是一种成对测试交互的系统的统计的方法。它提供了一种能对所有变量对的组合进行典型覆盖（均匀分布）的方法。这种技术对软件组件的集成测试尤其有用（尤其对面向对象的系统，因为对客户端来说很多子类能够被代替做为服务器）。它在配置选项组合的测试中也很有用（比如一个让用户选择字体、背景颜色及页面布局的网页）。

## 2. 背景

最初发现正交表方面知识的是一些对数字有强烈好奇心的一群修道士[Copeland2001]。这项技术一直静静在躺在这些修道士的笔记里，直到十九世纪五十年代才被广泛的重视。

就是在那时，统计学协会才对这方面的数字感兴趣，并把它用到统计测试设计。Genichi Taguchi 是第一个支持把正交表运用到测试设计方面的人。几十年来，他的技术被称为 Taguchi 方法，已经成为制造业中实验设计的中流砥柱。

正交表是一个二维数字表格，它有一个有趣的特性：即选择表中任何两列时成对组合的值的分布是平均分布的。在下面的图 1[Sloane2001]的例子中将介绍正交表的几个术语：

次数（Runs）：正交表中的行的个数。它直接对应到用正交表测试策略设计成的测试的个数。

因素数（Factorss）：正交表中列的个数。它直接对应到用这种技术设计测试案例时的变量的最大个数。

水平数（Levels）：任何单个因素能够取得的值的最大个数。正交表中的包含的值为从 0 到数“水平数-1”。

强度（Strength）：矩阵的列的个数，对任何以此个数的为列的子矩阵，所有可能的

组合出现的次数相同（ $L_{\text{Levels}^{\text{Strength}}}$  个可能的组合）。

正交表通常用下面的形式表示： $L_{\text{次数}}(\text{水平数}^{\text{因素数}}) . L_{\text{Runs}}(\text{Levels}^{\text{Factors}})$ 。

**Figure 1: An  $L_9(3^4)$  orthogonal array with 9 runs, 4 factors, 3 levels, and strength of 2.**

	Factors			
	0	1	2	3
R u n s	0	0	0	0
	0	1	1	2
	0	2	2	1
	1	0	1	1
	1	1	2	0
	1	2	0	2
	2	0	2	2
	2	1	0	1
	2	2	1	0

### 3. 为什么使用这个技术

对软件专业人员来说，测试案例的选择是即有趣又困难的选择。几乎人人都知道：产品的质量不是测试出来的；测试仅仅证明软件存在缺陷，但从来不能证明不存在缺陷；即使在一个小的系统中全面快速的测试也是不可能的。然而，测试是必须的。在选择测试案例的时候用一些智慧就能够从无休止的执行那些不可能发现缺陷的，并且不能增加你对系统的自信的测试状态转变到一种执行简单、能够发现大部分（不是全部）的缺陷并且对软件的质量更有舒心的测试状态。

下面是最基本的缺陷模型：

- 1 相互和集成是最主要的缺陷来源。
- 2 很多的缺陷并非出现在复杂的交互情况下，比如，“当背景是蓝色的，字体是 Arial，页面右侧有菜单，图像是一个大的图像和时间是星期四时，表格没有排列好。”而是，很多的缺陷都是发生在成对的交互中，比如，“当字体是 Arial 和页面右侧有菜单时表格没有排列好。”
- 3 考虑到组件或背景的很多可能组合，很容易漏掉一个
- 4 随机的选取值而生成的组合一定是生成一个没有什么效率的测试集，而这些值的分布是随机的并且是无意义的。

OATS 提供了一个选择测试集的方法：

- 1 保证了对所有被选变量的成对组合
- 2 生成一个有效且精简的测试集，这个测试集包括最少的测试用例但是测试所有变量的所有组合
- 3 生成所有的成对组合的均匀分布的测试集
- 4 训练一些所有变量的复杂组合
- 5 与用手工生成的测试集相比，它的生成更简单且更不易出错

