Fall 2019

# Exhaustive Search Algorithms to Find **α**(0,1) Designs with High Harmonic Mean Canonical Efficiency Factors

Zihao Chen
zc78@iastate.edu

# Exhaustive Search Algorithms to Find $\alpha(0,1)$ Designs with High Harmonic Mean Canonical Efficiency Factors

Zihao Chen, Dan Nettleton

December 5, 2019

**Abstract**

Resolvable incomplete block designs have been of interest in many fields for decades. Considerable research has been done on how to construct highly efficient resolvable incomplete block designs with respect to various efficiency criteria. Introduced by E.R Williams in 1975, $\alpha$ designs with zeros or ones in the off-diagonal of the concurrence matrix (denoted as $\alpha(0,1)$ designs) comprise one class of resolvable incomplete block designs with $v$ varieties, $b$ blocks, $r$ replicates and block size of $k$. It can be challenging to find an $\alpha(0,1)$ design with high efficiency, and when $v$ and $b$ become large, an exhaustive search is required. This paper proposes algorithms for constructing $\alpha(0,1)$ designs for $r=2$, $r=3$ and $r=4$ with highest possible harmonic mean canonical efficiency factors. The algorithms are based on the ideas of the combinatorial and factorial number systems. The performance of the algorithms for large $v$ ($v \geq 100$) is evaluated, and harmonic mean canonical efficiency factors are compared to theoretical upper bounds. A simulation study was carried out to compare randomized complete block designs and $\alpha(0,1)$ designs with different efficiency factors.

*Keywords:* resolvable design; exhaustive search; harmonic mean canonical efficiency factors; combinatorial number system; factorial number system.

# 1 Introduction

A block design is resolvable if the blocks can be partitioned into replicates such that each replicate is essentially one complete block with one experimental unit for each treatment. Let $D(v,b,r,k)$ denote the class of resolvable incomplete block designs with $v$ treatments, $b$ blocks of size $k$ and $r$ replicates. By definition, the class $D(v,b,r,k)$ is nonempty only when $\frac{v}{k} = \frac{b}{r} = s$ where $s$ is an integer representing the number of blocks per replication. A typical problem is to find a design in class $D(v,b,r,k)$ that has the highest possible efficiency. One efficiency criterion is the harmonic mean canonical efficiency factor (see Raghavarao (1971) and section 2 below). John (1966) pointed out that reducing the range of the off-diagonal elements of the concurrence matrix (defined in Section 2) improves the harmonic mean canonical efficiency factor. The class of $\alpha(0,1)$ designs is of interest because such designs have only zeros or ones in the off-diagonal elements. Williams (1975) presented the specific forms of the generating array for $\alpha(0,1)$ designs with $k = s$ and $k = s-1$ for $r = 2,3,4$. However, for $r \leq k < s-1$ , there are no specific forms, and $\alpha(0,1)$ designs are numerous when the number of varieties $v$ becomes large. Therefore, an exhaustive search is needed to find an $\alpha(0,1)$ design with highest possible efficiency among all the $\alpha(0,1)$ designs given $(v,b,r,k)$. Williams (1975) stated for large value of $v$ ($v \geq 100$) and certain values of $r, s$ and $k$, there are too many different generating arrays (see section 2 below) for an exhaustive search to be carried out. In this paper we will show that with the combinatorial and factorial number systems, it is possible to order all generating arrays and corresponding $\alpha(0,1)$ designs sequentially. This sequential ordering permits an exhaustive search so that, given $(v,b,r,k)$, we are able to identify the most efficient $\alpha(0,1)$ design eventually.

# 2 Background Review

The incidence matrix $N$ of a design in $D(v,b,r,k)$ is a $v \times b$ matrix with entry $n_{ij}$ being the number of experimental units assigned to variety $i$ in block $j$. The concurrence matrix of a design is defined as the product $NN'$ where $N'$ is the transpose of the incidence matrix $N$. The harmonic mean canonical efficiency factor (denoted as $\bar{E}$) of a $(v,b,r,k)$ design is the harmonic mean of the non-zero eigenvalues of matrix $C$, i.e., $\bar{E} = \frac{v-1}{\sum_{i=1}^{v-1}\frac{1}{\theta_i}}$ where $C = I - \frac{1}{rk}NN'$, $\theta_i, i = 1,2,...,v-1$ are the non-zero eigenvalues of $C$ (Raghavarao, 1971).

The design $D_d$ with varieties and blocks interchanged is called the dual of a design $D$. The harmonic mean canonical efficiency factor $\bar{E}_d$ for the dual design $D_d$ can be also defined in terms of the $b \times b$ matrix $C_d$ where $C_d = I - \frac{1}{rk}N'N$.

Because $C$ and $C_d$ have the same non-unit eigenvalues (Williams, 1975), we have $\bar{E} = \frac{v-1}{v-b+(b-1)\bar{E}_d^{-1}} = \frac{v-1}{v-b+\sum_{j=1}^{b-1}\gamma_j^{-1}}$ where $\bar{E}_d$ is the harmonic mean canonical efficiency factor of the dual design and $\gamma_j, j = 1, 2, ..., b-1$ are the eigenvalues of $C_d$.

The theoretical upper bound for the harmonic mean canonical efficiency factor of a resolvable design is $\frac{(v-1)(r-1)}{(v-1)(r-1)+b-r} = \frac{(v-1)(r-1)}{(v-1)(r-1)+r(s-1)}$ (Patterson and Williams, 1975).

Given the class $D(v, b, r, k)$ where $\frac{v}{k} = \frac{b}{r} = s$, a design constructed as follows is called an $\alpha$ design:

1. Let a generating array $\alpha$ be a $k \times r$ array whose entries can take values from $0, 1, ..., s-1$. For each of the columns in $\alpha$, we generate another $s-1$ columns by cyclically adding 1 to the elements in that column and reducing modulus $s$. This results in a $k \times rs$ array $\alpha^{\star}$.

2. For the $i$th row $(i = 1, 2, ..., k)$ in the array $\alpha^*$, we add $(i-1)s$ to the elements in that row. The resulting array is a resolvable design with $r$ replicates; i.e., each of the $r$ columns of $\alpha$ generates a complete replicate.

The dual generating array $\alpha'$ of a generating array $\alpha$ is a $r \times k$ array whose $(m, l)$th element is defined by $a'_{ml} = \dot{-}a_{lm}$ $(l = 1, ..., k; m = 1, ..., r)$ where $a'_{ml}$ is the $(m, l)$th element in array $\alpha'$, $a_{lm}$ is the $(l, m)$th element in array $\alpha$, and $\dot{-}$ represents the operation of subtracting from $s$ and performing modulo $s$ (Williams, 1975).

For example, if we randomly pick a generating array $\alpha$ from a resolvable design with $v = 16, b = 12, r = 3, k = 4, s = 4$, we can generate the corresponding $\alpha$ design as follows:

$$\alpha = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \end{pmatrix} \Rightarrow \alpha^* = \begin{pmatrix} 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 & 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 & 3 & 0 & 1 & 2 & 1 & 2 & 3 & 0 \\ 0 & 1 & 2 & 3 & 1 & 2 & 3 & 0 & 2 & 3 & 0 & 1 \\ 0 & 1 & 2 & 3 & 2 & 3 & 0 & 1 & 0 & 1 & 2 & 3 \end{pmatrix}$$

Add $(i-1) \times 4$ to $i$th row in the array $\alpha^*$, then we have

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 7 | 4 | 5 | 6 | 5 | 6 | 7 | 4 |
| 8 | 9 | 10 | 11 | 9 | 10 | 11 | 8 | 10 | 11 | 8 | 9 |
| 12 | 13 | 14 | 15 | 14 | 15 | 12 | 13 | 12 | 13 | 14 | 15 |

and every 4 columns form a complete replicate.

