Variety Trials in Two–Dimensional Layouts

A thesis presented to
The University of Waikato
in fulfilment of the requirement for the degree
of

Doctor of Philosophy

by

Carole Wright

The University of Waikato
Te Whare Wānanga o Waikato

2004

"Luck is the residue of design."

Branch Rickey

GM of the Brooklyn Dodgers and Pittsburgh Pirates

(1881 – 1965)

Abstract

This thesis shows that efficient resolvable row–column designs can be constructed quickly by using properties of both the contraction and the dual design. The method of constructing the contraction of a r–replicate resolvable row–column design is described and is an extension of the research undertaken by Bailey and Patterson (1991) and Jarrett, Piper and Wild (1997). The structural properties of the contraction are investigated, including the row and column incidence matrices. It is shown that the information matrix for the dual design can be expressed in terms of the row and column incidence matrices for the contraction.

A connection is made between the eigenvalues of the resolvable row–column design and the dual design. This relationship can be used to enable the average efficiency factor of the resolvable row–column design to be expressed in terms of the canonical efficiency factors of the dual design.

Existing optimisation algorithms available for the construction of resolvable row–column designs work well for small experiments. For experiments with a large number of treatments they are computationally expensive. The dimensionality of the problem of constructing resolvable row–column designs is reduced by working with the dual design. By expressing the information matrix for the dual design in terms of the contraction, the result is a computationally faster algorithm. For large designs it is shown to be quicker to update properties of the contraction than those of the resolvable row–column design. The computational effort required for generating resolvable row–column designs with a large number of treatments can be reduced further by using the (M,S)–optimality criterion.

When generating resolvable row–column designs using an optimisation algorithm, it is helpful to have a tight upper bound for the average efficiency factor. By comparing the average efficiency factor with the upper bound it is possible to determine when a good design has been obtained and a decision can then be made on whether to terminate the algorithm. Upper bounds for resolvable row–column designs are known not to be tight and a new upper bound is developed by exploiting

some of the properties of the contraction and dual design.

Acknowledgements

After venturing across the Tasman and experiencing the 'real' world for more than four years this exciting opportunity was presented. Besides Australia having all the major sporting trophies I was never in any doubt that it was time to return home. I would like to thank Prof. J.A. (Nye) John, Ass. Prof. David Whitaker and Dr Emlyn Williams for giving me the opportunity and support to fulfil one of my lifetime goals.

Also a big thank you and hug to my sister Fiona and my brother–in–law Chris for their support over the past three years. Thank you for opening your home as a place of refuge from the roller coaster ride that this research has taken me on.

I must also thank my nieces Jordon and Myshka. Both have offered unconditional love and, when needed, given me a fresh outlook on life.

Finally, thank you to family and friends for their continuous encouragement.

This research was supported by a grant from the Marsden Fund.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Resolvable Row–Column Designs

Researchers conducting experiments are constantly seeking more efficient use of their resources. In agricultural and forestry trials, scientists want to compare a large number of varieties while achieving greater precision in their variety comparisons. More recently, with the development of gene expression microarrays, geneticists want to study gene expression for thousands of genes at one time. In these types of situations, a design which controls variability in two directions is likely to be more efficient than any one–dimensional blocking structure. It may also be desirable to arrange the treatments (for example varieties) in groups such that each treatment occurs exactly once within each group. Designs incorporating this two–dimensional blocking structure and grouping feature are known as *resolvable row–column designs.*

A resolvable row–column design has $v = ks$ treatments set out in r replicates each with k rows and s columns. The property of resolvability requires that each of the v treatments occurs exactly once within each of the r replicates. The design in Table 1.1 is an example of a $r = 3$ replicate resolvable row–column design for $v = 15$ treatments, $k = 3$ rows and $s = 5$ columns. It can be seen that this design is resolvable, as the treatments numbered 1 to 15 occur exactly once within each replicate.

Row–column designs consist of two block designs. The block design given by the rows of the row–column design is referred to as the *row component design.* Similarly, the block design given by the columns is called the *column component design.* The

Table 1.1: Resolvable row–column design for $v = 15$, $k = 3$, $s = 5$, $r = 3$

Replicate		1					2					3				
Column		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
Row	1	3	13	11	4	10	2	6	4	12	14	7	15	1	6	10
	2	15	6	14	5	8	3	8	9	10	1	8	13	12	3	14
	3	12	9	7	1	2	7	11	15	5	13	4	2	11	5	9

row component design in Table 1.1 has $v = 15$ treatments with $rk = 9$ blocks of size $s = 5$, and the column component design has $v = 15$ treatments with $rs = 15$ blocks of size $k = 3$.

Designs of the same size can be compared using the *average efficiency factor* E which gives a measure of the precision with which treatment contrasts are estimated. The higher the value of E, the better the design. The average efficiency factor and how it is calculated are discussed in Section 1.2.

The average efficiency factor is a suitable design criterion but it can be computationally expensive to calculate. A simpler and computationally cheaper design criterion, which can be used in conjunction with the average efficiency factor, is provided by the *(M,S)-optimality criterion* (Shah, 1960; Eccleston and Hedayat, 1974). When a design is found which improves the (M,S)–optimality objective function, the average efficiency factor is then calculated. This approach is discussed in Section 1.3.

Existing optimisation algorithms are able to quickly construct efficient resolvable row–column designs for small experiments, but are often slow at constructing large designs. For example, it takes CycDesigN version 2.0 (Whitaker, Williams and John, 2002) over 10 minutes to find an initial solution for a four–replicate design with 9000 treatments laid out in 100 rows and 90 columns. Experimenters who often plan many large experiments need algorithms that will rapidly produce efficient designs. The overall aim of this research is to develop methods for quickly generating highly efficient resolvable row–column designs that meet the needs and requirements of experimenters. This means taking into account the practical features of the experiment with regard to the way the treatments are to be grouped and the size of the experiment. The aim is to have flexible methods of design construction, combined with improved computer optimisation techniques.

In the early stages of this project three different approaches were investigated for generating resolvable row–column designs, of which two were unsuccessful. One unsuccessful approach generated a class of resolvable row–column designs which will be called α–α–designs. These designs are an extension of α–designs (Patterson and Williams, 1976a) and use one α–array to generate the columns of the design and a second α–array to generate the rows. This approach is briefly outlined in Section 1.4. The second unsuccessful approach is briefly outlined in Section 1.5 and attempted to merge two one–dimensional block designs to form the dual design of an efficient resolvable row–column design. A third approach which aimed at exploring relationships based on the contractions of resolvable row–column designs proved successful and forms the focus of this research.

1.2 Average Efficiency Factor

The precision with which treatment effects are estimated will often vary for different designs of the same size. The aim is to construct a design in which every treatment contrast is estimated as precisely as possible, under the assumption that all contrasts are of equal interest. An appropriate criterion to find such a design is to minimise the average variance, $\bar{v}\sigma^2$ say, of all pairwise differences, where σ^2 is the error variance. The average efficiency factor of the design is obtained by comparing the average variance for the design with that of a design with the same replication per treatment and no blocking structure. In the absence of blocking the average variance is $2\sigma^2/r$. Assuming the error variance is the same for both designs, the average efficiency factor E is defined as

$$E = \frac{2\sigma^2/r}{\bar{v}\sigma^2} = \frac{2}{r\bar{v}}$$

John and Williams (1995, p30) show that the average efficiency factor is given by the harmonic mean of the non–zero eigenvalues of the matrix $\mathbf{A}/r$, where $\mathbf{A}$ is the information matrix for the design. The non–zero eigenvalues are called the canonical efficiency factors of the design and will be denoted by $e_1, e_2, \ldots, e_n$, where $n = \text{rank}(\mathbf{A})$; see Pearce, Calinski and Marshall (1974). If the design is *connected* then all treatment contrasts are estimable and $n = (v-1)$. This research is concerned only with connected resolvable row–column designs and the average efficiency factor for such designs is defined as

$$E = \frac{v-1}{\sum_{i=1}^{v-1} e_i^{-1}} \tag{1.1}$$

A design whose average efficiency factor is at least as large as any other design of the same size is called *A–optimal* (Kiefer, 1959).

In computer search algorithms it is computationally expensive to calculate the canonical efficiency factors of the design. An alternative expression to (1.1) for E requires the calculation of the Moore–Penrose inverse of $\mathbf{A}$. For a resolvable row–column design the information matrix $\mathbf{A}$ is given by

$$\mathbf{A} = r\mathbf{I}_v - \tfrac{1}{s}\mathbf{N}_k\mathbf{N}_k' - \tfrac{1}{k}\mathbf{N}_s\mathbf{N}_s' + \tfrac{r}{v}\mathbf{J}_{vv} \tag{1.2}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{N}_k$ and $\mathbf{N}_s$ are the incidence matrices of the row and column component designs respectively and $\mathbf{J}_{vv}$ is a $v \times v$ matrix of ones; see John and Williams (1995, p107).

The $v \times v$ information matrix $\mathbf{A}$ with rank n $(n < v)$ can be written in canonical form as

$$\mathbf{A} = r\sum_{i=1}^{n} e_i\mathbf{x}_i\mathbf{x}_i' \tag{1.3}$$

where $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ are the orthonormal eigenvectors of $\mathbf{A}$ corresponding to the non–zero eigenvalues, e_i is the non–zero eigenvalue of $\mathbf{A}/r$ corresponding to $\mathbf{x}_i$, and $\mathbf{x}_i'\mathbf{x}_j = 1$ if $i = j$, else $\mathbf{x}_i'\mathbf{x}_j = 0$. The Moore–Penrose inverse of $\mathbf{A}$, $\mathbf{A}^+$, is defined as

$$\mathbf{A}^+ = \frac{1}{r}\sum_{i=1}^{n} e_i^{-1}\mathbf{x}_i\mathbf{x}_i' \tag{1.4}$$

Since trace($\mathbf{x}_i'\mathbf{x}_i$)=trace($\mathbf{x}_i\mathbf{x}_i'$) $= 1$, it follows that

$$\text{trace}(\mathbf{A}^+) = \frac{1}{r}\sum_{i=1}^{n} e_i^{-1}\text{trace}(\mathbf{x}_i\mathbf{x}_i') = \frac{1}{r}\sum_{i=1}^{n} e_i^{-1}$$

Therefore the average efficiency factor in (1.1) can be written as

$$E = \frac{v-1}{r\ \text{trace}(\mathbf{A}^+)} \tag{1.5}$$

It is computationally less expensive to invert a matrix than to find the eigenvalues and eigenvectors of the matrix. Therefore, rather than calculating $\mathbf{A}^+$ using (1.4) the following calculation is performed

$$\mathbf{A}^+ = (\mathbf{A} + \mathbf{Z}\mathbf{Z}')^{-1} - \mathbf{Z}\mathbf{Z}' \tag{1.6}$$

where $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_{v-n})$ is of full column rank such that $\mathbf{A}\mathbf{z}_j = \mathbf{0}$ $(j = 1, 2, \ldots, v-n)$. Since $\mathbf{z}_j'\mathbf{x}_i = 0$, it follows that

$$(\mathbf{A} + \mathbf{Z}\mathbf{Z}')\mathbf{x}_i = \mathbf{A}\mathbf{x}_i + \mathbf{Z}\mathbf{Z}'\mathbf{x}_i = \mathbf{A}\mathbf{x}_i$$

and as $\mathbf{z}_j'\mathbf{z}_k = 1$ if $j = k$, otherwise $\mathbf{z}_j'\mathbf{z}_k = 0$

$$(\mathbf{A} + \mathbf{Z}\mathbf{Z}')\mathbf{z}_j = \mathbf{A}\mathbf{z}_j + \mathbf{Z}\mathbf{Z}'\mathbf{z}_j = \mathbf{z}_j$$

Therefore, using (1.3)

$$(\mathbf{A} + \mathbf{Z}\mathbf{Z}') = r\sum_{i=1}^{n} e_i\mathbf{x}_i\mathbf{x}_i' + \sum_{j=1}^{v-n} \mathbf{z}_j\mathbf{z}_j'$$

so that $(\mathbf{A} + \mathbf{Z}\mathbf{Z}')$ is non-singular. Hence

$$(\mathbf{A} + \mathbf{Z}\mathbf{Z}')^{-1} = \frac{1}{r}\sum_{i=1}^{n} e_i^{-1}\mathbf{x}_i\mathbf{x}_i' + \sum_{j=1}^{v-n} \mathbf{z}_j\mathbf{z}_j' = \mathbf{A}^+ + \mathbf{Z}\mathbf{Z}'$$

which gives (1.6).

As an example, for a connected resolvable row–column design rank$(\mathbf{A}) = (v-1)$, where $\mathbf{A}$ is given in (1.2). The eigenvector corresponding to the zero eigenvalue is $\mathbf{1}_v$, therefore

$$\mathbf{A}^+ = (\mathbf{A} + \mathbf{J})^{-1} - \mathbf{J}$$

and

$$\text{trace}(\mathbf{A}^+) = \text{trace}[(\mathbf{A} + \mathbf{J})^{-1}] - v$$

1.3 (M,S)–Optimality

In searching for optimal or near optimal r–replicate resolvable row–column designs, it remains computationally expensive to maximise the average efficiency factor. The search can be simplified by finding the set of designs which minimise the (M,S)–optimality criterion (Shah, 1960; Eccleston and Hedayat, 1974), which is a two stage procedure. Firstly, the M–optimality step finds the subclass of designs that maximises $\sum_i e_i$. Then, within this subclass, those designs that minimise $\sum_i e_i^2$ are obtained. This is the S–optimality step. For resolvable designs of a given size, $\sum_i e_i$

is fixed, and by minimising $\sum_i e_i^2$ the variance of the canonical efficiency factors is being minimised (John and Williams, 1995, p35). This also equates to minimising trace($\mathbf{A}^2/r^2$), or equivalently minimising trace($\mathbf{A}^2$). Using (1.2), trace($\mathbf{A}^2$) for resolvable row–column designs is given by

$$\begin{aligned}\text{trace}(\mathbf{A}^2) &= \text{trace}[\tfrac{1}{s^2}(\mathbf{N}_k\mathbf{N}_k')^2 + \tfrac{1}{k^2}(\mathbf{N}_s\mathbf{N}_s')^2 + \tfrac{2}{v}\mathbf{N}_k\mathbf{N}_k'\mathbf{N}_s\mathbf{N}_s')] + \\ &\quad r^2v - 2r^2k - 2r^2s - r^2 \\ &= \text{trace}(\mathbf{W}^2) + r^2(v - 2k - 2s - 1)\end{aligned}$$

where $\mathbf{W} = \frac{1}{s}\mathbf{N}_k\mathbf{N}_k' + \frac{1}{k}\mathbf{N}_s\mathbf{N}_s'$. Hence, for resolvable row–column designs, the (M,S)–optimality objective function of minimising trace($\mathbf{A}^2$) is equivalent to minimising trace($\mathbf{W}^2$).

