

Itaú Unibanco

Itaú

Programa de formação

ITAú analytics.

Módulo I – Fundamentos Computacionais
Sessão 5 - Aula 4 – Teste de Software –
Pairwise – Parte 3
Prof. Dr. Luiz Alberto Vieira Dias
Prof. Dr. Lineu Mialaret



Teste de Software
Pairwise Testing - Parte 3
Matrizes Ortogonais

CEDS -111 - Teste de Software - Semana 5 - Aula 4

Prof. Dr. Luiz Alberto **Vieira Dias** (ITA)
Prof. Dr. Adilson Marques da **Cunha** (ITA)
Prof. Dr. **Lineu** Mialaret (ITA/IFSP)
Prof. Dr. Paulo Marcelo **Tasinaffo** (ITA)



A preparação deste minicurso contou com o apoio dos Alunos do ITA

- Lucas Nadalete, MSc
 - D. Montini, DSc
 - G. Matuck, MSc
 - E. Mineda, MSc

As técnicas **Pairwise Testing**

- Duas técnicas são utilizadas para determinar o número de Casos de Teste em Pairwise Testing:
 - *Orthogonal Arrays* (matrizes ortogonais)
 - *Algoritmo Allpairs* (todos os pares)

Base matemática (1/2)

- **Não há base matemática** (distribuição estatística) para o aparecimento de defeitos no SW
- Para **Orthogonal Arrays** (OA) há uma bem formulada teoria!
- **Hipótese, não comprovada matematicamente, mas observada e medida estatisticamente, na qual a maioria de defeitos é:**
 - *single-mode* (Ex: a função sob teste não funciona) ou
 - *double-mode* (Ex: é o par que é testado, **esta** função/método com **aquela** função/método)

Base matemática (2/2)

- **Orthogonal Arrays** (OA), ou **Matrizes Ortogonais**, é uma técnica que determina **todas** as funções do software, que devem ser testadas, em duplas, para localização de defeitos de SW, gerados por *single e double-modes*
- O **sucesso** desta técnica, estatisticamente comprovado, documentado ou não, é a grande motivação para seu uso

Orthogonal Arrays (OA)

- Começou com **Euler** (quadrados latinos), passou por **Evariste Galois** (teoria de grupos) e foi popularizado por **Genichi Taguchi*** (Yamaha - vide referência), com a sua aplicação para teste de **hardware** (inicialmente) e depois outros autores aplicaram a técnica para **software**

Referência: Phadke, M. S. **Quality Engineering Using Robust Design**. New York, NY: Prentice-Hall, 1989

Definição de Orthogonal Array (OA)

Um **Orthogonal Array** é uma matriz bidimensional (elementos 1 a n_1 , 1 a n_2 , ..., 1 a n_m , em cada coluna) com as seguintes propriedades:

- Escolha duas colunas quaisquer; em cada dupla de colunas todas as combinações dos pares irão aparecer
- Se houver n ($n = 1 \dots N$) repetição(ões) de um par, em um par de colunas, estes pares aparecerão repetidos, em igual n° , em todos os pares de colunas

Para os matemáticos somente...

- Provar que a matriz **transposta** de um **Orthogonal Array** é igual à sua matriz **inversa** (onde **I** é a matriz identidade)

sendo $A A^{-1} = I$

e $A^T = A^{-1}$

ou $A A^T = I$

Exemplo de notação: $L_4(2^3)$

Linhas = 4

Colunas = 3

nMax = 2

	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Aparecem em cada par de colunas:
 $\{1,1\}$ $\{1,2\}$ $\{2,1\}$ $\{2,2\}$

$$L_9(3^4)$$

	1	2	3	4
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

Mixed Orthogonal Array (OA)