The dual generating array $\alpha'$ of the generating array $\alpha$ will be

$$\alpha' = \begin{pmatrix} (4-0)\,mod\,4 & (4-0)\,mod\,4 & (4-0)\,mod\,4 & (4-0)\,mod\,4 \\ (4-0)\,mod\,4 & (4-3)\,mod\,4 & (4-1)\,mod\,4 & (4-2)\,mod\,4 \\ (4-0)\,mod\,4 & (4-1)\,mod\,4 & (4-2)\,mod\,4 & (4-0)\,mod\,4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 2 \\ 0 & 3 & 2 & 0 \end{pmatrix}$$

Two generating arrays with same parameters are equivalent if the $\alpha$ designs produced have the same efficiency factor. Every generating array $\alpha$ is equivalent to a special array called the reduced array for $\alpha$, with all elements in the first row and first column equal to zero (Williams, 1975).

Isomorphic designs are designs with the same reduced array $\alpha$. Isomorphic operations on a generating array $\alpha$ include permutation of rows and columns. (Patterson and Williams, 1976).

# 3 Introduction of combinatorial and factorial number systems

Later on in the algorithms in section 4, we will show that we need a way to decode a natural number in $(0, 1, 2, ..., C(N, k) - 1)$ to a specific combination that is in the $C(N, k)$ combinations and decode a natural number in $(0, 1, 2, ..., P(N, k) - 1)$ to a specific permutation that is in the $P(N, k)$ permutations.

The combinatorial number system of degree $k$ gives a one-to-one correspondence between natural numbers $(0, 1, 2, ..., C(N, k) - 1)$ and $C(N, k)$ combinations where $C(N, k) = \frac{N!}{(N-k)!k!}$.

Specifically, we can represent a natural number $n \in \{0, 1, 2, ..., C(N, k) - 1\}$ by $(C_k, C_{k-1}, ..., C_2, C_1)$ where $C_k > C_{k-1} > ... > C_2 > C_1 \geq 0$,

$$n = \begin{pmatrix} C_k \\ k \end{pmatrix} + \begin{pmatrix} C_{k-1} \\ k-1 \end{pmatrix} + ... + \begin{pmatrix} C_2 \\ 2 \end{pmatrix} + \begin{pmatrix} C_1 \\ 1 \end{pmatrix}$$

To find $(C_k, C_{k-1}, ..., C_2, C_1)$, take $C_k$ as the maximum number satisfying $\begin{pmatrix} C_k \\ k \end{pmatrix} \leq n$, then take $C_{k-1}$ as the maximum number satisfying $\begin{pmatrix} C_{k-1} \\ k-1 \end{pmatrix} \leq n - \begin{pmatrix} C_k \\ k \end{pmatrix}$ until we find $C_1$. Hence, we have a one-to-one correspondence between $n$ and $(C_k, C_{k-1}, ..., C_2, C_1)$ which is one of the combinations. One of the proofs for the correspondence was shown by Abu et al. (2016).

The factorial number system converts natural numbers $0, 1, 2, ..., n! - 1$ to factorial representations through which we can obtain permutations of $0, 1, 2, ..., n - 1$.

We adopt zero-based numbering, and for a given natural number $k$ ($0 \leq k < n!$), suppose we are interested in finding the $k$th permutation of the $n$ numbers labelled as $0, 1, ..., n - 1$.

First, we represent $k$ as $(F_n, F_{n-1}, ..., F_2, F_1)$ such that $k = F_n(n-1)! + F_{n-1}(n-2)! + ... + F_2 1! + F_1 0!$ where the maximum number $F_i$ can take is $i - 1$, and the minimum is 0 for $i = 1, 2, ..., n$. $(F_n, F_{n-1}, ..., F_2, F_1)$ is called the factorial representation of $k$. We argue that there are one-to-one correspondences between the natural numbers from 0 to $n! - 1$ and $(F_n, F_{n-1}, ..., F_2, F_1)$.

The one-to-one correspondence can be shown as follows:

1. We have $n!$ permutations in total, and we also have $n!$ representations from $(F_n, F_{n-1}, ..., F_2, F_1)$ due to the fact that $F_n$ can take $n$ different values, $F_{n-1}$ can take $n - 1$ different values and so forth.

2. The maximum number that factorial representation can give is $(n-1) \times (n-1)! + (n-2) \times (n-2)! + ... + 1 \times 1! + 0 \times 0! = n! - 1$ which can be proved by induction as follows:

   Suppose we want to prove $\sum_{i=0}^{n} i \cdot i! = (n+1)! - 1$

   For $n = 1$, $\sum_{i=0}^{n=1} i \cdot i! = 0 + 1 = (1+1)! - 1$.

   Assume the equation holds for $n = k$, i.e., $\sum_{i=0}^{k} i \cdot i! = (k+1)! - 1$.

   $\sum_{i=0}^{k+1} i \cdot i! = \sum_{i=0}^{k} i \cdot i! + (k+1)(k+1)! = (k+1)! - 1 + (k+1)(k+1)! = (k+2)! - 1$.

   Hence, it also holds for $n = k + 1$.

For example, with $n = 3$, the mapping between $0, 1, ..., 3! - 1$ and the factorial representation $(F_3, F_2, F_1)$ is

| Natural Number | Factorial Representation |
|---|---|
| 0 | 000 |
| 1 | 010 |
| 2 | 100 |
| 3 | 110 |
| 4 | 200 |
| 5 | 210 |

To transform a natural number to a factorial number, we successively divide the number by $1, 2, 3, ...$ until the quotient becomes zero. The remainders from each step give the factorial representation (McCaffrey, 2004).

For example, natural number $531 (0 \leq 531 < 6!)$ can be transformed into a factorial representation as follows:

| | Quotient | Remainder |
|---|---|---|
| 531/1 | 531 | 0 |
| 531/2 | 265 | 1 |
| 265/3 | 88 | 1 |
| 88/4 | 22 | 0 |
| 22/5 | 4 | 2 |
| 4/6 | 0 | 4 |

Factorial representation of 531 is $(4, 2, 0, 1, 1, 0)$.

Second, we transform $(F_n, F_{n-1}, ..., F_{2,} F_1)$ to the corresponding permutation, i.e., $k$th permutation.

To transform a factorial representation of length $n$ i.e., $(F_n, F_{n-1}, ..., F_{2,} F_1)$ to a permutation of natural numbers $0, 1, 2, ..., n-1$, we can follow several steps (Kazunori, 2007):

1. Initialize a null vector $s$ of length $n$. Let vector $x = (0, 1, 2, ..., n-1)$, the position index start from the left and the initial position index to be zero.
2. Let $i = 0$.
3. Store element $x_i$ in vector $s$ at position indexed by $F_{n-i}$.
4. $i = i + 1$.
5. If $i = n$, stop. Otherwise, exclude the positions in $s$ that already have values stored, and reindex the leftovers from the left as $0, 1, 2, ..., n-i-1$.
6. Repeat steps from 3 to 5.

For example, to generate a permutation of $(0, 1, 2, 3, 4)$ (of size 5) from a $5-$digit factorial representation $(F_5, F_4, F_3, F_{2,} F_1) = (4, 1, 1, 0, 0)$, let the vector $x = (0, 1, 2, 3, 4)$ denoted by $x = (a, b, c, d, e)$ where $a = 0, b = 1, ..., e = 4$ to distinguish the position index. We will determine the permutation by the steps as above.