The subclass of (M,S)–optimal designs will, in general, still contain a large number of designs, so it is necessary to apply further optimality criteria to this subclass. It has been conjectured that A–optimal designs can be found within the subclass of (M,S)–optimal designs; see John and Williams (1982). The average efficiency factor could be calculated for the designs within the subclass of (M,S)–optimal designs in order to find the best designs. In terms of a search algorithm, the average efficiency factor would only be calculated if the (M,S)–optimality objective function value decreased or was equal to the best value.

1.4 α–α–Designs

Traditionally, resolvable block designs, in which blocks are grouped so that treatments occur exactly once in each group, have been used to control variability in one direction. For this purpose Patterson and Williams (1976a) introduced a flexible class of resolvable block designs known as α–designs. These designs are based on cyclic methods of construction and can be generated quickly. The average efficiency factor can be easily computed, and for designs with more than 1000 treatments, it is generally at least 99% of a theoretical upper bound for a resolvable design (John, Ruggiero and Williams, 2002).

It was investigated whether resolvable row–column designs could be constructed more efficiently using α–designs. An efficient α–design could be used to construct the column component design, and a second α–design used to form the row component design. These designs will be called α–α–designs.

An α–design for the column component is constructed from a $k \times r$ α–array whose elements are in the set of residues modulo s. A further $(s-1)$ columns can be generated from each column of the α–array by cyclic substitution, that is, adding one to the elements in each column and reducing modulo s as necessary. The resulting $k \times rs$ array is called the intermediate array. The α–design can be obtained by adding 1 to each element in row 1 of the intermediate array, $(s+1)$ to the elements in row 2, $(2s+1)$ to all elements in row 3, and in general, adding $[(i-1)s+1]$ to each element in the ith row of the intermediate array $(i = 1, 2, \ldots, k)$. Each set of columns generated from the same column in the α–array constitutes a replicate. The columns within each replicate of the α–design are the blocks and by changing an element in the α–array a different α–design can be generated.

Example

Given the α–array in Table 1.2 an α–design for three replicates of 12 treatments with blocks of size four can be constructed. The intermediate array, obtained by cycling the elements of the α–array, is given in Table 1.2 and the resulting α–design is in Table 1.3.

Table 1.2: α–array and intermediate array for $v = 12$, $k = 4$, $r = 3$

α–array			Intermediate array								
0	2	0	0	1	2	2	0	1	0	1	2
2	1	0	2	0	1	1	2	0	0	1	2
1	1	1	1	2	0	1	2	0	1	2	0
1	2	2	1	2	0	2	0	1	2	0	1

As a row–column design, the row component design of the α–design in Table 1.3 is clearly disconnected as the same three treatments occur together in a row within each of the three replicates. For instance, treatments 1, 2 and 3 occur in row 1 of each replicate, and treatments 4, 5 and 6 occur in row 2 of all three replicates. In order to improve the row component design, the treatments within each column

Table 1.3: α–design

Replicate	1			2			3		
Block	1	2	3	1	2	3	1	2	3
	1	2	3	3	1	2	1	2	3
	6	4	5	5	6	4	4	5	6
	8	9	7	8	9	7	8	9	7
	11	12	10	12	10	11	12	10	11

need to be permuted. This is achieved by using an $r \times s$ α–array whose elements determine the extent of the cyclic rotation of the treatments within each column. The (ij)th element of the $r \times s$ α–array is applied to column j in replicate i. A 0 in the α–array results in the column remaining unchanged. A 1 in the α–array will cause the treatment in row 2 to move up to row 1, the treatment in row 3 to move to row 2, and so on. The treatment in row 1 will be forced down to row k. In general, an x in the α–array will result in the treatment in row i moving to row $(p+1)$, where $p = (i + k - x - 1)$ modulo k. The resulting design is an α–α–design, which is a subclass of resolvable row–column designs.

Example (continued)

The row component of the resolvable row–column design in Table 1.3 can be improved by applying the cyclic permutations given by the α–array in Table 1.4. The resulting resolvable row–column design is given in Table 1.5.

Table 1.4: $r \times s$ α–array

1	0	0
2	2	3
0	3	2

The first row of the α–array in Table 1.4 gives the cyclic permutation to be applied to the three columns of replicate 1. The entry in column 1 of the first row of the α–array is 1, therefore the entries in column 1 of replicate 1 will shift up by one place. A 0 in the α–array means that the order of the treatments in the column are

Table 1.5: Resolvable row–column design

Replicate		1			2			3		
Column		1	2	3	1	2	3	1	2	3
Row	1	6	2	3	8	9	11	1	10	7
	2	8	4	5	12	10	2	4	2	11
	3	11	9	7	3	1	4	8	5	3
	4	1	12	10	5	6	7	12	9	6

to remain unchanged, hence columns 2 and 3 of replicate 1 remain the same. The second row of the α–array is applied to the three columns of replicate 2, and the third row is applied to replicate 3.

A computer algorithm was developed to construct resolvable row–column designs using a two stage method. At the first stage changes are made to the elements of the $k \times r$ α–array until an efficient column component design is generated. The row component design is then improved by perturbing elements in the $r \times s$ α–array to enable cyclic rotations of the treatments within each column. This approach was compared to an algorithm which produces the same column component design at the end of the first stage, but during the second stage allows for the random interchange of treatments within a column, rather than cyclic rotations within a column. For a given parameter set, each algorithm was run numerous times with different random number seeds and the best average efficiency factor obtained is given in Table 1.6.

This approach of applying two α–arrays to form a resolvable row–column design can generate designs quickly in two stages. This is due to the number of treatment interchanges being made per iteration. However, as the full search space of designs is often not explored, near optimal and optimal row–column designs are harder to locate. The average efficiency factors of the α–α–designs, $E_{\alpha-\alpha}$, were much lower than those for the designs produced from the algorithm using random treatment interchanges.

Various other forms of randomising the treatments within the blocks of the α–design were considered. These included using an α_2–array (John *et al.*, 2002) to improve the row component design of the α–design. α_2–arrays were also applied

Table 1.6: $E_{\alpha-\alpha}$ and E_{random} for resolvable row–column designs

v	k	s	r	$E_{\alpha-\alpha}$	E_{random}
18	6	3	3	0.5030	0.5418
40	8	5	3	0.6527	0.6717
60	6	10	4	0.7297	0.7357
120	10	12	4	0.8022	0.8069
225	15	15	3	0.8339	0.8376

at the second stage to the blocks of a resolvable block design which was not an α–design. Of the approaches considered, none were capable of generating resolvable row–column designs with an average efficiency factor sufficiently close to that of the method of randomly interchanging treatments to warrant further research.

1.5 Merging Block Designs

The average efficiency factor for the resolvable row–column design has a monotonically increasing relationship with the average efficiency factor of the dual design (John and Williams, 1995, p41). Given an efficient dual design, the corresponding resolvable row–column design will also be efficient. Investigations were aimed at whether it was possible to combine two efficient block designs to produce an efficient dual design, and hence, an efficient resolvable row–column design.

The dual design for a r–replicate resolvable row–column design with v treatments consists of v blocks of size r, with each plot containing two factors. The first factor has rk levels replicated s times and the second factor has rs levels replicated k times.

The two block designs to be merged are given by the two factors in the dual design. The first block design represents the first factor of the dual design, and the second block design represents the second factor. In order to form a valid dual design there are several constraints on the structure of each block design. For the first block design, the first entry in each block must contain an element from the set $\{1, 2, \ldots, k\}$. The second entry in each block must be from the set $\{k+1, k+2, \ldots, 2k\}$. In general, the ith entry in each block of the first design must be from the set $\{(i-1)k+1, (i-1)k+2, \ldots, ik\}$ $(i = 1, 2, \ldots, r)$. Similarly, the

jth entry in each block of the second block design must contain an element from the set $\{(j-1)s+1, (j-1)s+2, \ldots, js\}$ $(j = 1, 2, \ldots, r)$. A further constraint is that every element in the r sets for the first block design must occur s times in the design, and each element in the r sets for the second block design must occur k times.

Example

The dual design of a three–replicate resolvable row–column design with 12 treatments set out in 3 rows and 4 columns can be formed by merging two block designs. Two possible block designs are given in Tables 1.7 and 1.8. The first block design is constructed using the elements from the sets $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8\}$ and $\{9, 10, 11, 12\}$. Each block must contain one element from each of the three sets and the 12 elements must be replicated three times in the block design. The second block design contains elements from the sets $\{1, 2, 3\}$, $\{4, 5, 6\}$ and $\{7, 8, 9\}$. These 9 elements must be replicated four times and each block must contain one element from each of the three sets.

Table 1.7: The first block design

Block	1	2	3	4	5	6	7	8	9	10	11	12
	1	1	1	2	2	2	3	3	3	4	4	4
	6	7	8	5	7	8	5	5	8	6	6	7
	11	11	12	11	10	12	9	10	9	10	12	9

Table 1.8: The second block design

Block	1	2	3	4	5	6	7	8	9	10	11	12
	1	1	1	1	2	2	2	2	3	3	3	3
	4	5	5	6	4	5	6	6	4	4	5	6
	7	7	9	9	7	8	7	8	8	9	8	9

Initial investigations into the merging of two block designs showed several problems. In combining two block designs there must be exactly one occurrence of each combination of the levels of the two factors in the merged design. This is because each factor combination refers to a position in the row–column design. The first factor refers to the row position and the second factor refers to the column position. The rows in the first replicate of the row–column design are given by factor 1 levels $1, 2, \ldots, k$, those in the second replicate are represented by $k+1, k+2, \ldots, 2k$, and in general, the rows of the ith replicate are represented by $(i-1)k+1, (i-1)k+2, \ldots, ik$ $(i = 1, 2, \ldots, r)$. Similarly, the columns in the jth replicate of the row–column design are given by factor 2 levels $(j-1)s+1, (j-1)s+2, \ldots, js$ $(j = 1, 2, \ldots, r)$. Every position in the row–column design must be represented exactly once in the merged design, that is, the dual design. For example, the s replications of the elements from the set $\{1, 2, \ldots, k\}$ must occur exactly once with each element in the set $\{1, 2, \ldots, s\}$. Therefore, the first element in each block of the dual design must be a unique member of the set $\{11, 12, \ldots, 1s, 21, 22, \ldots, k1, k2, \ldots, ks\}$. This severely restricts the number of mergers possible with any two block designs. Given two efficient block designs, it was often not possible to merge the two designs to form a legitimate dual design.

Example (continued)

Given the block designs in Tables 1.7 and 1.8 it was possible to merge these designs to form a valid dual design. The resulting two factor block design is given in Table 1.9 and this is in fact the dual design for the resolvable row–column design in Table 1.5.

Table 1.9: Merged block design using Table 1.7 and Table 1.8

Block	1	2	3	4	5	6	7	8	9	10	11	12
	41	12	13	22	23	11	33	21	32	43	31	42
	75	66	74	76	84	85	86	54	55	65	56	64
	97	108	119	107	118	129	99	117	128	98	109	127

The average efficiency factor E_{merge} using this method of generating resolvable row–column designs is given in Table 1.10 for a few examples. E_{merge} is compared to the average efficiency factor E_{random} obtained from an algorithm which allows random permutations of treatments within the blocks of an α–design. In the few instances where it was possible to form a legitimate dual design, it can be seen from Table 1.10 that the resulting resolvable row–column designs were not efficient. The merging of two efficient block designs to form an efficient resolvable row–column is not a suitable method of construction.

Table 1.10: E_{merge} and E_{random} for resolvable row–column designs

v	k	s	r	E_{merge}	E_{random}
12	3	4	3	0.4868	0.5012
15	3	5	3	0.5188	0.5218
20	5	4	3	0.5395	0.5977

1.6 Thesis Outline

Early investigations into the construction of resolvable row–column designs showed that relationships based on the contraction had the most potential. This method is an extension of the work by Bailey and Patterson (1991) and Jarrett, Piper and Wild (1997) and is the main focus for the remainder of this thesis.

In Chapter 2 the concept of the contraction is outlined and the theory of Bailey and Patterson (1991) and Jarrett *et al.* (1997) for two–replicate resolvable row–column designs is redeveloped. The structural properties of the contraction are investigated, including the row and column incidence matrices. It is shown that the information matrix for the dual design can be expressed in terms of the row and column incidence matrices for the contraction. This theory is then extended to r–replicate resolvable row–column designs in Chapter 3, where the eigenvalues for the dual design are identified and a relationship is established with the non–zero eigenvalues of the information matrix $\mathbf{A}$. This relationship then leads to developing a connection between the average efficiency factors of the dual design and the row–column design for $r > 2$.

