Journal of Southeast University (English Edition) Vol. 27, No. 3, pp. 340 - 342 Sept. 2011 ISSN 1003—7985

Recursive constructions for t-covering arrays

Zha Rijun[1] Zhang Deping[2] Xu Baowen[3]

([1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China)
([2] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)
([3]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract: A t-covering array of size N, degree k, order v and strength t is an $N\times k$ array with entries from a set of v symbols such that any $N\times t$ subarray contains a t-tuple of v symbols at least once as a row. This paper presents a new algebraic recursive method for constructing covering arrays based on difference matrices. The method can extend parameter factors on the existing covering arrays and cover all the combinations of any t parameter factors ($t\geqslant 2$). The method, which recursively generates high strength covering arrays, is practical. Meanwhile, the theoretical derivation and realization of the proposed algebraic recursive algorithm are given.

Key words: covering array; orthogonal array; difference matrix
doi: 10.3969/j.issn.1003 - 7985.2011.03.022

A t-covering array of size N, degree k, order v and strength t is an $N\times k$ array with entries from a set of v symbols such that any $N\times t$ subarray contains a t-tuple of v symbols at least once as a row ($t\geqslant 2$). We denote such an array by CA(N; t, k, v). The covering array is optimal if it has the smallest possible number of N rows. This number is the covering array number, that is CAN(t, k, v) = min{N: $\exists$ CA(N; t, k, v)}. Covering arrays can be viewed as a generalization of orthogonal arrays. In fact, if we require that each $N\times t$ subarray contains a t-tuple of symbols exactly λ times, then we have a t-orthogonal array, denoted as $\mathrm{OA}_\lambda(N; t, k, v)$. In particular, if $\lambda=1$, we denote it as OA(N; t, k, v), which is an optimal covering array. Covering arrays have attracted attention in recent years due to the fact that covering arrays have a number of applications in software testing[1-2], data compression and intersecting codes[3], etc. Existing works on covering arrays mainly focus on covering arrays of strength $t=2$ or $t=3$[4-7]. However, there has not been much research on the covering arrays for strength of an arbitrary value t[8-11]. Meanwhile, the construction methods of t-covering arrays are either heuristic[10-11] or mathematic[8-9]. The algorithm complexity of heuristic methods is usually exponential[10-11] and the mathematics methods are usually hard and complicated[8-9]. In this paper, we present a new algebraic and recursive method for constructing covering arrays for strength of an arbitrary value t based on difference matrices. Our method is simple in mathematics construction and the complexity of the algorithm is polynomial.

1 Definitions and Recursive Construction Method

Considering an additive group Z_s of orders with elements denoted by $0, 1, \ldots, s-1$, we first give some definitions on difference matrices[12].

Definition 1 An $r\times c$ matrix with entries from Z_s is called a difference matrix if all the elements of Z_s appear equally among the entry-wise differences, modulus s, of any two columns of the matrix. We denote such a matrix by $D(r, c, s)$.

Lemma 1[12] If $\boldsymbol{D}=(d_{ij})_{r\times c}$ is a difference matrix $D(r, c, s)$, then

$$\boldsymbol{A}=\begin{bmatrix}\boldsymbol{D}_0\\ \boldsymbol{D}_1\\ \vdots\\ \boldsymbol{D}_{s-1}\end{bmatrix}$$

is an orthogonal array OA(rs; 2, c, s), where $\boldsymbol{D}_i=(d_{ij}+i)_{r\times c}(i\in\{0, 1, \ldots, s-1\})$.

Definition 2 Let $\boldsymbol{A}=(a_{ij})_{m\times n}$ and $\boldsymbol{B}=(b_{ij})_{u\times v}$ be respectively $m\times n$ and $u\times v$ matrices with entries from an Abelian group Z_s with binary operation $*$ ($*$ usually denotes addition or multiplication). Their Kronecker product, denoted by $\boldsymbol{A}\otimes\boldsymbol{B}$, is an $mu\times nv$ matrix,

$$\boldsymbol{A}\otimes\boldsymbol{B}=\begin{bmatrix}a_{11}*\boldsymbol{B} & \ldots & a_{1n}*\boldsymbol{B}\\ \vdots & & \vdots\\ a_{m1}*\boldsymbol{B} & \ldots & a_{mn}*\boldsymbol{B}\end{bmatrix}$$

where $a_{ij}*\boldsymbol{B}$ stands for the $u\times v$ matrix with entries $a_{ij}*b_{rs}$ ($1\leqslant r\leqslant u, 1\leqslant s\leqslant v$). In this paper, $*$ always denotes addition.

