Chapter 2
Mathematical Construction Methods

Abstract This chapter presents some representative mathematical methods that are commonly used in the construction of orthogonal arrays and covering arrays, as well as some bounds on the size of such arrays.

As we mentioned in Chap. 1, combinatorial testing is closely related to combinatorics, which is a branch of mathematics. For an introduction to combinatorial theory and combinatorial designs, see [3, 5]. Mathematicians have obtained many results in this field. Some of them can be used to construct combinatorial designs directly. Mathematical methods, in particular product or recursive constructions, can be employed to build large instances of orthogonal arrays and covering arrays. However, many of these methods are applicable only in specific cases.

2.1 Mathematical Methods for Constructing Orthogonal Arrays

As a combinatorial design with beautiful balancing property, the orthogonal array has long been the interest of mathematicians. There are many mathematical results about OA, either dealing with construction or proving its nonexistence given some parameters. For simplicity, here we just review a few ones that are easy to understand.

2.1.1 Juxaposition

Theorem 2.1 *If an $OA(N', s'_1 \cdot s_2 \cdots s_k, t)$ and an $OA(N'', s''_1 \cdot s_2 \cdots s_k, t)$ both exist, and $\frac{N'}{s'_1} = \frac{N''}{s''_1}$, then an $OA(N' + N'', (s'_1 + s''_1) \cdot s_2 \cdots s_k, t)$ exists.*

The proof of this theorem is trivial. Given an $\mathrm{OA}(N', s'_1 \cdot s_2 \cdots s_k, t)$ and an $\mathrm{OA}(N'', s''_1 \cdot s_2 \cdots s_k, t)$, we just need to relabel the elements in the first column of one array, and put it underneath the other array. Obviously the resulting array is an $\mathrm{OA}(N' + N'', (s'_1 + s''_1) \cdot s_2 \cdots s_k, t)$.

J. Zhang et al., *Automatic Generation of Combinatorial Test Data*,
SpringerBriefs in Computer Science, DOI 10.1007/978-3-662-43429-1_2

Fig. 2.1 Transform an $OA(8, 4 \cdot 2^4, 2)$ to an $OA(8, 2^6, 2)$ by splitting

$$
\begin{array}{|c|cccc}
\hline
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 \\
2 & 0 & 1 & 0 & 1 \\
2 & 1 & 0 & 1 & 0 \\
3 & 0 & 1 & 1 & 0 \\
3 & 1 & 0 & 0 & 1 \\
\hline
\end{array}
\qquad
\begin{array}{|cc|cccc}
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 \\
1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 \\
\hline
\end{array}
$$

2.1.2 Splitting

Theorem 2.2 *If an* $OA(N, (p \times s) \cdot s_2 \cdots s_k, t)$ *exists, then an* $OA(N, p \cdot s \cdot s_2 \cdots s_k, t)$ *also exists.*

For each element $i \in \{0, 1, \ldots, p \times s - 1\}$, let $v = i \bmod s$, $u = (i - v)/s$. Then split the first column of $\mathrm{OA}(N, (p \times s) \cdot s_2 \cdots s_k, t)$ into two columns by replacing element i with two elements u and v. The resulting array is an $\mathrm{OA}(N, p \cdot s \cdot s_2 \cdots s_k, t)$. For example, we can construct an $\mathrm{OA}(8, 2^6, 2)$ from an $\mathrm{OA}(8, 4 \cdot 2^4, 2)$ by splitting the first column, as illustrated in Fig. 2.1.

2.1.3 Hadamard Construction

An $n \times n$ $(-1, 1)$-matrix H_n is an Hadamard matrix if $H_n H_n^T = nI$.

Example 2.1 The matrix

$$
H_4 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}
$$

is an Hadamard matrix because:

$$
\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}.
$$

Theorem 2.3 *When an Hadamard matrix of order* $n \geq 4$ *exists, an* $OA(2n, 2^n, 3)$ *and an* $OA(8n, 2^{2n-4} \cdot 4^2, 3)$ *exist.*

Due to space limitation, we just elaborate the first case. Suppose H_n is an Hadamard matrix of order n $(n \geq 4)$, then $\frac{H_n}{-H_n}$ is an $\mathrm{OA}(2n, 2^n, 3)$. Figure 2.2 shows an $\mathrm{OA}(8, 2^4, 3)$ constructed with the Hadamard matrix H_4 in the above example. By convention, we replace the element '−1' with '0'.