- Nem sempre uma OA tem, em cada coluna, um mesmo número máximo de níveis ($n_{\text{Max}} = 2 \dots N$)
- Neste caso é chamado de *Mixed Orthogonal Array*
- A notação, por exemplo, fica:

$$L_{18}(2^1 3^7)$$

- Significando que há uma coluna na matriz, com valor máximo 2 e 7 colunas com valor máximo 3

Passos da OA para teste

1. Identificar as variáveis (funções)
2. Determinar o nº de valores para cada variável
3. Localizar um OA que tenha uma variável para cada coluna e nas colunas o valor máximo da variável esteja contido, a partir de:
 - i) um *site* apropriado; ou
 - ii) lista de Matrizes Ortogonais (OA); ou
 - iii) um fornecedor de OA (Taguchi)
4. Mapear as variáveis (funções) nas células da OA
5. Construir os Casos de Teste

Relembrando o Exemplo 1 de uso de Pairwise Testing

- Um *website* deve operar corretamente com **8 browsers** diferentes:
 - Internet Explorer 6, 7, 8
 - Mozilla 1, 2, 3
 - Opera 10
 - Chrome 9
- Com **3 plug-ins**: RealPlayer, Media Player ou nenhum
- Com **6 sistemas operacionais (SO) cliente**: Windows ME, NT, 2000, XP, Vista e 7
- Recebendo páginas de **3 servers** diferentes: IIS, Apache e Weblogic
- Rodando em **3 SO para servers** diferentes: Windows NT, 2003 e Linux

Passos 1 e 2 (variáveis e valores)

- Browsers: 8
- *Plug-ins*: 3
- SO cliente: 6
- *Servers*: 3
- *SO servers*: 3

Matriz Ortogonal

Colunas = 5

C1 = 1...8

C2 = 1...3

C3 = 1...6

C4 = 1...3

C5 = 1...3

Linhas = ?

- Total = **1296** combinações!
- Teria que fazer **1296** Casos de Teste

Passo 3 – escolha da OA

- A OA “perfeita” seria $L_x(8^1 6^1 3^3)$
- Como não existem **matematicamente** OA's para todos os índices, procura-se dentre as existentes a que mais se aproxima
- No caso é a $L_{64}(8^2 4^3)$, e então $x = 64$
- OA's maiores podem ser utilizadas, como será visto
- A OA é maior do que o necessário, mas mesmo assim, em vez de **1296** Casos de Teste tem-se **apenas 64!**

OA Perfeita (não existe) $\Rightarrow L_x(8^1 6^1 3^3)$

OA Possível $\Rightarrow L_{64}(8^2 4^3)$



	1	2	3	4	5
1	1	1	1	1	1
2	1	4	3	4	4
3	1	4	2	4	4
4	1	1	4	1	1
5	1	3	5	3	3
6	1	2	7	2	2
...	...	...	...	...	...
61	8	3	5	3	2
62	8	2	7	2	3
63	8	2	6	2	3
64	8	3	8	3	2

Passo 4 (mapear problema no OA 1/10)

- Exemplo coluna 1 (valores de 1 a 8):

Sistema Operacional Cliente (coluna 1, com max = 8)

- IE 6 = 1
- IE 7 = 2
- IE 8 = 3
- Mozilla 1 = 4
- Mozilla 2 = 5
- Mozilla 3 = 6
- Opera 10 = 7
- Chrome 9 = 8

Passo 4 (2/10)

	1	2	3	4	5
1	1	1	1	1	1
2	1	4	3	4	4
3	1	4	2	4	4
4	1	1	4	1	1
5	1	3	5	3	3
6	1	2	7	2	2
...	...	...	...	...	...
61	8	3	5	3	2
62	8	2	7	2	3
63	8	2	6	2	3
64	8	3	8	3	2

Passo 4 (3/10)

	1	2	3	4	5
1	IE 6	1	1	1	1
2	IE 6	4	3	4	4
3	IE 6	4	2	4	4
4	IE 6	1	4	1	1
5	IE 6	3	5	3	3
6	IE 6	2	7	2	2
...	...	...	...	...	...
61	Chrome 9	3	5	3	2
62	Chrome 9	2	7	2	3
63	Chrome 9	2	6	2	3
64	Chrome 9	3	8	3	2