Using Definition 2 we may rewrite the array $\boldsymbol{A}$ in Lemma 1 as

$$\boldsymbol{A}=\boldsymbol{E}\otimes\boldsymbol{D}=\begin{bmatrix}\delta_0*\boldsymbol{D}\\ \delta_1*\boldsymbol{D}\\ \vdots\\ \delta_{s-1}*\boldsymbol{D}\end{bmatrix}$$

where $\boldsymbol{E}=\{\delta_0, \delta_1, \ldots, \delta_{s-1}\}^{\mathrm{T}}(\delta_i\in Z_s=\{0, 1, \ldots, s-1\}, i=0, 1, \ldots, s-1)$.

Theorem 1 If $\boldsymbol{D}=(d_{ij})_{r\times c}$ is a difference matrix $D(r, c, s)$ and $\boldsymbol{B}=(b_{ij})_{N\times k}$ is a covering array CA(N; 2, k, s), where $d_{ij}, b_{ij}\in Z_s=\{0, 1, \ldots, s-1\}$ (Z_s is an additive group), then the array $\boldsymbol{C}=\boldsymbol{B}\otimes\boldsymbol{D}$ is a covering array

Received 2010-10-24.

Biographies: Zha Rijun (1971—), male, graduate; Xu Baowen (corresponding author), male, doctor, professor, bwxu@nju.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 90818027, 61003020, 91018005, 60873050), the National High Technology Research and Development Program of China (863 Program) (No. 2009AA01Z147), the National Basic Research Program of China (973 Program) (No. 2009CB320703).

Citation: Zha Rijun, Zhang Deping, Xu Baowen. Recursive constructions for t-covering arrays [J]. Journal of Southeast University (English Edition), 2011, 27(3): 340 - 342. [doi: 10.3969/j.issn.1003 - 7985.2011.03.022]

$CA(Nr; 2, kc, s)$.

Proof Let $C = B \otimes D = \begin{bmatrix} b_{11} * D & \dots & b_{1k} * D \\ \vdots & & \vdots \\ b_{N1} * D & \dots & b_{Nk} * D \end{bmatrix}$, where $B = (b_{ij})_{N \times k}$ is a covering array $CA(N; 2, k, s)$ and $D = (d_{ij})_{r \times c}$ is a difference matrix $D(r, c, s)$. In the following paragraphs, we will prove that the array C is a covering array $CA(Nr; 2, kc, s)$.

According to the definition of the Kronecker product, we know that the array C is an $Nr \times kc$ matrix and it is partitioned into Nk blocks subarrays. If $B = (b_{ij})_{N \times k}(b_{ij} \in Z_s = \{0, 1, \dots, s-1\})$ is a covering array $CA(N; 2, k, s)$, then each column of the array B includes each element of Z_s at least once. In other words, since each column of the array B includes $E = \{\delta_0, \delta_1, \dots, \delta_{s-1}\}^T (\delta_i \in Z_s = \{0, 1, \dots, s-1\}, i = 0, 1, \dots, s-1)$, then each column block subarray of the array C includes an $E \otimes D$ subarray. We know from Lemma 1 that $E \otimes D$ is an orthogonal array. Thus, each column block subarray of array C includes an orthogonal subarray with entries from the additive group Z_s.

We select any two columns l and m of the array C and consider the following two cases.

Case 1 Suppose that columns l and m occur in the same column block subarray of the array C. Since each column block subarray of the array C includes an orthogonal subarray with entries from the additive group Z_s, then two columns l and m will include any pair $(x, y), x, y \in Z_s = \{0, 1, \dots, s-1\}$ at least once.

Case 2 Suppose that columns l and m occur in different i and j column block subarray of the array C, respectively. We consider two different conditions for this case:

The first is the condition where the columns l and m come from the same column of the difference matrix D. Since the array B is a covering array with entries from the additive group Z_s, then any pair $(x, y), x, y \in Z_s = \{0, 1, \dots, s-1\}$ occurs in the i and j columns of the array B. The columns of l and m of the array C include any pair $(x + d_{ij}, y + d_{ij}) = (x, y) \bmod s$, $(x, y) \in Z_s = \{0, 1, \dots, s-1\}$, $d_{ij} \in Z_s = \{0, 1, \dots, s-1\}$. In other words, the columns l and m of the array C include any pair $(x, y), x, y \in Z_s = \{0, 1, \dots, s-1\}$.

The second is the condition where columns l and m come from different columns of the difference matrix D. As the array B is a covering array with entries from the additive group Z_s, the pair $(x, y), x, y \in Z_s = \{0, 1, \dots, s-1\}$ occurs in the i and j columns of the array B. Then the i and j columns of the array B include the same $E = \{\delta_0, \delta_1, \dots, \delta_{s-1}\}^T$ $(\delta_i \in Z_s = \{0, 1, \dots, s-1\}, i = 0, 1, \dots, s-1)$. In this case, columns l and m come from different columns of the difference matrix D, which is similar to Case 1. Then, columns l and m of the array C include any pair $(x, y), x, y \in Z_s = \{0, 1, \dots, s-1\}$.

