Available online at www.isas.org.in/jisas
JOURNAL OF THE INDIAN SOCIETY OF
AGRICULTURAL STATISTICS 69(2) 2015 201-225

Row-Column Designs for Diallel Cross Experiments with Specific Combining Abilities

Cini Varghese, Eldho Varghese, Seema Jaggi and Rajender Parsad
ICAR-Indian Agricultural Statistics Research Institute, New Delhi

Received 22 October 2014; Revised 16 July 2015; Accepted 6 August 2015

SUMMARY

In plant breeding programmes, Mating-Environmental (ME) designs are commonly used under one-way blocking setup to get an estimate for general and specific combining abilities (gca and sca) of inbred lines involved in the crosses. When a large number of crosses are to be compared, which lead to a large experimental area, it may be important to account for or eliminate the effects of fertility trends in the land in two directions. In such situations, designs with crosses arranged in a row-column (RC) set up can be more advantageously used. Though information on both gca and sca effects are important to the breeders, most of the statistical papers dealing with these designs assume sca effects as negligible for reducing the complexity in mathematical derivations. In this paper, a linear fixed effects model under a row-column setup with gca and sca components has been defined and the information matrix for estimating gca effects free from sca effects has been derived. Further, some classes of efficient MERC designs have been obtained for complete diallel cross (CDC) experiments and the designs obtained are found to be variance balanced for estimating the contrasts pertaining to gca effects. Macros have been developed using PROC IML of SAS software for the generation of the MERC designs so constructed.

Keywords: Diallel cross, General combining ability effects, Specific combining ability effects, Row-column design, MERC designs.

1. INTRODUCTION

A major objective of plant and animal breeding programmes is to improve the genetic potential of plants and animals. The breeding experiments comprise two types of designs namely, mating designs and environmental designs. Diallel and partial diallel cross plans are two of the commonly used mating designs. Instead of constructing crossing plans (Mating design), which is a procedure of producing the progenies, and then subjecting these progenies to the environmental conditions in a systematic manner (Environmental design), it is advisable to construct a combined Mating-Environmental (ME) design. Mating designs are commonly used to get an estimate for general and specific combining abilities (gca and sca) of inbred lines involved in the crosses.

Schmidt (1919) introduced the method of diallel crossing as a means of comparing the breeding values of parents. Complete diallel cross (CDC) may be defined as a set of all possible mating between several genotypes which may be individuals, clones, homozygous lines, etc. These crosses are frequently used in both plant and animal breeding for estimating genetic components of total variance of a quantitative character. These are also used in estimating general and specific combining abilities of inbred lines involved in the crosses. The terms gca and sca were originally defined by Sprague and Tatum (1942). The term

Corresponding author: Eldho Varghese
E-mail address: eldho@iasri.res.in

general combining ability (gca) is used to designate the average performance of a line in the hybrid combination whereas specific combining ability (sca) is used to designate those cases in which certain combinations do relatively better or worse than would be expected on the basis of the average performance of the lines involved.

The theory and analysis of diallel crosses were developed by Hayman (1954a, 1954b, 1958, 1960), Griffing (1956 a, b) and Kempthorne (1956). For more details on diallel cross experiments and analysis one may refer to (Kempthorne 1969, Narain 1990 and Singh and Choudhary 2004). Experimental design issues in the context of diallel cross experiments has received considerable attention in the literature; see for reference, Curnow (1963), Hinkelmann (1975) and Gupta *et al*. (1994).

Gupta and Kageyama (1994) investigated optimality of CDC in incomplete blocks and also characterized the optimal CDC. They derived such plans using nested balanced incomplete block designs.

Dey and Midha (1996) proposed optimal block designs for the Type IV mating designs of Griffing (1956). These incomplete block designs are derived by using triangular designs with two associate classes. Mukerjee (1997) investigated optimality aspects of partial diallel cross designs and showed that unblocked partial diallel are E-optimal. He also discussed optimal or efficient blocking of partial diallel cross plans. Das *et al.* (1998) gave some optimal designs for diallel cross experiments. Gupta and Choi (1998) studied optimality aspects of row-column designs for complete diallel crosses. They have constructed universally optimal designs for CDC assuming sca effects are negligible.

Das and Ghosh (1999) have given some methods of construction of balanced incomplete block diallel designs for CDC plans in which the number of experimental units is equal to the number of crosses in complete diallel. Parsad *et al*. (1999) gave some methods of construction of universally optimal block designs for estimating gca effects for both proper as well as non-proper settings and prepared a catalogue of these designs.

Xiang and Li (2001) suggested a method of analyzing diallel cross experiments through PROC MIXED of SAS assuming random gca and sca effects. Choi *et al*. (2004) studied diallel crosses for comparing a control line with test line and listed designs that estimate control versus test comparisons with a minimum variance. Hsu and Ting (2005) studied A-optimality of diallel cross experiments for comparing two or three test lines with a control line. Parsad *et al*. (2005) studied the optimality aspects of designs for 2-line and 4-line crosses and obtained universally optimal designs for 0-way, 1-way and 2-way elimination of heterogeneity.

Sharma and Fanta (2010) proposed a method of construction of balanced and orthogonal block designs for CDC plan connected for cross effects through cyclic permutation of initial blocks.

All the work on diallel/partial diallel cross experiments discussed above has been done under the assumption that the specific combining ability is negligible. Apart from inferring on gca effects, often an experimenter is also interested in inference on “cross effects” or, “sca effects”. Chai and Mukerjee (1999) studied the optimality aspect of dialllel cross experiments when specific combining abilities are also included in the model. Choi *et al.* (2002), Das and Dey (2004) and Dey (2010) investigated optimality of orthogonally blocked complete diallel crosses for estimating general combining abilities when the model also includes specific combining abilities.

Heterogeneity in the experimental material is the most important problem to be reckoned while laying out mating environmental designs. Usually, row-column (RC) designs are practised when there are two sources of variability present in the experimental material and such situations are very common in agricultural, horticultural, laboratory and forestry trials.

For example, in forestry trials it is very common to compare a large number of hybrid varieties (which represent the treatments), which lead to a large experimental area, and it may be important to account for or eliminate the effects of fertility trends in the land in two directions as mentioned by Hinkelmann and Kempthorne (2005). In experimental situations, viz., a field trial where there is a dominant fertility gradient in the experimental field in one direction and the prevailing wind has a predictable direction perpendicular to the fertility gradient and a green house trial wherein pots are arranged in rows perpendicular to the screen wall, the rows of pots and distance of pots from the screen wall are the two sources of variability among the experimental pots and these two sources of variations other than due to combining alibility effects of lines are to be controlled.

As the number of lines increases, the number of crosses increases rapidly and hence designs, where the crosses arranged in a row-column set up, can be more advantageously used. It will enable the experimenter to get a more precise comparison among gca as well as sca effects.

Here, we propose to use the term MERC designs for combined Mating-Environmental Row-Column designs. These are designs where crosses are arranged in rows and columns in such a way that each row and each column contains the crosses at most once which ensures the balanced estimation of contrasts pertaining to gca effects free from sca effects.

In this paper, Section 2 deals with the orthogonal estimation of gca and sca effects considering a linear fixed effects model under a row-column setup, Section 3 deals with the construction some classes of MERC designs that are variance balanced for estimating the elementary contrast pertaining to gca effects, Section 4 gives a catalogue of designs with efficiency factor (in a range varying from 0.66 to 1.00) and Section 5 deals with an illustration using hypothetical data set.

2. METHODOLOGY

Consider a row-column setup with p rows and q columns. Let t be the number of inbred lines resulting in $v = \binom{t}{2}$ crosses of the form $i \times j$, $i < j$, $i, j = 1, 2, \ldots, t$ [Griffing's Method IV, Griffing (1956a)].

The row-column model for mating experiments can be expressed in the form,

$$y_{kl(m)} = \mu + \tau_{ij(m)} + \alpha_k + \beta_l + \mathbf{e}_{kl(m)}, \quad (l \le i, j \le t = \text{number of lines}) \tag{2.1}$$

where $y_{kl(m)}$ is the response from m^{th} cross ($m = 1, 2, \ldots, v$) in the $(kl)^{th}$ cell, μ is the grand mean, $\tau_{ij(m)}$ is the effect of m^{th} cross, α_k is the k^{th} row effect ($k = 1, 2, \ldots, p$), β_l is the l^{th} column effect ($l = 1, 2, \ldots, q$) and $e_{kl(m)}$ is iid $N(0, \sigma^2)$.