In the search for optimal or near optimal resolvable row–column designs, using the average efficiency factor as the optimality criteria can be computationally expensive. An alternative, using the (M,S)–optimality criterion, is developed from properties of the contraction and is discussed in Chapter 3. By exploiting these properties it is possible to reduce the computational effort required to generate resolvable row–column designs. Hence, optimal or near optimal designs can be generated quickly and an algorithm for doing so is outlined in Chapter 4. This algorithm is compared with the design generation package CycDesigN version 2.0.

The computational effort required when constructing resolvable row–column designs can be reduced even further by estimating the average efficiency factor, rather than calculating the exact value. This approach is discussed in Chapter 5. The efficiency factors for the pairwise differences are also considered in Chapter 5. The distribution of the pairwise efficiency factors provide further information to the usefulness of a design and can aid in the design selection process.

When considering the average efficiency factor it is helpful to have an upper bound in order to assess the possibility for further improvement. An upper bound based on the third moment bound for block designs of Jarrett (1983), the corrected second moment bound for block designs from Jarrett (1989) and the corrected third moment bound for block designs from Williams and Patterson (1977) is developed in Chapter 6. This new bound is compared to the upper bound given by CycDesigN version 2.0.

Chapter 2

Two–Replicate Resolvable Row–Column Designs

2.1 Introduction

A resolvable row–column design has $v = ks$ treatments set out in r replicates each consisting of k rows and s columns. For the design to be resolvable every treatment must occur exactly once in each replicate. Bailey and Patterson (1991) show that two–replicate resolvable row–column designs are combinatorially equivalent to a single replicate row–column design for two factors. This single replicate design is called the *contraction.* The number of levels for the two factors in the contraction are k and s respectively.

Example

As an example consider the row–column design given in Table 2.1 where $v = 12$, $k = 3$, $s = 4$ and $r = 2$. The contraction will be a single replicate row–column design with 3 rows and 4 columns, with the first factor having $k = 3$ levels and the second factor having $s = 4$ levels. Following the method of John and Williams (1995, p123) the contraction can be formed and is given in Table 2.2. The entry in the row–column design in row 1 column 2 of replicate 2 is treatment 6. In replicate 1 treatment 6 is found in row 3 column 3, therefore the entry in row 1 column 2 of the contraction will be 33. Similarly, treatment 7 in replicate 2 is in row 3 column 1, and in replicate 1 it is in row 3 column 4. Therefore the entry in the contraction

Table 2.1: Resolvable row–column design for $v = 12$, $k = 3$, $s = 4$, $r = 2$

Replicate		1				2			
Column		1	2	3	4	1	2	3	4
Row	1	10	12	5	4	12	**6**	11	4
	2	1	9	3	8	1	8	5	2
	3	11	2	**6**	7	7	10	9	3

Table 2.2: The contraction of the design in Table 2.1

Column		1	2	3	4
Row	1	12	**33**	31	14
	2	21	24	13	32
	3	34	11	22	23

in row 3 column 1, will be 34.

It can be seen from the contraction that the first factor is orthogonal to columns, in that each of the $k = 3$ levels of the factor occur once in each column. Similarly, the second factor is orthogonal to rows. In such a case, the two–replicate resolvable row–column design will be adjusted orthogonal (John and Eccleston, 1986). The property of adjusted orthogonality will be discussed in detail in section 2.4.

Jarrett, Piper and Wild (1997) refer to the contraction as the *reduced design.* They relate the eigenvalues of the contraction to the eigenvalues of the two–replicate resolvable row–column design, and hence imply a direct relationship between the average efficiency factor of the row–column design and that of the contraction.

Jarrett *et al.* (1997) also consider the dual design of the two–replicate resolvable row–column design. The dual design is a factorial design with v blocks of size $r = 2$, with the rows of the row–column design defining factor 1 with rk levels and the columns defining factor 2 with rs levels; see Section 1.5. Jarrett *et al.* (1997) state that the non–unit canonical efficiency factors of the dual design, for a main effects model, are the same as those of the two–replicate row–column design. This will be investigated further in section 2.2.

One of the advantages of using the contraction and dual design is that it reduces the dimensionality of the problem of constructing two–replicate resolvable row–column designs. The canonical efficiency factors of the row–column design are obtained from a symmetric matrix of order $v = ks$, while those for the contraction and dual design are obtained from matrices of order $(k+s)$ and $2(k+s)$ respectively. For large designs this saving can be considerable, for example with $v = 300$, $k = 10$ and $s = 30$, the matrices are of order 300, 40 and 80 respectively.

2.2 Canonical Efficiency Factors

In order to derive the coefficient matrix of the normal equations for the dual design, we define the main effects model as

$$y_{ijm} = \mu + \alpha_i + \gamma_j + \beta_m + \epsilon_{ijm}$$

where y_{ijm} is the response within block m for factor 1 at level i and factor 2 at level j, μ is the general mean effect, α_i is the effect of factor 1 at the ith level, γ_j is the effect of factor 2 at the jth level, β_m is the effect of the mth block, and ϵ_{ijm} are uncorrelated random variables with mean 0 and variance σ^2. This model can be rewritten in matrix notation as

$$\mathbf{y} = \mathbf{1}\mu + \mathbf{W}_1\boldsymbol{\alpha} + \mathbf{W}_2\boldsymbol{\gamma} + \mathbf{Z}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are the design matrices for factor 1 and 2 respectively, and $\mathbf{Z}$ is the design matrix for the blocks.

The normal equations are

$$n\hat{\mu} + s\mathbf{1}'\hat{\boldsymbol{\alpha}} + k\mathbf{1}'\hat{\boldsymbol{\gamma}} + r\mathbf{1}'\hat{\boldsymbol{\beta}} = G$$

$$s\mathbf{1}\hat{\mu} + s\hat{\boldsymbol{\alpha}} + (\mathbf{I}_2 \otimes \mathbf{J}_{ks})\hat{\boldsymbol{\gamma}} + \mathbf{N}_k'\hat{\boldsymbol{\beta}} = \mathbf{T}_1 \qquad (2.1)$$

$$k\mathbf{1}\hat{\mu} + (\mathbf{I}_2 \otimes \mathbf{J}_{sk})\hat{\boldsymbol{\alpha}} + k\hat{\boldsymbol{\gamma}} + \mathbf{N}_s'\hat{\boldsymbol{\beta}} = \mathbf{T}_2 \qquad (2.2)$$

$$r\mathbf{1}\hat{\mu} + \mathbf{N}_k\hat{\boldsymbol{\alpha}} + \mathbf{N}_s\hat{\boldsymbol{\gamma}} + r\hat{\boldsymbol{\beta}} = \mathbf{B} \qquad (2.3)$$

where G is the overall total, $\mathbf{T}_1$ and $\mathbf{T}_2$ are vectors of the treatment totals, $\mathbf{B}$ is a vector of the block totals, $\otimes$ denotes the Kronecker product, and $n = rsk$.

The reduced normal equations involving only the treatment parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are obtained by eliminating $\hat{\mu}$ and the block effects $\hat{\boldsymbol{\beta}}$ from the full set of normal equations. Premultiplying (2.3) by $\mathbf{N}_k'/2$ and $\mathbf{N}_s'/2$, and subtracting from (2.1) and (2.2) respectively, gives

$$\mathbf{A}_d^* \begin{pmatrix} \hat{\alpha} \\ \hat{\gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{T}_1 - \frac{1}{2}\mathbf{N}_k'\mathbf{B} \\ \mathbf{T}_2 - \frac{1}{2}\mathbf{N}_s'\mathbf{B} \end{pmatrix}$$

where the matrix of coefficients for the reduced normal equations is given by

$$\mathbf{A}_d^* = \begin{pmatrix} s\mathbf{I} - \frac{1}{2}\mathbf{N}_k'\mathbf{N}_k & (\mathbf{I}_2 \otimes \mathbf{J}_{ks}) - \frac{1}{2}\mathbf{N}_k'\mathbf{N}_s \\ (\mathbf{I}_2 \otimes \mathbf{J}_{sk}) - \frac{1}{2}\mathbf{N}_s'\mathbf{N}_k & k\mathbf{I} - \frac{1}{2}\mathbf{N}_s'\mathbf{N}_s \end{pmatrix}$$

In the dual design, the first factor is replicated s times and the second factor k times. Following Ceranka and Mejza (1979) the canonical efficiency factors of the dual design can be obtained from the symmetric matrix $\mathbf{A}_d = \mathbf{Q}^{-1/2}\mathbf{A}_d^*\mathbf{Q}^{-1/2}$, where

$$\mathbf{Q} = \begin{pmatrix} s\mathbf{I}_{2k} & \mathbf{0} \\ \mathbf{0} & k\mathbf{I}_{2s} \end{pmatrix} \tag{2.4}$$

Therefore

$$\mathbf{A}_d = \begin{pmatrix} \mathbf{I}_{2k} - \frac{p^2}{2}\mathbf{N}_k'\mathbf{N}_k & pq(\mathbf{I}_2 \otimes \mathbf{J}_{ks}) - \frac{pq}{2}\mathbf{N}_k'\mathbf{N}_s \\ pq(\mathbf{I}_2 \otimes \mathbf{J}_{sk}) - \frac{pq}{2}\mathbf{N}_s'\mathbf{N}_k & \mathbf{I}_{2s} - \frac{q^2}{2}\mathbf{N}_s'\mathbf{N}_s \end{pmatrix} \tag{2.5}$$

where $p^2 = 1/s$ and $q^2 = 1/k$.

To establish the relationship between the canonical efficiency factors of $\mathbf{A}_d$ for the dual design and the two–replicate resolvable row–column design given by $\mathbf{A}/2$, let

$$\mathbf{Y} = \begin{pmatrix} p\mathbf{N}_k & q\mathbf{N}_s \end{pmatrix}$$

Hence

$$\mathbf{Y}'\mathbf{Y} = \begin{pmatrix} p^2\mathbf{N}_k'\mathbf{N}_k & pq\mathbf{N}_k'\mathbf{N}_s \\ pq\mathbf{N}_s'\mathbf{N}_k & q^2\mathbf{N}_s'\mathbf{N}_s \end{pmatrix}$$

so that from (2.5)

$$\mathbf{A}_d = \begin{pmatrix} \mathbf{I}_{2k} & pq(\mathbf{I}_2 \otimes \mathbf{J}_{ks}) \\ pq(\mathbf{I}_2 \otimes \mathbf{J}_{sk}) & \mathbf{I}_{2s} \end{pmatrix} - \frac{1}{2}\begin{pmatrix} p^2\mathbf{N}_k'\mathbf{N}_k & pq\mathbf{N}_k'\mathbf{N}_s \\ pq\mathbf{N}_s'\mathbf{N}_k & q^2\mathbf{N}_s'\mathbf{N}_s \end{pmatrix}$$

$$= \mathbf{V} - \tfrac{1}{2}\mathbf{Y}'\mathbf{Y}\text{ , say.} \tag{2.6}$$

Note from (1.2)

$$\tfrac{1}{2}\mathbf{A} = \mathbf{I}_v - \tfrac{1}{2}\mathbf{Y}\mathbf{Y}' + \tfrac{1}{v}\mathbf{J}_{vv} \tag{2.7}$$

By obtaining the non–zero eigenvalues of $\mathbf{Y}'\mathbf{Y}$ we can establish the relationship between the eigenvalues of $\mathbf{A}/2$ and $\mathbf{A}_d$ as $\mathbf{Y}'\mathbf{Y}$ and $\mathbf{Y}\mathbf{Y}'$ have the same non–zero eigenvalues.

For two–replicate row–column designs $\mathbf{N}_k$ and $\mathbf{N}_s$ can be expressed as

$$\mathbf{N}_k = \begin{pmatrix} \mathbf{N}_{k1} & \mathbf{N}_{k2} \end{pmatrix} \qquad \mathbf{N}_s = \begin{pmatrix} \mathbf{N}_{s1} & \mathbf{N}_{s2} \end{pmatrix}$$

where $\mathbf{N}_{k1}$ and $\mathbf{N}_{s1}$ are the row and column incidence matrices respectively for the first replicate, and $\mathbf{N}_{k2}$ and $\mathbf{N}_{s2}$ are the row and column incidence matrices for the second replicate.

The elements of the submatrices in $\mathbf{Y}'\mathbf{Y}$ can be expressed in terms of the two–replicate resolvable row–column design and also in terms of the contraction. $\mathbf{N}_k'\mathbf{N}_k$ is a symmetric matrix which can be expressed as

$$\mathbf{N}_k'\mathbf{N}_k = \begin{pmatrix} \mathbf{N}_{k1}'\mathbf{N}_{k1} & \mathbf{N}_{k1}'\mathbf{N}_{k2} \\ \mathbf{N}_{k2}'\mathbf{N}_{k1} & \mathbf{N}_{k2}'\mathbf{N}_{k2} \end{pmatrix}$$

The (ij)th element in $\mathbf{N}_{k1}'\mathbf{N}_{k1}$ is the number of treatments common to the ith and jth row of replicate 1, and will be s if $i = j$ and 0 otherwise. Similarly for $\mathbf{N}_{k2}'\mathbf{N}_{k2}$ where the entries correspond to the number of treatments common in the ith and jth row of replicate 2. $\mathbf{N}_{k1}'\mathbf{N}_{k2}$ is a $k \times k$ matrix. With respect to the two–replicate resolvable row–column design, the (ij)th element of this matrix is the number of treatments common to the ith row of replicate 1 and the jth row of replicate 2. In terms of the contraction, this element is the number of times level i of factor 1 occurs in the jth row of the contraction. This is equivalent to the (ij)th element of the row incidence matrix for factor 1 in the contraction, $\mathbf{R}_1$, say. Hence

$$\mathbf{N}_k'\mathbf{N}_k = \begin{pmatrix} s\mathbf{I} & \mathbf{R}_1 \\ \mathbf{R}_1' & s\mathbf{I} \end{pmatrix}$$

Similarly in terms of columns it can be shown that

$$\mathbf{N}_s'\mathbf{N}_s = \begin{pmatrix} \mathbf{N}_{s1}'\mathbf{N}_{s1} & \mathbf{N}_{s1}'\mathbf{N}_{s2} \\ \mathbf{N}_{s2}'\mathbf{N}_{s1} & \mathbf{N}_{s2}'\mathbf{N}_{s2} \end{pmatrix} = \begin{pmatrix} k\mathbf{I} & \mathbf{C}_2 \\ \mathbf{C}_2' & k\mathbf{I} \end{pmatrix}$$

where $\mathbf{C}_2$ is the column incidence matrix for factor 2 in the contraction.

