Mappable Nearly Orthogonal Arrays Using Projective Geometry

POONAM SINGH[a], MUKTA D. MAZUMDER[b] and **SANTOSH BABU[c]**

[a]*Department of Statistics, University of Delhi, Delhi 110007, India.*

(pbs_93@yahoo.co.in)

[b]*Department of Statistics, Ram Lal Anand College, University of Delhi, New Delhi 110021, India.*

(munroy2003@yahoo.com)

[c]*Department of Statistics, University of Delhi, Delhi 110007, India.*

(santosh.yadav0506@gmail.com)

Abstract: The finite projective geometry is used to construct many series of symmetric orthogonal arrays of strength two and more. In this paper, we propose a method of construction of Nearly orthogonal arrays mappable into fully symmetric orthogonal arrays of strength two finite projective geometry.

Key Words: Orthogonal arrays, symmetric orthogonal array, Mappable nearly orthogonal arrays, Finite Projective Geometry.

1. Introduction The concept of orthogonal arrays introduced by **Rao (1946)** is very useful in the field of design of experiments**. Rao (1947)** obtained upper bound on the maximal number of factors for a symmetric orthogonal array. **Bose and Bush (1952)** constructed orthogonal arrays of strength two and three using difference schemes, Galois field and finite projective geometry. Different methods for the construction of orthogonal arrays with different levels and strengths using orthogonal Latin squares, Hadamard matrices, group theory, finite fields, coding theory and finite projective geometry are discussed in **Hedayat et al. (1999).**

Bose and Bush (1952) constructed orthogonal arrays of strength two and three using difference schemes, Galois field and finite projective geometry. Different methods for the construction of orthogonal arrays with different levels and strengths using orthogonal Latin squares, Hadamard matrices, group theory, finite fields, coding theory and finite projective geometry are discussed in **Hedayat et al. (1999).**

Mukerjee et al. (2014) constructed some series of nearly mappable orthogonal arrays of strength two using resolvable orthogonal arrays. In these types of arrays, each column of a group is orthogonal to a large proportion of the other columns, and these arrays are also easily convertible to fully orthogonal arrays using a mapping of the large symbols in each column to a possibly smaller or same set of symbols. The importance and applications of this type of arrays have been of considerable interest because of their inherent better space filling properties.

Mukerjee et al. (2014) illustrated through an example how an MNOA with 81 runs, 40 factors each with 9 symbols can achieve a stratification on a 9 x 9 grid in 720 out of 780 two-dimensions and on a 3 x 3 grid in the remaining 60 two-dimensions. Thus, having a better space filling properties than an OA with 81 runs, 40 factors each at 3 levels and accommodating more factors than an OA with 81 runs, 10 factors with 9 symbols. An important property of MNOA's is that an another MNOA can be obtained with the same number of runs but less columns after deleting one or more columns from a MNOA **(Mukerjee et al. 2014).** The main intent is to increase the number of groups for attaining better amount of orthogonality degree instead of obtaining other orthogonal array after deleting columns. **Liu et al. (2023)** constructed mappable nearly orthogonal arrays with column - orthogonality and enhance the MNOA's projection uniformity on any one dimensional by using the construction (nearly) column orthogonal MNOAs and rotation matrices. **Singh et al. (2023)** constructed many new series of mappable nearly orthogonal arrays using difference matrix.

In this paper, we propose a method to construct mappable nearly orthogonal arrays using finite projective geometry. The constructed nearly orthogonal arrays are mappable to fully symmetric orthogonal arrays of strength two.

In section 2, we give notations and definitions of orthogonal array, mappable nearly orthogonal array (MNOA) and finite projective geometry. In section 3, we present a new method and some newly constructed mappable nearly orthogonal array of strength two using the concept of finite projective geometry. We also provide MNOA tables that are obtained using our propose method

2. PRELIMINARIES

Following results and definitions are useful and important for the present study.

2.1. Orthogonal array: An $N \times k$ matrix A, with entries from a set G of s (≥ 2) elements, is called a symmetric orthogonal array of strength t, size N, k constraints and s levels if every $(N \times t)$ submatrix of A contains all possible $(1 \times t)$ row vectors with the same frequency λ. The orthogonal array is denoted by $OA[N, k, s, t]$. The number λ is called the index of the array and it satisfies $N = \lambda s^t$.