为了说明使用 OATS 技术的好处，把用 OATS 生成的测试用例与测试变量的每一种组合的测试集相比较，假设系统有 4 个变量选择，每一个都有三种取值。全部的测试集有 81 个测试用例（ $3 \times 3 \times 3 \times 3$  或者是所有选项的迪卡尔积）。用 OATS 技术生成（使用图 1 的正交表）

的测试集只有 9 个测试用例，但它能够测试所有的成对组合。用 OATS 生成的测试集只有测试全集的 11% 那么大，但它能够覆盖大部分的相互方面的缺陷。

## 4. 怎样使用这个技术

OATS 技术是非常简单的，步骤大致描述如下：

- 1 确定交互测试中有多少个相互独立的变量，这映射到表中的因素数 (Factors)
- 2 确定每个变量可以取值的个数的最大数，这映射到表中的水平数 (Levels)
- 3 选择一个次数 (Run) 数最少的最适合的正交表。一个最合适的正交表是至少满足第一步说明的因素数且至少满足第二步说明的水平数
- 4 把因素和值映射到表中
- 5 为剩下的水平数选取值
- 6 把次数中所描述的组合转化成测试用例，再增加一些没有生成的但可疑的测试用例

## 5. 实际例子

### 5.1 一个简单的例子

考虑一个网页有三个不同的部分 (Top, Middle, Bottom), 并且我们可以把其中的一个单独的显示及隐藏。而你要测试这三个不同部分的相互。按照前面给出的步骤，让我们给系统生成一个测试集

1. 有三个独立的变量 (网面的三个部分)
2. 每个变量能够取两个值 (hidden 或 visible)
3. 选择  $L_4(2^3)$  正交表——变量为三因素，值为二水平。注意，次数 (run) 的多少是选择合适正交表的必要条件
4. 把变量的值映射到表中，将如下图 2 所示，其中 Hidden=0, Visible=1

Figure 2

OA before mapping factors			
	Factor 1	Factor 2	Factor 3
Run 1	0	0	0
Run 2	0	1	1
Run 3	1	0	1
Run 4	1	1	0

OA after mapping factors			
	Top	Middle	Bottom
Test 1	Hidden	Hidden	Hidden
Test 2	Hidden	Visible	Visible
Test 3	Visible	Hidden	Visible
Test 4	Visible	Visible	Hidden

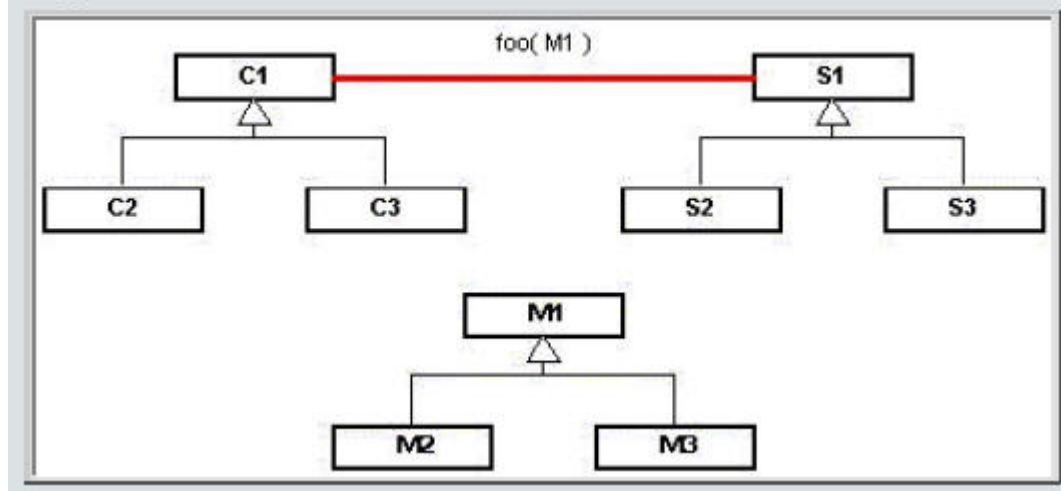
5. 此种情况没有剩下的水平数。换句话说，表中的每个水平都有一个值被映射过来
6. 把表中每一行（次数，run）转换成测试用例，可以得到四个测试用例。这是测试三个变量成对交互时需要测试的。测试用例可以描述成如下：
- A. 显示首页，隐藏这三个部分
  - B. 显示首页，显示除 Top 部分外的其它部分
  - C. 显示首页，显示除 Middle 部分外的其它部分
  - D. 显示首页，显示除 Bottom 部分外的其它部分