Finally, $\mathbf{N}_k'\mathbf{N}_s$ is a $2k \times 2s$ matrix which can be expressed as

$$\mathbf{N}_k'\mathbf{N}_s = \begin{pmatrix} \mathbf{N}_{k1}'\mathbf{N}_{s1} & \mathbf{N}_{k1}'\mathbf{N}_{s2} \\ \mathbf{N}_{k2}'\mathbf{N}_{s1} & \mathbf{N}_{k2}'\mathbf{N}_{s2} \end{pmatrix} = \begin{pmatrix} \mathbf{J}_{ks} & \mathbf{C}_1 \\ \mathbf{R}_2' & \mathbf{J}_{ks} \end{pmatrix}$$

To show this, the (ij)th element of $\mathbf{N}'_{k1}\mathbf{N}_{s1}$ is the number of treatments common in row i and column j of replicate 1. Similarly, the (ij)th element of $\mathbf{N}'_{k2}\mathbf{N}_{s2}$ is the number of treatments common in row i and column j of replicate 2. As these row–column designs are resolvable there will only ever be one treatment common, hence each element in these matrices will be equal to 1. $\mathbf{N}'_{k1}\mathbf{N}_{s2}$ is a matrix representing the number of treatments common to the rows of replicate 1 and the columns of replicate 2, and will be equal to the number of times levels of factor 1 occur in columns of the contraction. Hence, $\mathbf{N}'_{k1}\mathbf{N}_{s2} = \mathbf{C}_1$, say. Similarly, the (ij)th element of $\mathbf{N}'_{k2}\mathbf{N}_{s1}$ represents the number of times level j of factor 2 occurs in the ith row of the contraction. Thus $\mathbf{N}'_{k2}\mathbf{N}_{s1} = \mathbf{R}'_2$, say.

Therefore in terms of the row and column incidence matrices for the contraction, $\mathbf{Y}'\mathbf{Y}$ can be written as

$$\mathbf{Y}'\mathbf{Y} = \left(\begin{array}{cc|cc} \mathbf{I}_k & p^2\mathbf{R}_1 & pq\mathbf{J}_{ks} & pq\mathbf{C}_1 \\ p^2\mathbf{R}'_1 & \mathbf{I}_k & pq\mathbf{R}'_2 & pq\mathbf{J}_{ks} \\ \hline pq\mathbf{J}_{sk} & pq\mathbf{R}_2 & \mathbf{I}_s & q^2\mathbf{C}_2 \\ pq\mathbf{C}'_1 & pq\mathbf{J}_{sk} & q^2\mathbf{C}'_2 & \mathbf{I}_s \end{array}\right)$$

Note that

$$\mathbf{C}_1\mathbf{1} = s\mathbf{1}, \quad \mathbf{C}'_1\mathbf{1} = k\mathbf{1}, \quad \mathbf{C}_2\mathbf{1} = \mathbf{C}'_2\mathbf{1} = k\mathbf{1} \tag{2.8}$$

$$\mathbf{R}_2\mathbf{1} = k\mathbf{1}, \quad \mathbf{R}'_2\mathbf{1} = s\mathbf{1}, \quad \mathbf{R}_1\mathbf{1} = \mathbf{R}'_1\mathbf{1} = s\mathbf{1} \tag{2.9}$$

Example (continued)

Returning to the two–replicate resolvable row–column design in Table 2.1, the incidence matrices $\mathbf{N}_k$ and $\mathbf{N}_s$ for the rows and columns respectively are

$$\mathbf{N}_k = \left(\begin{array}{ccc|ccc} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{array}\right) \qquad \mathbf{N}_s = \left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right)$$

which lead to the row and column incidence matrices for the two factors being

$$\mathbf{R}_1 = \begin{pmatrix} 2 & 1 & 1 \\ 0 & 2 & 2 \\ 2 & 1 & 1 \end{pmatrix} \qquad \mathbf{R}_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{C}_1 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{C}_2 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

By permuting the columns of $\mathbf{Y}$ such that it is now expressed in terms of replicates rather than rows and columns, it becomes

$$\mathbf{Y} = \begin{pmatrix} p\mathbf{N}_{k1} & q\mathbf{N}_{s1} & p\mathbf{N}_{k2} & q\mathbf{N}_{s2} \end{pmatrix}$$

Hence

$$\mathbf{Y}'\mathbf{Y} = \begin{pmatrix} \mathbf{L} & \mathbf{H} \\ \mathbf{H}' & \mathbf{L} \end{pmatrix} \tag{2.10}$$

where

$$\mathbf{L} = \begin{pmatrix} \mathbf{I}_k & pq\mathbf{J}_{ks} \\ pq\mathbf{J}_{sk} & \mathbf{I}_s \end{pmatrix} \quad \text{and} \quad \mathbf{H} = \begin{pmatrix} p^2\mathbf{R}_1 & pq\mathbf{C}_1 \\ pq\mathbf{R}_2 & q^2\mathbf{C}_2 \end{pmatrix} \tag{2.11}$$

Under the same permutation $\mathbf{V}$ from (2.6) can be expressed as

$$\mathbf{V} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix}$$

Now define $\mathbf{x}_1 = (q\mathbf{1}'_k \;\; p\mathbf{1}'_s)'$ and $\mathbf{x}_2 = (q\mathbf{1}'_k \;\; -p\mathbf{1}'_s)'$ so then $\mathbf{Hx}_1 = \mathbf{H}'\mathbf{x}_1 = \mathbf{Lx}_1 = 2\mathbf{x}_1$ and $\mathbf{Hx}_2 = \mathbf{H}'\mathbf{x}_2 = \mathbf{Lx}_2 = \mathbf{0}$. Using (2.8) and (2.9) it can be verified that

$$\mathbf{z}_{01} = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_1 \end{pmatrix} \quad \mathbf{z}_{02} = \begin{pmatrix} \mathbf{0} \\ \mathbf{x}_2 \end{pmatrix} \quad \mathbf{z}_{03} = \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{0} \end{pmatrix} \quad \mathbf{z}_{04} = \begin{pmatrix} \mathbf{x}_1 \\ -\mathbf{x}_1 \end{pmatrix}$$

are eigenvectors of $\mathbf{Y}'\mathbf{Y}$ with eigenvalues 4, 0, 0 and 0 respectively. Since $\mathbf{x}'_1\mathbf{x}_2 = 0$, these four eigenvectors are mutually orthogonal.

Let $\mathbf{z} = (\mathbf{z}'_1 \;\; \mathbf{z}'_2)'$ where $\mathbf{z}_1$ and $\mathbf{z}_2$ are vectors of length $(k+s)$. If $\mathbf{z}$ is another eigenvector of $\mathbf{Y}'\mathbf{Y}$ then it must be orthogonal to $\mathbf{z}_{01}$, $\mathbf{z}_{02}$, $\mathbf{z}_{03}$ and $\mathbf{z}_{04}$. Hence, $\mathbf{z}'_i\mathbf{x}_1 = 0$ and $\mathbf{z}'_i\mathbf{x}_2 = 0$ $(i = 1, 2)$, which implies that

$$\mathbf{z}'_1\mathbf{1} = \mathbf{z}'_2\mathbf{1} = 0 \tag{2.12}$$

It also follows that $\mathbf{z}_{01}$, $\mathbf{z}_{02}$, $\mathbf{z}_{03}$ and $\mathbf{z}_{04}$ are eigenvectors of $\mathbf{V}$ with eigenvalues 2, 0, 0 and 2 respectively. Using (2.12) it can be shown that $\mathbf{Vz} = \mathbf{z}$, so that the remaining $2(k+s-2)$ eigenvalues of $\mathbf{V}$ are equal to 1.

If (2.10) is expressed as

$$\begin{aligned} \mathbf{Y}'\mathbf{Y} &= \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{H} \\ \mathbf{H}' & \mathbf{0} \end{pmatrix} \\ &= \mathbf{V} + \mathbf{W}\text{ , say,} \end{aligned}$$

then

$$\mathbf{Y}'\mathbf{Yz} = \mathbf{Vz} + \mathbf{Wz} \tag{2.13}$$

If $\mathbf{z}$ is an eigenvector of $\mathbf{W}$ then

$$\mathbf{Hz}_2 = \omega\mathbf{z}_1 \tag{2.14}$$

$$\mathbf{H}'\mathbf{z}_1 = \omega\mathbf{z}_2 \tag{2.15}$$

for some constant ω. Premultiplying (2.14) by $\mathbf{H}'$ and using (2.15) gives

$$\mathbf{H}'\mathbf{H}\mathbf{z}_2 = \omega\mathbf{H}\mathbf{z}_1 = \omega^2\mathbf{z}_2 \tag{2.16}$$

The eigenvalue of $\mathbf{H}'\mathbf{H}$ corresponding to $\mathbf{z}_2$ is ω^2, which gives $\pm\omega$ as two eigenvalues of $\mathbf{W}$. Similarly it can be shown that the eigenvalue of $\mathbf{H}\mathbf{H}'$ corresponding to $\mathbf{z}_1$ is also ω^2. It is known that two eigenvalues of $\mathbf{H}'\mathbf{H}$ are 0 and 4, since $\mathbf{H}'\mathbf{H}\mathbf{x}_2 = \mathbf{0}$ and $\mathbf{H}'\mathbf{H}\mathbf{x}_1 = 4\mathbf{x}_1$. Let the $(k+s-2)$ eigenvalues of $\mathbf{H}'\mathbf{H}$ that satisfy (2.12) and (2.16) be denoted by $\omega_1^2,\ \omega_2^2,\ \ldots,\ \omega_{k+s-2}^2$.

Then using (2.13), the remaining eigenvalues of $\mathbf{Y}'\mathbf{Y}$, which will be referred to as non–trivial eigenvalues, are $1 \pm \omega_i$ $(i = 1, 2, \ldots, k+s-2)$. Using (2.12) it can be shown that the eigenvector $\mathbf{z}_{01}$ of $\mathbf{Y}'\mathbf{Y}$ with eigenvalue 4 corresponds to an eigenvalue of 0 for $\mathbf{A}_d$. Given the eigenvectors and eigenvalues of $\mathbf{V}$ and $\mathbf{Y}'\mathbf{Y}$, from (2.6) the eigenvalues of $\mathbf{A}_d$ for the dual design are therefore

$$0, 0, 0, \tfrac{1}{2}(1 \pm \omega_1), \tfrac{1}{2}(1 \pm \omega_2), \ldots, \tfrac{1}{2}(1 \pm \omega_{k+s-2}), 2 \tag{2.17}$$

The canonical efficiency factors of the two–replicate resolvable row–column design are the non–zero eigenvalues of $\mathbf{A}/2$ and can now be obtained. Let $b = 2(k+s)$ and assume that the row–column design is connected so that $\text{rank}(\mathbf{A}) = (v-1)$ and $\mathbf{A}\mathbf{1} = \mathbf{0}$, where $\mathbf{A}/2$ is given by (2.7). Since $\mathbf{A}\mathbf{1} = \mathbf{0}$, $\mathbf{z}_{01}$ which corresponds to the eigenvalue 4 of $\mathbf{Y}'\mathbf{Y}$ will correspond to the zero eigenvalue of $\mathbf{A}/2$. Recall that the non–zero eigenvalues of $\mathbf{Y}\mathbf{Y}'$ are the same as those for $\mathbf{Y}'\mathbf{Y}$.

Two cases need to be considered. Firstly, consider designs where $v \geq b$. For such designs the eigenvalues of $\mathbf{Y}\mathbf{Y}'$ are 4, $1 \pm \omega_i$ $(i = 1, 2, \ldots, k+s-2)$ and 0 with multiplicity $(3 + v - b)$. Therefore, using (2.7) the canonical efficiency factors of the two–replicate resolvable row–column design are

$$\tfrac{1}{2}(1 \pm \omega_1), \tfrac{1}{2}(1 \pm \omega_2), \ldots, \tfrac{1}{2}(1 \pm \omega_{k+s-2}) \tag{2.18}$$

and 1 with multiplicity $(3 + v - b)$.

Now consider the case where $v < b$. The non–zero eigenvalues of $\mathbf{Y}\mathbf{Y}'$ are still 4 and $1 \pm \omega_i$ $(i = 1, 2, \ldots, k+s-2)$, but the 0 eigenvalue will now have multiplicity less than 3. The multiplicity of the 0 eigenvalue is given by $[3 - (b - v)] = (3 + v - b)$. Hence the canonical efficiency factors of the two–replicate resolvable row–column design are unchanged from (2.18).

Connected two–replicate resolvable row–column designs can only be found for designs where $3 + v - b \geq 0$. This constraint can be rewritten as $(k-2)(s-2) \geq 1$, suggesting the design parameters k and s both need to be greater than 2.

Example (continued)

For the two–replicate resolvable row–column design in Table 2.1, the non–zero eigenvalues of $\mathbf{H}'\mathbf{H}$, $\mathbf{Y}'\mathbf{Y}$, $\mathbf{A}_d$ and $\mathbf{A}/2$ are given in Table 2.3.