The model in equ.(2.1) can be rewritten as

$$\mathbf{Y} = \mu\mathbf{1} + \Delta'\tau + \mathbf{D}'_1\alpha + \mathbf{D}'_2\beta + \mathbf{e}, \tag{2.2}$$

where $\mathbf{Y}$ is a $n \times 1$ vector of observations, $\mathbf{1}$ is a $n \times 1$ vector of ones, Δ' is a $n \times v$ incidence matrix of observations versus cross, τ is a $v \times 1$ vector of cross effects, $\mathbf{D}'_1$ is a $n \times p$ incidence matrix of observations versus rows, α is a $p \times 1$ vector of row effects, $\mathbf{D}'_2$ is a $n \times q$ incidence matrix of observations versus columns, β is a $q \times 1$ vector of column effects and $\mathbf{e}$ is a $n \times 1$ vector of errors. Now, the design matrix $\mathbf{X}_{n\times(v+p+q+1)}$ can be partitioned into parameters of interest ($\mathbf{X}_1$) and nuisance parameters ($\mathbf{X}_2$).

$$\mathbf{X}_1 = [\Delta'],\ \mathbf{X}_2 = [\mathbf{1}\ \mathbf{D}'_1\ \mathbf{D}'_2]$$

The information matrix for τ can be obtained as $\mathbf{C}_\tau = \mathbf{X}'_1\mathbf{X}_1 - \mathbf{X}'_1\mathbf{X}_2(\mathbf{X}'_2\mathbf{X}_2)^- \mathbf{X}'_2\mathbf{X}_1$, where $v \times v$ matrix $\mathbf{C}$ is symmetric, non-negative definite with zero row and column sums. Hence the information matrix for estimating cross effects is obtained as,

$$\mathbf{C}_\tau = \mathbf{R}_\tau - \frac{1}{q}\mathbf{N}_r\mathbf{N}'_r - \frac{1}{p}\mathbf{N}_c\mathbf{N}'_c + \frac{1}{pq}\mathbf{r}\mathbf{r}',$$

where $\mathbf{R}_\tau = \text{diag}(r_1, r_2, \ldots, r_v)$; $\mathbf{r}' = (r_1, r_2, \ldots, r_v)$ is the $v \times 1$ replication vector of crosses with r_s ($s = 1, 2, \ldots, v$) being the number of times the s^{th} cross appears in the design; $\mathbf{N}_r$ is $v \times p$ incidence matrix of crosses versus rows and $\mathbf{N}_c$ is a $v \times q$ incidence matrix of crosses versus columns.

Now, the cross effect τ_{ij} can be expressed as,

$$\tau_{ij} = \bar{\tau} + g_i + g_j + s_{ij},\ (1 \le i < j \le t) \tag{2.3}$$

where $\bar{\tau}$ is the mean effect of crosses, $\{g_i\}$ denote the gca effects and $\{s_{ij}\}$ denote the sca effects such that

$$\sum_{i=1}^{t} g_i = 0, (1 \le i \le t) \tag{2.4}$$

$$\sum_{i=1}^{t-1}\sum_{j(>i)=2}^{t} s_{ij} = 0, (1 \le i \le j \le t) \text{ and } s_{ij} = s_{ji} \tag{2.5}$$

We arrange the crosses in the order (1, 2), (1, 3), …, (1, t), (2, 3), …, (2, t), …, (t – 1, t). Let $\mathbf{g} = (g_1, g_2, \ldots, g_t)'$ and let τ and $\mathbf{s}$ be $v \times 1$ vectors with elements $\{\tau_{ij}\}$ and $\{s_{ij}\}$ respectively. We follow Chai and Mukerjee (1999) and Das and Dey (2004) to express the general and specific combining abilities, *i.e.*, $\mathbf{g}$ and $\mathbf{s}$ in terms of τ. We define $\mathbf{Q}$ to be a $t \times v$ matrix with rows indexed by 1, 2, …, t and columns by the pairs (i, j), $(1 \le i < j \le t)$, such that the $\{u, (i, j)\}^{th}$ entry of $\mathbf{Q}$ is 1 if $u = i$ or j and zero, otherwise. We then have

$$\mathbf{QQ'} = (t-2)\mathbf{I}_t + \mathbf{J}_{tt},$$

$$(\mathbf{QQ'})^{-1} = (t-2)^{-1}\{\mathbf{I}_t - (2(t-1))^{-1}\mathbf{J}_{tt}\} \quad (2.6)$$

$$\mathbf{Q1}_v = (t-1)\mathbf{1}_t \text{ and } \mathbf{Q'1}_t = 2\mathbf{1}_v, \quad (2.7)$$

where for positive integers c and d, $\mathbf{I}_c$ is the c^{th} order identity matrix, $\mathbf{1}_c$ is the $c \times 1$ vector of all ones and $\mathbf{J}_{cd} = \mathbf{1}_c\mathbf{1}'_d$. Therefore (2.3) can be represented in matrix notation as

$$\tau = \bar{\tau}\mathbf{1}_v + \mathbf{Q'g} + \mathbf{s}, \quad (2.8)$$

where from (2.4) and (2.5), we have

$$\mathbf{1}'_t\mathbf{g} = 0 \text{ and } \mathbf{Qs} = 0 \quad (2.9)$$

Premultiplying (2.8) by $\mathbf{Q}$ and using (2.9), one has

$$\mathbf{g} = \mathbf{H}_1\tau \text{ and } \mathbf{s} = \tau - \bar{\tau}\mathbf{1}_v - \mathbf{Q'g} = \mathbf{H}_2\tau, \quad (2.10)$$

where

$$\mathbf{H}_1 = (\mathbf{QQ'})^{-1}\mathbf{Q} - (2v)^{-1}\mathbf{J}_{tv} = (t-2)^{-1}(\mathbf{Q} - 2t^{-1}\mathbf{J}_{tv}), \quad (2.11)$$

and

$$\mathbf{H}_2 = \mathbf{I}_v - \mathbf{Q'}(\mathbf{QQ'})^{-1}\mathbf{Q} = \mathbf{I}_v - (t-2)^{-1}(\mathbf{Q'Q} - 2(t-1)^{-1}\mathbf{J}_{vv}). \quad (2.12)$$

Since $\mathbf{H}_i\mathbf{1}_v = 0$, $\mathbf{H}_2\mathbf{1}_v = 0$, $\mathbf{H}_1\mathbf{H}'_2 = 0$,

Rank $(\mathbf{H}_1) = t - 1$ and Rank $(\mathbf{H}_2) = v - t$, (2.13)

it is clear that $\mathbf{g}$ and $\mathbf{s}$ represent treatment contrasts carrying $t - 1$ and $v - t$ degrees of freedom respectively and the contrasts representing $\mathbf{g}$ are orthogonal to those representing $\mathbf{s}$. But for $t = 3$, $\mathbf{s} = \mathbf{0}$ and hence t should be greater than 3.

Now, under the model (2.2), the joint information matrix for $\begin{pmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{pmatrix}\tau$ is given by

$$\mathbf{C} = \begin{bmatrix}\mathbf{H}_1\mathbf{C}_\tau\mathbf{H}'_1 & \mathbf{H}_1\mathbf{C}_\tau\mathbf{H}'_2\\ \mathbf{H}_2\mathbf{C}_\tau\mathbf{H}'_1 & \mathbf{H}_2\mathbf{C}_\tau\mathbf{H}'_2\end{bmatrix},$$

Following $\mathbf{Q}_4 = \begin{bmatrix}\mathbf{1}'_3 & \mathbf{0}'_3\\ \mathbf{I}_3 & \mathbf{M}\end{bmatrix}$ where $\mathbf{M} = \begin{bmatrix}1 & 1 & 0\\ 1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}$,

define $\mathbf{Q}_t = \begin{bmatrix}\mathbf{1}'_{t-1} & \mathbf{0}'_{v-t+1}\\ \mathbf{I}_{t-1} & \mathbf{Q}_{t-1}\end{bmatrix}$ for t lines and using (2.6) we obtain

$$\{(\mathbf{QQ'})^{-1}\mathbf{Q}\}_{ij} = \begin{cases}\dfrac{1}{t-1}, & \text{if } \{\mathbf{Q}\}_{ij} = 1\\ -\dfrac{1}{(t-1)(t-2)}, & \text{if } \{\mathbf{Q}\}_{ij} = 0\end{cases};$$

$i = 1, 2, \ldots, t$ and $j = 1, 2, \ldots, v$.

$$\text{Therefore, } \{\mathbf{H}_1\}_{ij} = \begin{cases}\dfrac{1}{t}, & \text{if } \{\mathbf{Q}\}_{ij} = 1\\ -\dfrac{2}{t(t-2)}, & \text{if } \{\mathbf{Q}\}_{ij} = 0\end{cases}$$

Before discussing the methods of construction of designs, a trivial case (rows as well as columns are complete with respect to crosses) is explained here.