Theorem 1: (Rao, 1947) In an $OA(N, k, s, t)$ the inequalities

$$N - 1 \geq \binom{k}{1}(s-1) + \ldots\ldots.. + \binom{k}{u}(s-1)^u \qquad if \;\; t = 2u$$

and

$$N - 1 \geq \binom{k}{1}(s-1) + \ldots\ldots.. + \binom{k}{u}(s-1)^u + \binom{k-1}{u}(s-1)^{u+1} \qquad if \;\; t = 2u + 1$$

must hold.

An orthogonal array is said to be a tight orthogonal array, if equality holds inequalities.

2.2. Mappable nearly orthogonal array

A mappable nearly orthogonal array $MNOA\left[N, \prod_{i=1}^{m} s_i{}^{c_i}, \prod_{i=1}^{m}\prod_{j=1}^{c_i} r_{ij}\right]$ is an $N \times \tilde{c}$ array whose $\tilde{c} = c_1 + \cdots + c_m$ columns can be partitioned into m disjoint groups of $c_1, \ldots.., c_m$ columns with the following properties:

(i) for $i = 1, \ldots\ldots m$, every column of the ith group is populated by s_i symbols;

(ii) any two columns from different groups are orthogonal;

(iii) for $i = 1, \ldots\ldots m$ and $j = 1, \ldots\ldots. c_i$, the s_i symbols in the jth column of the ith group can be mapped to a set of $r_{ij} \leq s_i$ symbols such that these mapping convert the array into an orthogonal array $OA[N, \prod_{i=1}^{m}\prod_{j=1}^{c_i} r_{ij}]$. In particular, if $s_i = s$, $c_i = c$ and $r_{ij} = r$ for every i and j, then a mappable nearly orthogonal array is denoted as $A = MNOA[N, (s^c)^m, \; (r^c)^m]$.

In a mappable nearly orthogonal array, before mapping each column $c_i$in the ith group is orthogonal to at least a proportion $\pi_i = \left(\frac{\tilde{c}-c_i}{\tilde{c}-1}\right), i = 1,2, \ldots.., m$ of the other columns. Here we construct only symmetric mappable nearly orthogonal arrays, so every column c_i contains the same symbols. Also, if $c_1 = c_2 = \cdots = c_m = c$, then $\tilde{c} = mc$ and we have

$$\pi = \pi_{min} = (m-1)c/(mc-1) \qquad (1)$$

where m and c are number of groups and number of columns respectively.

2.3. Finite Projective Geometry: A finite projective geometry $PG(r,m)$ over Galois field of order m or $GF(m)$, where m is a prime or a power of a prime number, consists of ordered sets $(y_0, y_1, \dots\dots\dots, y_r)$ called points where y_i, $i = 0,1, \dots\dots, r$, are elements of $GF(m)$ and all of them are not simultaneously zero. The point $(ay_0, ay_1, \dots\dots\dots, ay_r)$ represents the same point as$(y_0, y_1, \dots\dots\dots, y_r)$, for any $a \in GF(m)$, $(a \neq 0)$. The collection of all those points which satisfy a set of $(r-t)$ linearly independent homogeneous equations with coefficients from $GF(m)$ (all of them are not simultaneously zero within the same equation) is said to represent a t-flat in $PG(r,m)$.

In particular, a 0-flat, 1-flat,........., a $(r-1)$-flat in $PG(r,m)$ is known as a point, a line,......, a hyperplane respectively. The number of points lying on a t-flat in $PG(r,m)$ is $\frac{(m^{t+1}-1)}{(m-1)}$ and the number of independent points lying on a t-flat is $(t+1)$.

3. Method of construction

Mukerjee et al. (2014) constructed mappable nearly orthogonal arrays of strength two using resolvable orthogonal arrays. Here, we use projective geometry to construct new series of mappable nearly orthogonal arrays of strength two. The total number of distinct points in $PG(r,m)$ is $|\mathrm{PG(r,m)}| = \left[\frac{m^{r+1}-1}{m-1}\right]$. $PG(r,m)$ has disjoint t- flats if and only if $(t+1)|(r+1)$ and there are $\left[\frac{m^{r+1}-1}{m^{t+1}-1}\right]$ number of disjoint t- flats. disjoint t-flats. The $PG(r,m)$can be used to construct symmetric orthogonal array $OA[m^{r+1}, p,\ m^{t+1}, 2]$, where $p = \left[\frac{m^{r+1}-1}{m^{t+1}-1}\right]$.