Table 2.3: Derivation of the canonical efficiency factors of the design in Table 2.1

$\mathbf{H}'\mathbf{H}$	$\mathbf{Y}'\mathbf{Y}$	$\mathbf{A}_d$	$\mathbf{A}/2$
ω_i^2	$1 \pm \omega_i$	$\frac{1}{2}(1 \pm \omega_i)$	$\frac{1}{2}(1 \pm \omega_i)$
0	1	$\frac{1}{2}$	$\frac{1}{2}$
	1	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{9}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
$\frac{1}{9}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
$\frac{1}{9}$	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
	$\frac{4}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$
	$\frac{3}{2}$	$\frac{3}{4}$	$\frac{3}{4}$
4	4	2	1

2.3 Average Efficiency Factors

In the previous section it was established that a relationship exists between the non–zero, non–unit canonical efficiency factors of the two–replicate resolvable row–column design and the non–zero, non–trivial eigenvalues of the dual design. The average efficiency factor of the row–column design can now be expressed in terms of the average efficiency factor of the dual design.

The average efficiency factor E for the two–replicate resolvable row–column design is given by the harmonic mean of the canonical efficiency factors in (2.18), namely

$$
\begin{aligned}
E &= \frac{v-1}{3+v-b+\sum_{i=1}^{k+s-2}[\frac{1}{2}(1\pm\omega_i)]^{-1}} \\
&= \frac{v-1}{3+v-b+4\sum_{i=1}^{k+s-2}(1-\omega_i^2)^{-1}} \qquad (2.19)
\end{aligned}
$$

The average efficiency factor E_d for the dual design is defined as the harmonic mean of the $2(k+s-2)$ non–zero, non–trivial eigenvalues given in (2.17). The eigenvalue equal to 2 is excluded from the calculations of E_d as the corresponding eigenvector $\mathbf{z}_{04}$ is not a member of the treatment contrast space; see Appendix A. This gives

$$
\begin{aligned}
E_d &= \frac{2(k+s)-4}{\sum_{i=1}^{k+s-2}[\frac{1}{2}(1\pm\omega_i)]^{-1}} \\
&= \frac{2(k+s)-4}{4\sum_{i=1}^{k+s-2}(1-\omega_i^2)^{-1}} \qquad (2.20)
\end{aligned}
$$

Using (2.19) and (2.20) the average efficiency factor of the two–replicate resolvable row–column design can be expressed as a function of E_d, namely

$$
E = \frac{v-1}{2(k+s-2)E_d^{-1}+(v-1)-2(k+s-2)} \qquad (2.21)
$$

Recall from Section 2.2 that ω_i^2 $(i=1,2,\ldots,k+s-2)$ are the non–zero, non–trivial eigenvalues of $\mathbf{H}'\mathbf{H}$ which is a $(k+s)$ square matrix. Thus the quantities $(1-\omega_i^2)$ in (2.20) are the non–zero eigenvalues of $\mathbf{I}-\mathbf{H}'\mathbf{H}$. These quantities also correspond to the non–zero canonical efficiency factors of the symmetric matrix $\mathbf{A}_c = \mathbf{Q}^{-1/2}\mathbf{A}_c^*\mathbf{Q}^{-1/2}$, where $\mathbf{A}_c^*$ is the information matrix for the contraction based on the main effects model and $\mathbf{Q}$ is given by (2.4). $\mathbf{A}_c$ can be expressed as

$$
\mathbf{A}_c = \mathbf{I} - \mathbf{H}\mathbf{H}' + \begin{pmatrix} q^2\mathbf{J}_{kk} & 2pq\mathbf{J}_{ks} \\ 2pq\mathbf{J}_{sk} & p^2\mathbf{J}_{ss} \end{pmatrix}
$$

and is equivalent to the matrix $\mathbf{F}$ as given by Jarrett *et al.* (1997).

These relationships show that E can be calculated directly from the non–zero eigenvalues of $\mathbf{H}'\mathbf{H}$. Therefore, in order to calculate E for two–replicate resolvable row–column designs it is not necessary to invert a $v \times v$ matrix but a $(k+s)$ square matrix. For large v this will result in considerable reductions in computational effort.

Example (continued)

From Table 2.3 the values of $(1-\omega_i^2)$ are 1, 8/9, 8/9, 8/9 and 3/4 such that, using

(2.20), the average efficiency factor of the dual design E_d is 0.437956. Substituting this value into (2.21) gives the average efficiency factor E of the two–replicate resolvable row–column design as 0.461538.

2.4 Adjusted Orthogonality

The property of adjusted orthogonality for row–column designs was introduced by Eccleston and Russell (1975). This concept ensures that no treatment is confounded with both rows and columns. Expressed mathematically, a two–replicate resolvable row–column design is adjusted orthogonal if

$$\mathbf{C}_1 = \mathbf{J} \quad \text{and} \quad \mathbf{R}_2 = \mathbf{J}$$

It will now be shown that for the adjusted orthogonal case the eigenvalues of $\mathbf{Y}'\mathbf{Y}$ comprise the eigenvalues of the row contraction and those of the column contraction. With the property of adjusted orthogonality, (2.11) becomes

$$\mathbf{H} = \begin{pmatrix} p^2\mathbf{R}_1 & pq\mathbf{J}_{ks} \\ pq\mathbf{J}_{sk} & q^2\mathbf{C}_2 \end{pmatrix}$$

so that

$$\mathbf{H}'\mathbf{H} = \begin{pmatrix} p^4\mathbf{R}_1'\mathbf{R}_1 & \mathbf{0} \\ \mathbf{0} & q^4\mathbf{C}_2'\mathbf{C}_2 \end{pmatrix} + \begin{pmatrix} q^2\mathbf{J}_{kk} & 2pq\mathbf{J}_{ks} \\ 2pq\mathbf{J}_{sk} & p^2\mathbf{J}_{ss} \end{pmatrix}$$

Given a block diagonal matrix $\mathbf{B}$ with square diagonal blocks $\mathbf{B}_1$ and $\mathbf{B}_2$, it is known that the eigenvalues of $\mathbf{B}$ are given by the eigenvalues of the matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ (Harville, 1997, p523). Using this fact and (2.12), it can be seen that the eigenvalues of $\mathbf{H}'\mathbf{H}$ form two groups, corresponding to the matrices $p^4\mathbf{R}_1'\mathbf{R}_1$ and $q^4\mathbf{C}_2'\mathbf{C}_2$. The non–zero eigenvalues of $p^4\mathbf{R}_1'\mathbf{R}_1$ will be denoted by ξ_i^2 $(i = 1, 2, \ldots, k-1)$ and those of $q^4\mathbf{C}_2'\mathbf{C}_2$ will be denoted by δ_i^2 $(i = 1, 2, \ldots, s-1)$. These relate to the canonical efficiency factors of the row and column contractions respectively. The first factor which is orthogonal to columns is replicated s times, and the second factor which is orthogonal to rows is replicated k times. As $\mathbf{R}_1'\mathbf{R}_1$ and $\mathbf{R}_1\mathbf{R}_1'$ have the same non–zero eigenvalues, it follows that the canonical efficiency factors of the row contraction based on the information matrix $\mathbf{A}/s = \mathbf{I} - p^4\mathbf{R}_1\mathbf{R}_1'$ will be

$$1 - \xi_1^2, 1 - \xi_2^2, \ldots, 1 - \xi_{k-1}^2$$

Therefore, the average efficiency factor E_k for the row contraction is the harmonic mean of the $(k-1)$ canonical efficiency factors

$$E_k = \frac{k-1}{\sum_{i=1}^{k-1}(1-\xi_i^2)^{-1}} \tag{2.22}$$

Similarly, for the column contraction the canonical efficiency factors are

$$1-\delta_1^2, 1-\delta_2^2, \ldots, 1-\delta_{s-1}^2$$

Hence the average efficiency factor E_s for the column contraction is the harmonic mean of the $(s-1)$ canonical efficiency factors

$$E_s = \frac{s-1}{\sum_{i=1}^{s-1}(1-\delta_i^2)^{-1}} \tag{2.23}$$

Therefore with the condition of adjusted orthogonality and using (2.20), it can be seen that

$$(b-4)E_d^{-1} = 4(s-1)E_s^{-1} + 4(k-1)E_k^{-1}$$

In terms of the row and column contractions, the average efficiency factor of an adjusted orthogonal two–replicate resolvable row–column design simplifies to

$$E = \frac{v-1}{v-1-2(k-1)-2(s-1)+4(k-1)E_k^{-1}+4(s-1)E_s^{-1}} \tag{2.24}$$

as given by John and Williams (1995, p124).

Example (continued)

As shown earlier for the two–replicate resolvable row–column design in Table 2.1, $\mathbf{C}_1 = \mathbf{J}$ and $\mathbf{R}_2 = \mathbf{J}$ so that the design is adjusted orthogonal. The canonical efficiency factors of the row contraction are 1 and 3/4. Substituting these into (2.22) gives the average efficiency factor for the row contraction as 6/7. For the column contraction the canonical efficiency factors are 8/9, 8/9 and 8/9, which, using (2.23), gives an average efficiency factor of 8/9. Substituting these values into (2.24) gives $E = 0.461538$, as before.

Chapter 3

r–Replicate Resolvable Row–Column Designs

3.1 Introduction

In agricultural and forestry field trials it is common for each variety to be replicated more than twice. In this chapter the theory presented in Chapter 2 will be extended to cover resolvable row–column designs with more than two replicates. The structure and method of constructing the contraction for such designs is discussed in Section 3.2.

Of primary importance in this chapter is to understand the properties of the contraction in order to construct optimal or near optimal resolvable row–column designs. The model of interest is, however, the main effects model of the dual design. The structure of the dual design is discussed in Section 3.3.

In the previous chapter it was shown that the non–zero eigenvalues of $\mathbf{A}_d$ for a two–replicate resolvable row–column design can be expressed as a function of the non–zero eigenvalues of $\mathbf{H}'\mathbf{H}$. In Section 3.4 the structure of $\mathbf{A}_d$ is obtained and the corresponding canonical efficiency factors are determined in Section 3.5. The average efficiency factor of the resolvable row–column design is calculated as a function of the non–zero eigenvalues of $\mathbf{A}_d$. In Section 3.6 expressions for the average efficiency factors of the resolvable row–column design and the dual design are obtained. It is also shown that the average efficiency factor of the resolvable row–column design can be expressed in terms of the average efficiency factor of the dual design.

An alternative approach to finding a subclass of designs which are optimal or near optimal is to consider the (M,S)–optimality criterion. This approach is computationally more economical in a search algorithm than calculating the average efficiency factor for resolvable row–column designs with more than two replicates. A suitable (M,S)–optimality criterion is developed in Section 3.7.

Often it is important to find a resolvable row–column design which not only has a good row–column design, but also good row and/or column component designs. Average efficiency factors for the row and column component designs are discussed in Section 3.8.

3.2 Contraction

The structure of the contraction for a r–replicate resolvable row–column design is a direct extension of the two–replicate case. In Chapter 2 it was shown that the contraction of a two–replicate resolvable row–column design contains a single array representing the relationship between the two replicates of the row–column design. For three–replicate resolvable row–column designs the contraction will contain two arrays. These represent the relationships between one of the replicates and the remaining two replicates. Without loss of generality, the contraction will be considered with respect to the first replicate of the row–column design. The first array in the contraction gives the relationship between replicate 1 and replicate 2 of the row–column design, and the second array represents the relationship between replicates 1 and 3. In general, a r–replicate resolvable row–column design will have a corresponding contraction containing $(r-1)$ arrays.

Each array in the contraction has k rows and s columns and is a replicate in a row–column design with two factors. The first factor relates to the rows, and the second factor to the columns of the resolvable row–column design. Therefore, the first factor has k levels and the second factor s levels.

The elements of each array are determined by the positioning of treatments within a replicate of the row–column design relative to the first replicate. For example, consider replicates 1 and m in a r–replicate resolvable row–column design $(m = 2, \dots, r)$. Suppose that treatment t occurs in row r_1 column c_1 of replicate 1, and in row r_2 column c_2 of replicate m. The entry in row r_2 column c_2 of the

$(m-1)$th array will be the two-tuple r_1c_1.

Example

The three-replicate resolvable row–column design in Table 3.1 was generated using CycDesigN version 2.0. The contraction of this row–column design is shown in Table 3.2.

Table 3.1: Resolvable row–column design for $v = 12$, $k = 3$, $s = 4$, $r = 3$

Replicate		1				2				3			
Column		1	2	3	4	1	2	3	4	1	2	3	4
Row	1	11	12	9	10	**7**	9	10	6	*5*	2	**7**	12
	2	*5*	6	4	1	12	8	*5*	11	9	6	3	8
	3	2	3	**7**	8	1	2	3	4	1	10	11	4

Table 3.2: The contraction of the design in Table 3.1

Array		1				2			
Column		1	2	3	4	1	2	3	4
Row	1	**33**	13	14	22	*21*	31	**33**	12
	2	12	34	*21*	11	13	22	32	34
	3	24	31	32	23	24	14	11	23

Consider treatment 7 in replicate 1 of the row–column design in Table 3.1. It occurs in row 3 column 3. Hence, 33 is the entry in arrays 1 and 2 of Table 3.2 corresponding to the location of treatment 7 in replicates 2 and 3 respectively.

As another example, treatment 5 is in row 2 column 1 of replicate 1. Therefore 21 is the entry in row 2 column 3 of array 1 and in row 1 column 1 of array 2.

Associated with each array in the contraction is a row incidence matrix and a column incidence matrix for each factor. These incidence matrices will play an important role in the construction of resolvable row–column designs. Let $\mathbf{R}_{1j}$ represent the row incidence matrix for factor 1 in array j, and $\mathbf{R}_{2j}$ the row incidence matrix for factor 2 in array j $(j = 1, 2, \ldots, r-1)$. Similarly, let $\mathbf{C}_{ij}$ represent the column incidence matrix for factor i in array j ($i = 1, 2$ and $j = 1, 2, \ldots, r-1$). The (wx)th element of $\mathbf{R}_{1j}$ is the number of times level w of factor 1 occurs in the xth row of array j. Similarly, the (wx)th element of $\mathbf{R}_{2j}$ is the number of times level w of factor 2 occurs in the xth row of array j. The column incidence matrices can be constructed in a similar way. The (wx)th element of $\mathbf{C}_{ij}$ is the number of times level w of factor i occurs in the xth column of array j ($i = 1, 2$ and $j = 1, 2, \ldots, r-1$).