Passo 4 (4/10)

- Exemplo coluna 3 (valores de 1 a 6):

Sistema Operacional Cliente (coluna 3, com max = 6)

- Win ME = 1
 - Win NT = 2
 - Win 2000 = 3
 - Win XP = 4
 - Win Vista = 5
 - Win 7 = 6
- E assim por diante para as demais variáveis

Passo 4 (5/10)

	1	2	3	4	5
1	IE 6	1	1	1	1
2	IE 6	4	3	4	4
3	IE 6	4	2	4	4
4	IE 6	1	4	1	1
5	IE 6	3	5	3	3
6	IE 6	2	7	2	2
...	...	...	...	...	...
61	Chrome 9	3	5	3	2
62	Chrome 9	2	7	2	3
63	Chrome 9	2	6	2	3
64	Chrome 9	3	8	3	2

Passo 4 (6/10)

	1	2	3	4	5
1	IE 5.0	1	Win ME	1	1
2	IE 5.0	4	Win 2000	4	4
3	IE 5.0	4	Win NT	4	4
4	IE 5.0	1	Win XP	1	1
5	IE 5.0	3	Win Vista	3	3
6	IE 5.0	2	7	2	2
...	...	...	...	...	...
61	Opera 7	3	Win Vista	3	2
62	Opera 7	2	7	2	3
63	Opera 7	2	Win 7	2	3
64	Opera 7	3	8	3	2

Passo 4 (7/10)

- Complete todas as colunas do *array*
- Nos espaços em branco (números da OA) coloque valores **válidos**, de forma aleatória
- Se houverem colunas a mais delete-as (não é o caso deste exemplo)

Passo 4 (8/10)

	1	2	3	4	5
1	IE 6	1	Win ME	1	1
2	IE 6	4	Win 2000	4	4
3	IE 6	4	Win NT	4	4
4	IE 6	1	Win XP	1	1
5	IE 6	3	Win Vista	3	3
6	IE 6	2	7	2	2
...	...	...	...	...	...
61	Chrome 9	3	Win Vista	3	2
62	Chrome 9	2	7	2	3
63	Chrome 9	2	Win 7	2	3
64	Chrome 9	3	8	3	2

Passo 4 (9/10)

	1- Browser	2- Plug-in	3- Client OS	4- Server	5- Server OS
1	IE 6	Real PL	Win ME	IIS	Win NT
2	IE 6	4	Win 2000	4	4
3	IE 6	4	Win NT	4	4
4	IE 6	Real PL	Win XP	IIS	Win NT
5	IE 6	nenhum	Win Vista	WebLogic	Linux
6	IE 6	Media PL	7	Apache	Win 2003
...	...	...	...	...	...
61	Chrome 9	nenhum	Win Vista	WebLogic	Win 2003
62	Chrome 9	Media PL	7	Apache	Linux
63	Chrome 9	Media PL	Win 7	Apache	Linux
64	Chrome 9	nenhum	8	WebLogic	Win 2003

Passo 4 (10/10)

	1- Browser	2- Plug-in	3- Client OS	4- Server	5- Server OS
1	IE 6	Real PL	Win ME	IIS	Win NT
2	IE 6	Real PL	Win 2000	WebLogic	Linux
3	IE 6	Media PL	Win NT	WebLogic	Linux
4	IE 6	Real PL	Win XP	IIS	Win NT
5	IE 6	nenhum	Win Vista	WebLogic	Linux
6	IE 6	Media PL	Win NT	Apache	Win 2003
...	...	...	...	...	...
61	Chrome 9	nenhum	Win Vista	WebLogic	Win 2003
62	Chrome 9	Media PL	Win NT	Apache	Linux
63	Chrome 9	Media PL	Win 7	Apache	Linux
64	Chrome 9	nenhum	Win Vista	WebLogic	Win 2003



