

Itaú Unibanco

Itaú

Programa de formação

ITAÚ analytics.

Módulo I – Fundamentos Computacionais
Sessão 5 - Aula 2 – Teste de Software –
Pairwise – Parte 1
Prof. Dr. Luiz Alberto Vieira Dias
Prof. Dr. Lineu Mialaret



ITAú Analytics – Fundamento de Computação – CEDS 111
2º SEMESTRE 2018

Teste de Software

Pairwise Testing - Parte 2

Matrizes Ortogonais

CEDS -111 - Teste de Software - Semana 5 - Aula 3

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

Considerações iniciais para *Pairwise Testing* – Exemplo 1

- 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

Considerações iniciais para Pairwise Testing

- *Browsers: 8*
 - *Plug-ins: 3*
 - *SO cliente: 6*
 - *Servers: 3*
 - *SO servers: 3*
-
- Quantas escolhas (*arranjos*, combinações, permutações) são possíveis?

Um pouco de Matemática:

Permutações – Arranjos – (1/2)

- “Arranjo Permutação simples de n elementos tomados r a r , onde $n > 0$ e $r \leq n$, é qualquer ordenação de r elementos dentre os n elementos, em que cada maneira de tomar os elementos se diferenciam pela **ordem** e **natureza** dos elementos”

$$P(n,r) = n! / (n-r)!$$

$$P(n,n) = P(n) = n!$$

Onde n é o total de elementos e r o nº de elementos escolhidos.” Onde: n e $r \in \{N\}$

Fonte: Wikipédia



Permutações – Arranjos (2/2)

- “A permutação com **repetição** é usada quando a **ordem** dos elementos **importa** e cada elemento pode ser **contado mais de uma** vez.

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

Onde **n** é o total de elementos e **r** o nº de elementos escolhidos.”

Fonte: Wikipédia

Exemplo de permutação com repetição



PS- Este tipo de cadeado deveria ser chamado de **cadeado de permutação com repetições** e não “cadeado de combinação”...

Permutações e Combinações

- **Combinações:** a ordem **não importa**
- **Arrajos e Permutações:** a ordem **importa**
- Arranjos, Combinações e Permutações podem ser, ou não, com **repetições**

Regras gerais de Análise Combinatória

“Regra do Produto: A Regra do Produto diz que se um elemento **H** pode ser escolhido de **m** formas diferentes e se depois de cada uma dessas escolhas, um outro elemento **M** pode ser escolhido de **n** formas diferentes, a escolha do par **(H,M)** nesta ordem poderá ser realizada de **m x n** formas”

- Dá o número de combinações, sem repetição

Nº de escolhas para o Exemplo 1

Pairwise Testing

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

$$\text{Escolhas} = n_1 \times n_2 \times \dots \times n_n =$$

$$8 \times 3 \times 6 \times 3 \times 3 = 1296$$

Total = **1296** possibilidades!

Exemplo 2

- Um sistema tem 4 **parâmetros** de entrada e cada um deles pode tomar um de 3 valores diferentes {1,2,3}.
- Caso de **permutação com repetições**, a ordem importa!

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

Permutações e Combinações

- **Combinações** (a ordem **não importa**):

Sem repetições: $C(n,r) = n! / (r!(n-r)!)$

Com repetições: $CR(n, r) = C(n + r - 1, r)$

- **Permutações**: (a ordem **importa**):

$$P(n,r) = n!/(n-r)! \quad P(n,n) = n! \quad PR(n,r) = n^r$$

Permutações com repetições

- Sendo **n** o número de elementos a escolher, e você escolhe **r** dentre eles (Repetição **permitida**, a **ordem importa**)

$$PR = n^r$$

- No caso do Exemplo:

$$PR = (Niv)^{par}$$

- O número total de permutações c/ repetição, ou de **casos de teste**, é:

$$3^4 = 81$$

- Utilizando o **pairwise testing**, apenas **9 (nove)** **casos de teste** são necessários! **Como?**

Orthogonal Array

$L_9(3^4) \rightarrow$ Resultado = 9 experimentos
(4 parâmetros: 3 níveis (*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

Taguchi Orthogonal Array Selector

Number of Parameters

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

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

Instruções: Clique no quadrado apropriado, de acordo com o número de parâmetros e níveis de seu experimento

Site: www.freequality.org

TAGUCHI ARRAY SELECTOR

Levels->		2	3	4	5			
Parameters	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						

Permutações e Combinações

- **Combinações:** a ordem **não importa**
- **Arranjos e Permutações:** a ordem **importa**
- Cada um pode ser:
 - Com repetições
 - Sem repetições

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 assumi-
dos por cada parâ-
metro

$L_9(3^4)$

9 Casos de Teste!

Outro exemplo

- 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

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!

Base matemática (1/2)

- **Não há** (para o aparecimento de defeitos no SW) para OA há uma bem formulada teoria!
- **Hipótese:** a maioria dos defeitos é single-mode (a função sob teste não funciona – usa-se PyUnit) ou double-mode (é o par: **esta** função/modo com **aquela** função/modo que não funciona, embora os outros pares funcionem adequadamente)

Base matemática (2/2)

- **Pairwise** é uma técnica que determina **todas** as funções que devem ser testadas para localização de defeitos de SW gerados por *single* e *double-modes*
- O **sucesso** desta técnica, comprovado estatisticamente, documentado e não documentado, é a grande motivação para seu uso e resultado prático positivo!

Aplicação

- Infelizmente esta técnica **não** resolve todos os casos
- Quando o número de níveis (*levels*) é “**muito**” grande (por exemplo **>30**), fica difícil
- Outro problema é para níveis **contínuos**, pois a técnica só funciona para valores **discretos** de níveis e parâmetros (estes em geral são **discretos** mesmo)
- **IMPORTANTE** – Se puder utilizar classes de equivalência, a técnica **se aplica** mesmo para **níveis contínuos**

Cuidados

- Em casos **críticos**, onde se espera que **triplets** possam causar defeitos e consequentemente falhas, estes devem ser identificados e **testados**
- Existem software que identificam e enumeram os casos de teste de **triplets** e acima, de forma automatizada. Exemplo:

Hexawise