Example (continued)

From the contraction in Table 3.2 the row and column incidence matrices are

$$\mathbf{R}_{11} = \begin{pmatrix} 2 & 2 & 0 \\ 1 & 1 & 2 \\ 1 & 1 & 2 \end{pmatrix} \qquad \mathbf{R}_{12} = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 2 & 0 \end{pmatrix}$$

$$\mathbf{R}_{21} = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 2 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{R}_{22} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

$$\mathbf{C}_{11} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \\ 1 & 2 & 1 & 0 \end{pmatrix} \qquad \mathbf{C}_{12} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & 0 & 1 \\ 0 & 1 & 2 & 1 \end{pmatrix}$$

$$\mathbf{C}_{21} = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad \mathbf{C}_{22} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

Each pair of arrays in the contraction yields a secondary array which is constructed in the same way as a contraction array. For a three–replicate resolvable row–column design, one secondary array is constructed from arrays 1 and 2 of the contraction. This third array is equivalent to an array produced by considering the relationship of replicate 2 in the row–column design with respect to replicate 3. In general, for a r–replicate resolvable row–column design, $(r-1)(r-2)/2$ secondary arrays can be constructed from pairs of the $(r-1)$ arrays in the contraction. This results in a total of $h = r(r-1)/2$ arrays.

Row and column incidence matrices can be also constructed for each secondary array. For a r–replicate resolvable row–column design, the row and column sums of all the incidence matrices satisfy the following conditions

$$\mathbf{C}_{1j}\mathbf{1} = s\mathbf{1}, \quad \mathbf{C}'_{1j}\mathbf{1} = k\mathbf{1}, \quad \mathbf{C}_{2j}\mathbf{1} = \mathbf{C}'_{2j}\mathbf{1} = k\mathbf{1} \tag{3.1}$$

$$\mathbf{R}_{1j}\mathbf{1} = \mathbf{R}'_{1j}\mathbf{1} = s\mathbf{1}, \quad \mathbf{R}_{2j}\mathbf{1} = k\mathbf{1}, \quad \mathbf{R}'_{2j}\mathbf{1} = s\mathbf{1} \tag{3.2}$$

for $j = 1, 2, \ldots, h$.

Example (continued)

The array in Table 3.3 has been constructed from the arrays of the contraction in Table 3.2. For instance, the two–tuple 33 occurs in row 1 column 1 of array 1, so that 11 will occur in the secondary array according to the position of 33 in array 2, that is, row 1 column 3.

Table 3.3: Secondary array of the design in Table 3.1

Column		1	2	3	4
Row	1	23	32	**11**	21
	2	12	14	33	22
	3	31	13	24	34

The row and column incidence matrices for this secondary array are

$$\mathbf{R}_{13} = \begin{pmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \qquad \mathbf{R}_{23} = \begin{pmatrix} 2 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

$$\mathbf{C}_{13} = \begin{pmatrix} 1 & 2 & 1 & 0 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 1 & 1 \end{pmatrix} \qquad \mathbf{C}_{23} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

3.3 Dual Design

Although this chapter is concerned with the properties of the contraction, the model of interest is the main effects model of the dual design. The dual design is formed by interchanging the block and treatments labels. The relationship between r–replicate resolvable row–column designs and their dual designs is a natural extension of the two–replicate case. The dual design consists of v blocks of size r, with each plot containing two factors. The rows of the row–column design define one factor with rk levels and the columns define the second factor with rs levels. The rows in the first replicate of the row–column design are labelled $1, 2, \dots, k$, those in the second replicate are labelled $k+1, k+2, \dots, 2k$, and in general, the rows of the ith replicate are labelled $(i-1)k+1, (i-1)k+2, \dots, ik$. Similarly, the columns in the first replicate are labelled $1, 2, \dots, s$, in the second replicate the columns are labelled $s+1, s+2, \dots, 2s$, and in the ith replicate they are labelled $(i-1)s+1, (i-1)s+2, \dots, is$.

Example (continued)

The dual design for the resolvable row–column design in Table 3.1 is given in Table 3.4. For instance, as treatment 1 is in row 2 column 4 of replicate 1, in row $k+3=6$ and column $s+1=5$ of replicate 2, and in row $2k+3=9$ and column $2s+1=9$ of replicate 3, the entries in block 1 of the dual design will be 24, 65 and 99.

Table 3.4: The dual of the design in Table 3.1

Block	1	2	3	4	5	6	7	8	9	10	11	12
	24	31	32	23	21	22	33	34	13	14	11	12
	65	66	67	68	57	48	45	56	46	47	58	55
	99	$7\underline{10}$	$8\underline{11}$	$9\underline{12}$	79	$8\underline{10}$	$7\underline{11}$	$8\underline{12}$	89	$9\underline{10}$	$9\underline{11}$	$7\underline{12}$

3.4 Information Matrix of the Dual Design

In the previous chapter for two–replicate resolvable row–column designs, $\mathbf{A}_d$ was expressed first in terms of the row and column incidence matrices and then in terms of $\mathbf{Y}'\mathbf{Y}$. From the main effects model for the dual design, the structure of $\mathbf{A}_d$ for

r–replicate resolvable row–column designs will be determined in this section.

The main effects model for the dual design of a r–replicate resolvable row–column design is

$$y_{ijm} = \mu + \alpha_i + \gamma_j + \beta_m + \epsilon_{ijm}$$

where y_{ijm} is the response within block m for factor 1 at level i and factor 2 at level j, μ is the general mean effect, α_i is the effect of factor 1 at the ith level, γ_j is the effect of factor 2 at the jth level, β_m is the effect of the mth block, and ϵ_{ijm} are uncorrelated random variables with mean 0 and variance σ^2.

The normal equations are

$$\begin{aligned} n\hat{\mu} + s\mathbf{1}'\hat{\boldsymbol{\alpha}} + k\mathbf{1}'\hat{\boldsymbol{\gamma}} + r\mathbf{1}'\hat{\boldsymbol{\beta}} &= G \\ s\mathbf{1}\hat{\mu} + s\hat{\boldsymbol{\alpha}} + (\mathbf{I}_r \otimes \mathbf{J}_{ks})\hat{\boldsymbol{\gamma}} + \mathbf{N}_k'\hat{\boldsymbol{\beta}} &= \mathbf{T}_1 \\ k\mathbf{1}\hat{\mu} + (\mathbf{I}_r \otimes \mathbf{J}_{sk})\hat{\boldsymbol{\alpha}} + k\hat{\boldsymbol{\gamma}} + \mathbf{N}_s'\hat{\boldsymbol{\beta}} &= \mathbf{T}_2 \\ r\mathbf{1}\hat{\mu} + \mathbf{N}_k\hat{\boldsymbol{\alpha}} + \mathbf{N}_s\hat{\boldsymbol{\gamma}} + r\hat{\boldsymbol{\beta}} &= \mathbf{B} \end{aligned}$$

where n, G, $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{B}$ are as defined in Section 2.2. $\mathbf{N}_k$ and $\mathbf{N}_s$ are the row and column incidence matrices respectively of the resolvable row–column design. For a r–replicate resolvable row–column design they can be written as

$$\mathbf{N}_k = \begin{pmatrix} \mathbf{N}_{k1} & \mathbf{N}_{k2} & \ldots & \mathbf{N}_{kr} \end{pmatrix} \qquad \mathbf{N}_s = \begin{pmatrix} \mathbf{N}_{s1} & \mathbf{N}_{s2} & \ldots & \mathbf{N}_{sr} \end{pmatrix}$$

where $\mathbf{N}_{ki}$ and $\mathbf{N}_{si}$ are the row and column incidence matrices respectively of the ith replicate $(i = 1, 2, \ldots, r)$.

The reduced normal equations after the removal of $\hat{\mu}$ and the block effects $\hat{\boldsymbol{\beta}}$ are

$$\mathbf{A}_d^* \begin{pmatrix} \hat{\boldsymbol{\alpha}} \\ \hat{\boldsymbol{\gamma}} \end{pmatrix} = \begin{pmatrix} \mathbf{T}_1 - \frac{1}{r}\mathbf{N}_k'\mathbf{B} \\ \mathbf{T}_2 - \frac{1}{r}\mathbf{N}_s'\mathbf{B} \end{pmatrix}$$

where $\mathbf{A}_d^*$ is

$$\mathbf{A}_d^* = \begin{pmatrix} s\mathbf{I} - \frac{1}{r}\mathbf{N}_k'\mathbf{N}_k & (\mathbf{I}_r \otimes \mathbf{J}_{ks}) - \frac{1}{r}\mathbf{N}_k'\mathbf{N}_s \\ (\mathbf{I}_r \otimes \mathbf{J}_{sk}) - \frac{1}{r}\mathbf{N}_s'\mathbf{N}_k & k\mathbf{I} - \frac{1}{r}\mathbf{N}_s'\mathbf{N}_s \end{pmatrix}$$

As in Chapter 2, following Ceranka and Mejza (1979), the canonical efficiency factors of the dual design are a subset of the non–zero eigenvalues of $\mathbf{A}_d = \mathbf{Q}^{-1/2}\mathbf{A}_d^*\mathbf{Q}^{-1/2}$ where $\mathbf{Q}$ is now

$$\mathbf{Q} = \begin{pmatrix} s\mathbf{I}_{rk} & \mathbf{0} \\ \mathbf{0} & k\mathbf{I}_{rs} \end{pmatrix}$$

Therefore

$$\mathbf{A}_d = \begin{pmatrix} \mathbf{I}_{rk} - \frac{p^2}{r}\mathbf{N}_k'\mathbf{N}_k & pq(\mathbf{I}_r \otimes \mathbf{J}_{ks}) - \frac{pq}{r}\mathbf{N}_k'\mathbf{N}_s \\ pq(\mathbf{I}_r \otimes \mathbf{J}_{sk}) - \frac{pq}{r}\mathbf{N}_s'\mathbf{N}_k & \mathbf{I}_{rs} - \frac{q^2}{r}\mathbf{N}_s'\mathbf{N}_s \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{I}_{rk} & pq(\mathbf{I}_r \otimes \mathbf{J}_{ks}) \\ pq(\mathbf{I}_r \otimes \mathbf{J}_{sk}) & \mathbf{I}_{rs} \end{pmatrix} - \frac{1}{r}\begin{pmatrix} p^2\mathbf{N}_k'\mathbf{N}_k & pq\mathbf{N}_k'\mathbf{N}_s \\ pq\mathbf{N}_s'\mathbf{N}_k & q^2\mathbf{N}_s'\mathbf{N}_s \end{pmatrix} \tag{3.3}$$

$$= \mathbf{V} - \tfrac{1}{r}\mathbf{Y}'\mathbf{Y} \text{ , say,} \tag{3.4}$$

where $p^2 = 1/s$, $q^2 = 1/k$ and $\mathbf{Y} = (p\mathbf{N}_k \;\; q\mathbf{N}_s)$.

The submatrices of the $\mathbf{Y}'\mathbf{Y}$ matrix can be expressed in terms of the row and column incidence matrices of the contraction and the secondary arrays. Now

$$\mathbf{N}_{ki}'\mathbf{N}_{km} = \begin{cases} s\mathbf{I}_k & \text{if } i = m \\ \mathbf{R}_{1(m-1)} & \text{if } i = 1,\; m > 1 \\ \mathbf{R}_{1j} & \text{if } 1 < i < m, \text{ where } j = m - r - i(i+1-2r)/2 \end{cases}$$

where $\mathbf{R}_{1j}$ is the row incidence matrix for factor 1 in array j ($j = 1, 2, \ldots, h$ and $h = r(r-1)/2$). Similarly in terms of columns

$$\mathbf{N}_{si}'\mathbf{N}_{sm} = \begin{cases} k\mathbf{I}_s & \text{if } i = m \\ \mathbf{C}_{2(m-1)} & \text{if } i = 1,\; m > 1 \\ \mathbf{C}_{2j} & \text{if } 1 < i < m, \text{ where } j = m - r - i(i+1-2r)/2 \end{cases}$$

and

$$\mathbf{N}_{ki}'\mathbf{N}_{sm} = \begin{cases} \mathbf{J}_{ks} & \text{if } i = m \\ \mathbf{C}_{1(m-1)} & \text{if } i = 1,\; m > 1 \\ \mathbf{R}_{2(i-1)}' & \text{if } m = 1,\; i > 1 \\ \mathbf{C}_{1j} & \text{if } 1 < i < m, \text{ where } j = m - r - i(i+1-2r)/2 \\ \mathbf{R}_{2j}' & \text{if } 1 < m < i, \text{ where } j = i - r - m(m+1-2r)/2 \end{cases}$$

where $\mathbf{C}_{ij}$ is the column incidence matrix for factor i in array j and $\mathbf{R}_{2j}$ is the row incidence matrix for factor 2 in array j ($j = 1, 2, \ldots h$).