We now describe a new method for constructing mappable nearly orthogonal array using projective geometry.

Step I: Using all points of $PG(t+1,\ m)$ obtain the orthogonal array $A = OA[m^{t+2}, q, m, 2]$, where $q = \left[\frac{m^{t+2}-1}{m-1}\right]$.

Step II: Write the array A as $A = [C_1' \quad C_2' \dots\dots \quad C_m']'$, where $C_j, j = 1, 2, \dots\dots, m$ is of order $(m^{t+1} \times q)$. Delete the column, having single symbol from C_j to obtain C_j^*, for $j = 1, 2, \dots\dots, m$. So that C_j^* has $(q-1)$ columns.

Step III: For $j = 1, 2, \dots m$ and $k = 1,2, \dots., (q-1)$, replace the m^{t+1} occurrences of each of the m symbols in the kth column of C_j^* by m^{t+1} symbols from the set $S = (0, 1, 2, \dots\dots.., (m^{t+1}-1))$ as follows to obtain D_j

For $k = 1,2, \dots\dots., (q-1)$, define

$$t_{kh} = \{hm, hm+1, \dots\dots., hm + (m^t - 1)\}, \quad h = 0,1,2, \dots\dots.., (m-1) \qquad (2)$$

and replace the m^{t+1} occurrences of symbol h by the m^t members of t_{kh} in order as obtained in (2), that is, the first occurrence of h is replaced by hm and second occurrence by $hm+1$ and so on.

Step IV: Write $D = [D_1',\ D_2', \dots\dots.., D_m']'$, where D_j is the the $(m^{t+1} \times (q-1))$ array obtained from C_j^* after changing symbols of C_j^* according to (2), such that each column of D_j is a permutation of $\{0, 1, \dots\dots\dots, (m^{t+1} - 1)\}$ symbols. Let R_i^j denote the ith row of D_j, for $i = 1\ , 2, \dots\dots. (m^{t+1} - 1)$.

Step V: Consider orthogonal arrays $B^j = OA[m^{r+1}, p,\ m^{t+1}, 2], j = 1, 2, \dots\dots, m$ obtained from $p = \left[\frac{m^{r+1}-1}{m^{t+1}-1}\right]$ disjoint t-flats of $PG(r,m)$. Let $0, 1, \dots\dots\dots, (m^{t+1} - 1)$ be the symbols in the ith column of orthogonal arrays $B^j = [b_{li}]$; $l = 1, 2, \dots\dots\dots m^{r+1}$ and $i = 1, 2,\ \dots\dots p$.

Step VI: Construct the pre - mapping array using array A as described below

Write the array B^j as $B^j = [b_1 \colon b_2 \colon \dots\dots b_p]$, where b_e, $e = 1,2, \dots\dots, p$ are the column vectors of array B^j of order $(m^{r+1} \times 1)$ with m^{t+1} symbols . Replace the symbols $\{0, 1, 2, \dots\dots.., (m^{t+1} - 1)\}$ of the column vector b_e of

B^j respectively by $R_0^j,\ R_1^j, \dots\dots\dots\dots\dots.. R^j_{(m^{t+1}-1)}$ rows of D_j respectively to obtain t_e, where t_e is of order $m^{r+1} \times (q-1)$, having rows $R_0^j,\ R_1^j, \dots\dots\dots\dots\dots.. R^j_{(m^{t+1}-1)}$.

(i). For $j = 1, 2, \dots., m$ write $T_j = [t_1 : t_2 : \dots\dots : t_p]$; where T_j is of order $(m^{r+1} \times p(q-1))$ with each column having symbols $0,1,2, \dots\dots.., (m^{t+1}-1)$.

(ii). Juxtaposition T_j, $j = 1, 2, \dots..., m$ to obtain the pre mapping array $T = [T_1{}' \quad \dots\dots \quad T_m{}']'$, which is of order $m^{r+2} \times p(q-1)$, having p groups of $(q-1)$ columns each.