Corollary 1 If $D = (d_{ij})_{r \times c}$ is a difference matrix $D(r, c, s)$ and $B = (b_{ij})_{N \times k}$ is a covering array $CA(N; 2, k, s)$, where $d_{ij}, b_{ij} \in Z_s = \{0, 1, \dots, s-1\}$ (Z_s is an additive group), then there is a covering array $CA(Nr^j; 2, kc^j, s)$ $(j \geqslant 2)$.

Proof According to Theorem 1, we know that $B \otimes D$ is a covering array $CA(Nr; 2, kc, s)$; then $(B \otimes D) \otimes D$ is still a covering array $CA(Nr^2; 2, kc^2, s)$. We do it for j times iteratively. $(B \otimes D) \underbrace{\otimes \dots \otimes}_{j-1} D$ is a covering array $CA(Nr^j; 2, kc^j, s)$ $(j \geqslant 2)$.

According to Theorem 1 and Corollary 1, we know that we can use the algebraic operation "$\otimes$" to generate arbitrary parameter factors of 2-covering arrays $CA(Nr^j; 2, kc^j, s)$ $(j \geqslant 2)$ from the existing few parameter factors of 2-covering arrays $CA(N; 2, k, s)$ and difference matrix $D(r, c, s)$. Therefore, we can obtain the following recursive algorithm for extension parameter factors of covering arrays of strength 2.

Algorithm 1

Input: Difference matrix $D = D(r, c, s)$ and covering array $B_1 = CA(N; 2, k, s)$;

Output: Covering array $B_{j+1} = CA(Nr^j; 2, kc^j, s)$ $(j \geqslant 1)$.

For $l = 1, 2, \dots, j$,

$B_{l+1} = B_l \otimes D$

End

The implementation complexity of Algorithm 1 is $O(jrcNk)$. In the following of this paper, we generalize the recursive method of covering arrays of strength 2 to covering arrays of strength $t (t \geqslant 2)$.

Let Z be an Abelian group of order s. By Z^t, for $t \geqslant 1$, we will denote the Abelian group of order s^t consisting of all the t-tuples of elements from Z with the usual vector addition as the binary operation. Let $Z_0^t = \{(x_1, \dots, x_t): x_1 = \dots = x_t \in Z\}$. Then Z_0^t is a subgroup of Z^t of order s, and we denote its cosets by $Z_i^t (i = 1, \dots, s^{t-1} - 1)$ [12].

Definition 3[12] An $r \times c$ matrix D based on Z_s is a difference matrix of strength t if for every $r \times t$ subarray, each set $Z_i^t (i = 0, 1, \dots, s^{t-1} - 1)$ is represented equally when the rows subarray are viewed as elements of Z^t.

We denote such a matrix D by $D_t(r, c, s)$. For $t = 2$, Definition 3 is equivalent to Definition 1.

The following results are generalized from Theorem 1 and Corollary 1 of $t \geqslant 2$. Their proof methods are similar.

Theorem 2 If $D = (d_{ij})_{r \times c}$ is a difference matrix $D_t(r, c, s)$ of strength t, and $B = (b_{ij})_{N \times k}$ is a covering array $CA(N; t, k, s)$ of strength t, then $C = B \otimes D$ is a covering array $CA(Nr; t, kc, s)$ of strength t.

Corollary 2 If $B = (d_{ij})_{r \times c}$ is a difference matrix $D_t(r, c, s)$ of strength t, and $B = (b_{ij})_{N \times k}$ is a covering array $CA(N; t, k, s)$ of strength t, then there is a covering array $CA(Nr^j; t, kc^j, s)$ $(j \geqslant 2)$ of strength t.

We also give the recursive algorithm for the extension of parameter factors of covering arrays of strength $t (t \geqslant 2)$.

Algorithm 2

Input: Difference matrix $D = D_t(r, c, s)$ and covering array $B_1 = CA(N; t, k, s)$;

Output: Covering array $B_{j+1} = CA(Nr^j; t, kc^j, s)$ $(j \geqslant 1)$.

For $l = 1, 2, \dots, j$,

$B_{1+1} = B_1 \otimes D$

End

The implementation complexity of Algorithm 2 is also $O(jrcNk)$.

2 Examples

In this section, we give two examples to demonstrate our recursive construction method.