Hence, for a three-replicate resolvable row-column design

$$\mathbf{N}_k'\mathbf{N}_k = \begin{pmatrix} s\mathbf{I} & \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{11}' & s\mathbf{I} & \mathbf{R}_{13} \\ \mathbf{R}_{12}' & \mathbf{R}_{13}' & s\mathbf{I} \end{pmatrix} \tag{3.5}$$

$$\mathbf{N}_s'\mathbf{N}_s = \begin{pmatrix} k\mathbf{I} & \mathbf{C}_{21} & \mathbf{C}_{22} \\ \mathbf{C}_{21}' & k\mathbf{I} & \mathbf{C}_{23} \\ \mathbf{C}_{22}' & \mathbf{C}_{23}' & k\mathbf{I} \end{pmatrix} \tag{3.6}$$

$$\mathbf{N}_k'\mathbf{N}_s = \begin{pmatrix} \mathbf{J}_{ks} & \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{R}_{21}' & \mathbf{J}_{ks} & \mathbf{C}_{13} \\ \mathbf{R}_{22}' & \mathbf{R}_{23}' & \mathbf{J}_{ks} \end{pmatrix}$$

so that

$$\mathbf{Y}'\mathbf{Y} = \left(\begin{array}{ccc|ccc} \mathbf{I} & p^2\mathbf{R}_{11} & p^2\mathbf{R}_{12} & pq\mathbf{J}_{ks} & pq\mathbf{C}_{11} & pq\mathbf{C}_{12} \\ p^2\mathbf{R}_{11}' & \mathbf{I} & p^2\mathbf{R}_{13} & pq\mathbf{R}_{21}' & pq\mathbf{J}_{ks} & pq\mathbf{C}_{13} \\ p^2\mathbf{R}_{12}' & p^2\mathbf{R}_{13}' & \mathbf{I} & pq\mathbf{R}_{22}' & pq\mathbf{R}_{23}' & pq\mathbf{J}_{ks} \\ \hline pq\mathbf{J}_{sk} & pq\mathbf{R}_{21} & pq\mathbf{R}_{22} & \mathbf{I} & q^2\mathbf{C}_{21} & q^2\mathbf{C}_{22} \\ pq\mathbf{C}_{11}' & pq\mathbf{J}_{sk} & pq\mathbf{R}_{23} & q^2\mathbf{C}_{21}' & \mathbf{I} & q^2\mathbf{C}_{23} \\ pq\mathbf{C}_{12}' & pq\mathbf{C}_{13}' & pq\mathbf{J}_{sk} & q^2\mathbf{C}_{22}' & q^2\mathbf{C}_{23}' & \mathbf{I} \end{array}\right)$$

By permuting the rows and columns, $\mathbf{Y}'\mathbf{Y}$ can be expressed as

$$\mathbf{Y}'\mathbf{Y} = \begin{pmatrix} \mathbf{L} & \mathbf{H}_1 & \mathbf{H}_2 \\ \mathbf{H}_1' & \mathbf{L} & \mathbf{H}_3 \\ \mathbf{H}_2' & \mathbf{H}_3' & \mathbf{L} \end{pmatrix}$$

where

$$\mathbf{L} = \begin{pmatrix} \mathbf{I}_k & pq\mathbf{J}_{ks} \\ pq\mathbf{J}_{sk} & \mathbf{I}_s \end{pmatrix}$$

and for $j = 1, 2, 3$

$$\mathbf{H}_j = \begin{pmatrix} p^2\mathbf{R}_{1j} & pq\mathbf{C}_{1j} \\ pq\mathbf{R}_{2j} & q^2\mathbf{C}_{2j} \end{pmatrix} \tag{3.7}$$

It can be seen that $\mathbf{H}_j$ involves the row and column incidence matrices of the jth array ($j = 1, 2, 3$).

Under the same permutation, $\mathbf{V}$ in (3.4) can be expressed as

$$\mathbf{V} = \begin{pmatrix} \mathbf{L} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{L} \end{pmatrix}$$

Hence

$$\mathbf{A}_d = \mathbf{V} - \tfrac{1}{3}\mathbf{Y}'\mathbf{Y} = \frac{1}{3}\begin{pmatrix} 2\mathbf{L} & -\mathbf{H}_1 & -\mathbf{H}_2 \\ -\mathbf{H}_1' & 2\mathbf{L} & -\mathbf{H}_3 \\ -\mathbf{H}_2' & -\mathbf{H}_3' & 2\mathbf{L} \end{pmatrix} \tag{3.8}$$

As a natural extension of the three–replicate case, for r–replicate resolvable row–column designs it follows that $\mathbf{A}_d$ can be written as

$$\mathbf{A}_d = \frac{1}{r}\begin{pmatrix} (r-1)\mathbf{L} & -\mathbf{H}_1 & -\mathbf{H}_2 & & -\mathbf{H}_{r-1} \\ -\mathbf{H}_1' & (r-1)\mathbf{L} & -\mathbf{H}_r & & -\mathbf{H}_{2r-3} \\ -\mathbf{H}_2' & -\mathbf{H}_r' & (r-1)\mathbf{L} & \ldots & -\mathbf{H}_{3r-6} \\ & & & & \\ -\mathbf{H}_{r-1}' & -\mathbf{H}_{2r-3}' & -\mathbf{H}_{3r-6}' & \ldots & (r-1)\mathbf{L} \end{pmatrix} \tag{3.9}$$

In the two–replicate case $\mathbf{A}_d$ is given by

$$\mathbf{A}_d = \frac{1}{2}\begin{pmatrix} \mathbf{L} & -\mathbf{H}_1 \\ -\mathbf{H}_1' & \mathbf{L} \end{pmatrix}$$

In Section 2.2 it was shown that the canonical efficiency factors of a two–replicate resolvable row–column design are functions of the non–zero eigenvalues of the matrix $\mathbf{H}_1'\mathbf{H}_1$. The average efficiency factor of the row–column design can therefore be obtained from the inverse of a square matrix of order $(k+s)$. This is computationally cheaper than inverting $\mathbf{A}_d$, which is a square matrix of order $2(k+s)$.

For resolvable row–column designs with more than two replicates, such a simplification is not possible. That is, the canonical efficiency factors of the row–column design cannot be obtained from a simple function involving the $\mathbf{H}_j$ matrices $(j = 1, 2, \ldots, h)$.

In the next section the canonical efficiency factors of the row–column design will be obtained by considering the non–zero eigenvalues of $\mathbf{Y}'\mathbf{Y}$.

3.5 Eigenvalues

The average efficiency factor for the resolvable row–column design can be calculated from the canonical efficiency factors of $\mathbf{A}/r$ or $\mathbf{A}_d$. By obtaining the non–zero eigenvalues of $\mathbf{Y}'\mathbf{Y}$ the relationship between the eigenvalues of $\mathbf{A}/r$ and $\mathbf{A}_d$ can be determined, as $\mathbf{Y}'\mathbf{Y}$ and $\mathbf{Y}\mathbf{Y}'$ have the same non–zero eigenvalues. Recall that $\mathbf{Y}\mathbf{Y}'$ is a square matrix of order v and $\mathbf{Y}'\mathbf{Y}$ is a square matrix of order b, where $b = r(k+s)$, and

$$\begin{aligned} \tfrac{1}{r}\mathbf{A} &= \mathbf{I}_v - \tfrac{1}{rs}\mathbf{N}_k\mathbf{N}_k' - \tfrac{1}{rk}\mathbf{N}_s\mathbf{N}_s' + \tfrac{1}{v}\mathbf{J}_{vv} \\ &= \mathbf{I}_v - \tfrac{1}{r}\mathbf{Y}\mathbf{Y}' + \tfrac{1}{v}\mathbf{J}_{vv} \end{aligned} \qquad (3.10)$$

and

$$\mathbf{A}_d = \mathbf{V} - \tfrac{1}{r}\mathbf{Y}'\mathbf{Y} \qquad (3.11)$$

where $\mathbf{V}$ and $\mathbf{Y}'\mathbf{Y}$ are given in (3.3).

Using (3.1) and (3.2) it can be seen that $\mathbf{x} = (q\mathbf{1}_{rk}' \;\; p\mathbf{1}_{rs}')'$ is an eigenvector of $\mathbf{Y}'\mathbf{Y}$ with eigenvalue $2r$ and is also an eigenvector of $\mathbf{V}$ with eigenvalue 2. It follows that $\mathbf{A}_d\mathbf{x} = \mathbf{0}$. Since $\mathbf{x}$ is the eigenvector corresponding to the grand mean and $\mathbf{x}'\mathbf{1} \neq 0$, $\mathbf{x}$ is not a member of the treatment contrast space.

Now let

$$\mathbf{X}_2 = \begin{pmatrix} q\mathbf{T} \otimes \mathbf{1}_k \\ p\mathbf{T} \otimes \mathbf{1}_s \end{pmatrix}$$

where $\mathbf{T}$ is an $r \times (r-1)$ matrix whose columns comprise a set of orthonormal contrasts, that is, $\mathbf{T}'\mathbf{T} = \mathbf{I}$ and $\mathbf{T}'\mathbf{1} = \mathbf{0}$. The columns of $\mathbf{X}_2$ are orthogonal to $\mathbf{x}$, as $\mathbf{X}_2'\mathbf{x} = \mathbf{0}$. Using (3.1) and (3.2) it can be shown that $\mathbf{Y}'\mathbf{Y}\mathbf{X}_2 = \mathbf{0}$, so that the columns of $\mathbf{X}_2$ are eigenvectors of $\mathbf{Y}'\mathbf{Y}$ with zero eigenvalues. The columns of $\mathbf{X}_2$ are also eigenvectors of $\mathbf{V}$ with eigenvalues 2, since $\mathbf{V}\mathbf{X}_2 = 2\mathbf{X}_2$. It follows that $\mathbf{A}_d\mathbf{X}_2 = 2\mathbf{X}_2$. The columns of $\mathbf{X}_2$ correspond to the between replicate contrasts. The effect of replicates is not included in the reduced normal equations, therefore these eigenvectors are not members of the treatment contrast space; see Appendix A.

Now consider

$$\mathbf{X}_0 = \begin{pmatrix} q\mathbf{I}_r \otimes \mathbf{1}_k \\ -p\mathbf{I}_r \otimes \mathbf{1}_s \end{pmatrix}$$

The r columns of $\mathbf{X}_0$ are orthogonal to $\mathbf{x}$ and to the columns of $\mathbf{X}_2$, since $\mathbf{X}_0'\mathbf{x} = \mathbf{0}$ and $\mathbf{X}_0'\mathbf{X}_2 = \mathbf{0}$. Using (3.1) and (3.2) it can be shown that $\mathbf{Y}'\mathbf{Y}\mathbf{X}_0 = \mathbf{0}$ and that $\mathbf{V}\mathbf{X}_0 = \mathbf{0}$. Hence, the columns of $\mathbf{X}_0$ are eigenvectors of $\mathbf{Y}'\mathbf{Y}$ and $\mathbf{V}$ with zero eigenvalues. It follows that $\mathbf{A}_d\mathbf{X}_0 = \mathbf{0}$. These eigenvectors correspond to the between rows and columns within replicate contrasts and are therefore not members of the treatment contrast space.

Any other eigenvector of $\mathbf{V}$ must be orthogonal to $\mathbf{x}$, $\mathbf{X}_0$ and $\mathbf{X}_2$. Let $\mathbf{z} = (\mathbf{w}' \ \ \mathbf{y}')'$ where $\mathbf{w} = (\mathbf{w}_1' \ \ \mathbf{w}_2' \ldots \mathbf{w}_r')'$ and $\mathbf{y} = (\mathbf{y}_1' \ \ \mathbf{y}_2' \ldots \mathbf{y}_r')'$. Let $\mathbf{w}_i$ be vectors of length k and $\mathbf{y}_i$ be vectors of length s $(i = 1, 2, \ldots, r)$. If $\mathbf{z}$ is an eigenvector of $\mathbf{V}$, then for $i = 1, 2, \ldots, r$

$$\begin{aligned}
\mathbf{z}'\mathbf{x} &= q\mathbf{w}'\mathbf{1}_{rk} + p\mathbf{y}'\mathbf{1}_{rs} = 0 \\
\mathbf{z}'\mathbf{X}_0 &= q\mathbf{w}_i'\mathbf{1}_k - p\mathbf{y}_i'\mathbf{1}_s = \mathbf{0} \\
\mathbf{z}'\mathbf{X}_2 &= q\mathbf{w}'(\mathbf{T} \otimes \mathbf{1}_k) + p\mathbf{y}'(\mathbf{T} \otimes \mathbf{1}_s) = \mathbf{0}
\end{aligned}$$

From the first two equations above it can be shown that $\mathbf{w}'\mathbf{1}_{rk} = \mathbf{y}'\mathbf{1}_{rs} = 0$. Given that a column of $\mathbf{T}$ can be an elementary contrast such as $(1/\sqrt{2} \ \ -1/\sqrt{2} \ \ 0 \ldots 0)'$, it follows that $\mathbf{w}_1'\mathbf{1}_k = \mathbf{w}_2'\mathbf{1}_k = \ldots = \mathbf{w}_r'\mathbf{1}_k$ and $\mathbf{y}'\mathbf{1}_s = \mathbf{y}_2'\mathbf{1}_s = \ldots = \mathbf{y}_r'\mathbf{1}_s$, which then implies

$$\mathbf{w}_i'\mathbf{1}_k = \mathbf{y}_i'\mathbf{1}_s = 0 \tag{3.12}$$

Using (3.12) it then follows that $\mathbf{Vz} = \mathbf{z}$, so the remaining $(b - 2r)$ eigenvalues of $\mathbf{V}$ are equal to 1.

Hence, it has been shown that some of the eigenvalues of $\mathbf{Y}'\mathbf{Y}$ are $2r$, and 0 with multiplicity $(2r - 1)$. As mentioned earlier, these $2r$ known eigenvectors are not members of the treatment contrast space. In order to determine the remaining $(b - 2r)$ eigenvalues of $\mathbf{Y}'\mathbf{Y}$ two cases need to be considered, namely designs with $v < b$ and those with $v \geq b$.

First consider a r–replicate resolvable row–column design with $v < b$. As $\mathbf{YY}'$ and $\mathbf{Y}'\mathbf{Y}$ have the same non–zero eigenvalues, $2r$ must also be an eigenvalue of $\mathbf{YY}'$. The remaining $(v - 1)$ eigenvalues $\mathbf{YY}'$ and $\mathbf{Y}'\mathbf{Y}$ have in common will be referred to as non–trivial and will be denoted by λ_i $(i = 1, 2, \ldots, v - 1)$. The eigenvalues of $\mathbf{Y}'\mathbf{Y}$ are therefore

$$\lambda_1, \lambda_2, \ldots, \lambda_{v-1}, 2r$$

with the remaining $(b - v)$ eigenvalues equal to 0. Note that λ_i can be equal to 0 $(i = 1, 2, \ldots, v - 1)$.