2.1 The Case of a Trivial Class of Designs

Consider a Latin Square design with $v = \binom{t}{2}$ rows, v columns and v crosses, each cross replicated v times. The crosses are arranged such that each cross appears in each row and in each column once. Following is the arrangement:

1 × 2	1 × 3	…	$(t-2) \times (t-1)$	$(t-1) \times t$
1 × 3	1 × 4	…	$(t-1) \times t$	1 × 2
. . .				
$(t-1) \times t$	1 × 2	…	$(t-3) \times (t-2)$	$(t-2) \times (t-1)$

For this class of MERC designs, $\mathbf{C}_\tau = \dfrac{t(t-1)}{2}\left[\mathbf{I} - \dfrac{2\mathbf{J}}{t(t-1)}\right]$ and the information matrix for estimating gca is $\mathbf{C}_{gca} = \dfrac{t(t-1)}{2(t-2)}\left[\mathbf{I} - \dfrac{\mathbf{J}}{t}\right]$. The variance of estimated elementary treatment contrast pertaining to gca effects is given by

$$V(\hat{g}_i - \hat{g}_j) = \frac{4(t-2)}{t(t-1)}\sigma^2; i \neq j = 1, 2, \ldots, t.$$

Note: As it is difficult to get a general expression for the information matrix for sca effects ($\mathbf{C}_{sca}$), variance of estimated elementary contrasts pertaining to sca effects can be worked out empirically by writing a contrast of interest using computed $\mathbf{C}_{sca}$.

Example 2.1.1: For t = 4, the MERC design (before randomization of rows and columns) with parameters $v = p = q = r = 6$ is as follows:

1 × 2	1 × 3	1 × 4	2 × 3	2 × 4	3 × 4
1 × 3	1 × 4	2 × 3	2 × 4	3 × 4	1 × 2
1 × 4	2 × 3	2 × 4	3 × 4	1 × 2	1 × 3
2 × 3	2 × 4	3 × 4	1 × 2	1 × 3	1 × 4
2 × 4	3 × 4	1 × 2	1 × 3	1 × 4	2 × 3
3 × 4	1 × 2	1 × 3	1 × 4	2 × 3	2 × 4

For this design, $V(\hat{g}_i - \hat{g}_j) = 0.6667\sigma^2$.

For this design, the variance for comparing sca effects of (1 × 2) with (1 × 3) is $V(\hat{s}_{1\times 2} - \hat{s}_{1\times 3}) = 0.1667\sigma^2$.

Remark: Deleting the last row (column) from the above class of design, will result in incomplete MERC design with parameters $v = \frac{t(t-1)}{2}$, $p = \frac{(t-2)(t+1)}{2}$, $q = \frac{t(t-1)}{2}$ and $r = \frac{(t-2)(t+1)}{2}$.

For this class of Incomplete MERC designs,

$$\mathbf{C}_\tau = \frac{t(t-1)(t^2-t-4)}{2(t-2)(t+1)}\left[\mathbf{I} - \frac{2\mathbf{J}}{t(t-1)}\right],$$

and the information matrix for estimating gca is

$$\mathbf{C}_{gca} = \frac{(t-1)^2(t^2-t-4)}{2(t-2)^2(t+1)}\left[\mathbf{I} - \frac{\mathbf{J}}{t}\right].$$

The variance of elementary contrast pertaining to gca effects is given by

$$V(\hat{g}_i - \hat{g}_j) = \frac{4(t-2)^2(t+1)}{(t-1)^2(t^2-t-4)}\sigma^2; i \neq j = 1, 2, ..., t.$$

3. METHOD OF CONSTRUCTION

Method 3.1: *t* prime

Obtain an array of size $(t-2) \times t$ by taking an initial row 1, 2, ..., t and cyclically generating the columns by taking a difference of one. Generate a second array using the same initial row by taking a difference of two. Obtain $(t-1)/2$ such arrays by taking difference upto $(t-1)/2$. Now append each array to the right of the initial array. This will result in an array of size $(t-2) \times t(t-1)/2$. Generate crosses from each column of the array by taking all possible distinct crosses using the lines in each column, which result in a MERC design with parameters $v = \frac{t(t-1)}{2}$, $p = \frac{(t-3)(t-2)}{2}$, $q = \frac{t(t-1)}{2}$ and $r = \frac{(t-3)(t-2)}{2}$.

For this class of MERC designs, considering columns as blocks, the crosses form a PBB design with triangular association scheme and the information matrix for estimating gca is $\mathbf{C}_{gca} = \frac{t(t-3)(t-4)}{2(t-2)^2}\left[\mathbf{I} - \frac{\mathbf{J}}{t}\right]$.

The variance of elementary treatment contrast pertaining to gca effects is given by

$$V(\hat{g}_i - \hat{g}_j) = \frac{4(t-2)^2}{t(t-3)(t-4)}\sigma^2; i \neq j = 1, 2, ..., t.$$

Example 3.1.1: For t = 5, we obtain two arrays of size 3 × 5 each as follows:

Array I					Array II				
1	2	3	4	5	1	2	3	4	5
2	3	4	5	1	3	4	5	1	2
3	4	5	1	2	5	1	2	3	4

By forming all possible distinct crosses within each column, the MERC design with parameters v = 10, p =3, q = 10 and r = 3 can be obtained as:

1 × 2	2 × 3	3 × 4	4 × 5	5 × 1	1 × 3	2 × 4	3 × 5	4 × 1	5 × 2
1 × 3	2 × 4	3 × 5	4 × 1	5 × 2	1 × 5	2 × 1	3 × 2	4 × 3	5 × 4
2 × 3	3 × 4	4 × 5	5× 1	1 × 2	3 × 5	4 × 1	5 × 2	1 × 3	2 × 4

The crosses in the above design follow a two associate class triangular association scheme as shown below:

*	1 × 2	1 × 3	1 × 4	1× 5
1 × 2	*	2 × 3	2 × 4	2 × 5
1 × 3	2 × 3	*	3 × 4	3 × 5
1 × 4	2 × 4	3 × 4	*	4 × 5
1× 5	2 × 5	3 × 5	4 × 5	*

Here, the crosses in the same row and column are first associates and rests are second associates.

For this design, $V(\hat{g}_i - \hat{g}_j) = 3.6\sigma^2$.

For this design, the variance of the estimated elementary contrast for comparing sca effects with one common parent is $V(\hat{s}_{1\times2} - \hat{s}_{1\times5}) = 0.5\sigma^2$ and with no common parent is $V(\hat{s}_{1\times2} - \hat{s}_{3\times4}) = 0.25\sigma^2$.

Method 3.2: $t = 2^m$ (m >1)

For a given t, consider the following crosses as given below:

1 × 2	1 × 3	…	1 × (t – 1)	1 × t

Now, rearrange the order of the position of the crosses. Firstly, starting from the first position of the above row, group the crosses present in positions at distance 2 together, and then take crosses at distance 4 together, and then crosses at distances 8 together, and so on upto crosses at distances (t/2) + 1 together as shown below: *i.e.*,

1 × 2	1 × 4	…	1 × t	1 × 3	1 × 7	…	1 × (t–1)	…	1 × [(t/2)+1]

Consider this row as the initial column. Now, generate the (t/2)–1 more columns by using the following steps:

Step 1: Generate an array of one's of order (t–1)× (t/2–1)

Step 2: In the first 2^{m-1} rows of the array, 2^0 must be added to every entry, in next 2^{m-2} rows, 2^1 must be added to every $(2^{m-2})^{th}$ column. Similarly, in the next 2^{m-3} rows, 2^2 must be added to every $(2^{m-3})^{th}$ column and so on upto the $(t–2)^{th}$ row.

An array of size (t – 1) × t/2 is obtained by adding each column of the above array, taking one column at a time, to both the parental lines in the initial column and taking modulus t. Now, rotating the columns cyclically (t/2) – 1 times, by one position at a time, and appending the arrays one below the other an array of order $\frac{t(t-1)}{2} \times t/2$. Obtain another array of the same size, from the previous array, starting from the second row and ending with the first row by rotating the rows circularly. Append this new array to the right hand side of the previous array [*i.e.*, i^{th} and $(i + 1)^{th}$ rows of the previous array are appended side by side taking mod(p)]. This will result in an array of order $\frac{t(t-1)}{2} \times t$, which is a MERC design with parameters $v = \frac{t(t-1)}{2}$, $p = \frac{t(t-1)}{2}$, $q = t$ and $r = t$.