Step VII: Now for post mapping array, m^{t+1} symbols $t_{kh} = \{hm, hm+1, \dots\dots., hm+(m^t-1)\}$, are in each of the $(q-1)$ columns of t_e mapped to h for $h = 0,1,2, \dots\dots.., (m-1)$ to get the array of order $m^{r+1} \times p(q-1)$ with symbols $\{0, 1, \dots, (m-1)\}$. Hence, we get the following mappable nearly orthogonal array

$$MNOA[m^{r+2}, \{(m^{t+1})^{q-1}\}^p, \{(m)^{q-1}\}^p\ ].$$

Thus ,we have the following result.

Theorem 2: The existence of orthogonal arrays $A = OA[m^{t+2}, q, m, 2]$ and $B = OA[m^{r+1}, p,\ m^{t+1}, 2]$ with $p = \left[\frac{m^{r+1}-1}{m^{t+1}-1}\right]$, $q = \left[\frac{m^{t+2}-1}{m-1}\right]$ and $(t+1)$ divides $(r+1)$. Implies the existence of a $MNOA[m^{r+2}, \{(m^{t+1})^{q-1}\}^p, \{(m)^{q-1}\}^p\ ]$.

We illustrate this result through the examples

Example 2.1: Let $t = 1, r = 3$ and $m = 2$ in $PG(r, m)$. Using step I, we obtain the array $A = OA[8, 7, 2, 2]$ of order (8×7) using the $PG(t+1, m)$ or $PG(2, 2)$ as

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

For $j = 1, 2$ the array A can be written as $A = [C_1' \quad C_2']'$ through the step II,

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad \text{and} \quad C_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now we can see that $C_1$and C_2 have third column , which has same symbols 0 and 1. So this column will be deleted from C_1 and C_2 and remaining columns consisting only two symbols with equal occurrences.

$$C_1^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \text{and } C_2^* = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, replace the occurrences of two symbols 0 and 1 in the every column of C_1^* and C_2^* by set of four symbols $S = (0, 1, 2, 3)$ according to (2), to get D_1 , D_2 and D as displayed below:

$$D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & 2 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 \\ 3 & 3 & 1 & 3 & 3 & 1 \end{bmatrix} \text{ and } D_2 = \begin{bmatrix} 0 & 0 & 0 & 2 & 2 & 2 \\ 2 & 1 & 2 & 0 & 3 & 0 \\ 1 & 2 & 3 & 3 & 0 & 1 \\ 3 & 3 & 1 & 1 & 1 & 3 \end{bmatrix}$$

and the array

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & 2 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 \\ 3 & 3 & 1 & 3 & 3 & 1 \\ 0 & 0 & 0 & 2 & 2 & 2 \\ 2 & 1 & 2 & 0 & 3 & 0 \\ 1 & 2 & 3 & 3 & 0 & 1 \\ 3 & 3 & 1 & 1 & 1 & 3 \end{bmatrix}$$

Here $R_0^1, R_1^1, R_2^1, R_3^1$ and $R_0^2, R_1^2, R_2^2, R_3^2$ are the rows of D_1 and D_2 respectively.

According to step V, consider an orthogonal array $B = OA(16,\ 5, 4, 2)$, obtaining using disjoint $1-$flats in $PG(3, 2)$ and is displayed below:

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 2 & 1 \\ 2 & 1 & 2 & 0 & 3 \\ 2 & 2 & 1 & 2 & 0 \\ 0 & 2 & 2 & 1 & 2 \\ 3 & 1 & 0 & 2 & 2 \\ 3 & 2 & 3 & 0 & 1 \\ 1 & 2 & 0 & 3 & 3 \\ 0 & 3 & 3 & 2 & 3 \\ 2 & 3 & 0 & 1 & 1 \\ 2 & 0 & 3 & 3 & 2 \\ 1 & 3 & 1 & 0 & 2 \\ 3 & 3 & 2 & 3 & 0 \\ 0 & 1 & 1 & 3 & 1 \\ 3 & 0 & 1 & 1 & 3 \\ 1 & 1 & 3 & 1 & 0 \end{bmatrix}$$

We can write the array $B^j = [b_{li}]$, here $l = 1, 2, \dots \dots \dots \dots 16$, $j = 1, 2$ and $i = 1, 2, \dots \dots . . ,5$.

