5. Generation from Combinatorial Design II

Andrea Polini

Software Engineering II – Software Testing
MSc in Computer Science
University of Camerino

January 20^{th}, 2015

Pairwise design for multivalued factors

In most practical cases factors can assume more than just two levels

- SAMNA cannot be applied
- MOLS(n) can be used to derive test set to satisfy the pairwise criterion

PDMOLS algorithm

Input: n - number of factors
Output: a test set satisfying the pairwise criterion

1. Label the factors as $F_1, F_2, \ldots, F_n$ such that the following ordering constraint is satisfied: $|F_1| \geq |F_2| \geq \ldots \geq |F_{n-1}| \geq |F_n|$. Let $b = |F_1|$ and $k = |F_2|$.
2. Prepare a table containing n columns and $b \times k$ rows divided into b blocks. Label the columns as $F_1, F_2, \ldots, F_n$. Each block contains k rows.
3. Fill column F_1 with 1s in block 1, 2s in block 2 and so on. Fill block 1 of columns F_2 with the sequence 1,2,. . . ,k.
4. Find $s = n(k)$ MOLS of order k. Denote them as $M_1, M_2, \ldots, M_s$. Note that $s < k$ for $k > 1$.
5. Fill block 1 of colum F_3 with entried from column 1 of M_1, block 2 with entries from column 2 of M_1, and so on. If the number of blocks $b = b_1 > k$ then reuse columns of M_1 to fill rows in the remaining $b_1 - k$ blocks. Repeat the procedure for the remaining columns. If $s < (n - 2)$ then fill columns by randomlyselecting the values.
6. Generate the test set from the rows of the resulting filled table.

PDMOLS and combination constraints

In most real cases it is not meaningful/possible to use all the possible tests generated according to PDMOLS.

- If the factor X assumes level *x* than factor Y cannot assume level *y*

The AGTCS system

Factor	**Levels**			
F'_1:Hardware (H)	PC	Mac		
F'_2:OS (O)	Win2000	Win XP	OS9	OS10
F'_3:Browser(B)	Explorer	Netscape 4.x	Firefox	Chrome
F'_4:PI(P)	New	Existing		

How to handle constraints

- The “PC” level is incompatible with “OSx” families.
- The “Mac” level is incompatible with “Win OS” families.
- there are invalid levels

PDMOLS and combination constraints

In most real cases it is not meaningful/possible to use all the possible tests generated according to PDMOLS.

- If the factor X assumes level *x* than factor Y cannot assume level *y*

The AGTCS system

Factor	Levels			
F_1':Hardware (H)	PC	Mac		
F_2':OS (O)	Win2000	Win XP	OS9	OS10
F_3':Browser(B)	Explorer	Netscape 4.x	Firefox	Chrome
F_4':PI(P)	New	Existing		

How to handle constraints

- The "PC" level is incompatible with "OSx" families.
- The "Mac" level is incompatible with "Win OS" families.
- there are invalid levels

MOLS shortcomings

- A sufficient number of MOLS might not exist for the problem at hand
- MOLS assist with the generation of balanced design but the number of configuration could be larger than necessary

To address such issues other approaches have been proposed:

- Orthogonal Arrays (and variants)
- Covering Arrays

Orthogonal Arrays

Definition

An Orthogonal Array is an $N \times k$ matrix in which the entries are from a finite set S of s symbols such that any $N \times t$ subarray contains each t-uple exactly the same number of times. Such an orthogonal array is denoted by $OA(N, k, s, t)$. The index of an orthogonal array, denoted by λ, is equal to N/s^t.

Example

Run	F_1	F_2	F_3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Orthogonal arrays assume that each factor assumes values from the same set of s values. This is not generally the case and Mixed Level Orthogonal Arrays can be used in such contexts.

Orthogonal Arrays

Definition

An Orthogonal Array is an $N \times k$ matrix in which the entries are from a finite set S of s symbols such that any $N \times t$ subarray contains each t-uple exactly the same number of times. Such an orthogonal array is denoted by $OA(N, k, s, t)$. The index of an orthogonal array, denoted by λ, is equal to N/s^t.

Example

Run	F_1	F_2	F_3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

Orthogonal arrays assume that each factor assumes values from the same set of s values. This is not generally the case and Mixed Level Orthogonal Arrays can be used in such contexts.

Covering Arrays

A Covering Array, denoted as $CA(N, k, s, t)$ is an $N \times k$ matrix in which entries are from a finite set S of s symbols such that each $N \times t$ subarray contains each possible t-uple at least λ times. In this case we have an unbalanced design.

Run	F_1	F_2	F_3	F_4	F_5
1	1	1	1	1	1
2	2	1	1	2	2
3	1	2	1	2	1
4	1	1	2	1	2
5	2	2	1	1	2
6	2	1	2	2	1
7	1	2	2	2	2
8	2	2	2	1	1

Run	F_1	F_2	F_3	F_4	F_5
1	1	1	1	1	1
2	2	2	1	2	1
3	1	2	2	1	2
4	2	1	2	2	2
5	2	2	1	1	2
6	1	1	1	2	2

Generation of Covering Arrays

IPO Procedure permits the derivation of mixed-level covering arrays for pairwise designs.

Summary from test generation strategies

- Generation from requirements
- Generation from formal models
- Generation using combinatorial design