Let $s$ be a null vector of size 5. ($s = (0,0,0,0,0)$)

| Step | Assignment | vector $s$ | Reindex $s$ |
|---|---|---|---|
| 1 | $F_5 = 4$; $s_4 = x_0 = a.$ | $(0,0,0,0,a)$ | $(\underbrace{0}_{0},\underbrace{0}_{1},\underbrace{0}_{2},\underbrace{0}_{3},a)$ |
| 2 | $F_4 = 1$; $s_1 = x_1 = b$ | $(0,b,0,0,a)$ | $(\underbrace{0}_{0},b,\underbrace{0}_{1},\underbrace{0}_{2},a)$ |
| 3 | $F_3 = 1$; $s_1 = x_2 = c$ | $(0,b,c,0,a)$ | $(\underbrace{0}_{0},b,c,\underbrace{0}_{1},a)$ |
| 4 | $F_2 = 0$; $s_0 = x_3 = d$ | $(d,b,c,0,a)$ | $(d,b,c,\underbrace{0}_{0},a)$ |
| 5 | $F_1 = 0$; $s_0 = x_4 = e$ | $(d,b,c,e,a)$ | |

Therefore, the permutation corresponding to $(F_5, F_4, F_3, F_2, F_1) = (4,1,1,0,0)$ is $(d,b,c,e,a) = (3,1,2,4,0)$.

# 4 Construction of $\alpha(0,1)$-design

An $\alpha$ design is called an $\alpha(0,1)$ design if the off-diagonal elements of the concurrence matrix $NN'$ consist of only zeros and ones.

Williams (1975) showed that a generating array $\alpha$ produces an $\alpha(0,1)$ design if and only if $a_{l,m} \dot{-} a_{l',m} \neq a_{l,m'} \dot{-} a_{l',m'}$ where $l \neq l'; m \neq m'; l, l' = 1, ..., k; m, m' = 1, ..., r$, and a dual generating array $\alpha'$ produce an $\alpha(0,1)$ design if and only if

$$a'_{m,l'} \dot{-} a'_{m,l} \neq a'_{m',l'} \dot{-} a'_{m',l} \quad (\star)$$

where $l \neq l'; m \neq m'; l, l' = 1, ..., k; m, m' = 1, ..., r$ and $a \dot{-} b \equiv (a-b)\, mod\, s$.

Williams (1975) showed that for an $\alpha(0,1)$ design to exist, $k \leq s$ is a necessary condition and presented the specific forms for $\alpha(0,1)$ designs for $k = s$ and $k = s-1$ for $r = 2,3,4$.

However, the $\alpha(0,1)$ design with highest possible efficiency for $r \leq k < s-1$ need an exhaustive search, and we propose algorithms to conduct the exhaustive search.

In section 2, we have $\bar{E} = \frac{v-1}{v-b+(b-1)E_d^{-1}} = \frac{v-1}{v-b+\sum_{j=1}^{b-1} \gamma_j^{-1}}$. Therefore it is easier to get the harmonic mean canonical efficiency factor through the dual design because $b \leq v$, and calculating the eigenvalues of a $b \times b$ matrix $C_d$ requires less effort than doing so for a $v \times v$ matrix $C$.

**Case I:** For $r = 2$, a dual generating array $\alpha'$ is a $2 \times k$ matrix, and the equivalent reduced array is of the form

$$\begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & a'_{2,2} & a'_{2,3} & ... & a'_{2,k} \end{pmatrix}_{2\times k}$$

From equation ($\star$), $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ must take values without replacement from $1, 2, ..., s-1$. By the isomorphy property, we have $C(s-1, k-1) = \frac{(s-1)!}{(k-1)!(s-k)!}$ choices in total because permutation of columns is an isomorphic operation. Based on the combinatorial number system, we construct a combination-generating function $COMBINADIC(N, k, i)$ that generates the $i$th combination case in $C(N, k)$ combinations without the knowledge of previous $i-1$ combinations.

**Algorithm 1** Exhaustive Search for $\alpha(0,1)$ designs with highest efficiency when $r = 2$

```
Input: k, s
Output: Design_matrix; Ē_update
function COMBINADIC(N, k, i)
return (c_k, c_{k−1}, ..., c_2, c_1) that corresponds to ith combination from C(N, k).
end function

Initialize: α_dual; Ē_update = 0
for i from 1 to (s−1)!/((k−1)!(s−k)!) do
    a'_{2,2}, a'_{2,3}, .....a'_{2,k} <-COMBINADIC(s − 1, k − 1, i)
    Set a'_{2,2}, a'_{2,3}, .....a'_{2,k} as the row nested in the reduced dual generating array α'
    Construct the dual design incidence matrix N'
    Construct C_d = I − (1/(rk)) N'N
    Calculate the non-zero eigenvalues of C_d γ_j, j = 1, 2, ..., b − 1
    Calculate the Efficiency factor as Ē = (v−1) / (v−b+Σ_{j=1}^{b−1} γ_j^{−1})
    if Ē > Ē_update then
        α_dual < −α'
        Ē_update < −Ē
    end if
end for
Transform the dual generating array α_dual to the generating array α by
a_{lm} = (s − a'_{ml}) mod s (l = 1, ..., k; m = 1, ...., r)
Generate Design_matrix by the procedure of generating α design.
```

For example, if we have $v = 24, b = 12, r = 2, k = 4, s = 6$, then $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ can take one of 10 combinations $(3, 2, 1), (4, 2, 1), (4, 3, 1), (4, 3, 2), (5, 2, 1), (5, 3, 1), (5, 3, 2), (5, 4, 1), (5, 4, 2), (5, 4, 3)$. The COMBINADIC function will go through each choice.

**Case II:** For $r = 3$, a dual generating array $\alpha'$ is a $3 \times k$ matrix, and the equivalent reduced array is of the form

$$\begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & a'_{2,2} & a'_{2,3} & ... & a'_{2,k} \\ 0 & a'_{3,2} & a'_{3,3} & ... & a'_{3,k} \end{pmatrix}_{3\times k}$$

From equation ($\star$), both vectors $(a'_{2,2}, a'_{2,3}, ..., a'_{2,k})$ and $(a'_{3,2}, a'_{3,3}, ..., a'_{3,k})$ must take values without replacement from $1, 2, ..., s-1$. Suppose $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ is the $i$th of the $C(s-1, k-1)$ combinations, and given that $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ has taken the $i$th combination where $i = 1, 2, ..., \frac{(s-1)!}{(k-1)!(s-k)!}$, we consider that $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ takes $u$th permutation of $j$th combination where $j = i+1, i+2, ..., \frac{(s-1)!}{(k-1)!(s-k)!}$, $u = 1, 2, ..., (k-1)!$. We also need $a'_{2,l} \neq a'_{3,l}$ $(u = 2, ..., k)$ because the elements in the first column of reduced array are 0.

Now, we will conduct an exhaustive search with the help of a permutation-generating function $PERM(N, k, i)$ based on the ideas of combinatorial and factorial number systems. The function $PERM(N, k, i)$ will generate the $i$th permutation case in $P(N, k)$ permutations without the knowledge of previous $i-1$ permutations.