Using step VI, construct the pre - mapping array as follows:

First denote the arrays B_1 and B_2 as $B_1 = [b_1 \colon b_2 \colon b_3 \colon b_4 \colon b_5]$ and $B_2 = [b_1 \colon b_2 \colon b_3 \colon b_4 \colon b_5]$, where B_1 and B_2 are same as array B and $b_e, e = 1,2, \dots . ,5$ are the columns or group of array B. Now construct the arrays $T_j, j = 1, 2$ each of order 16×6, after each symbols b_e is replaced by the row R_i^J for every $j = 1, 2$ by following step VI and hence we have pre-mapping array $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$, where $T_1 = [t_1 \colon t_2 \colon t_3 \colon t_4 \colon t_5]$ and $T_2 = [t_1 \colon t_2 \colon t_3 \colon t_4 \colon t_5]$. For post mapping array, map the symbols (0,1,2,3) to (0,1) using (2) we obtain the resultant $MNOA[32, (4^6)^5, (2^6)^5]$

Pre-mapping

000000	000000	000000	000000	000000
212212	000000	123123	123123	212212
123123	212212	123123	000000	331331
123123	123123	212212	123123	000000
000000	123123	123123	212212	123123
331331	212212	000000	123123	123123
331331	123123	331331	000000	212212
212212	123123	000000	331331	331331
000000	331331	331331	123123	331331
123123	331331	000000	212212	212212
123123	000000	331331	331331	123123
212212	331331	212212	000000	123123
331331	331331	123123	331331	000000
000000	212212	212212	331331	212212
331331	000000	212212	212212	331331
212212	212212	331331	212212	000000
000222	000222	000222	000222	000222
212030	000222	123301	123301	212030
123301	212030	123301	000222	331113
123301	123301	212030	123301	000222
000222	123301	123301	212030	123301
331113	212030	000222	123301	123301
331113	123301	331113	000222	212030
212030	123301	000222	331113	331113
000222	331113	331113	123301	331113
123301	331113	000222	212030	212030
123301	000222	331113	331113	123301
212030	331113	212030	000222	123301
331113	331113	123301	331113	000222
000222	212030	212030	331113	212030
331113	000222	212030	212030	331113
212030	212030	331113	212030	000222

Post-mapping

000000	000000	000000	000000	000000
101101	000000	011011	011011	101101
011011	101101	011011	000000	110110
011011	011011	101101	011011	000000
000000	011011	011011	101101	011011
110110	101101	000000	011011	011011
110110	011011	110110	000000	101101
101101	011011	000000	110110	110110
000000	110110	110110	011011	110110
011011	110110	000000	101101	101101
011011	000000	110110	110110	011011
101101	110110	101101	000000	011011
110110	110110	011011	110110	000000
000000	101101	101101	110110	101101
110110	000000	101101	101101	110110
101101	101101	110110	101101	000000
000111	000111	000111	000111	000111
101010	000111	011100	011100	101010
011100	101010	011100	000111	110001
011100	011100	101010	011100	000111
000111	011100	011100	101010	011100
110001	101010	000111	011100	011100
110001	011100	110001	000111	101010
101010	011100	000111	110001	110001
000111	110001	110001	011100	110001
011100	110001	000111	101010	101010
011100	000111	110001	110001	011100
101010	110001	101010	000111	011100
110001	110001	011100	110001	000111
000111	101010	101010	110001	101010
110001	000111	101010	101010	110001
101010	101010	110001	101010	000111

This is required mappable nearly tight orthogonal array $MNOA[32,(4^6)^5,(2^6)^5]$.

Example 2.2: Let $t=2, r=5$ and $m=2$ in $PG(r,m)$. Using step I, we obtain the array $A=OA[16,15,2,2]$ of order (16×15) using the $PG(t+1,m)$ or $PG(3,2)$ as

$$A=\begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&1&0&1&0&1&0&1&0&1&0&1&0&1\\
0&1&1&0&0&1&1&0&0&1&1&0&0&1&1\\
1&1&0&0&1&1&0&0&1&1&0&0&1&1&0\\
0&0&0&1&1&1&1&0&0&0&0&1&1&1&1\\
1&0&1&1&0&1&0&0&1&0&1&1&0&1&0\\
0&1&1&1&1&0&0&0&0&1&1&1&1&0&0\\
1&1&0&1&0&0&1&0&1&1&0&1&0&0&1\\
0&0&0&0&0&0&0&1&1&1&1&1&1&1&1\\
1&0&1&0&1&0&1&1&0&1&0&1&0&1&0\\
0&1&1&0&0&1&1&1&1&0&0&1&1&0&0\\
1&1&0&0&1&1&0&1&0&0&1&1&0&0&1\\
0&0&0&1&1&1&1&1&1&1&1&0&0&0&0\\
1&0&1&1&0&1&0&1&0&1&0&0&1&0&1\\
0&1&1&1&1&0&0&1&1&0&0&0&0&1&1\\
1&1&0&1&0&0&1&1&0&0&1&0&1&1&0
\end{bmatrix}$$