Fig. 2.2 Construct an $OA(8, 2^4, 3)$ from an Hadamard matrix of order 4

$$\left(\begin{array}{rrrr} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ \hline -1 & -1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 \end{array}\right) \qquad \begin{array}{cccc} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}$$

2.1.4 Zero-Sum Construction

Theorem 2.4 *Suppose Z_s is the additive group of integers modulo s. For each of the s^t t-tuples over Z_s, form a row vector of length $t+1$ by adjoining in the last column the negative of the sum of the elements in the first t columns. Use these vectors to form an $s^t \times (t+1)$ array, and the array is an $OA(s^t, s^{t+1}, t)$.*

Example 2.2 Assume that we would like to construct an $OA(16, 2^5, 4)$. Firstly, we enumerate all tuples of length 4 over the set $\{0, 1\}$, i.e., the 16 vectors ranging from $\langle 0, 0, 0, 0\rangle$ to $\langle 1, 1, 1, 1\rangle$. Secondly, for each vector we add an extra element in the end, so that the sum of all elements is divisible by 2. For instance, $\langle 0, 0, 0, 0\rangle$ is extended to $\langle 0, 0, 0, 0, 0\rangle$, and $\langle 0, 1, 1, 1\rangle$ is extended to $\langle 0, 1, 1, 1, 1\rangle$. Finally, all these vectors are used as row vectors to form the target array, as Fig. 2.3 illustrates.

2.1.5 Construction from Mutually Orthogonal Latin Squares

Theorem 2.5 *There exist k mutually orthogonal Latin squares of order n (k-MOLS(n)) if and only if there exists an $OA(n^2, n^{k+2}, 2)$.*

Fig. 2.3 $OA(16, 2^5, 4)$

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{array}$$

Fig. 2.4 Construct an OA(9, 3^4, 2) from 2-MOLS(3)

$$\begin{matrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{matrix} \qquad \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 1 & 0 & 2 \\ 1 & 2 & 1 & 0 \\ 2 & 0 & 1 & 2 \\ 2 & 1 & 2 & 0 \\ 2 & 2 & 0 & 1 \end{matrix}$$

Suppose that $\{LS_1 \cdots LS_k\}$ is a set of k mutually orthogonal Latin squares of order n, and $LS_f(i, j)$ denotes the element in the ith row, jth column of LS_f. For each combination of i and j ($0 \le i \le n-1, 0 \le j \le n-1$), we form a $(k+2)$-tuple $\langle i, j, LS_1(i, j), \ldots, LS_k(i, j)\rangle$. Then we use these tuples as row vectors to form a matrix. Obviously the resulting matrix is an OA(n^2, n^{k+2}, 2).

Example 2.3 In Fig. 2.4, the matrix on the right is an OA(9, 3^4, 2) constructed from the 2-MOLS(3) on the left. The first column represents the row indices of 2-MOLS(3); the second column represents the column indices of 2-MOLS(3); the third column contains the elements in the first latin square, and the last column contains the elements in the second latin square.

2.2 Mathematical Methods for Constructing Covering Arrays

A covering array is optimal if it has the smallest possible number N of rows. As mentioned in Chap. 1, this smallest number is called the covering array number (CAN). Formally, $\text{CAN}(d_1 \cdot d_2 \cdots d_k, t) = \min\{N | \exists \text{CA}(N, d_1 \cdot d_2 \cdots d_k, t)\}$. CAN is a vital attribute of covering arrays. It serves as the lower bound of the size of the test suite. CAN can always be obtained as the by-product of the construction of CAs.

2.2.1 Simple Constructions

2.2.1.1 Column-Collapsing

Given a $\text{CA}(N, d_1 \cdots d_i \cdots d_k, t)$, if we delete an arbitrary column i, we will get a $\text{CA}(N, d_1 \cdots d_{i-1} \cdot d_{i+1} \cdots d_k, t)$. So

$$\text{CAN}(d_1 \cdots d_{i-1} \cdot d_{i+1} \cdots d_k, t) \le \text{CAN}(d_1 \cdots d_i \cdots d_k, t).$$

2.2.1.2 Symbol-Collapsing

Given a $\mathrm{CA}(N, d_1 \cdots d_i \cdots d_k, t)$, if we replace a symbol v in the ith column with any symbol in this set $\{0, 1, \ldots, d_i - 1\}$ other than v itself, we will get a $\mathrm{CA}(N, d_1 \cdots (d_i - 1) \cdots d_k, t)$. So

$$\mathrm{CAN}(d_1 \cdots (d_i - 1) \cdots d_k, t) \le \mathrm{CAN}(d_1 \cdots d_i \cdots d_k, t).$$