The basic idea of function $PERM(N, k, i)$ is that we view $P(N, k)$ as the combination of all $k!$ permutations from each of the $C(N, k)$ pieces, i.e., $P(N, k) = C(N, k) \times k!$. Given an index $i$ $(i = 1, 2, ..., P(N, k))$, we first identify the piece that $i$th permutation belongs to, and then identify the position of the permutation in the $k!$ permutations in that piece.

**Algorithm 2** Exhaustive search for $\alpha(0,1)$ designs with highest efficiency when $r = 3$

**Input:** $k, s$

**Output:** $Design\_matrix$; $\bar{E}\_update$

**function** PERM($N, k, i$)

    $index\_piece$ <- position index of the piece that the $i$th permutation belongs to

    $array\_comb$ <- the combination identified by $index\_piece$

    $index\_perm$ <- position index of permutation in that piece

    $i\_fac$ <- factorial representation for index $index\_perm$

    $array\_perm$ <- the permutation converted from $i\_fac$ based on $array\_comb$

**return** $array\_perm$

**end function**

**Initialize:** $\alpha\_dual$; $\bar{E}\_update$**=0**

**for** $i$ from 1 to $\frac{(s-1)!}{(k-1)!(s-k)!}$ **do**

    **for** $j$ from 1 to $\frac{(s-1)!}{(k-1)!}$ **do**

        **for** $u$ from 1 to $(k-1)!$ **do**

            $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ <-$COMBINADIC(s-1, k-1, i)$

            $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ <- $PERM(COMBINADIC(s-1, k-1, j), k-1, u)$

            Combine $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ and $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ as the two rows nested in the reduced dual generating array $\alpha'$

            **if** $a'_{2,l} \neq a'_{3,l}$ and $a'_{2,l'} \dot{-} a'_{2,l} \neq a'_{3,l'} \dot{-} a'_{3,l}$ $(l = 2, ..., k)$ **then**

                Construct the dual design incidence matrix $N'$

                Construct $C_d = I - \frac{1}{rk} N'N$

                Calculate the non-zero eigenvalues of $C_d$ $\gamma_j, j = 1, 2, ..., b-1$

                Calculate the Efficiency factor as $\bar{E} = \frac{v-1}{v-b+\sum_{j=1}^{b-1} \gamma_j^{-1}}$

                **if** $\bar{E} > \bar{E}\_update$ **then**

                    $\alpha\_dual < -\alpha'$

                    $\bar{E}\_update < -\bar{E}$

                **end if**

            **end if**

        **end for**

    **end for**

    Transform the dual generating array $\alpha\_dual$ to the generating array $\alpha$ by $a_{lm} = (s - a'_{ml}) \, mod \, s$ $(l = 1, ..., k; m = 1, ...., r)$

    Generate $Design\_matrix$ by the procedure of generating $\alpha$ design

**Case III:** For $r=4$, a dual generating array $\alpha'$ is a $4\times k$ matrix, and the equivalent reduced array is of the form

$$\begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ 0 & a'_{2,2} & a'_{2,3} & ... & a'_{2,k} \\ 0 & a'_{3,2} & a'_{3,3} & ... & a'_{3,k} \\ 0 & a'_{4,2} & a'_{4,3} & ... & a'_{4,k} \end{pmatrix}_{4\times k}$$

From equation ($\star$), vectors $(a'_{2,2}, a'_{2,3}, ..., a'_{2,k})$, $(a'_{3,2}, a'_{3,3}, ..., a'_{3,k})$ and $(a'_{4,2}, a'_{4,3}, ..., a'_{4,k})$ must take values without replacement from $1, 2, ..., s-1$. Suppose $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ is the $i$th of the $C(s-1, k-1)$ combinations, and given that $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ has taken the $i$th combination where $i = 1, 2, ..., \frac{(s-1)!}{(k-1)!(s-k)!}$, we consider that $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ takes $u$th permutation of $j$th combination where $j = i+1, i+2, ..., \frac{(s-1)!}{(k-1)!(s-k)!}, u = 1, 2, ..., (k-1)!$ and $a'_{4,2}, a'_{4,3}, ..., a'_{4,k}$ takes $u'$th permutation of $t$th combination where $t = j+1, j+2, ..., \frac{(s-1)!}{(k-1)!(s-k)!}, u' = 1, 2, ..., (k-1)!$.

We also need $a'_{2,l} \neq a'_{3,l}$, $a'_{3,l} \neq a'_{4,l}$, $a'_{2,l} \neq a'_{4,l}$ $(l = 2, ..., k)$ because the elements in the first column of reduced array are 0.

A R program has been written to conduct the exhaustive search algorithms. The program will pause when an $\alpha(0,1)$ design is found, and ask the user whether to stop or continue the sequential search until an $\alpha(0,1)$ design with higher efficiency is encountered.

**Algorithm 3** Exhaustive search for $\alpha(0,1)$ designs with highest efficiency when $r = 4$

**Input:** $k, s$

**Output:** *Design_matrix*; $\bar{E}$*_update*

**function** PERM$(N, k, i)$
    *index_piece* <- position index of the piece that the $i$th permutation belongs to
    *array_comb* <- the combination identified by *index_piece*
    *index_perm* <- position index of permutation in that piece
    *i_fac* <- factorial representation for index *index_perm*
    *array_perm* <- the permutation converted from *i_fac* based on *array_comb*
**return** *array_perm*
**end function**

**Initialize:** $\alpha$*_dual*; $\bar{E}$*_update*=**0**
**for** $i$ from 1 to $\frac{(s-1)!}{(k-1)!(s-k)!}$ **do**
    **for** $j$ from 1 to $\frac{(s-1)!}{(k-1)!}$ **do**
        **for** $t$ from 1 to $\frac{(s-1)!}{(k-1)!}$ **do**
            **for** $u$ from 1 to $(k-1)!$ **do**
                **for** $u'$ from 1 to $(k-1)!$ **do**
                    $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$ <-$COMBINADIC(s-1, k-1, i)$
                    $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ <- $PERM(COMBINADIC(s-1, k-1, j), k-1, u)$
                    $a'_{4,2}, a'_{4,3}, ..., a'_{4,k}$ <- $PERM(COMBINADIC(s-1, k-1, t), k-1, u')$
                    Combine $a'_{2,2}, a'_{2,3}, ..., a'_{2,k}$, $a'_{3,2}, a'_{3,3}, ..., a'_{3,k}$ and $a'_{4,2}, a'_{4,3}, ..., a'_{4,k}$ as the three rows nested in the reduced dual generating array $\alpha'$
                    **if** $a'_{m,l} \neq a'_{m,l'}$ and $a'_{m,l'} \dot{-} a'_{m,l} \neq a'_{m',l'} \dot{-} a'_{m',l}$ $(l \neq l'; m \neq m'; l, l' = 2, ..., k; m, m' = 2, 3, 4)$. **then**
                        Construct the dual design incidence matrix $N'$
                        Construct $C_d = I - \frac{1}{rk} N'N$
                        Calculate the non-zero eigenvalues of $C_d$ $\gamma_j, j = 1, 2, ..., b-1$
                        Calculate the Efficiency factor as $\bar{E} = \frac{v-1}{v-b+\sum_{j=1}^{b-1} \gamma_j^{-1}}$
                        **if** $\bar{E} > \bar{E}$*_update* **then**
                            $\alpha$*_dual* $< -\alpha'$
                            $\bar{E}$*_update* $> \bar{E}$
                        **end if**
                    **end if**
                **end for**
            **end for**
        **end for**
    **end for**
**end for**
Transform the dual generating array $\alpha$*_dual* to the generating array $\alpha$ by $a_{lm} = (s - a'_{ml}) \, mod \, s$ $(l = 1, ..., k; m = 1, ...., r)$
Generate *Design_matrix* by the procedure of generating $\alpha$ design. =0