For this class of MERC designs, the crosses are partially balanced and and the information matrix for estimating gca is $\mathbf{C}_{gca} = \frac{t}{(t-2)}\left[\mathbf{I} - \frac{\mathbf{J}}{t}\right]$.

The variance of elementary treatment contrasts pertaining to gca effects is given by

$$V(\hat{g}_i - \hat{g}_j) = \frac{2(t-2)}{t}\sigma^2; i \neq j = 1, 2, ..., t.$$

Example 3.2.1: For t = 8, m = 3 and number of crosses is 28. The initial column is arranged as follows:

1 × 2
1 × 4
1 × 6
1 × 8
1 × 3
1 × 7
1 × 5

The array generated at the end of Step 2 is of order 7 × 3

2	2	2
2	2	2
2	2	2
2	2	2
1	3	1
1	3	1
1	1	1

Adding this array taking one column at a time to both the parental lines in the initial column (taking modulus 8) will result in the following array of order 7 × 4

1 × 2	3 × 4	5 × 6	7 × 8
1 × 4	3 × 6	5 × 8	7 × 2
1 × 6	3 × 8	5 × 2	7 × 4
1 × 8	3 × 2	5 × 4	7 × 6
1 × 3	2 × 4	5 × 7	6 × 8
1 × 7	2 × 8	5 × 3	6 × 4
1 × 5	2 × 6	3 × 7	4 × 8

The array of order 28 × 4 obtained after rotating the columns cyclically 3 times, by one position at a time, and appending below the array is

1 × 2	**3 × 4**	**5 × 6**	**7 × 8**
1 × 4	**3 × 6**	**5 × 8**	**7 × 2**
1 × 6	**3 × 8**	**5 × 2**	**7 × 4**
1 × 8	**3 × 2**	**5 × 4**	**7 × 6**
1 × 3	**2 × 4**	**5 × 7**	**6 × 8**
1 × 7	**2 × 8**	**5 × 3**	**6 × 4**
1 × 5	**2 × 6**	**3 × 7**	**4 × 8**
3 × 4	5 × 6	7 × 8	1 × 2
3 × 6	5 × 8	7 × 2	1 × 4
3 × 8	5 × 2	7 × 4	1 × 6
3 × 2	5 × 4	7 × 6	1 × 8
2 × 4	5 × 7	6 × 8	1 × 3
2 × 8	5 × 3	6 × 4	1 × 7
2 × 6	3 × 7	4 × 8	1 × 5
5 × 6	7 × 8	1 × 2	3 × 4
5 × 8	7 × 2	1 × 4	3 × 6
5 × 2	7 × 4	1 × 6	3 × 8
5 × 4	7 × 6	1 × 8	3 × 2
5 × 7	6 × 8	1 × 3	2 × 4
5 × 3	6 × 4	1 × 7	2 × 8
3 × 7	4 × 8	1 × 5	2 × 6
7 × 8	1 × 2	3 × 4	5 × 6
7 × 2	1 × 4	3 × 6	5 × 8
7 × 4	1 × 6	3 × 8	5 × 2
7 × 6	1 × 8	3 × 2	5 × 4
6 × 8	1 × 3	2 × 4	5 × 7
6 × 4	1 × 7	2 × 8	5 × 3
4 × 8	1 × 5	2 × 6	3 × 7

Rotating the rows circularly, starting from the second row and ending with the first row, and appending to the above array as discussed in the method leads to MERC design with parameters $v = 28$, $p = 28$, $q = 8$ and $r = 8$ is obtained as follows:

1 × 2	3 × 4	5 × 6	7 × 8	1 × 4	3 × 6	5 × 8	7 × 2
1 × 4	3 × 6	5 × 8	7 × 2	1 × 6	3 × 8	5 × 2	7 × 4
1 × 6	3 × 8	5 × 2	7 × 4	1 × 8	3 × 2	5 × 4	7 × 6
1 × 8	3 × 2	5 × 4	7 × 6	1 × 3	2 × 4	5 × 7	6 × 8
1 × 3	2 × 4	5 × 7	6 × 8	1 × 7	2 × 8	5 × 3	6 × 4
1 × 7	2 × 8	5 × 3	6 × 4	1 × 5	2 × 6	3 × 7	4 × 8
1 × 5	2 × 6	3 × 7	4 × 8	3 × 4	5 × 6	7 × 8	1 × 2
3 × 4	5 × 6	7 × 8	1 × 2	3 × 6	5 × 8	7 × 2	1 × 4
3 × 6	5 × 8	7 × 2	1 × 4	3 × 8	5 × 2	7 × 4	1 × 6
3 × 8	5 × 2	7 × 4	1 × 6	3 × 2	5 × 4	7 × 6	1 × 8
3 × 2	5 × 4	7 × 6	1 × 8	2 × 4	5 × 7	6 × 8	1 × 3
2 × 4	5 × 7	6 × 8	1 × 3	2 × 8	5 × 3	6 × 4	1 × 7
2 × 8	5 × 3	6 × 4	1 × 7	2 × 6	3 × 7	4 × 8	1 × 5
2 × 6	3 × 7	4 × 8	1 × 5	5 × 6	7 × 8	1 × 2	3 × 4
5 × 6	7 × 8	1 × 2	3 × 4	5 × 8	7 × 2	1 × 4	3 × 6
5 × 8	7 × 2	1 × 4	3 × 6	5 × 2	7 × 4	1 × 6	3 × 8
5 × 2	7 × 4	1 × 6	3 × 8	5 × 4	7 × 6	1 × 8	3 × 2
5 × 4	7 × 6	1 × 8	3 × 2	5 × 7	6 × 8	1 × 3	2 × 4
5 × 7	6 × 8	1 × 3	2 × 4	5 × 3	6 × 4	1 × 7	2 × 8
5 × 3	6 × 4	1 × 7	2 × 8	3 × 7	4 × 8	1 × 5	2 × 6
3 × 7	4 × 8	1 × 5	2 × 6	7 × 8	1 × 2	3 × 4	5 × 6
7 × 8	1 × 2	3 × 4	5 × 6	7 × 2	1 × 4	3 × 6	5 × 8
7 × 2	1 × 4	3 × 6	5 × 8	7 × 4	1 × 6	3 × 8	5 × 2
7 × 4	1 × 6	3 × 8	5 × 2	7 × 6	1 × 8	3 × 2	5 × 4
7 × 6	1 × 8	3 × 2	5 × 4	6 × 8	1 × 3	2 × 4	5 × 7
6 × 8	1 × 3	2 × 4	5 × 7	6 × 4	1 × 7	2 × 8	5 × 3
6 × 4	1 × 7	2 × 8	5 × 3	4 × 8	1 × 5	2 × 6	3 × 7
4 × 8	1 × 5	2 × 6	3 × 7	1 × 2	3 × 4	5 × 6	7 × 8

For this design, $V(\hat{g}_i - \hat{g}_j) = 1.5\sigma^2$.

For this design, the variance for comparing sca effects of (1×2) with (1×3) is $V(\hat{s}_{1\times 2} - \hat{s}_{1\times 3}) = 0.2530\sigma^2$.

Remark: It is also possible to construct MERC designs for all even values of t. For that, one has to first generate an initial array of size $(t–1) \times t/2$ which consists of all $t(t–1)/2$ crosses arranged in such way that in each row, each line should occur exactly once.

Obtain another array of the same size, from the previous array, starting from the $\left(\frac{t}{2}+1\right)^{th}$ row and ending with the row by rotating the $\left(\frac{t}{2}\right)^{th}$ rows circularly. Append this new array to the right hand side of the previous array [*i.e.*, i^{th} and $\left(\frac{t}{2}+i\right)^{th}$ rows of the previous array are appended side by side taking mod(p)]. This leads to a MERC design with parameters $v=\frac{t(t-1)}{2}$, $p=\frac{t(t-1)}{2}$, $q=t$ and $r = t$.

For this class of designs, the characterization properties are the same as those for the series for $t = 2^m$ ($m > 1$).