2.2.1.3 Derivation

Given a $\mathrm{CA}(N, d_1 \cdots d_i \cdots d_k, t)$, if we select a symbol v from the ith column, then extract the rows with a symbol v on the ith column, and finally delete the ith column, we will get a $\mathrm{CA}(M, d_1 \cdots d_{i-1} \cdot d_{i+1} \cdots d_k, t-1)$, where M is no less than the product of the $t-1$ largest levels (excluding d_i). And we have

$$d_i \times \mathrm{CAN}(d_1 \cdots d_{i-1} \cdot d_{i+1} \cdots d_k, t-1) \le \mathrm{CAN}(d_1 \cdots d_i \cdots d_k, t).$$

2.2.1.4 Juxaposition

Given a $\mathrm{CA}(N', d'_1 \cdot d_2 \cdots d_k, t)$ and a $\mathrm{CA}(N'', d''_1 \cdot d_2 \cdots d_k, t)$, we can construct a $\mathrm{CA}(N' + N'', (d'_1 + d''_1) \cdot d_2 \cdots d_k, t)$ by relabeling the symbols in the first column of one CA and putting it underneath the other.

In particular, we can construct a $\mathrm{CA}(\ell N, \ell d_1 \cdot d_2 \cdots d_k, t)$ from a $\mathrm{CA}(N, d_1 \cdot d_2 \cdots d_k, t)$, so we have

$$\mathrm{CAN}(\ell d_1 \cdot d_2 \cdots d_k, t)) \le \ell \cdot \mathrm{CAN}(d_1 \cdot d_2 \cdots d_k, t).$$

For example, it is not difficult to find an instance of CA$(25, 5^2 4^1 3^2 2^7, 2)$. Then we can produce a CA$(100, 10^2 4^1 3^2 2^7, 2)$ by applying juxaposition twice. Obviously this result is optimal.

2.2.2 Recursive Constructions

For many combinatorial objects, we can use mathematical results to produce large objects from smaller ones. In this subsection, we briefly describe some of the results for covering arrays.

$$
\begin{array}{r}
\\ \\ N\ \textbf{rows} \\ \\ \\ \\ M\ \textbf{rows} \\ \\
\end{array}
\left|\begin{array}{ccc|ccc|c|ccc}
a_{11} & a_{12} \cdots & a_{1k} & a_{11} & a_{12} \cdots & a_{1k} & \cdots & a_{11} & a_{12} \cdots & a_{1k} \\
a_{21} & a_{22} \cdots & a_{2k} & a_{21} & a_{22} \cdots & a_{2k} & \cdots & a_{21} & a_{22} \cdots & a_{2k} \\
 & \vdots & & & \vdots & & \cdots & & \vdots & \\
a_{N1} & a_{N2} \cdots & a_{Nk} & a_{N1} & a_{N2} \cdots & a_{Nk} & \cdots & a_{N1} & a_{N2} \cdots & a_{Nk} \\
\hline
b_{11} & b_{11} \cdots & b_{11} & b_{12} & b_{12} \cdots & b_{12} & \cdots & b_{1\ell} & b_{1\ell} \cdots & b_{1\ell} \\
b_{21} & b_{21} \cdots & b_{21} & b_{22} & b_{22} \cdots & b_{22} & \cdots & b_{2\ell} & b_{2\ell} \cdots & b_{2\ell} \\
 & \vdots & & & \vdots & & \cdots & & \vdots & \\
b_{M1} & b_{M1} \cdots & b_{M1} & b_{M2} & b_{M2} \cdots & b_{M2} & \cdots & b_{M\ell} & b_{M\ell} \cdots & b_{M\ell}
\end{array}\right|
$$

Fig. 2.5 The product of $A \otimes B$ (Reprinted with permission from [9]. Copyright 2008, Elsevier Inc.)

Fig. 2.6 $\mathrm{CA}(N + (v - 1) \cdot M, v^{2k}, 3)$

$$
\boxed{\begin{array}{ll}
A & A \\
B & B^{\pi^1} \\
B & B^{\pi^2} \\
 & \cdots \\
B & B^{\pi^{v-1}}
\end{array}}
$$

2.2.2.1 Strength-2 Covering Arrays

For strength $t = 2$ (i.e., pairwise testing), Stevens and Mendelsohn proved the following theorem [8]:

Theorem 2.6 (Products of Strength-2 CAs) *If a* $\mathrm{CA}(N, v^k, 2)$ *and a* $\mathrm{CA}(M, v^\ell, 2)$ *both exist, then a* $\mathrm{CA}(N + M, v^{k\ell}, 2)$ *also exists.*

Let $A = (a_{ij})$ be $\mathrm{CA}(N, v^k, 2)$ and $B = (b_{ij})$ be $\mathrm{CA}(M, v^\ell, 2)$. Then the $(N + M) \times k\ell$ array $C = (c_{ij}) = A \otimes B$ in Fig. 2.5 is a $\mathrm{CA}(N + M, v^{k\ell}, 2)$.