注意，并不是所有的可能组合都被测试。如果测试所有可能的组合将得到八个测试用例。你可以增加一些很可疑的，但没有有表中出现的测试用例。例如，如果你觉得显示所有部分或仅仅显示 middle 部分等这些组合很有可能出错，你就可以增加测试用例来测试他们。

## 5.2 一个与正交表不匹配的例子

下面看一个与可用的正交表不完全相配的一个例子。它描述了一个面向对象系统包含一个 client 类 (C1) 和它的两个子类 (C2 和 C3)。这些 client 类又与一个 server 类 S1 相互，这个 server 类有两个子类 S2 和 S3。这个 server 类包括一个方法 foo()，它把 M1 的一个实例做为参数。M1 又有两个子类，M2 和 M3。下图 3 描述了这些类的关系。

Figure 3



为了测试这些类的所有组合，我们必须设计 27 个测试用例（三个 client 类可以把送三个消息到三个 server 类—— $3 \times 3 \times 3 = 27$ ）。令人更难容忍的是，这仅仅是用单一的测试用例来测试方法 `foo()`。在很多情况下，我们必须设计很多测试用例来测试某一个特殊的方法。当然，这个相互也才仅仅是整个被测系统的一个很小的部分。使用 OATS 技术可以大幅度的减少测试用例的数量。

- 1、有三个独立的变量（client，server，和消息类）
- 2、每个变量可以到三个值

3、理想情况下，我们将使用三水平三因素的正交表  $L_9(3^3)$ 。但是，没有这种类型的正交表被设计并公布出来。因此，我们需要的找一个最小的能够处理问题的正交表。

$L_9(3^4)$  正好满足这个要求。它有三水平的值，四因素，用它来处理我们的三个变量已经够了。

- 4、把值映射到正交表中，我们将得到如下图四：
  - A. 对 Client, C1=0; C2=1; C3=2。
  - B. 对 Server, S1=0; S2=1; S3=2。
  - C. 对 Message, M1=0; M2=1; M3=2。

Figure 4

OA before mapping factors				
	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	0	0	0	0
Run 2	0	1	1	2
Run 3	0	2	2	1
Run 4	1	0	1	1
Run 5	1	1	2	0
Run 6	1	2	0	2
Run 7	2	0	2	2
Run 8	2	1	0	1
Run 9	2	2	1	0

OA after mapping factors				
	Client	Server	Message	
Test 1	C1	S1	M1	
Test 2	C1	S2	M2	
Test 3	C1	S3	M3	
Test 4	C2	S1	M2	
Test 5	C2	S2	M3	
Test 6	C2	S3	M1	
Test 7	C3	S1	M3	
Test 8	C3	S2	M1	
Test 9	C3	S3	M2	

5. 此种情况没有剩下的水平数。然而，你会注意到在原始的表中有一个额外的因素。可以简单的忽略这个因素；而不影响由此表生成的测试集。你将仍旧得到一个成对的均匀分布组合。
6. 把表中每一行的值提取出来就可以得到 9 个测试用例。像前面提到的一样，这 9 个组合映射到一个更大的测试用例集，而这个测试集必须执行这 9 个组合中的每一个。