Example 3.2.2: Let $t = 6$

Initial array

1 × 2	3 × 4	5 × 6
1 × 3	2× 5	4 × 6
1 × 4	2 × 6	3 × 5
1 × 5	2 × 4	3 × 6
1 × 6	2 × 3	4 × 5

For the given example, MERC design with parameters $v = 15$, $p =15$, $q = 6$ and $r = 6$

1 × 2	3 × 4	5 × 6	1 × 3	2× 5	4 × 6
3 × 4	5 × 6	1 × 2	2× 5	4 × 6	1 × 3
5 × 6	1 × 2	3 × 4	4 × 6	1 × 3	2× 5
1 × 3	2× 5	4 × 6	1 × 4	2 × 6	3 × 5
2× 5	4 × 6	1 × 3	2 × 6	3 × 5	1 × 4
4 × 6	1 × 3	2× 5	3 × 5	1 × 4	2 × 6
1 × 4	2 × 6	3 × 5	1 × 5	2 × 4	3 × 6
2 × 6	3 × 5	1 × 4	2 × 4	3 × 6	1 × 5
3 × 5	1 × 4	2 × 6	3 × 6	1 × 5	2 × 4
1 × 5	2 × 4	3 × 6	1 × 6	2 × 3	4 × 5
2 × 4	3 × 6	1 × 5	2 × 3	4 × 5	1 × 6
3 × 6	1 × 5	2 × 4	4 × 5	1 × 6	2 × 3
1 × 6	2 × 3	4 × 5	1 × 2	3 × 4	5 × 6
2 × 3	4 × 5	1 × 6	3 × 4	5 × 6	1 × 2
4 × 5	1 × 6	2 × 3	5 × 6	1 × 2	3 × 4

For this design, $V(\hat{g}_i - \hat{g}_j) = 1.33\sigma^2$.

For this design, the variance for comparing *sca* effects of (1 × 2) with (1 × 3) is $V(\hat{s}_{1\times2} - \hat{s}_{1\times3}) = 0.3167\sigma^2$.

Macros have been developed using PROC IML of SAS software for generation of various classes of MERC designs and is given in the Appendix.

4. LIST OF DESIGNS

Considering a row-column model, the per cross canonical efficiency of these designs as compared to an orthogonal design with same number of crosses and same number of replications has been studied by developing a SAS code in PROC IML assuming same σ^2 for developed MERC design and the orthogonal design to which it is compared and listed along with other parameters in Table 4.1 (Method 3.1) and Table 4.2 (Method 3.2).

Table 4.1. List of designs obtained under Method 3.1

S.N.	t	v	p	q	r	$\bar{V}_{(i,i)}$	$E_{canonical}$
1	5	10	3	10	3	3.60	0.7018
2	7	21	10	21	10	1.19	0.9397
3	11	55	36	55	36	0.56	0.9899
4	13	78	55	78	55	0.41	0.9944
5	17	136	105	136	105	0.29	0.9978
6	19	171	136	171	136	0.25	0.9985

Table 4.2. List of designs obtained under Method 3.2

S.N.	t	v	p	q	r	$\bar{V}_{(i,i)}$	$E_{canonical}$
1	4	6	6	4	4	1.00	0.8823
2	6	15	15	6	6	1.33	0.7778
3	8	28	28	8	8	1.50	0.7297
4	10	45	45	10	10	1.60	0.7021
5	12	66	66	12	12	1.67	0.6842
6	14	91	91	14	14	1.71	0.6716
7	16	120	120	16	16	1.75	0.6623

Here, t = Number of lines, v = Total number of crosses, p = Number of rows, q = Number of columns,

r = Number of replications, $\bar{V}_{(i,i)}\sigma^2$ = Variance of the estimated elementary contrasts pertaining to general combining ability effects and $E_{canonical}$ = Canonical efficiency factor. $E_{canonical}$ of the MERC design is computed relative to an orthogonal design with the same number of crosses by working out the harmonic mean of (1/r) times the non-zero eigen values of the information matrix pertaining to cross effects. Here, r is the number of replications of the crosses and it is assumed that σ^2 is the same for the developed MERC design and the orthogonal design to which it is compared.

It can be seen that the efficiency of the designs constructed through Method 3.1 is quite high and those constructed through Method 3.2 is fairly high.

5. ILLUSTRATION

Illustration using hypothetical data set has been given in Appedix. A programme has been written using SAS IML for doing the analysis of data obtained from MERC designs. User should enter the data in the format given in section 'SAS code for the analysis' to get the analysis done. This can be done by printing the 'xx' in the program. User can also get the data file in Excel format by submitting the given code:

```
ods html body = 'data.xls';
print xx;
ods html close;
```

Data file consists of five columns where in the first column of data file represents row number, second column represents the column number, third is representing line 1, fourth column representing line 2 and fifth column representing the cross number. Once the code is submitted through a SAS software, a MSWORD file named 'anova' will be created. User can save /open the file where one can find all the results.

6. CONCLUSIONS

Diallel and partial diallel crosses are frequently used in breeding programmes for estimating genetic components of total variance of a quantitative character. The interest of the experimenter lies in the orthogonal estimation of the general combining abilities free from specific combining ability effects. Some methods of construction of efficient MERC designs for CDC have been given. The developed designs would help the experimenter to estimate the contrasts pertaining to the general combining ability effects free of specific combining ability effects when two cross classified sources of variations are present in the experimental material. All the designs lead to a balanced estimation of contrasts pertaining to the general combining ability effects when specific combining ability effects are included in the model.

ACKNOWLEDGEMENTS

Authors are grateful to the Editor and the referee for their constructive comments which helped in improving the quality of the manuscript. The facilities provided by Director, ICAR-Indian Agricultural Statistics Research Institute to carry out the research are duly acknowledged.

REFERENCES

Chai, F.S. and Mukerjee, R. (1999). Optimal designs for diallel crosses with specific combining abilities. *Biometrika*, **86(2)**, 453-458.

Choi, K.C., Chatterjee, K., Das, A. and Gupta, S. (2002). Optimality of orthogonally blocked diallels with specific combining abilities. *Statist. Prob. Lett.*, **57**, 145-150.

Choi, K.C., Gupta, S. and Kageyama, S. (2004). Designs for diallel crosses for test versus control comparisons. *Utilitas Mathematica*, **65**, 167-180.

Curnow, R.N. (1963). Sampling the diallel cross. *Biometrics*, **19**, 287-306.

Das, A. and Dey, A. (2004). Designs for diallel cross experiments with specific combining abilities. *J. Ind. Soc. Agril. Statist.*, **57** (Special volume), 247-256.

Das, A., Dey, A. and Dean, A.M. (1998). Optimal designs for diallel cross experiments. *Statist. Prob. Lett.*, **36**, 427-436.

Das, A. and Ghosh, D.K. (1999). Balanced incomplete block diallel cross designs. *Statist. Appl.*, **1**, 1-16.

Dey, A. (2010). *Incomplete Block Designs*. Hindustan Book Agency, New Delhi.

Dey, A. and Midha, C.K. (1996). Optimal block designs for diallel crosses. *Biometrika*, **83(2)**, 484-489.

Griffing (1956a). Concept of general and specific combining ability in relation to diallel crossing system. *Austral. J. Biol. Sci.*, **9**, 463-493.

Griffing (1956b). A generalized treatment of diallel crosses in quantitative inheritance. *Heredity*, **10**, 31-50.

Gupta, S. and Choi, K.C. (1998). Optimal row-column designs for complete diallel crosses. *Comm. Statist. – Theory Methods*, **27(11)**, 2827-2835.

Gupta, S. and Kageyama, S. (1994). Optimal complete diallel crosses. *Biometrika*, **81**,420-424.

Gupta, S., Das, A. and Kageyama, S. (1994). Single replicate orthogonal block designs for circulant partial diallel crosses. *Comm. Statist. – Theory Methods*, **24**, 2601-2607.

Hayman, B.I. (1954 a). The analysis of variance of diallel tables. *Biometrics*, **10**, 235-244.

Hayman, B.I. (1954 b). The theory and analysis of diallel crosses. *Genetics*, **39**, 789-809.

Hayman, B.I. (1958). The theory and analysis of diallel crosses. II. *Genetics*, **43**, 63-85.

Hayman, B.I. (1960). The theory and analysis of diallel crosses. III. *Genetics*, **45**, 155-172.

Hinkelmann, K. (1975). Design of genetical experiments. In: *A Survey of Statistical Design and Linear Models*, Ed. J.N. Srivastava, 243-269, Amsterdam, North Holland.

Hinkelmann, K. and Kempthorne, O. (2005). *Design and Analysis of Eperiments: Advanced Design of Experiments* (Volume II), John Wiley and Sons, Inc., Hoboken, New Jersey.

Hsu, Yug-Feng and Ting, Chao-Ping (2005). A-Optimal and efficient diallel cross experiments for comparing test treatments with a control. *Statist. Prob. Lett.*, **71(1)**, 99-110.

Kempthorne, O. (1956). The theory of the diallel cross. *Genetics*, **41**, 451-459.

Kempthorne, O. (1969). Introduction to Genetic Statistics. Iowa State University Press, Iowa.

Mukerjee, R. (1997). Optimal partial diallel crosses. *Biometrika*, **84(4)**, 939-948.