Similarly, there are product methods for pairwise mixed covering arrays. For details, see [4]. The methods can be used recursively.

2.2.2.2 Strength-3 Covering Arrays

For covering arrays of strength 3, Chateauneuf and Kreher proved the following theorem [1]:

Theorem 2.7 *If a* $\mathrm{CA}(N, v^k, 3)$ *and a* $\mathrm{CA}(M, v^k, 2)$ *both exist, then a* $\mathrm{CA}(N+(v-1) \cdot M, v^{2k}, 3)$ *also exists.*

Suppose A is a $\mathrm{CA}(N, v^k, 3)$, and B is a $\mathrm{CA}(M, v^k, 2)$. Let $\{\pi^i | 1 \le i \le v - 1\}$ be the cyclic group of permutations generated by $\pi = (0, 1, \ldots, v - 1)$. In other words, π^i is a bijection that maps symbol s to $(s + i)$ **mod** v. Let B^{π^i} be the matrix obtained by applying the permutation π^i to B. Then a $\mathrm{CA}(N + (v - 1) \cdot M, v^{2k}, 3)$ can be constructed in the way illustrated in Fig. 2.6.

$$\begin{array}{cccc} 0&0&0&0\\ 1&1&1&1\\ 1&0&0&1\\ 1&0&1&0\\ 0&1&0&1\\ 1&1&0&0\\ 0&0&1&1\\ 0&1&1&0 \end{array} \qquad \begin{array}{cccc} 0&0&0&0\\ 0&1&1&1\\ 1&0&1&1\\ 1&1&0&1\\ 1&1&1&0 \end{array} \qquad \left|\begin{array}{cccc|cccc} 0&0&0&0&0&0&0&0\\ 1&1&1&1&1&1&1&1\\ 1&0&0&1&1&0&0&1\\ 1&0&1&0&1&0&1&0\\ 0&1&0&1&0&1&0&1\\ 1&1&0&0&1&1&0&0\\ 0&0&1&1&0&0&1&1\\ 0&1&1&0&0&1&1&0\\ \hline 0&0&0&0&1&1&1&1\\ 0&1&1&1&1&0&0&0\\ 1&0&1&1&0&1&0&0\\ 1&1&0&1&0&0&1&0\\ 1&1&1&0&0&0&0&1 \end{array}\right|$$

A=CA$(8, 2^4, 3)$ $\qquad$ B=CA$(5, 2^4, 2)$

Fig. 2.7 Product Construction for CA$(13, 2^8, 3)$

Example 2.4 Suppose we are given two covering arrays, A is a CA$(8, 2^4, 3)$ and B is a CA$(5, 2^4, 2)$. Since $v = 2$, we have only one permutation π^1, which is $\pi^1(0) = 1$ and $\pi^1(1) = 0$. So B^{π^1} is actually obtained by permuting symbol '0' and '1' in B. We can form a CA$(13, 2^8, 3)$ with A, B and B^{π^1}, as shown in Fig. 2.7.

2.2.2.3 Covering Arrays of Arbitrary Strength

For covering arrays of arbitrary strength, we have the following theorem [1]:

Theorem 2.8 *If a* CA(N, v^k, t) *and a* CA(M, w^k, t) *both exist, then a* CA$(NM, (vw)^k, t)$ *also exists.*

Suppose A is a CA(N, v^k, t), and B is a CA(M, w^k, t). We construct a series of $N \times k$ matrices C_l with entries $C_l(i, j) =< A(i, j), B(l, j) >$, where $1 \le i \le N$, $1 \le j \le k$, and $1 \le l \le M$. Using these matrices, we form an $NM \times k$ matrix $C = [C_1, \ldots, C_M]^T$. Obviously C is a CA$(NM, (vw)^k, t)$.

2.2.3 Construction Based on Difference Covering Arrays

A difference covering array, or a DCA$(k, n; v)$ is an $n \times k$ array (d_{ij}) with entries from an Abelian group G of order v, such that for any two distinct columns l and h, the difference list

$$\delta_{l,h} = \{d_{1l} - d_{1h}, d_{2l} - d_{2h}, \ldots, d_{nl} - d_{nh}\}$$

contains every element of G at least once.

Example 2.5 The array in Fig. 2.8 is a DCA$(4, 3; 2)$ over Z_2 [10].