Lembre-se do que diz o **testador** de software **Sir Mick Jagger**:

*“You can’t always get what you want,
But if you try sometimes,
You just might find,
You get what you need”*

Citado em [Copeland, 2007]

Passo 5

- Construa os Casos de Teste (cada linha)
- Neste exemplo os 64 Casos de Teste, em vez dos 1296, são as linhas da OA

Passos da OA para teste

1. Identificar as variáveis (funções)
2. Determinar o nº de valores para cada variável
3. Localizar um OA que tenha uma variável para cada coluna e nas colunas o valor máximo da variável esteja contido, a partir de:
 - i) um *site* apropriado; ou
 - ii) lista de Matrizes Ortogonais (OA); ou
 - iii) um fornecedor de OA (Taguchi)
4. Mapear as variáveis (funções) nas células da OA
5. Construir os Casos de Teste

Exemplo 2

Utilização da Tabela Taguchi

- Um sistema tem 4 parâmetros (**par**) de entrada e cada parâmetro pode assumir um dos 3 valores (**val**) diferentes possíveis.
- É um caso de **permutação com repetições** e neste caso, **a ordem conta!**

$$PR_{n,r} = n^r$$

Onde **n** é o total de elementos e **r** o nº de elementos escolhidos que repetem

Arranjos com repetições

- Sendo o nº de coisas a escolher = **n**, e você escolhe **r** dentre elas (repetição permitida, a ordem é considerada)

$$PR(n, r) = n^r$$

No caso do Exemplo 4 → **PR = (val)^{par}**

- O nº total de arranjos c/ repetição, ou os **Casos de Teste**, é:

$$3^4 = 81$$

- Utilizando o **Pairwise Testing** apenas **9 (nove) Casos de Teste** são necessários! **Como?**

Taguchi Orthogonal Array Selector

Number of Levels

values

	2	3	4	5
2	P=2, L=2	P=2, L=3	P=2, L=4	P=2, L=5
3	P=3, L=2	P=3, L=3	P=3, L=4	P=3, L=5
4	P=4, L=2	P=4, L=3	P=4, L=4	P=4, L=5
5	P=5, L=2	P=5, L=3	P=5, L=4	P=5, L=5
6	P=6, L=2	P=6, L=3	P=6, L=4	P=6, L=5
7	P=7, L=2	P=7, L=3	P=7, L=4	P=7, L=5
8	P=8, L=2	P=8, L=3	P=8, L=4	P=8, L=5
9	P=9, L=2	P=9, L=3	P=9, L=4	P=9, L=5
10	P=10, L=2	P=10, L=3	P=10, L=4	P=10, L=5
11	P=11, L=2	P=11, L=3		P=11, L=5
12	P=12, L=2	P=12, L=3		P=12, L=5
13	P=13, L=2	P=13, L=3		
14	P=14, L=2	P=14, L=3		
15	P=15, L=2	P=15, L=3		
16	P=16, L=2	P=16, L=3		
17	P=17, L=2	P=17, L=3		
18	P=18, L=2	P=18, L=3		
19	P=19, L=2	P=19, L=3		
20	P=20, L=2	P=20, L=3		
21	P=21, L=2	P=21, L=3		
22	P=22, L=2	P=22, L=3		
23	P=23, L=2	P=23, L=3		
24	P=24, L=2			
25	P=25, L=2			
26	P=26, L=2			
27	P=27, L=2			
28	P=28, L=2			
29	P=29, L=2			
30	P=30, L=2			
31	P=31, L=2			

Directions: Click on the appropriate square, according to the number of parameters and levels in your experiment.