Narain, P. (1990). *Statistical Genetics*. Wiley Eastern Ltd., New Delhi.

Parsad, R., Gupta, V.K and Srivastava, R. (1999). Universally optimal block designs for diallel crosses. *Stats. Appl.*, **1(1)**, 35-52.

Parsad, R., Gupta, V.K. and Gupta, S.C. (2005): Optimal designs for experiments on two-line and four-line crosses. *Utilitas Mathematica*, **68**, 11-32.

Schmidt, J. (1919). La valeur de l'individu a titre de generateur appreciee suivant la methode du croisement diallele. *Compt. Rend. Lab. Carlsberg*, **14(6)**, 1-33.

Sharma, M.K. and Fanta, Sileshi (2010). Variance balanced designs for complete diallel cross. *J. Agric. Biotech. Sustain. Dev.*, **2(4)**, 56-60.

Singh, R.K. and Choudhary, B.D. (2004). *Biometrical Methods in Quantitative Genetic Analysis*. Kalyani Publishers, New Delhi.

Sprague, G.F. and Tatum, L.A. (1942). General vs. specific combing ability in single crosses of corn. *J. Amer. Soc. Agron.*, **34**, 923-932.

Xiang, Bin and Li, Bailian (2001). A new mixed analytical method for genetic analysis of diallel data. *Canad. J. Forest Res.*, **31(12)**, 2252-2259

APPENDIX

SAS Macro for generation of complete MERC

```
%let t=5;/*number of lines*/
proc iml; c_no=comb(&t,2);
n_no=&t; k=1; d0=j(c_no,3,0);
do i=1 to n_no-1;
do j=i+1 to n_no;
d0[k,1]=i;
d0[k,2]=j;
d0[k,3]=k;
k=k+1;
end;
end;
print d0; /*d0 is a matrix with three columns viz., line1 line2 and cross number*/
d1=j((c_no*c_no),3,0);
do l=1 to c_no;
do i=1 to c_no;
if i+(l-1)<(c_no+1) then do;
d1[((l-1)*c_no)+i, ]=d0[(l-1)+i, ];
end;
else do;
d1[((l-1)*c_no)+i, ]=d0[(i+(l-1))-c_no, ];
end;
end;
end;
print d1; /*d1 is a matrix with three columns viz., line1 line2 and cross number for the entire design*/
k=1;
x=j((c_no*c_no),2,0);
do i=1 to c_no;
do j=1 to c_no;
x[k,1]=i;
x[k,2]=j;
k=k+1;
end;
end;
*print x; /*x is a matrix with two columns representing row numbers and column numbers for the entire design*/

/****************code for generating designs*******/
ww=j(c_no,c_no,0);
k=1;
do i=1 to c_no;
do j=1 to c_no;
ww[i,j]=d1[k,1];
k=k+1;
end;
end;
*print ww;
ww1=j(c_no,c_no,0);
```

```
k=1;
do i=1 to c_no;
do j=1 to c_no;
ww1[i,j]=d1[k,2];
k=k+1;
end;
end;
*print ww1;
ww_=char(ww,4,0);
ww1_=char(ww1,4,0);
www=j(nrow(ww),ncol(ww),' x');
design=ww_+www+ww1_;
*print design; /***************************************************/
/*********Randmization*****************************/

/*****row-randomization*******/
r=j(1,nrow(ww),0);
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
ra=j(nrow(ww),ncol(ww),0);
random_row=char(ra,10,0);
do i=1 to nrow(ww);
do j=1 to ncol(ww);
random_row[i,j]= design[rr[i],j];
end;
end;
*print random_row;
/*******/

/******column-randomization****/
r=j(ncol(ww),1,0);
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
randomization=char(ra,10,0);
do i=1 to nrow(ww);
do j=1 to ncol(ww);
randomization[i,j]= random_row[i,rr[j]];
end;
end;
print randomization; /*it will give a randomized layout of the design*/
/******/
```

SAS Macro for generation of designs using Method 3.1

```
%let t=5;/*t should be prime*/
proc iml;
```

```
c_no=comb(&t,2);
n_no=&t;
k=1;
d0=j(n_no-2,((n_no-1)/2)*n_no,0);
do k=1 to (n_no-1)/2;
do i=1 to n_no-2;
do j=1 to n_no;
d0[i,j+((k-1)*n_no)]=mod((j+(i-1)+(i-1)*(k-1)),n_no);
if d0[i,j+((k-1)*n_no)]=0 then d0[i,j+((k-1)*n_no)]=n_no;
end;
end;
end;
*print d0[format=3.0]; /*d0 is creating the arrays given in Example 3.1*/
d1=j((n_no-2)*(n_no-3)/2,(n_no-1)*n_no,0);
do j=1 to (n_no-1/2)*n_no;
l=1;
do i=1 to (n_no-2);
do q=1 to (n_no-2)-i;
d1[l,(2*j)-1]=d0[i,j];
d1[l,(2*j)]=d0[i+q,j];
l=l+1;
end;
end;
end;
*print d1; /*d1 is creating all the possible pairs for each of the columns in d0 (in two consecutive columns)*/
d2=j(((n_no-2)*(n_no-3)/2)*c_no,2,0);
l=1;
do j=1 to ((n_no-1)*n_no)/2;
do i=1 to (n_no-2)*(n_no-3)/2;
d2[l,1]=d1[i,(2*j)-1];
d2[l,2]=d1[i,2*j];
l=l+1;
end;
end;
*print d2; /*d2 is arranging the crosses generated d1 in two columns (column 1 represents line1 and column
 two represents line 2*/
k=1;
d00=j(c_no,3,0);
do i=1 to n_no-1;
do j=i+1 to n_no;
d00[k,1]=i;
d00[k,2]=j;
d00[k,3]=k; k=k+1;
end;
end;
print d00; /*d0 is a matrix with three columns viz., line1 line2 and cross number*/
d22=j(((n_no-2)*(n_no-3)/2)*c_no,1,0);
do i=1 to ((n_no-2)*(n_no-3)/2)*c_no;
do k=1 to c_no;
```

```
if (d2[i,1]=d00[k,1] & d2[i,2]=d00[k,2])|(d2[i,1]=d00[k,2] & d2[i,2]=d00[k,1])
then d22[i,1]=d00[k,3];
end;
end;
print d22; /*d22 is the corresponding cross numbers of crosses in d2 (Eg: line 1 line 2 = cross 1 as per d00)*/
x=j(((n_no-2)*(n_no-3)/2)*c_no,2,0);
k=1;
do j=1 to c_no;
do i=1 to ((n_no-2)*(n_no-3)/2);
x[k,1]=i;
x[k,2]=j;
k=k+1;
end;
end;
print x; /*x is a matrix with two columns representing row numbers and column numbers for the entire design*/
/*****************code for generating designs*******/
ww=j(((n_no-2)*(n_no-3)/2),c_no,0);
k=1;
do j=1 to c_no;
do i=1 to ((n_no-2)*(n_no-3)/2);
ww[i,j]=d2[k,1];
k=k+1;
end;
end;
*print ww;
ww1=j(((n_no-2)*(n_no-3)/2),c_no,0);
k=1;
do j=1 to c_no;
do i=1 to ((n_no-2)*(n_no-3)/2);
ww1[i,j]=d2[k,2];
k=k+1;
end;
end;
*print ww1;
ww_=char(ww,4,0);
ww1_=char(ww1,4,0);
www=j(nrow(ww),ncol(ww),' x');
design=ww_+www+ww1_; *print design;
/**************************************************/
/*********Randmization****************************/

/*****row-randomization*******/
r=j(1,nrow(ww),0);
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
ra=j(nrow(ww),ncol(ww),0);
random_row=char(ra,10,0);
```

```
do i=1 to nrow(ww);
do j=1 to ncol(ww);
random_row[i,j]= design[rr[i],j];
end;
end;
*print random_row;
/*******/ /

/******column-randomization****/
r=j(ncol(ww),1,0);
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
randomization=char(ra,10,0);
do i=1 to nrow(ww);
do j=1 to ncol(ww);
randomization[i,j]= random_row[i,rr[j]];
end;
end;
*print randomization; /*it will give a randomized layout of the design*/

/******/
```