2.2.3.1 Strength-2 Covering Arrays

In [2], Colbourn showed that CAs of strength 2 can be easily constructed with DCAs.

Theorem 2.9 *If there exists a* DCA$(k, n; v)$*, then there exists a* CA$(nv, v^k, 2)$.

Given a DCA$(k, n; v)$ (d_{ij}) over the Abelian group G of order v. For each row vector

$$\langle d_{i1}, d_{i2}, \ldots, d_{ik} \rangle \quad (1 \le i \le n)$$

and each element $u \in G$, we construct a row vector

$$\langle (d_{i1} + u), (d_{i2} + u), \ldots, (d_{ik} + u) \rangle.$$

These $n \times v$ row vectors form a CA$(nv, v^k, 2)$.

Example 2.6 From the DCA(4, 3; 2) in Fig. 2.8, we can construct a CA$(6, 2^4, 2)$ in Fig. 2.9 using this method. Since the DCA is over Z_2, each row vector in the DCA would produce two row vectors in the CA by adding '0' and '1' respectively.

2.2.3.2 Strength-3 Covering Arrays

In [6], Ji and Yin proposed two constructive methods to build covering arrays of strength 3 based on DCAs. For simplicity we only introduce the first one.

Theorem 2.10 *If there exists a* DCA$(4, n; v)$*, then there exists a* CA$(nv^2, v^5, 3)$.

Given a DCA$(4, n; v)$ (d_{ij}) over the Abelian group G of order v. For each row vector

$$\langle d_{i1}, d_{i2}, d_{i3}, d_{i4} \rangle \quad (1 \le i \le n),$$

we construct a series of row vectors

Fig. 2.8 DCA(4, 3; 2)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{array}$$

Fig. 2.9 Construct a CA$(6, 2^4, 2)$ from a DCA(4, 3; 2)

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array}$$

Fig. 2.10 Construct a CA(12, 2^5, 3) from a DCA(4, 3; 2)

$$\begin{array}{ccccc}
0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 \\
\hline
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
\hline
1 & 1 & 1 & 1 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
\hline
1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1
\end{array}$$

$$R(i, u, e) = \langle (d_{i1} + u), (d_{i2} + u), (d_{i3} + u + e), (d_{i4} + u + e), e \rangle,$$

where $u, e \in G$. These nv^2 row vectors form a CA(nv^2, v^5, 3).

Example 2.7 From the DCA(4, 3; 2) in Fig. 2.8, we can construct a CA(12, 2^5, 3) in Fig. 2.10 using this method. Each row vector in the DCA would produce four row vectors in the CA since there are four value combinations of u and e.

There are also constructive methods to produce CAs of higher strength from DCAs. For example, CAs of strength 5 can be built with the methods in [7].

References

1. Chateauneuf, M., Kreher, D.: On the state of strength-three covering arrays. J. Comb. Des. **10**(4), 217–238 (2002)
2. Colbourn, C.J.: Combinatorial aspects of covering arrays. Le Matematiche (Catania) **58**, 121–167 (2004)
3. Colbourn, C.J., Dinitz, J.H. (eds.): Handbook of Combinatorial Designs, 2nd edn. Chapman & Hall / CRC, Boca Raton (2006)
4. Colbourn, C.J., Martirosyan, S.S., Mullen, G.L., Shasha, D., Sherwood, G.B., Yucas, J.L.: Products of mixed covering arrays of strength two. J. Comb. Des. **14**(2), 124–138 (2006)
5. Hall Jr, M.: Combinatorial Theory, 2nd edn. Wiley, New York (1998)
6. Ji, L., Yin, J.: Constructions of new orthogonal arrays and covering arrays of strength three. J. Comb. Theory Ser. A **117**, 236–247 (2010)
7. Ji, L., Li, Y., Yin, J.: Constructions of covering arrays of strength five. Des. Codes Cryptogr. **62**(2), 199–208 (2012)
8. Stevens, B., Mendelsohn, E.: New recursive methods for transversal covers. J. Comb. Des. **7**(3), 185–203 (1999)
9. Yan, J., Zhang, J.: A backtracking search tool for constructing combinatorial test suites. J. Syst. Softw. **81**(10), 1681–1693 (2008)
10. Yin, J.: Constructions of difference covering arrays. J. Comb. Theory Ser. A **104**(2), 327–339 (2003)

http://www.springer.com/978-3-662-43428-4

Automatic Generation of Combinatorial Test Data
Zhang, J.; Zhang, Z.; Ma, F.
2014, XI, 90 p. 31 illus., Softcover
ISBN: 978-3-662-43428-4