Site: www.freequality.org

THESE COMBINATIONS
ARE NOT AVAILABLE
– TRY RUNNING A
SMALLER EXPERIMENT

TAGUCHI ARRAY SELECTOR

Parameters →

Levels →

lines

	2	3	4	5
2	4	9	16	25
3	4	9	16	25
4	8	9	16	25
5	8	18	16	25
6	8	18	32	25
7	8	18	32	50
8	12	18	32	50
9	12	27	32	50
10	12	27	32	50
11	12	27		50
12	16	27		50
13	16	27		
14	16	36		
15	16	36		
16	32	36		
17	32	36		
18	32	36		
19	32	36		
20	32	36		
21	32			
22	32			
23	32			
24	32			
25	32			
26	32			
27	32			
28	32			

Number of Levels

	2	3	4	5
2	P=2, L=2	P=2, L=3	P=2, L=4	P=2, L=5
3	P=3, L=2	P=3, L=3	P=3, L=4	P=3, L=5
4	P=4, L=2	P=4, L=3	P=4, L=4	P=4, L=5
5	P=5, L=2	P=5, L=3	P=5, L=4	P=5, L=5
6	P=6, L=2	P=6, L=3	P=6, L=4	P=6, L=5
7	P=7, L=2	P=7, L=3	P=7, L=4	P=7, L=5
8	P=8, L=2	P=8, L=3	P=8, L=4	P=8, L=5
9	P=9, L=2	P=9, L=3	P=9, L=4	P=9, L=5
10	P=10, L=2	P=10, L=3	P=10, L=4	P=10, L=5
11	P=11, L=2	P=11, L=3		P=11, L=5
12	P=12, L=2	P=12, L=3		P=12, L=5
13	P=13, L=2	P=13, L=3		
14	P=14, L=2	P=14, L=3		
15	P=15, L=2	P=15, L=3		
16	P=16, L=2	P=16, L=3		
17	P=17, L=2	P=17, L=3		
18	P=18, L=2	P=18, L=3		
19	P=19, L=2	P=19, L=3		
20	P=20, L=2	P=20, L=3		
21	P=21, L=2	P=21, L=3		
22	P=22, L=2	P=22, L=3		
23	P=23, L=2	P=23, L=3		
24	P=24, L=2			
25	P=25, L=2			
26	P=26, L=2			
27	P=27, L=2			
28	P=28, L=2			
29	P=29, L=2			
30	P=30, L=2			
31	P=31, L=2			

THESE COMBINATIONS
ARE NOT AVAILABLE
– TRY RUNNING A
SMALLER EXPERIMENT

4 parâmetros

3 valores (*levels*)
podem ser
assumidos por cada
parâmetro

$L_9(3^4)$

9 Casos de Teste!

Orthogonal Array

$L_9(3^4) \rightarrow$ Resultado = 9 experimentos
(4 parâmetros: 3 valores (*levels*) para cada parâmetro)

Experimentos	Coluna 1 – P1	Coluna 2 – P2	Coluna 3 – P3	Coluna 4 – P4
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

Fonte: <http://www.york.ac.uk/depts/maths/tables/orthogonal.htm>

Exemplo 3

- Um sistema tem 13 parâmetros e cada um deles pode tomar um de 3 valores diferentes
- O número de combinações é 3^{13}
- **$3^{13} = 1.594.323$** combinações!
- Utilizando o *Pairwise Testing* apenas 27 Casos de Teste são necessários, segundo Copeland (2007). **Ver as tabelas de Taguchi, na Internet!**