For example, given a resolvable design with $v = 112, b = 42, r = 3, k = 8$, run code:

```
> design_alpha(v = 112, k = 8, r = 3)
```

```
Treatments    Blocks    Efficiency    Upper_bound
112           42        0.8391        0.882
```

```
$design
, , Replicate 1

  B1 B2  B3  B4  B5  B6  B7  B8  B9 B10 B11 B12 B13 B14
1  0  1   2   3   4   5   6   7   8   9  10  11  12  13
2 14 15  16  17  18  19  20  21  22  23  24  25  26  27
3 28 29  30  31  32  33  34  35  36  37  38  39  40  41
4 42 43  44  45  46  47  48  49  50  51  52  53  54  55
5 56 57  58  59  60  61  62  63  64  65  66  67  68  69
6 70 71  72  73  74  75  76  77  78  79  80  81  82  83
7 84 85  86  87  88  89  90  91  92  93  94  95  96  97
8 98 99 100 101 102 103 104 105 106 107 108 109 110 111

, , Replicate 2

   B1  B2  B3  B4  B5  B6  B7 B8 B9 B10 B11 B12 B13 B14
1   0   1   2   3   4   5   6  7  8   9  10  11  12  13
2  27  14  15  16  17  18  19 20 21  22  23  24  25  26
3  40  41  28  29  30  31  32 33 34  35  36  37  38  39
4  53  54  55  42  43  44  45 46 47  48  49  50  51  52
5  66  67  68  69  56  57  58 59 60  61  62  63  64  65
6  79  80  81  82  83  70  71 72 73  74  75  76  77  78
7  92  93  94  95  96  97  84 85 86  87  88  89  90  91
8 105 106 107 108 109 110 111 98 99 100 101 102 103 104

, , Replicate 3

   B1  B2  B3  B4  B5  B6  B7  B8  B9 B10 B11 B12 B13 B14
1   0   1   2   3   4   5   6   7   8   9  10  11  12  13
2  21  22  23  24  25  26  27  14  15  16  17  18  19  20
3  41  28  29  30  31  32  33  34  35  36  37  38  39  40
4  48  49  50  51  52  53  54  55  42  43  44  45  46  47
5  68  69  56  57  58  59  60  61  62  63  64  65  66  67
6  75  76  77  78  79  80  81  82  83  70  71  72  73  74
7  95  96  97  84  85  86  87  88  89  90  91  92  93  94
8 102 103 104 105 106 107 108 109 110 111  98  99 100 101
```

```
[1] "Do you want a design with higher efficiency ?"
> Enter y or n: y
```

| Treatments | Blocks | Efficiency | Upper_bound |
|---|---|---|---|
| 112 | 42 | 0.8393 | 0.882 |

, , Replicate 1

| | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | B12 | B13 | B14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 2 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 3 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 |
| 4 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 |
| 5 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 |
| 6 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 |
| 7 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 | 97 |
| 8 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |

, , Replicate 2

| | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | B12 | B13 | B14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 2 | 27 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 3 | 40 | 41 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 |
| 4 | 53 | 54 | 55 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 |
| 5 | 66 | 67 | 68 | 69 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 |
| 6 | 79 | 80 | 81 | 82 | 83 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 |
| 7 | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| 8 | 106 | 107 | 108 | 109 | 110 | 111 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 |

, , Replicate 3

| | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | B10 | B11 | B12 | B13 | B14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 2 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 3 | 36 | 37 | 38 | 39 | 40 | 41 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
| 4 | 52 | 53 | 54 | 55 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 |
| 5 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 56 | 57 | 58 | 59 | 60 |
| 6 | 81 | 82 | 83 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |
| 7 | 97 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 | 96 |
| 8 | 110 | 111 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |

Refer to the tables of examples of $\alpha(0,1)$ designs that includes harmonic mean canonical efficiency factors, and the theoretical upper bounds are provided for large $v >= 100$, $k <= 10$. The first column indicates the number of consecutive searches (the maximum number of consecutive searches within 20 minutes) that has been implemented.

Table 1. Efficiency check of $\alpha(0,1)$ designs with $r = 2, v \geq 100$

| No. of Searches | $v$ | $k$ | $r$ | $\bar{E}$ | Upper Bounds of $\bar{E}$ |
|---|---|---|---|---|---|
| 8 | 104 | 8 | 2 | 0.7943 | 0.8825 |
| 10 | 112 | 8 | 2 | 0.7916 | 0.8820 |
| 2 | 120 | 10 | 2 | 0.8391 | 0.9069 |
| 15 | 128 | 8 | 2 | 0.7872 | 0.8811 |
| 3 | 130 | 10 | 2 | 0.8363 | 0.9063 |
| 23 | 136 | 8 | 2 | 0.7852 | 0.8807 |
| 7 | 140 | 10 | 2 | 0.8341 | 0.9058 |
| 33 | 144 | 8 | 2 | 0.7837 | 0.8803 |
| 11 | 150 | 10 | 2 | 0.8320 | 0.9055 |
| 37 | 152 | 8 | 2 | 0.7824 | 0.8801 |
| 13 | 160 | 10 | 2 | 0.8302 | 0.9051 |
| 52 | 168 | 8 | 2 | 0.7798 | 0.8796 |
| 18 | 170 | 10 | 2 | 0.8286 | 0.9048 |
| 21 | 180 | 10 | 2 | 0.8272 | 0.9045 |
| 46 | 200 | 10 | 2 | 0.8248 | 0.9040 |
| 21 | 210 | 10 | 2 | 0.8176 | 0.9039 |
| 28 | 220 | 10 | 2 | 0.8167 | 0.9037 |
| 25 | 230 | 10 | 2 | 0.8151 | 0.9035 |
| 40 | 240 | 10 | 2 | 0.8148 | 0.9034 |
| 46 | 250 | 10 | 2 | 0.8143 | 0.9033 |
| 43 | 260 | 10 | 2 | 0.8136 | 0.9031 |
| 48 | 270 | 10 | 2 | 0.8130 | 0.9030 |
| 48 | 280 | 10 | 2 | 0.8123 | 0.9029 |
| 65 | 290 | 10 | 2 | 0.8118 | 0.9028 |
| 56 | 300 | 10 | 2 | 0.8113 | 0.9027 |
| 67 | 310 | 10 | 2 | 0.8108 | 0.9026 |
| 74 | 320 | 10 | 2 | 0.8102 | 0.9025 |
| 76 | 330 | 10 | 2 | 0.8096 | 0.9025 |
| 90 | 340 | 10 | 2 | 0.8093 | 0.9024 |
| 75 | 350 | 10 | 2 | 0.8089 | 0.9023 |
| 75 | 360 | 10 | 2 | 0.8083 | 0.9023 |