For $j=1,2$ the array A can be written as $A=[C_1' \quad C_2']'$ through the step II,

$$C_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

and $$C_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Now we can see that $C_1$and C_2 have seven column , which has same symbols 0 and 1. So this column will be deleted from C_1 and C_2 and remaining columns consisting only two symbols with equal occurrences.

$$C_1^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

and

$$C_2^* = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Now, replace the occurrences of two symbols 0 and 1 in the every column of C_1^* and C_2^* by set of four symbols $S = (0, 1, 2, 3,4,5,6,7)$ according to (2), to obtain D_1, D_2 and D as displayed below:

$$D_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 4 & 1 & 4 & 1 & 4 & 4 & 1 & 4 & 1 & 4 & 1 & 4 \\ 1 & 4 & 5 & 2 & 1 & 4 & 5 & 1 & 4 & 5 & 2 & 1 & 4 & 5 \\ 5 & 5 & 1 & 3 & 5 & 5 & 1 & 5 & 5 & 1 & 3 & 5 & 5 & 1 \\ 2 & 2 & 2 & 4 & 6 & 6 & 6 & 2 & 2 & 2 & 4 & 6 & 6 & 6 \\ 6 & 3 & 6 & 5 & 2 & 7 & 2 & 6 & 3 & 6 & 5 & 2 & 7 & 2 \\ 3 & 6 & 7 & 6 & 7 & 2 & 3 & 3 & 6 & 7 & 6 & 7 & 2 & 3 \\ 7 & 7 & 3 & 7 & 3 & 3 & 7 & 7 & 7 & 3 & 7 & 3 & 3 & 7 \end{bmatrix}$$

and

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 1 & 4 & 1 & 4 & 1 & 4 & 0 & 5 & 0 & 5 & 0 & 5 & 0 \\ 1 & 4 & 5 & 2 & 1 & 4 & 5 & 5 & 0 & 1 & 6 & 5 & 0 & 1 \\ 5 & 5 & 1 & 3 & 5 & 5 & 1 & 1 & 1 & 5 & 7 & 1 & 1 & 5 \\ 2 & 2 & 2 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 0 & 2 & 2 & 2 \\ 6 & 3 & 6 & 5 & 2 & 7 & 2 & 2 & 7 & 2 & 1 & 6 & 3 & 6 \\ 3 & 6 & 7 & 6 & 7 & 2 & 3 & 7 & 2 & 3 & 2 & 3 & 6 & 7 \\ 7 & 7 & 3 & 7 & 3 & 3 & 7 & 3 & 3 & 7 & 3 & 7 & 7 & 3 \end{bmatrix}$$

and the array

$$D = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 4 & 1 & 4 & 1 & 4 & 4 & 1 & 4 & 1 & 4 & 1 & 4 \\ 1 & 4 & 5 & 2 & 1 & 4 & 5 & 1 & 4 & 5 & 2 & 1 & 4 & 5 \\ 5 & 5 & 1 & 3 & 5 & 5 & 1 & 5 & 5 & 1 & 3 & 5 & 5 & 1 \\ 2 & 2 & 2 & 4 & 6 & 6 & 6 & 2 & 2 & 2 & 4 & 6 & 6 & 6 \\ 6 & 3 & 6 & 5 & 2 & 7 & 2 & 6 & 3 & 6 & 5 & 2 & 7 & 2 \\ 3 & 6 & 7 & 6 & 7 & 2 & 3 & 3 & 6 & 7 & 6 & 7 & 2 & 3 \\ 7 & 7 & 3 & 7 & 3 & 3 & 7 & 7 & 7 & 3 & 7 & 3 & 3 & 7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 4 & 1 & 4 & 1 & 4 & 1 & 4 & 0 & 5 & 0 & 5 & 0 & 5 & 0 \\ 1 & 4 & 5 & 2 & 1 & 4 & 5 & 5 & 0 & 1 & 6 & 5 & 0 & 1 \\ 5 & 5 & 1 & 3 & 5 & 5 & 1 & 1 & 1 & 5 & 7 & 1 & 1 & 5 \\ 2 & 2 & 2 & 4 & 6 & 6 & 6 & 6 & 6 & 6 & 0 & 2 & 2 & 2 \\ 6 & 3 & 6 & 5 & 2 & 7 & 2 & 2 & 7 & 2 & 1 & 6 & 3 & 6 \\ 3 & 6 & 7 & 6 & 7 & 2 & 3 & 7 & 2 & 3 & 2 & 3 & 6 & 7 \\ 7 & 7 & 3 & 7 & 3 & 3 & 7 & 3 & 3 & 7 & 3 & 7 & 7 & 3 \end{bmatrix}$$