Number of Levels

	2	3	4	5
2	P=2, L=2	P=2, L=3	P=2, L=4	P=2, L=5
3	P=3, L=2	P=3, L=3	P=3, L=4	P=3, L=5
4	P=4, L=2	P=4, L=3	P=4, L=4	P=4, L=5
5	P=5, L=2	P=5, L=3	P=5, L=4	P=5, L=5
6	P=6, L=2	P=6, L=3	P=6, L=4	P=6, L=5
7	P=7, L=2	P=7, L=3	P=7, L=4	P=7, L=5
8	P=8, L=2	P=8, L=3	P=8, L=4	P=8, L=5
9	P=9, L=2	P=9, L=3	P=9, L=4	P=9, L=5
10	P=10, L=2	P=10, L=3	P=10, L=4	P=10, L=5
11	P=11, L=2	P=11, L=3		P=11, L=5
12	P=12, L=2	P=12, L=3		P=12, L=5
13	P=13, L=2	P=13, L=3		
14	P=14, L=2	P=14, L=3		
15	P=15, L=2	P=15, L=3		
16	P=16, L=2	P=16, L=3		
17	P=17, L=2	P=17, L=3		
18	P=18, L=2	P=18, L=3		
19	P=19, L=2	P=19, L=3		
20	P=20, L=2	P=20, L=3		
21	P=21, L=2	P=21, L=3		
22	P=22, L=2	P=22, L=3		
23	P=23, L=2	P=23, L=3		
24	P=24, L=2			
25	P=25, L=2			
26	P=26, L=2			
27	P=27, L=2			
28	P=28, L=2			
29	P=29, L=2			
30	P=30, L=2			
31	P=31, L=2			

THESE COMBINATIONS
ARE NOT AVAILABLE
– TRY RUNNING A
SMALLER EXPERIMENT

$P = 13; L=3 ;$

$L_x(3^{13}) \rightarrow L_{27}(3^{13})$

$x = 27$

OBS:

Testando todas as
possibilidades =
1.594.323 permutações!

TAGUCHI ARRAY SELECTOR

Parameters→	Levels→	2	3	4	5
2		4	9	16	25
3		4	9	16	25
4		8	9	16	25
5		8	18	16	25
6		8	18	32	25
7		8	18	32	50
8		12	18	32	50
9		12	27	32	50
10		12	27	32	50
11		12	27		50
12		16	27		50
13		16	27		50
14		16	36		
15		16	36		
16		32	36		
17		32	36		
18		32	36		
19		32	36		
20		32	36		
21		32			
22		32			
23		32			
24		32			
25		32			
26		32			
27		32			
28		32			

lines



Exemplo 4 – Exercício para Casa (1/3)

Site: www.freequality.org

- Utilizando a Técnica de **Pairwise**, com **Matrizes Ortogonais**, mostre **todos** os Casos de Teste para o seu problema individual, seguindo os passos abaixo:

1. Modele o seu problema individualmente, na forma $L_x (N_1^{P1}, N_2^{P2}, \dots)$, onde x é o número de linhas da Matriz Ortogonal de Taguchi (a ser determinado por você) e, conseqüentemente, o número de Casos de Teste, e o **simplifique** para uso pela **Tabela Taguchi**

Exercício para casa (2/3)

(continuação)

2. Identifique seus $N_1, N_2, \dots$ (níveis)
3. Identifique seus $P_1, P_2, \dots$ (parâmetros)
4. Escolha sua Matriz Ortogonal (Taguchi)
5. Liste os Casos de Teste
6. **Desafio:** Se **não** utilizasse Pairwise, quantos Casos de Teste seriam necessários?

Exercício para casa (3/3)

Liste **todos** os **Casos de Teste Pairwise** para um programa em um *website* que deve operar corretamente com até:

- **4 browsers (B)** diferentes: Internet Explorer –B1, Safari –B2, Chrome –B3 e Opera –B4
- **3 sistemas operacionais (SC) cliente:** Windows-SC1, MacOS-SC2, Linux-SC3
- **2 Plug-ins (P):** Media Player-P1, Real Player-P2
- Rodando em **3 SO para servers (SS)** diferentes: Windows NT -SS1, MacSERVER-SS2 e LinuxSS3

DICA IMPORTANTE: Use as Matrizes Ortogonais da Tabela do Taguchi (internet) para **listar** os casos de teste. Seu caso pode requerer testar somente **alguns** dos 4 itens acima.