Table 2. Efficiency check of $\alpha(0,1)$ designs with $r=3, v \geq 100$

| No. of Searches | $v$ | $k$ | $r$ | $\bar{E}$ | Upper Bounds of $\bar{E}$ |
|---|---|---|---|---|---|
| 3 | 108 | 9 | 3 | 0.8613 | 0.8963 |
| 56 | 110 | 5 | 3 | 0.7306 | 0.8059 |
| 15 | 112 | 8 | 3 | 0.8397 | 0.8820 |
| 76 | 114 | 6 | 3 | 0.7777 | 0.8395 |
| 66 | 115 | 5 | 3 | 0.7294 | 0.8057 |
| 69 | 116 | 4 | 3 | 0.6546 | 0.7549 |
| 5 | 117 | 9 | 3 | 0.8594 | 0.8958 |
| 50 | 119 | 7 | 3 | 0.8117 | 0.8634 |
| 13 | 120 | 8 | 3 | 0.8380 | 0.8815 |
| 74 | 124 | 4 | 3 | 0.6529 | 0.7546 |
| 75 | 125 | 5 | 3 | 0.7274 | 0.8052 |
| 11 | 126 | 9 | 3 | 0.8578 | 0.8953 |
| 23 | 128 | 8 | 3 | 0.8364 | 0.8811 |
| 33 | 130 | 5 | 3 | 0.7151 | 0.8050 |
| 62 | 132 | 6 | 3 | 0.7731 | 0.8387 |
| 64 | 133 | 7 | 3 | 0.8088 | 0.8627 |
| 10 | 135 | 9 | 3 | 0.8564 | 0.8948 |
| 15 | 136 | 8 | 3 | 0.8342 | 0.8807 |
| 43 | 138 | 6 | 3 | 0.7677 | 0.8384 |
| 22 | 140 | 7 | 3 | 0.8013 | 0.8625 |
| 13 | 144 | 9 | 3 | 0.8550 | 0.8944 |
| 82 | 145 | 5 | 3 | 0.7233 | 0.8045 |
| 64 | 147 | 7 | 3 | 0.8055 | 0.8622 |
| 85 | 148 | 4 | 3 | 0.6490 | 0.7538 |
| 86 | 148 | 4 | 3 | 0.6490 | 0.7538 |
| 2 | 150 | 10 | 3 | 0.8709 | 0.9055 |
| 32 | 152 | 8 | 3 | 0.8317 | 0.8801 |
| 3 | 153 | 9 | 3 | 0.8530 | 0.8941 |
| 58 | 154 | 7 | 3 | 0.8038 | 0.8620 |
| 82 | 155 | 5 | 3 | 0.7215 | 0.8042 |
| 62 | 156 | 6 | 3 | 0.7678 | 0.8378 |

# 5 Simulation study

Consider simulating from the spherical covariance model (Stroup, 2002) $Y = X\boldsymbol{\beta} + Z\boldsymbol{\gamma} + \boldsymbol{e}$ where $\boldsymbol{\beta}$ represents the treatment effect, $\boldsymbol{\gamma}$ represents the replicate effect and error vector $\boldsymbol{e} \sim MVN(0, R)$. The $ij$th row-column element of R, denoted as $r_{ij}$ is

$$r_{ij} = \begin{cases} \sigma_e^2[1 - (\frac{3d}{2l}) + (\frac{d^3}{2l^3})] & \text{if } d < l \\ 0 & \text{if } d \geq l \end{cases}$$

where $d$ is the Euclidean distance between the $i$th and $j$th observation (the observations were recorded columnwise), and $l$ is the range that is the Euclidean distance between two observations below which spatial correlation is nonzero.

Suppose we have 216 treatments and 2 replicates, and we are interested in estimating the differences between the effect of each treatment and the overall average treatment effect, i.e., $\beta_i - \bar{\beta}, i = 1, 2, ..., 216$.

Suppose we have a $9 \times 48$ field available (432 plots in total) and there are resources to assign two plots to each treatment. We have three competing designs as shown in Figure 5.1 where the treatment numbers are zero-based. The first design is a randomized complete block design in which we randomly assign the 216 treatments to the 2 large complete blocks. The second design is an $\alpha(0, 1)$ design with $v = 216, b = 48, r = 2, k = 9$ where each column is an incomplete block. The last design is an $\alpha(0, 1)$ design with $v = 216, b = 48, r = 2, k = 9$ with $3 \times 3$ square shaped incomplete blocks.

For each iteration, we repeat the following procedure:

(1) Generate the treatment effects $\beta_1, \beta_2, ..., \beta_{216}$ and replicate effects $\gamma_1, \gamma_2$ from $N(0, 10)$. Generate $\boldsymbol{e} \sim MVN(0, R)$ where $\sigma_e = 10$ and range $l = 3$.

(2) Obtain three sets of simulated response $Y_C$, $Y_S$ and $Y_{RCBD}$ corresponding to the $\alpha(0, 1)$ design with incomplete blocks arranged as columns, the $\alpha(0, 1)$ design with incomplete blocks arranged as squares and the randomized complete block design because we need to match the generated treatment effects from (1) to the three designs respectively.

(3) Fit three models corresponding to the three competing designs and calculate the $RMSE = \sqrt{\frac{1}{216}\sum_{i=1}^{216}((\hat{\beta}_i - \bar{\hat{\beta}}) - (\beta_i - \bar{\beta}))^2}$ for each model.

(i) Model for randomized complete block design:

$Y_{ij} = \beta_i + \gamma_j + e_{ij}$, $i = 1, 2, ..., 216; j = 1, 2$

where $\beta_i$ is the $i^{th}$ treatment effect, $\gamma_j$ is the $j^{th}$ complete block effect, and $e_{ij}$ is the error term with $e_{ij} \overset{iid}{\sim} N(0, \sigma^2)$.

(ii) Model for the $\alpha(0,1)$ design with column shaped incomplete blocks:

$Y_{ij} = \beta_i + \gamma_j + \tau_k + e_{ij}$, $i = 1, 2, ..., 216; j = 1, 2; k = 1, 2, ..., 48$

where $\beta_i$ is the $i^{th}$ treatment effect, $\gamma_j$ is the $j^{th}$ complete block effect, $\tau_k$ is the $k^{th}$ incomplete block effect with $\tau_k \overset{iid}{\sim} N(0, \sigma_b^2)$, and $e_{ij}$ is the error term with $e_{ij} \overset{iid}{\sim} N(0, \sigma^2)$.

(iii) Model for the $\alpha(0,1)$ design with square shaped incomplete blocks:

The same model as the second but the value $e_{ij}$ may be different because the locations of some treatments are different.

(4) To further compare the design with $RCBD$ randomization and square shaped incomplete blocks to the $\alpha(0,1)$ design with the square shaped incomplete blocks, fit the model for the design with $RCBD$ randomization and square shaped incomplete blocks:

$Y_{ij} = \beta_i + \gamma_j + \tau_k + e_{ij}$, $i = 1, 2, ..., 216; j = 1, 2; k = 1, 2, ..., 48$

where $\beta_i$ is the $i^{th}$ treatment effect, $\gamma_j$ is the $j^{th}$ complete block effect, $\tau_k$ is the $k^{th}$ incomplete block effect with $\tau_k \overset{iid}{\sim} N(0, \sigma_b^2)$, and $e_{ij}$ is the error term with $e_{ij} \overset{iid}{\sim} N(0, \sigma^2)$.

(5) Calculate five log ratios : $\log(\frac{RMSE_C}{RMSE_S}), \log(\frac{RMSE_{RCBD}}{RMSE_S}), \log(\frac{RMSE_{RCBD}}{RMSE_C}), \log(\frac{RMSE_{RCBD_S}}{RMSE_S})$, $\log(\frac{RMSE_{RCBD_S}}{RMSE_C})$ where $RMSE_C$ is from the model fitted with the $\alpha(0,1)$ design with column shaped incomplete blocks, $RMSE_S$ is from the model fitted with the $\alpha(0,1)$ design with square shaped incomplete blocks, $RMSE_{RCBD}$ is from the model fitted with the randomized complete block design and $RMSE_{RCBD_S}$ is from the model fitted the design with $RCBD$ randomization and square shaped incomplete blocks.