Here R_0^1, R_1^1, R_2^1,................ and R_7^1 are rows of D_1 , similarly R_0^2, R_1^2, R_2^2,.......... and R_7^2) are rows of D_1 and D_2 respectively.

According to step V, consider an orthogonal array $B = OA(64,\ 9, 8, 2)$, obtained using disjoint $2-$flats in $PG(5, 2)$

We can write the array $B^j = [b_{li}]$, here $l = 1, 2, \dots \dots \dots 64$, $j = 1, 2$ and $i = 1, 2, \dots \dots ..,9$.

Using step VI, construct the pre - mapping array as follows:

First denote the arrays $B^1 = [b_1: b_2: b_3: \dots \dots \dots b_9]$ and $B^2 = [b_1: b_2: b_3: \dots . \dots \dots . b_9]$, where B_1 and B_2 are arrays same as array B and $b_e, e = 1,2, \dots ,9$ are the columns or groups of array B. Now construct the arrays T_j, $j = 1,2$ each of order 64×14, after each symbol of column b_e is replaced by the rows R_i^J of D_j by following step VI and hence, we have pre-mapping array $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$,where $T_1 = [t_1: t_2: t_3: t_4: t_5: \dots \dots t_9]$ and $T_2 = [t_1: t_2: t_3: t_4: t_5: \dots \dots t_9]$. For post mapping array, map the symbols (0,1,2,3,4,5,6,7) to (0,1) as described in Step VII. The resultant array is MNOA$[128, \{(8)^{14}\}^9, \{(2)^{14}\}^9]$.

Table 1: Some Mappable Nearly orthogonal arrays based on 1-flat and $PG(r, m)$.

r	m	$\mathbf{A = OA[m^{t+2}, q, m, 2]}$	$MNOA[m^{r+2}, \{(m^{t+1})^{q-1}\}^p, \{(m)^{q-1}\}^p]$	π
3	2	OA[8, 7 ,2, 2]	MNOA$[32, \{(4)^6\}^5, \{(2)^6\}^5]$	0.8275
5	2	OA[8, 7, 2, 2]	MNOA$[128, \{(4)^6\}^{21}, \{(2)^6\}^{21}]^*$	0.9600
7	2	OA[8, 7,2, 2]	MNOA$[512, \{(4)^6\}^{85}, \{(2)^6\}^{85}]^*$	0.9901
3	4	OA[64, 21,4, 2]	MNOA$[1024, \{(16)^{20}\}^{17}, \{(4)^{20}\}^{17}]^*$	0.9439

5	4	$\mathrm{OA}[64, 21,4, 2]$	$\mathrm{MNOA}[16384, \{(16)^{20}\}^{273}, \{(4)^{20}\}^{273}]^*$	0.9965
7	4	$\mathrm{OA}[64, 21,4, 2]$	$\mathrm{MNOA}[262144, \{(16)^{20}\}^{4369}, \{(4)^{20}\}^{4369}]^*$	0.9997
3	3	$\mathrm{OA}[27, 13,3, 2]$	$\mathrm{MNOA}[243, \{(9)^{12}\}^{10}, \{(3)^{12}\}^{10}]$	0.9075
5	3	$\mathrm{OA}[27, 13,3, 2]$	$\mathrm{MNOA}[2187, \{(9)^{12}\}^{91}, \{(3)^{12}\}^{91}]^*$	0.9899
7	3	$\mathrm{OA}[27, 13,3, 2]$	$\mathrm{MNOA}[19683, \{(9)^{12}\}^{820}, \{(3)^{12}\}^{820}]^*$	0.9988