Example 1 There is a difference matrix $\boldsymbol{D}=D(7,7,7)$,

$$\boldsymbol{D}=\begin{bmatrix}0&1&4&2&2&4&1\\0&2&6&5&6&2&0\\0&3&1&1&3&0&6\\0&4&3&4&0&5&5\\0&5&5&0&4&3&4\\0&6&0&3&1&1&3\\0&0&2&6&5&6&2\end{bmatrix}$$

and a covering array CA(49; 2, 8, 7)[12]. According to Theorem 1 and Corollary 1, there is a covering array CA(343; 2, 56, 7) = CA(49; 2, 8, 7) $\otimes D(7,7,7)$ and a covering array CA(2 401; 2, 392, 7) = CA(343; 2, 56, 7) $\otimes D(7,7,7)$, etc.

Example 2 There is a difference matrix $\boldsymbol{D}=D_3(9,3,3)$,

$$\boldsymbol{D}=\begin{bmatrix}0&0&0\\0&1&1\\0&2&2\\1&0&1\\1&1&2\\1&2&0\\2&0&2\\2&1&0\\2&2&1\end{bmatrix}$$

and a covering array CA(33; 3, 6, 3)[6]. According to Theorem 2 and Corollary 2, there is a covering array CA(297; 3, 18, 3) = CA(33; 3, 6, 3) $\otimes D_3(9,3,3)$ and a covering array CA(2 673; 3, 54, 3) = CA(297; 3, 18, 3) $\otimes D_3(9,3,3)$, etc.

3 Conclusion

This paper proposes a new algebraic recursive method for constructing t-covering arrays ($t\geqslant 2$) based on the t-difference matrix ($t\geqslant 2$). It can extend parameter factors on the existing t-covering arrays and cover all the combinations of all parameter factors. The method is simple and easy to realize. The complexity of the algorithm is polynomial. What remains a problem is to obtain the t-difference matrix. We can further work on the investigation of the t-difference matrix in the future.

References

[1] Hartman A. Software and hardware testing using combinatorial covering suites[J]. *Graph Theory, Combinatorics and Algorithms*, 2005, **34**(1): 237 − 266.

[2] Xu Baowen, Nie Changhai, Shi Liang, et al. A software failure debugging method based on combinatorial design approach for testing[J]. *Chinese Journal of Computers*, 2006, **29**(1): 132 − 138.

[3] Sloane N J A. Covering arrays and intersecting codes[J]. *Journal of Combinatorial Designs*, 1993, **1**(1): 51 − 63.

[4] Colbour C J. Strength two covering arrays: existence tables and projection[J]. *Discrete Mathematics*, 2008, **308**(6): 772 − 786.

[5] Chateaunenf M, Colbour C J, Kreher D L. Covering arrays of strength three[J]. *Designs, Codes and Cryptography*, 1999, **16**(3): 235 − 242.

[6] Chateaunenf M, Kreher D L. On the state of strength three covering arrays[J]. *Journal of Combinatorial Designs*, 2002, **10**(4): 217 − 238.

[7] Walker Ⅱ R A, Colbourn C J. Tabu search for covering arrays using permutation vectors[J]. *Journal of Statistical Planning and Inference*, 2009, **139**(1): 69 − 80.

[8] Godbole A P, Skipper D E, Sunley R A. *T*-covering arrays: upper bounds and Poission approximations[J]. *Combinatorics, Probability and Computing*, 1996, **5**(1): 105 − 117.

[9] Martirosyan S, van Trung T. On *t*-covering arrays[J]. *Designs, Codes and Cryptography*, 2004, **32**(1): 323 − 339.

[10] Bryce R C, Colbourn C J. A density-based greedy algorithm for higher strength covering arrays[J]. *Software Testing, Verification and Reliability*, 2009, **19**(1): 37 − 53.

[11] Lei Y, Kacker R, Kuhn R D, et al. IPOG: a general strategy for *t*-way software testing[C]//*Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*. Tucson, Arizona, USA, 2007: 549 − 556.

[12] Hedayat A S, Sloane N J A, Stufken J. *Orthogonal arrays: theory and applications*[M]. New York: Springer, 1999.

t-维组合覆盖阵的递归构建

查日军[1]　张德平[2]　徐宝文[3]

([1] 东南大学计算机科学与工程学院，南京 210096)

([2] 南京航空航天大学计算机科学与技术学院，南京 210016)

([3] 南京大学计算机软件新技术国家重点实验室，南京 210093)

摘要：一个大小为 N、度为 k、取值为 v、强度为 t 的 t-维组合覆盖阵是一个 $N\times k$ 矩阵，其取值为 v 的符号集，并且任意 $N\times t$ 子矩阵的行都至少包含取值为 v 符号集的任一 t 元组一次. 提出了一种基于差阵的 t-维组合覆盖阵代数递归构建新方法，该方法在已有的覆盖阵基础上可大规模地扩展参数个数，实现任意 t 个参数组合的有效覆盖($t\geqslant 2$)，是一种实用的生成高维组合覆盖阵代数递归新方法. 同时，给出了该新方法的理论推导与算法实现.

关键词：覆盖阵；正交阵；差阵

中图分类号：TP306