Figure 5.1: (*a*) *Randomized complete block design;* (*b*) $\alpha(0,1)$ *design with column shaped incomplete blocks* (*c*) $\alpha(0,1)$ *design with* $3 \times 3$ *square shaped incomplete blocks;*

(*a*) Randomized complete block design

$$\overset{Block\,1}{\begin{bmatrix} \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \end{bmatrix}_{9\times 24}} \quad \overset{Block\,2}{\begin{bmatrix} \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \\ \cdots & \cdots & & \cdots \end{bmatrix}_{9\times 24}}$$

(*b*) $\alpha(0,1)$ design with column shaped incomplete blocks

| *B*1 | *B*2 | *B*3 | *B*4 | *B*5 | ... | *B*24 | *B*25 | *B*26 | ... | *B*46 | *B*47 | *B*48 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | ... | ... | ... | ... | ... | ... | ... | 196 | 197 | 198 |
| 25 | 26 | 27 | ... | | | | | | | 212 | 213 | 214 |
| 50 | 51 | 52 | | ... | | | | | | 22 | 23 | 24 |
| 75 | 76 | 77 | | | ... | | | | | 47 | 48 | 49 |
| 100 | 101 | 102 | | | | ... | | | | 72 | 73 | 74 |
| 125 | 126 | 127 | | | | | ... | | | 97 | 98 | 99 |
| 150 | 151 | 152 | | | | | | ... | | 122 | 123 | 174 |
| 175 | 176 | 177 | | | | | | | ... | 147 | 148 | 199 |
| 200 | 201 | 202 | ... | ... | ... | ... | ... | ... | ... | 172 | 173 | 215 |

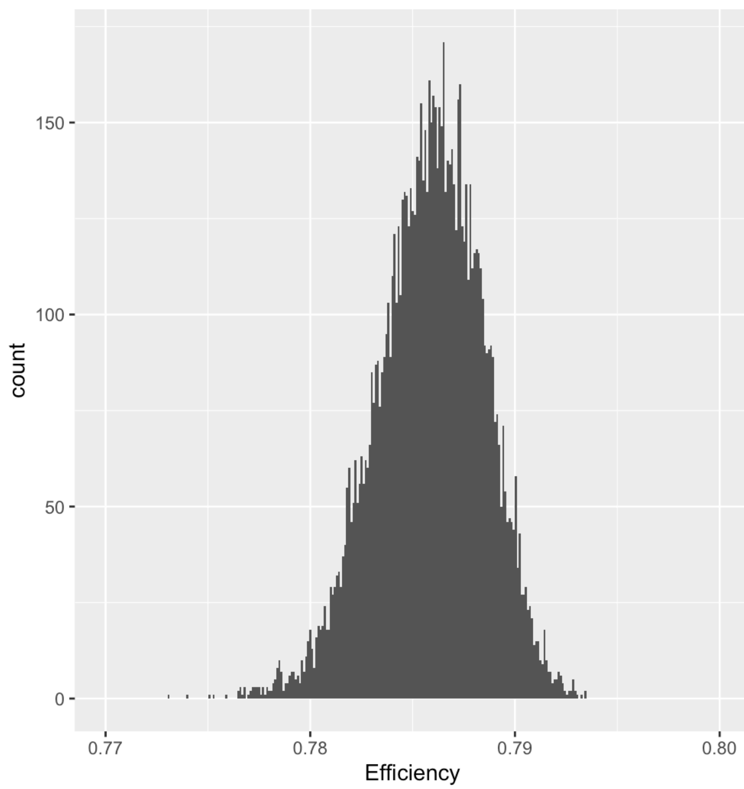
(*c*) $\alpha(0,1)$ design with $3 \times 3$ square shaped incomplete blocks

| | 1 | 2 | 3 | 4 | 5 | ... | 24 | 25 | 26 | ... | 46 | 47 | 48 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 25 | 50 | ... | ... | ... | ... | ... | ... | ... | 196 | 212 | 22 | |
| *B*1 | 75 | 100 | 125 | ... | | | | | | | 47 | 72 | 97 | *B*46 |
| | 150 | 175 | 200 | | ... | | | | | | 122 | 147 | 172 | |
| | 1 | 26 | 51 | | | ... | | | | | 197 | 213 | 23 | |
| *B*2 | 76 | 101 | 126 | | | | ... | | | | 48 | 73 | 98 | *B*47 |
| | 151 | 176 | 201 | | | | | ... | | | 123 | 148 | 173 | |
| | 2 | 27 | 52 | | | | | | ... | | 198 | 214 | 24 | |
| B3 | 77 | 102 | 127 | | | | | | | ... | 49 | 74 | 99 | *B*48 |
| | 157 | 177 | 202 | ... | ... | ... | ... | ... | ... | ... | 174 | 199 | 215 | |

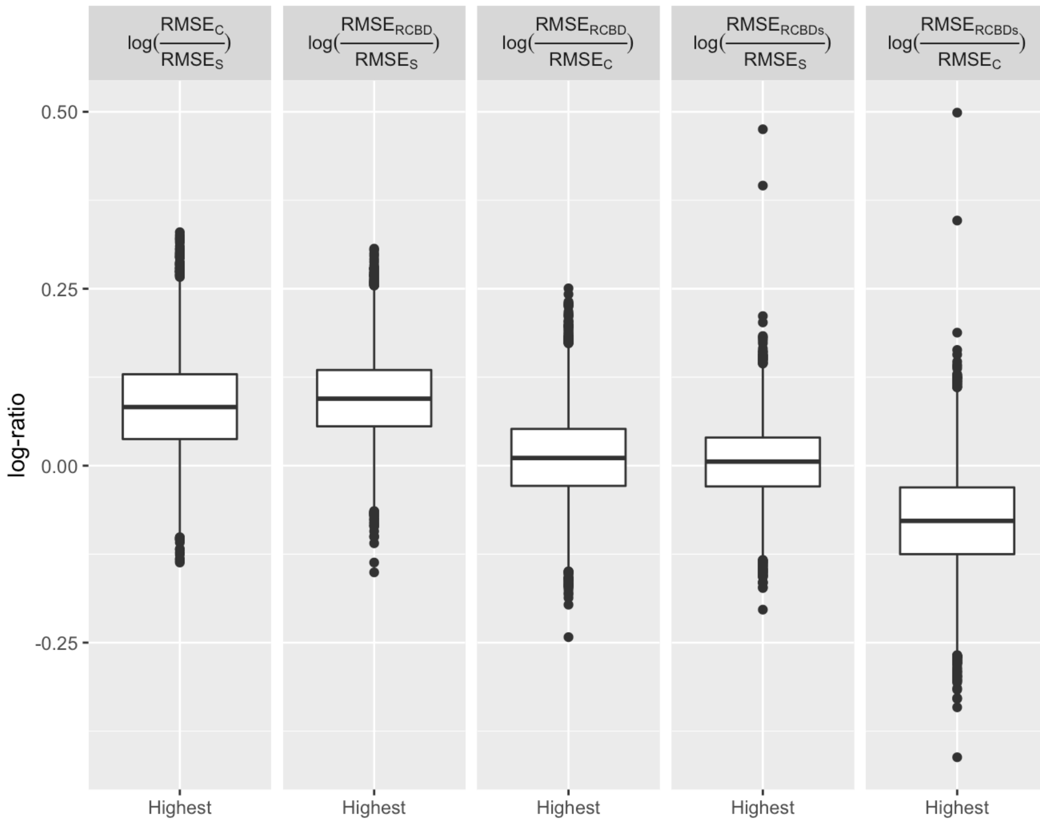
The columns of Table 3 shows the proportions of positive of the log ratios where the $\alpha(0,1)$ design has the highest mean canonical efficiency factor (0.8015); i.e., the search is exhaustive.