### 5.3 一个复杂的多水平的例子

下面是一个运用到混合多水平的复杂的例子。假设有一个系统有 5 个独立的变量 (A, B, C, D, E)。变量 A 和 B 都有两个取值 ( $A_{1-2}$  和  $B_{1-2}$ )。变量 C 和 D 都有三个可能的取值 ( $C_{1-3}$  和  $D_{1-3}$ )。变量 E 有六个可能的取值 ( $E_{1-6}$ )。为了测试所有的组合，我们必须设计 216 个测试用例 ( $2 \times 2 \times 3 \times 3 \times 6 = 216$ )。这个例子说明为了测试所有成对组合用 OATS 技术怎么样把测试用例的数量减少到 18 个。

1. 有五个独立的变量
2. 两个变量取二个值。二个变量取三个值。一个变量取六个值
3. 获得正交表的最简单方法是在正交表的列表中找到一个合适的满足至少六水平（任何一个变量中的最大水平数）和至少五个因素数的正交表。最小的并且水平数相容的

正交表就可能是  $L_{49}(7^8)$ 。这个正交表生成 49 个测试用例。这远比 216 个测试用例好，但仍旧是有很多测试用例。

你应该注意到前面提到的相容的水平数。这点很重要，因为实际上恰巧有很少

一部分的正交表是有混合的水平数的。 $L_{18}(3^6 6^1)$  就是一个这样的正交表。这个正交表的命名意味着运行次数 18 次，有 7 个因素 6 个 3 水平的 1 个 6 水平的水平数。我们的问题恰巧很适合这个正交表，测试集中的测试用例数从 49 个减少到 18 个。现在测试用例的数目比 216 更少了。

4. 在把值映射到正交表，如下图图 5 所示

A. 对 A,  $A_1=0; A_2=1;$

B. 对 B,  $B_1=0; B_2=1;$

C. 对 C,  $C_1=0; C_2=1; C_3=2;$

D. 对 D,  $D_1=0; D_2=1; D_3=2;$

E. 对 E,  $E_1=0; E_2=1; E_3=2; E_4=3; E_5=4; E_6=5;$

**Figure 5**

OA before mapping factors							
	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6	Factor 7
Run 1	0	0	0	0	0	0	0
Run 2	0	1	2	2	0	1	1
Run 3	0	2	1	2	1	0	2
Run 4	0	1	1	0	2	2	3
Run 5	0	2	0	1	2	1	4
Run 6	0	0	2	1	1	2	5
Run 7	1	1	1	1	1	1	0
Run 8	1	2	0	0	1	2	1
Run 9	1	0	2	0	2	1	2
Run 10	1	2	2	1	0	0	3



<b>Run 11</b>	1	0	1	2	0	2	4
<b>Run 12</b>	1	1	0	2	2	0	5
<b>Run 13</b>	2	2	2	2	2	2	0
<b>Run 14</b>	2	0	1	1	2	0	1
<b>Run 15</b>	2	1	0	1	0	2	2
<b>Run 16</b>	2	0	0	2	1	1	3
<b>Run 17</b>	2	1	2	0	1	0	4
<b>Run 18</b>	2	2	1	0	0	1	5