SAS Macro for generation of designs using Method 3.2

```
%let t=8;/*when t is a power of 2*/
proc iml;
p=log(&t)/log(2);
*print p;
aa=j(p,1,0);
aa1=j(p,1,0);
do i=1 to p;
aa[i,1]=2**(i-1);
aa1[i,1]=2**(p-i);
end;
*print aa aa1; /*These two vectors have been defined to generate the numbers starting from 2^0, 2^1,2^2,...,
2^(p-1) in ascending and descending order*/
aaa=j(p,(&t/2)-1,1);
do i=1 to (p-1);
do j=1 to (&t/2)-1;
if mod(j,aa[i,1])=0 then
aaa[i,j]=(aa[i,1]+1);
end;
end;
aaaa=j(&t-1,(&t/2)-1,0);
l=1;
do i=1 to p;
do k=1 to aa1[i,1];
aaaa[l, ]=aaa[i, ];
l=l+1;
```

```
end;
end;
*print aaaa [format=3.0]; /*This matrix is creating the array given in Example 3.2*/
k=1;
a1=j(&t/2,&t,0);
do i= 1 to &t/2;
a1[i,1]=k;
a1[i,2]=2*i;
end;
a2=j((&t/2)-1,&t,0);
l=1;
do i=2 to p;
do j=1 to aa1[i,1];
a2[l,1]=k;
a2[l,2]=(2*(aa[i,1]*(j-1)))+aa[i,1]+1;
l=l+1;
end;
end;
a=a1//a2; /* The first two columns represents the line 1 and line 2 given in first array of Example 3.2*/
do i=1 to (&t/2)-1;
do j=1 to 2;
a[,((2*i)+j)]=mod((a[,((2*i)+j)-2]+aaaa[,i]),&t);
end;
end;
do i=1 to &t-1;
do j=1 to &t;
if a[i,j]=0 then a[i,j]=&t;
end;
end;
rsuma=a[,+];
*print a[format=3.0]; /* This array represents third array of Example 3.2*/
x_=j((&t/2)*(&t-1),&t,0);
do k=1 to &t/2;
do i=1 to (&t-1);
do j=1 to &t;
if j+(2*(k-1))>&t then
                do;
                x_[i+((&t-1)*(k-1)),j]=a[i,j+(2*(k-1))-&t];
                end;
                else do;
                x_[i+((&t-1)*(k-1)),j]=a[i,j+(2*(k-1))];
                end;
end;
end;
end;
x_1=j(nrow(x_)-1,ncol(x_),0);
do i=2 to nrow(x_);
x_1[i-1, ]=x_[i, ];
end;
```

```
x_1=x_1//x_[1, ];
x_=x_||x_1;
c_no=comb(&t,2);
n_no=&t;
d00=j((c_no*&t),2,0);
l=1;
do j=1 to &t;
do i=1 to c_no;
do k=1 to 2;
d00[l,k]=x_[i,(2*(j-1))+k];
end;
l=l+1;
end;
end;
k=1;
d0=j(c_no,3,0);
do i=1 to c_no;
d0[k,1]=d00[i,1];
d0[k,2]=d00[i,2];
d0[k,3]=i;
k=k+1;
end;
d01=j((c_no*&t/2),1,0);
l=1;
do j=1 to &t/2;
do i=1 to c_no;
if (&t-1)*(j-1)+i < c_no+1 then
                                        do;
                                        d01[l,1]=(&t-1)*(j-1)+i;
                                        end;
                                        else do;
                                        d01[l,1]=(&t-1)*(j-1)+i-c_no;
                                        end;
l=l+1;
end;
end;
d02=j((c_no*&t/2),1,0);
do i=1 to c_no*&t/2;
d02[i,1]=mod(d01[i, ]+1,(&t*(&t-1)/2));
if d02[i,1]=0 then d02[i,1]=(&t*(&t-1)/2);
end;
d1=d00||(d01//d02);
*print d1; /*d1 represents the an array containing three columns line1 line 2 and cross number for the entire design*/
k=1;
x=j((c_no*&t),2,0);
do j=1 to &t;
do i=1 to c_no;
x[k,1]=i;
x[k,2]=j;
```

```
k=k+1;
end;
end;
*print x;/*x is a matrix with two columns representing row numbers and column numbers for the entire design*/
/*****************code for generating designs*******/
ww=j(c_no,&t,0);
k=1;
do j=1 to &t;
do i=1 to c_no;
ww[i,j]=d1[k,1];
k=k+1;
end;
end;
*print ww;
ww1=j(c_no,&t,0);
k=1;
do j=1 to &t;
do i=1 to c_no;
ww1[i,j]=d1[k,2];
k=k+1;
end;
end;
*print ww1;
ww_=char(ww,4,0);
ww1_=char(ww1,4,0);
www=j(nrow(ww),ncol(ww),' x');
design=ww_+www+ww1_;
*print design;
/**************************************************/                                /
*********Randmization****************************/

/*****row-randomization*******/
r=j(1,nrow(ww),0);
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
ra=j(nrow(ww),ncol(ww),0);
random_row=char(ra,10,0);
do i=1 to nrow(ww);
do j=1 to ncol(ww);
random_row[i,j]= design[rr[i],j];
end;
end;
*print random_row;
/*******/

/******column-randomization****/
r=j(ncol(ww),1,0);
```

```
call randgen(r,'uniform');
*print r;
rr=rank(r);
*print rr;
randomization=char(ra,10,0);
do i=1 to nrow(ww);
do j=1 to ncol(ww);
randomization[i,j]= random_row[i,rr[j]];
end;
end;
print randomization; /*it will give a randomized layout of the design*/
/******/
```

SAS code for the analysis

```
/*SAS code for the analysis */
Data MERC;
input row column line1 line2 cross yld;
cards;
1   1   1   2   1   31
1   2   1   3   2   58
1   3   1   4   3   76
1   4   1   5   4   101
1   5   2   3   5   118
1   6   2   4   6   21
1   7   2   5   7   137
1   8   3   4   8   156
1   9   3   5   9   173
1   10  4   5   10  193
2   1   1   3   2   57
2   2   1   4   3   78
2   3   1   5   4   99
2   4   2   3   5   120
2   5   2   4   6   81
2   6   2   5   7   88
2   7   3   4   8   157
2   8   3   5   9   179
2   9   4   5   10  193
2   10  1   2   1   113
3   1   1   4   3   75
3   2   1   5   4   99
3   3   2   3   5   116
3   4   2   4   6   81
3   5   2   5   7   146
3   6   3   4   8   106
3   7   3   5   9   178
3   8   4   5   10  197
3   9   1   2   1   120
3   10  1   3   2   146
4   1   1   5   4   100
```