| Table 3. Proportions of positive with 10000 iterations | | | | |
|---|---|---|---|---|
| $\log(\frac{RMSE_C}{RMSE_S})$ | $\log(\frac{RMSE_{RCBD}}{RMSE_S})$ | $\log(\frac{RMSE_{RCBD}}{RMSE_C})$ | $\log(\frac{RMSE_{RCBD_S}}{RMSE_S})$ | $\log(\frac{RMSE_{RCBD_S}}{RMSE_C})$ |
| 0.8959 | 0.9496 | 0.5741 | 0.5426 | 0.1304 |

The following graph shows the distribution of the efficiency factor values from the design with $RCBD$ randomization and the square shaped incomplete blocks with 10000 iterations. We notice that most efficiency factor values are between 0.78 and 0.79 which are close to but lower than 0.8015, the efficiency factor value for the $\alpha(0,1)$ design with highest efficiency.




The following boxplot shows the log ratios from 10000 iterations and $\alpha(0,1)$ design has the highest mean canonical efficiency factor (0.8015).




As we can see from Table 3 and the boxplot, the $\alpha(0,1)$ design with square shaped blocks has smaller $RMSE$ than the $\alpha(0,1)$ design with column shaped blocks as well as the randomized complete block designs about 90 percent of the time. The $\alpha(0,1)$ design with column shaped blocks has larger $RMSE$ than the randomized complete block designs more than 50 percent of the time.

The design with $RCBD$ randomization and square shaped incomplete blocks has larger $RMSE$ than the $\alpha(0,1)$ design with square shaped blocks 55 percent of the time because the efficiency factors of the designs with $RCBD$ randomization and square shaped incomplete blocks are slightly smaller than 0.8015. The $\alpha(0,1)$ design with column shaped blocks has larger $RMSE$ than the design with $RCBD$ randomization and square shaped incomplete blocks more than 80 percent of the time.

To examine whether the $\alpha(0,1)$ design with square blocks still have a larger proportion of positive in the log ratio compared to the randomized complete block design if we consider the spatial spherical covariance structure in the error term, we fit another two models as follows:

(i) Spatial model for the randomized complete block design:

$Y_{ij} = \beta_i + \tau_j + e_{ij}$, $i = 1, 2, ..., 216; j = 1, 2$

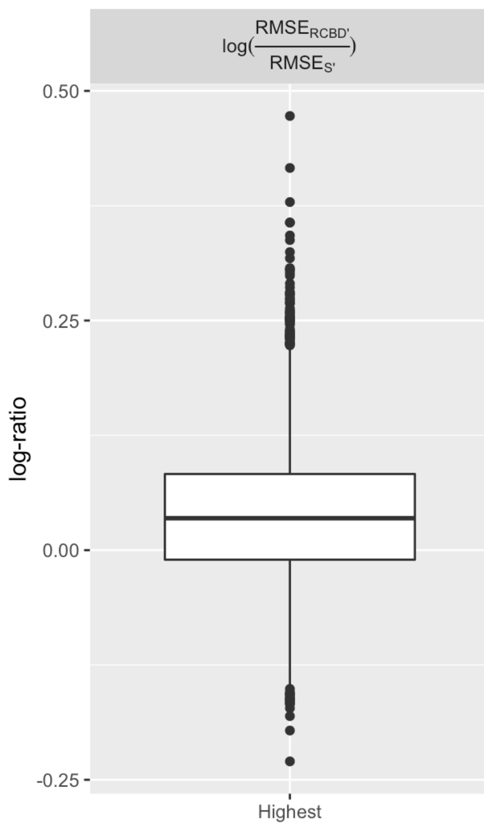
where $\beta_i$ is the $i^{th}$ treatment effect, $\tau_j$ is the $j^{th}$ complete block effect, and $\boldsymbol{e}$ is the error vector with $\boldsymbol{e} \sim MVN(0, R)$. $R$ is the spherical covariance structure defined as above.

(ii) Spatial model for the $\alpha(0,1)$ design with square shaped incomplete blocks:

$Y_{ij} = \beta_i + \gamma_j + \tau_k + e_{ij}$, $i = 1, 2, ..., 216; j = 1, 2; k = 1, 2, ..., 48$

where $\beta_i$ is the $i^{th}$ treatment effect, $\gamma_j$ is the $j^{th}$ complete block effect, $\tau_k$ is the $k^{th}$ incomplete block effect with $\tau_k \overset{iid}{\sim} N(0, \sigma_b^2)$, and $\boldsymbol{e}$ is the error vector with $\boldsymbol{e} \sim MVN(0, R)$.

| Table 4. Proportion of positive with 5000 iterations |
|---|
| $\log(\frac{RMSE_{RCBD'}}{RMSE_{S'}})$ |
| 0.7007 |




The result of comparison between the $\alpha(0,1)$ design with square shaped blocks and the randomized complete block design after accounting for the spherical spatial structure is shown above. $RMSE_{S'}$ is from the model fitted with the $\alpha(0,1)$ design with square shaped incomplete blocks, $RMSE_{RCBD'}$ is from the model fitted with the randomized complete block design.

It indicates that the $\alpha(0,1)$ design with square shaped incomplete blocks still has smaller $RMSE$ than the randomized complete block design around 70 percent of the time.

# 6 Conclusion

The combinatorial number system gives the correspondence between natural numbers and combinations, and factorial number system gives the correspondence between natural numbers and permutations. Based on the ideas of the combinatorial and factorial number systems, we start with the construction of generating array and then conduct an exhaustive search for $\alpha(0,1)$ designs with $r \leq k < s-1$ and $r = 2, 3, 4$. Finally, we are able to find the $\alpha(0,1)$ design with the highest possible efficiency. The simulation study showed that the $\alpha(0,1)$ design with high efficiency factor performed better in estimating the contrast between the effect of each treatment and the overall average effect than the randomized complete block design no matter we consider the spatial structure or not when fitting the model, and adding the square shaped incomplete blocks to the randomized complete block design will reduce the $RMSE$ of the contrast estimate.

## References


Abu, B. S., Saadia, F., and Muhammad, T. (2016). Proof of bijection for combinatorial number system. *arXiv preprint arXiv:1601.05794*.

John, J. A. (1966). Cyclic incomplete block designs. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(2):345–360.

Kazunori, W. (2007). A factorial representation of permutations and its application to flow-shop scheduling. *Systems and Computers in Japan*, 38(1):73–86.

McCaffrey, J. (2004). Generating the $m$th lexicographical element of a mathematical combination. *MSDN Library*.

Patterson, H. D. and Williams, E. R. (1975). Some theoretical results on general block designs. *Congressus Numerantium*, 5(1):489–496.

Patterson, H. D. and Williams, E. R. (1976). A new class of resolvable incomplete block designs. *Biometrika*, 63(1):83–92.

Raghavarao, D. (1971). *Constructions and combinatorial problems in design of experiments*. Wiley.

Stroup, W. (2002). Power analysis based on spatial effects mixed models: A tool for comparing design and analysis strategies in the presence of spatial variability. *Journal of Agricultural, Biological, and Environmental Statistics*, 7(4):491–511.

Williams, E. R. (1975). *A new class of resolvable block designs*. PhD thesis, University of Edinburgh.