Table 2: Some Mappable Nearly tight orthogonal arrays based on 2-flat and $PG(r, m)$.

r	m	$\mathbf{A = OA[m^{t+2}, q, m, 2]}$	$\boldsymbol{MNOA[m^{r+2}, \{(m^{t+1})^{q-1}\}^p, \{(m)^{q-1}\}^p]}$	$\boldsymbol{\pi}$
5	2	$\mathrm{OA}[16, 15,2, 2]$	$\mathrm{MNOA}[128, \{(8)^{14}\}^9, \{(2)^{14}\}^9]$	0.8960
8	2	$\mathrm{OA}[16, 15,2, 2]$	$\mathrm{MNOA}[1024, \{8)^{14}\}^{73}, \{(2)^{14}\}^{73}]^*$	0.9872
11	2	$\mathrm{OA}[16, 15,2, 2]$	$\mathrm{MNOA}[8192, \{(8)^{14}\}^{585}, \{(2)^{14}\}^{585}]^*$	0.9985
5	4	$\mathrm{OA}[256, 85,4, 2]$	$\mathrm{MNOA}[4096, \{(64)^{84}\}^{65}, \{(4)^{84}\}^{65}]^*$	0.9732
8	4	$\mathrm{OA}[256, 85, 4, 2]$	$\mathrm{MNOA}[262144, \{(64)^{84}\}^{4161}, \{(4)^{84}\}^{4161}]^*$	0.9997
5	3	$\mathrm{OA}[81,40,3, 2]$	$\mathrm{MNOA}[2187, \{(27)^{39}\}^{28}, \{(3)^{39}\}^{28}]^*$	0.9651
8	3	$\mathrm{OA}[81, 40,3, 2]$	$\mathrm{MNOA}[19683, \{(27)^{39}\}^{757}, \{(3)^{39}\}^{757}]^*$	0.9987
5	9	$\mathrm{OA}[6561, 820,9, 2]$	$\mathrm{MNOA}[531441, \{(729)^{819}\}^{730}, \{(9)^{819}\}^{730}]^*$	0.9986
5	5	$\mathrm{OA}[625, 156,5, 2]$	$\mathrm{MNOA}[15625, \{(125)^{155}\}^{126}, \{(5)^{155}\}^{126}]^*$	0.9921

Notes: All values in the last column of the above tables are obtained by using equation (1) and all designs marked by (*) are newly constructed else are same as Mukerjee et. al (2014).

4. Conclusion

In this paper, we constructed nearly orthogonal arrays mappable into fully orthogonal arrays of strength two. Some new designs are also constructed, these new designs can be useful as better space filling designs, since these designs give us better values of degree of orthogonality π.

References

Bose, R. C., and Bush, K. A., (1952). Orthogonal arrays of strength two and three. Ann. Math. Statist., **23**, 508-524.

Hedayat, A. S., Sloane, N. J. A. and Stufken., J (1999). Orthogonal arrays: Theory and Applications, Springer, New York.

Mukerjee, R., Sun, F. and Tang, B., (2014). Nearly orthogonal arrays mappable into fully orthogonal arrays, Biometrika, **101**, 957-963.

Liu, H., Sun, F., Lin, K. J. Dennis and Liu, M-Q., (2023). On the construction of mappable nearly orthogonal arrays with column - orthogonality, Communications in mathematics and statistics. https://doi.org/10.1007/s40304-023-00333-x.

Rao, C.R., (1946). Hypercubes of strength d, leading to confounded designs in factorial experiments, Bulletin of the Calcutta Mathematical Society, **38**, 67-78.

Rao, C.R., (1947). Factorial experiments derivable from combinatorial arrangements of arrays, Supplement to the Journal of the Royal Statistical Society, **9,** 128-139.

Raghavarao, D., (1971). Constructions and combinatorial problems in Designs of experiments, John Wiley, New York.

Singh, P., Mazumder, D. M., and Babu, S., (2023). Construction of nearly orthogonal arrays mappable into fully orthogonal arrays of strength two and three. International Journal of Mathematics and Statistics. 24, 37- 50 .