OA after mapping factors

	A	B	C	D		E
Test 1	A <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>	D <sub>1</sub>		E <sub>1</sub>
Test 2	A <sub>1</sub>	B <sub>2</sub>	C <sub>3</sub>	D <sub>3</sub>		E <sub>2</sub>
Test 3	A <sub>1</sub>	2	C <sub>2</sub>	D <sub>3</sub>		E <sub>3</sub>
Test 4	A <sub>1</sub>	B <sub>2</sub>	C <sub>2</sub>	D <sub>1</sub>		E <sub>4</sub>
Test 5	A <sub>1</sub>	2	C <sub>1</sub>	D <sub>2</sub>		E <sub>5</sub>
Test 6	A <sub>1</sub>	B <sub>1</sub>	C <sub>3</sub>	D <sub>2</sub>		E <sub>6</sub>
Test 7	A <sub>2</sub>	B <sub>2</sub>	C <sub>2</sub>	D <sub>2</sub>		E <sub>1</sub>
Test 8	A <sub>2</sub>	2	C <sub>1</sub>	D <sub>1</sub>		E <sub>2</sub>
Test 9	A <sub>2</sub>	B <sub>1</sub>	C <sub>3</sub>	D <sub>1</sub>		E <sub>3</sub>
Test 10	A <sub>2</sub>	2	C <sub>3</sub>	D <sub>2</sub>		E <sub>4</sub>
Test 11	A <sub>2</sub>	B <sub>1</sub>	C <sub>2</sub>	D <sub>3</sub>		E <sub>5</sub>
Test 12	A <sub>2</sub>	B <sub>2</sub>	C <sub>1</sub>	D <sub>3</sub>		E <sub>6</sub>
Test 13	2	2	C <sub>3</sub>	D <sub>3</sub>		E <sub>1</sub>
Test 14	2	B <sub>1</sub>	C <sub>2</sub>	D <sub>2</sub>		E <sub>2</sub>
Test 15	2	B <sub>2</sub>	C <sub>1</sub>	D <sub>2</sub>		E <sub>3</sub>
Test 16	2	B <sub>1</sub>	C <sub>1</sub>	D <sub>3</sub>		E <sub>4</sub>
Test 17	2	B <sub>2</sub>	C <sub>3</sub>	D <sub>1</sub>		E <sub>5</sub>
Test 18	2	2	C <sub>2</sub>	D <sub>1</sub>		E <sub>6</sub>

5. 像先前的例子一样，这个正交表有额外的不需要的因素。它们可以被安全的忽略，如图 5 所示的变灰的部分。这个正交表有剩下的水平数。变量 A 和 B 在原始表中有三个水平数，但是它们实际只有两个可能的取值。这将导致在把因素映射过之后变量 A 和 B 有多余的水平。这些多余的值在图 5 中用黄色标注出来了。

为了明确有多余水平的测试用例，你必须在表中相应的位置填写一个值。值的选择是任意的，但是为了帮助我们增加发现缺陷的机会，我们增加尽可能多样化的值会更有意义的 [Marick1995]。一个很好的方法是在填写剩下的水平时从列的顶部开始循环第一个可能的值。图 6 显示了使用循环方法填写剩下的水平的表格。