4	2	2	3	5	120
4	3	2	4	6	81
4	4	2	5	7	150
4	5	3	4	8	168
4	6	3	5	9	131
4	7	4	5	10	200
4	8	1	2	1	128
4	9	1	3	2	148
4	10	1	4	3	168
5	1	2	3	5	117
5	2	2	4	6	81
5	3	2	5	7	146
5	4	3	4	8	168
5	5	3	5	9	189
5	6	4	5	10	149
5	7	1	2	1	127
5	8	1	3	2	152
5	9	1	4	3	166
5	10	1	5	4	189
6	1	2	4	6	18
6	2	2	5	7	86
6	3	3	4	8	104
6	4	3	5	9	129
6	5	4	5	10	147
6	6	1	2	1	16
6	7	1	3	2	91
6	8	1	4	3	110
6	9	1	5	4	127
6	10	2	3	5	146
7	1	2	5	7	134
7	2	3	4	8	155
7	3	3	5	9	176
7	4	4	5	10	198
7	5	1	2	1	125
7	6	1	3	2	91
7	7	1	4	3	160
7	8	1	5	4	182
7	9	2	3	5	195
7	10	2	4	6	158
8	1	3	4	8	152
8	2	3	5	9	176
8	3	4	5	10	194
8	4	1	2	1	125
8	5	1	3	2	149
8	6	1	4	3	109
8	7	1	5	4	181
8	8	2	3	5	199
8	9	2	4	6	156
8	10	2	5	7	223

```
9    1    3    5    9    173
9    2    4    5    10   194
9    3    1    2    1    121
9    4    1    3    2    149
9    5    1    4    3    167
9    6    1    5    4    130
9    7    2    3    5    198
9    8    2    4    6    160
9    9    2    5    7    221
9    10   3    4    8    241
10   1    4    5    10   193
10   2    1    2    1    123
10   3    1    3    2    147
10   4    1    4    3    169
10   5    1    5    4    190
10   6    2    3    5    149
10   7    2    4    6    161
10   8    2    5    7    227
10   9    3    4    8    241
10   10   3    5    9    264
;
ods rtf file='anova.doc' startpage=no;
ods output overallanova=total_ss lsmeans=lsmeans;
proc glm data=MERC;
class row column cross;
model yld=row column cross/ss3;/*Type I sum of squares (ss1 instead of ss3)should be used for designs where columns/rows are incomplete with respect to crosses*/
lsmeans cross;
run;
ods output close;
proc iml;
use merc;
read all into xx;
*print xx;
row_no=nrow(xx);
row=xx[1:row_no,1];
column=xx[1:row_no,2];
cross=xx[1:row_no,3]||xx[1:row_no,4];
row_c=nrow(cross); /*number of rows in the matrix 'cross'*/
col_c=ncol(cross);/*number of columns in the matrix 'cross'*/
*print cross;
sca0=xx[1:row_no,5];
*print sca0;
n_no=max(xx[1:row_no,4]);/*number of lines*/
c_no=max(xx[1:row_no,5]);/*number of crosses*/
m=j(row_c,1,1);
/*print cross;*/
x11=j(row_c,max(cross),0);
k=1;
```

```
do i=1 to row_c;
do j=1 to col_c;
if cross[i,j]>0 then
x11[k,cross[i,j]]=x11[k,cross[i,j]]+1;
end;
k=k+1;
end;
*print x11;
sca=j(row_c,c_no,0);
k=1;
do i=1 to row_c;
if sca0[i, ]>0 then sca[k,sca0[i, ]]=1;
k=k+1;
end;
*print sca[format=3.0];
rep=sca‘*sca;
row_1=j(nrow(row),max(row),0);/*design matrix for observations VS rows*/
k=1;
do i=1 to nrow(row);
if row[i, ]>0 then
row_1[k,row[i, ]]=1;
k=k+1;
end;
*print row_1;
col_1=j(nrow(column),max(column),0);/*design matrix for observations VS columns */
k=1;
do i=1 to nrow(column);
if column[i, ]>0 then
col_1[k,column[i, ]]=1;
k=k+1;
end;
*print col_1;
x1=sca;
x2=m||row_1||col_1;
c=(x1‘*x1)-(x1‘*x2)*ginv(x2‘*x2)*(x2‘*x1); /*information matrix for cross effects*/
*print c[format=3.2];
q1=x11‘*sca;
q=q1/rep[1,1];/*Q matrix (line VS cross matrix) please refer to the paper*/
qq=q*q‘;
inv_qq=inv(qq);
qqq=inv_qq*q;
jpv=j(n_no,c_no,1);
jpv1=jpv*(.5/c_no);
h1=(inv_qq*q)-jpv1;
h2=i(c_no)-(q‘*inv_qq*q);
c11=h1*c*h1‘; /*information matrix for GCA effects*/
c22=h2*c*h2‘; /*information matrix for SCA effects*/
*print c11;
cross_t=j(max(xx[,5]),1,0);/*Vector of cross totals*/
```

```
do j=1 to max(xx[,5]);
do i= 1 to nrow(xx);
if xx[i,5]=j then cross_t[j,1]=cross_t[j,1]+xx[i,6];
end;
end;
*print cross_t;
row_t=j(max(xx[,1]),1,0);/*Vector of row totals*/
do j=1 to max(xx[,1]);
do i= 1 to nrow(xx);
if xx[i,1]=j then row_t[j,1]=row_t[j,1]+xx[i,6];
end;
end;
*print row_t;
col_t=j(max(xx[,2]),1,0);/*Vector ofcolumn totals */
do j=1 to max(xx[,2]);
do i= 1 to nrow(xx);
if xx[i,2]=j then col_t[j,1]=col_t[j,1]+xx[i,6];
end;
end;
*print col_t;
n1=sca‘*row_1;
n2=sca‘*col_1;
m=row_1‘*col_1;
k1=row_1‘*row_1;
k2=col_1‘*col_1;
DF_gca=max(xx[,4])-1;/*Degrees of freedom for gca effects*/
DF_sca=max(xx[,5])-max(xx[,4]);/*Degrees of freedom for sca effects*/
adj_cross=cross_t-(n1*inv(k1)*row_t)-(n2-n1*inv(k1)*m)*ginv(k2-m‘*inv(k1)*m)*(col_t-m‘*inv(k1)*row_t);/
*Adjusted cross totals*/
*print adj_cross;
SS_gca=adj_cross‘*h1‘*ginv(c11)*h1*adj_cross;/* sum of suares due to gca*/
SS_sca=adj_cross‘*h2‘*ginv(c22)*h2*adj_cross;/* sum of suares due to sca*/
*ss_cross=ss_gca+ss_sca;
*print ss_gca ss_sca;
MS_gca=ss_gca/df_gca;/* Mean square due to gca*/
MS_sca=ss_sca/df_sca;/* Mean square due to sca*/
*print ms_gca ms_sca;
use total_ss;
read all var{DF SS MS FValue ProbF} into y;
*print y;
FValue_gca=ms_gca/y[2,3];/*calculated value of F for testing gca effects*/
FValue_sca=ms_sca/y[2,3];/*calculated value of F for testing sca effects*/
*print fvalue_gca fvalue_sca;
FtabValue_gca=finv(1-.025,df_gca,y[2,1]);/*Table value of F for testing gca effects*/
FtabValue_sca=finv(1-.025,df_sca,y[2,1]);/*Table value of F for testing sca effects*/
*print FtabValue_gca FtabValue_sca;
ProbF_gca=1-probf(fvalue_gca,df_gca,y[2,1]);/* Significance of gca effects*/
if probf_gca<.00001 then probf_gca='<.0001';
ProbF_sca=1-probf(fvalue_sca,df_sca,y[2,1]);/* Significance of sca effects*/
```

```
if probf_sca<.00001 then probf_sca='<.0001';
*print probf_gca probf_sca;
print df_gca ss_gca ms_gca fvalue_gca FtabValue_gca probf_gca;
print df_sca ss_sca ms_sca fvalue_sca FtabValue_sca probf_sca;
use lsmeans;
read all into yy;
grand_mean=sum(xx[,6])/nrow(xx);
cross_est=yy-grand_mean;
*print cross_est;
gca_effects=h1*cross_est;
sca_effects=h2*cross_est;
Line_number=j(nrow(gca_effects),1,0);
do i=1 to nrow(gca_effects);
Line_number[i, ]=i;
end;
Cross_number=j(nrow(sca_effects),1,0);
do i=1 to nrow(sca_effects);
Cross_number[i, ]=i;
end;
gca_effects=Line_number||gca_effects;
sca_effects=Cross_number||sca_effects;
print gca_effects ;
print sca_effects;
zero=j(1,nrow(gca_effects)-2,0);
p1={1 -1};
p11=p1||zero;/* contrasts for pair-wise comparison of gca (or sca )effects*/
var=(p11*ginv(c11)*p11')*y[2,3];
Std_Error_Diff_gi_gj=sqrt(var);
print Std_Error_Diff_gi_gj;/*Standard Error of Difference of gca effects*/
TValue=tinv(1-.025,y[2,1]);/*Table value of T for C.D. Values*/
Critical_Difference=std_error_diff_gi_gj*TValue;/*Critical differece for pair-wise comparison*/
print Critical_Difference;
ods rtf close;
quit;
```

Output

The GLM Procedure

Class Level Information		
Class	**Levels**	**Values**
row	10	1 2 3 4 5 6 7 8 9 10
column	10	1 2 3 4 5 6 7 8 9 10
cross	10	1 2 3 4 5 6 7 8 9 10

Number of Observations Read	100
Number of Observations Used	100

The GLM Procedure
Dependent Variable: yld

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	27	240034.6700	8890.1730	10975.5	<.0001
Error	72	58.3200	0.8100		
Corrected Total	99	240092.9900			

R-Square	Coeff Var	Root MSE	yld Mean
0.999757	0.633758	0.900000	142.0100

Source	DF	Type III SS	Mean Square	F Value	Pr > F
row	9	81172.89000	9019.21000	11134.8	<.0001
column	9	79584.09000	8842.67667	10916.9	<.0001
cross	9	79277.69000	8808.63222	10874.9	<.0001

Least Squares Means

cross	yld LSMEAN
1	102.900000
2	118.800000
3	127.800000
4	139.800000
5	147.800000
6	99.800000
7	155.800000
8	164.800000
9	176.800000
10	185.800000

DF_gca	SS_gca	MS_gca	FValue_gca	FtabValue_gca	ProbF_gca
4	66188.973	16547.243	20428.695	2.9693207	<.0001

DF_sca	SS_sca	MS_sca	FValue_sca	FtabValue_sca	ProbF_sca
5	13088.717	2617.7433	3231.7819	2.7483203	<.0001

gca_effects	
1	–26.24667
2	–20.58
3	13.386667
4	3.3866667
5	30.053333

sca_effects	
1	7.7166667
2	–10.35
3	8.65
4	–6.016667
5	12.983333
6	–25.01667
7	4.3166667
8	6.0166667
9	–8.65
10	10.35