**Figure 6: OA after mapping left over levels**

	A	B	C	D		E
Test 1	A <sub>1</sub>	B <sub>1</sub>	C <sub>1</sub>	D <sub>1</sub>		E <sub>1</sub>
Test 2	A <sub>1</sub>	B <sub>2</sub>	C <sub>3</sub>	D <sub>3</sub>		E <sub>2</sub>
Test 3	A <sub>1</sub>	B <sub>1</sub>	C <sub>2</sub>	D <sub>3</sub>		E <sub>3</sub>
Test 4	A <sub>1</sub>	B <sub>2</sub>	C <sub>2</sub>	D <sub>1</sub>		E <sub>4</sub>
Test 5	A <sub>1</sub>	B <sub>2</sub>	C <sub>1</sub>	D <sub>2</sub>		E <sub>5</sub>
Test 6	A <sub>1</sub>	B <sub>1</sub>	C <sub>3</sub>	D <sub>2</sub>		E <sub>6</sub>
Test 7	A <sub>2</sub>	B <sub>2</sub>	C <sub>2</sub>	D <sub>2</sub>		E <sub>1</sub>
Test 8	A <sub>2</sub>	B <sub>1</sub>	C <sub>1</sub>	D <sub>1</sub>		E <sub>2</sub>
Test 9	A <sub>2</sub>	B <sub>1</sub>	C <sub>3</sub>	D <sub>1</sub>		E <sub>3</sub>
Test 10	A <sub>2</sub>	B <sub>2</sub>	C <sub>3</sub>	D <sub>2</sub>		E <sub>4</sub>
Test 11	A <sub>2</sub>	B <sub>1</sub>	C <sub>2</sub>	D <sub>3</sub>		E <sub>5</sub>
Test 12	A <sub>2</sub>	B <sub>2</sub>	C <sub>1</sub>	D <sub>3</sub>		E <sub>6</sub>
Test 13	A <sub>1</sub>	B <sub>1</sub>	C <sub>3</sub>	D <sub>3</sub>		E <sub>1</sub>
Test 14	A <sub>2</sub>	B <sub>1</sub>	C <sub>2</sub>	D <sub>2</sub>		E <sub>2</sub>
Test 15	A <sub>1</sub>	B <sub>2</sub>	C <sub>1</sub>	D <sub>2</sub>		E <sub>3</sub>
Test 16	A <sub>2</sub>	B <sub>1</sub>	C <sub>1</sub>	D <sub>3</sub>		E <sub>4</sub>
Test 17	A <sub>1</sub>	B <sub>2</sub>	C <sub>3</sub>	D <sub>1</sub>		E <sub>5</sub>
Test 18	A <sub>2</sub>	B <sub>2</sub>	C <sub>2</sub>	D <sub>1</sub>		E <sub>6</sub>

6. 如前所述，你从 216 个可能的情况中得到 18 个测试用例。这 18 个测试用例测试了这此独立变量的所有的成对组合。这证明了在测试时它比测试全部组合的方法省去了不少的工作，而我们的缺陷模型显示它能够发现交互中的大部分缺陷。

## 6. 行业结果

作者不知道任何使用该技术对软件测试的费用及效果的影响的任何成文的工业或研究结果。这里主要强调的是这只是在把 OATS 技术及与其它测试技术的比较中得出的经验结论。

## 7. 尾注

1. 这篇文章仅仅关注强度为 2 时的正交表的情况。不同强度的正交表将需要不同数目的列来选择测试，也不一定要关心成对组合。

2. 这个技术没有生成包括所有变量组合的测试集。它甚至还没有覆盖所有可能情况的四分之一。所有可能的组合数是每个因素的取值个数的迪卡尔积。这么多的数目对实际测试工作来说是很难处理的。然而，成对组合的数目是非常非常的小的。

3. 正交表的列表可以在附录中的很多高级统计书或有关 Taguchi 方法的书中找到。另外，Neil Sloan 的网站也是一个正交表的很好的来源。参见附件及网站上的其它选项。

## 8. 参考资料

[Copeland2001] Copeland, Lee. "Object-Oriented Testing." *Software Quality Engineering*. STAR East 2001. The Rosen Centre Hotel, Orlando, Florida. 14 May 2001

[Marick1995] Marick, Brian. *The Craft of Software Testing: Subsystem Testing Including Object-Based and Object-Oriented Testing*. Upper Saddle River, NJ: PTR Prentice Hall, 1995.

[Sloane2001] Sloane, Neil A. A Library of Orthogonal Arrays. Information Sciences Research Center, AT&T Shannon Labs. 9 Aug. 2001 <http://www.research.att.com/~njas/ordir/>.

Copyright © 2001 Seilevel, Inc. All rights reserved.

For problems or questions regarding this article, contact the author.

©Copyright 2001 Seilevel. Site created by Stylefish.

备注：本文由李正新翻译，翻译不当，敬请原谅

李正新 Zension Li

Email: [lzx123963@sina.com](mailto:lzx123963@sina.com)

[zensionli@gmail.com](mailto:zensionli@gmail.com)

MSN: [lizhengxin1981@hotmail.com](mailto:lizhengxin1981@hotmail.com)

QQ: 349390730