From (3.11) and (3.12) it follows that the eigenvalues of $\mathbf{A}_d$ for a resolvable row–column design with $v < b$ are

$$1 - \tfrac{1}{r}\lambda_1, 1 - \tfrac{1}{r}\lambda_2, \ldots, 1 - \tfrac{1}{r}\lambda_{v-1}, \tag{3.13}$$

$(r+1)$ eigenvalues equal to 0, $[(b-2r)-(v-1)] = [r(k+s-2)-(v-1)]$ eigenvalues equal to 1 and $(r-1)$ eigenvalues equal to 2. Note that, using (3.10), the eigenvalues of $\mathbf{A}/r$ are

$$0, 1 - \tfrac{1}{r}\lambda_1, 1 - \tfrac{1}{r}\lambda_2, \ldots, 1 - \tfrac{1}{r}\lambda_{v-1} \tag{3.14}$$

Now consider a r–replicate resolvable row–column design with $v \geq b$. For these designs $\mathbf{Y}'\mathbf{Y}$ will have b eigenvalues, of which it is known from above that one is equal to $2r$ and $(2r-1)$ will be equal to 0. The remaining $(b-2r)$ eigenvalues will be called non–trivial and will be denoted by $\lambda_1, \lambda_2, \ldots, \lambda_{b-2r}$. Using (3.11) and (3.12) it can be shown that the eigenvalues of $\mathbf{A}_d$ are

$$1 - \tfrac{1}{r}\lambda_1, 1 - \tfrac{1}{r}\lambda_2, \ldots, 1 - \tfrac{1}{r}\lambda_{b-2r}, \tag{3.15}$$

$(r+1)$ eigenvalues equal to 0 and $(r-1)$ eigenvalues equal to 2. Using (3.10) the eigenvalues of $\mathbf{A}/r$ are

$$0, 1 - \tfrac{1}{r}\lambda_1, 1 - \tfrac{1}{r}\lambda_2, \ldots, 1 - \tfrac{1}{r}\lambda_{b-2r} \tag{3.16}$$

and $[(v-1)-(b-2r)] = [(v-1)-r(k+s-2)]$ eigenvalues equal to 1.

3.6 Average Efficiency Factors

Having obtained the eigenvalues of $\mathbf{A}/r$ and $\mathbf{A}_d$ in Section 3.5, the average efficiency factors of the resolvable row–column design and the dual design can be calculated. The average efficiency factor E for a r–replicate resolvable row–column design is given by the harmonic mean of the non–zero eigenvalues of $\mathbf{A}/r$. For resolvable row–column designs with $v < b$, the average efficiency factor is given by the harmonic mean of the $(v-1)$ non–zero eigenvalues in (3.14), namely

$$E = \frac{v-1}{\sum_{i=1}^{v-1}(1 - \frac{1}{r}\lambda_i)^{-1}} \tag{3.17}$$

If $v \geq b$, E is still defined as the harmonic mean of the non–zero eigenvalues of $\mathbf{A}/r$, but these now include $[(v-1) - r(k+s-2)]$ eigenvalues equal to 1 and the $r(k+s-2)$ eigenvalues in (3.16). Hence

$$E = \frac{v-1}{\sum_{i=1}^{r(k+s-2)}(1-\frac{1}{r}\lambda_i)^{-1} + (v-1) - r(k+s-2)} \tag{3.18}$$

The average efficiency factor E_d for the dual design of a resolvable row–column design where $v < b$, is the harmonic mean of the $(v-1)$ non–zero eigenvalues of $\mathbf{A}_d$ given in (3.13) and $[r(k+s-2)-(v-1)]$ eigenvalues equal to 1. The $(r-1)$ eigenvalues equal to 2 are not canonical efficiency factors as the corresponding eigenvectors are not members of the treatment contrast space; see Appendix A. These $(r-1)$ eigenvalues are therefore not included in the calculation of E_d. Considering only the non–zero eigenvalues which form the contrast space, E_d is calculated as

$$E_d = \frac{r(k+s-2)}{\sum_{i=1}^{v-1}(1-\frac{1}{r}\lambda_i)^{-1} + r(k+s-2) - (v-1)} \tag{3.19}$$

If $v \geq b$, then E_d is the harmonic mean of the $r(k+s-2)$ canonical efficiency factors of $\mathbf{A}_d$ given in (3.15), and again excluding the $(r-1)$ eigenvalues equal to 2

$$E_d = \frac{r(k+s-2)}{\sum_{i=1}^{r(k+s-2)}(1-\frac{1}{r}\lambda_i)^{-1}} \tag{3.20}$$

The average efficiency factor E for a r–replicate resolvable row–column design can be expressed as a function of the relevant E_d. For resolvable row–column designs with $v < b$, substituting (3.19) into (3.17) gives

$$E = \frac{v-1}{r(k+s-2)E_d^{-1} + [(v-1) - r(k+s-2)]} \tag{3.21}$$

This is a rank adjustment formula analogous to that obtained for resolvable block designs; see John and Williams (1995, p82). The term in square brackets in the denominator represents the difference between the number of non–trivial eigenvalues $(v-1)$ in the row–column design and the number $r(k+s-2)$ in the dual design.

Substituting (3.20) into (3.18) results in an expression for E in terms of E_d for designs with $v \geq b$. This relationship is again given by (3.21).

An alternative expression of E_d requires the calculation of $\mathbf{A}_d^+$, which is the Moore–Penrose inverse of $\mathbf{A}_d$. The sum of the reciprocal of the non–zero eigenvalues of a matrix is equivalent to the trace of the Moore–Penrose inverse of the matrix. For both cases presented above the average efficiency factor of the dual design can be given by

$$E_d = \frac{r(k+s-2)}{\text{trace}(\mathbf{A}_d^+) - (r-1)/2} \tag{3.22}$$

In search algorithms it is often computationally cheaper to invert a matrix and calculate the trace, than to find the canonical efficiency factors of a matrix. The calculation of E_d given by (3.22) will be more efficient to calculate repeatedly in a search algorithm than the approach given by (3.19) and (3.20).

Example (continued)

The non–zero eigenvalues of $\mathbf{Y}'\mathbf{Y}$ for the three–replicate resolvable row–column design given in Table 3.1 are

0.5436,	0.5606,	1.0563,	1.2030,	1.3333,	1.4871,
1.5000,	1.6667,	1.6667,	1.8959,	2.0867,	6

Therefore, the non–zero eigenvalues of $\mathbf{A}_d$ are

0.3044,	0.3680,	0.4444,	0.4444,	0.5000,	0.5043,
0.5556,	0.5990,	0.6479,	0.8131,	0.8188,	1,
1,	1,	1,	2,	2	

E_d is given by the harmonic mean of the first $r(k+s-2) = 15$ non–zero eigenvalues of $\mathbf{A}_d$. As $v < r(k+s)$, E_d is calculated using (3.19), therefore $E_d = 0.578054$. Substituting E_d into (3.21) gives

$$E = \frac{11}{15(0.578054)^{-1} + 11 - 15} = 0.501159$$

The non–zero eigenvalues of $\mathbf{A}/r$ are

0.3044,	0.3680,	0.4444,	0.4444,	0.5000,	0.5043,
0.5556,	0.5990,	0.6479,	0.8131,	0.8188	

The harmonic mean of these eigenvalues is $E = 0.501159$, as before.

3.7 (M,S)–Optimality Criterion

Applying the (M,S)–optimality criterion to the dual of the resolvable row–column designs finds the subclass of designs that maximises $\sum_i e_{di}$, where e_{di} are the canonical efficiency factors of $\mathbf{A}_d$. This is equivalent to maximising trace($\mathbf{A}_d$). Then, within this subclass, those designs that minimise $\sum_i e_{di}^2$ are obtained and this is equivalent to minimising trace($\mathbf{A}_d^2$). Since the non-trivial eigenvalues of $\mathbf{A}_d$ and $\mathbf{A}/r$ are the same, it follows that if a dual design is (M,S)–optimal, then the corresponding resolvable row–column design is also (M,S)–optimal.

From (3.9) it can be shown that

$$\text{trace}(\mathbf{A}_d) = (r-1)(k+s)$$

and

$$\text{trace}(\mathbf{A}_d^2) = \tfrac{(r-1)^2}{r}(k+s+2) + \tfrac{2}{r^2}\text{trace}(\mathbf{HH}') \qquad (3.23)$$

where $\mathbf{H} = (\mathbf{H}_1 \ \mathbf{H}_2 \ \mathbf{H}_3 \ \ldots \ \mathbf{H}_h)$ and recalling that $h = r(r-1)/2$. Since trace($\mathbf{A}_d$) is a constant, (M,S)–optimal designs are obtained by minimising trace($\mathbf{HH}'$).

Working directly with the information matrix $\mathbf{A}$ of the resolvable row–column design, the objective of the (M,S)–optimality is to minimise trace($\mathbf{W}^2$) (see Section 1.3), where

$$\mathbf{W} = \tfrac{1}{s}\mathbf{N}_k\mathbf{N}_k' + \tfrac{1}{k}\mathbf{N}_s\mathbf{N}_s'$$

Trace($\mathbf{W}^2$) can be expressed in terms of the $\mathbf{H}_j$ matrices ($j = 1, 2, \ldots, h$), namely

$$\text{trace}(\mathbf{W}^2) = r(k+s+2) + 2\ \text{trace}(\mathbf{HH}') \qquad (3.24)$$

The (M,S)–optimality objective functions given by trace($\mathbf{A}_d^2$) and trace($\mathbf{W}^2$) are both minimised when trace($\mathbf{HH}'$) is minimised.

Trace($\mathbf{HH}'$) can be expressed in terms of the row and column incidence matrices for each of the contraction arrays and the secondary arrays. Let ss($\mathbf{B}$) represent the sum of squares of the elements of some matrix $\mathbf{B}$. Using (3.7), it follows that

$$\text{trace}(\mathbf{HH}') = \sum_{i=1}^{h}[\text{ss}(\mathbf{R}_{1i})/s^2 + \text{ss}(\mathbf{C}_{1i})/ks + \text{ss}(\mathbf{R}_{2i})/ks + \text{ss}(\mathbf{C}_{2i})/k^2] \qquad (3.25)$$

By working with the (M,S)–optimality function, no matrix inversion or eigenvalue calculations are required. However, to calculate trace($\mathbf{W}^2$) using the row and column concurrence matrices of the row–column design requires considerable computational effort. This effort can be reduced by expressing trace($\mathbf{W}^2$) in terms of trace($\mathbf{HH}'$) and calculating (3.25). From the group of (M,S)–optimal designs, the best design can be found by calculating the average efficiency factors of these designs.

Example (continued)

Using the row and column incidence matrices calculated for the contraction in Table 3.2, trace($\mathbf{HH}'$) is 15.9583. Therefore

$$\text{trace}(\mathbf{A}_d^2) \;=\; \tfrac{2^2}{3}(4+3+2) + \tfrac{2}{3^2}(15.9583) \;=\; 15.5463$$

and

$$\text{trace}(\mathbf{W}^2) \;=\; 3(3+4+2) + 2(15.9583) \;=\; 58.9166$$

3.8 Row and Column Component Designs

The block design given by the rows of the resolvable row–column design is referred to as the row component design. Similarly, the block design given by the columns is called the column component design. Often it is important to ensure a good row and/or column component design, as well as a good row–column design. For instance, if the column factor turns out to be unimportant, then the rows could be used as the blocking factor and the design analysed as a block design. It would be important in this case to ensure that the row component design is a good block design.

Just as the dual design for a resolvable row–column design can be constructed, so can the dual design for a block design. The dual design of a block design is constructed by swapping the treatment and block labels in the original design (John and Williams, 1995, p39). Alternatively, the dual design of the row component design is obtained by simply deleting the second factor from the dual design of the resolvable row–column design. Therefore, the row component dual design is a block design with rk treatments set out in v blocks of size r.

Example (continued)

The dual of the row component design from the row–column design in Table 3.1 is given by deleting the second factor in the design in Table 3.4. This row component

Table 3.5: The dual of the row component design in Table 3.1

Block	1	2	3	4	5	6	7	8	9	10	11	12
	2	3	3	2	2	2	3	3	1	1	1	1
	6	6	6	6	5	4	4	5	4	4	5	5
	9	7	8	9	7	8	7	8	8	9	9	7

dual design is given in Table 3.5.

The information matrix for the dual design of the row component design is given by the square matrix $\mathbf{A}^*_{dk}$ of order rk, namely

$$\mathbf{A}^*_{dk} = s\mathbf{I} - \tfrac{1}{r}\mathbf{N}'_k\mathbf{N}_k$$

The canonical efficiency factors are given by the non–zero, non–trivial eigenvalues of $\mathbf{A}_{dk}$. These exclude the $(r-1)$ eigenvalues equal to 1 which represent the between replicate contrasts. $\mathbf{A}_{dk}$ is given by the top left submatrix of (3.3), that is

$$\mathbf{A}_{dk} = \mathbf{I} - \tfrac{1}{rs}\mathbf{N}'_k\mathbf{N}_k$$

Further from (3.5), $\mathbf{A}_{dk}$ can be expressed in terms of the row incidence matrices of the first factor in the contraction of the resolvable row–column design, that is, $\mathbf{R}_{1j}$ $(j = 1, 2, \ldots, h)$.

The average efficiency factor of the row component dual design is given by the harmonic mean of the canonical efficiency factors of $\mathbf{A}_{dk}$ and will be denoted by E_{dk}. The average efficiency factor E_k of the row component design is given by

$$E_k = \frac{v-1}{r(k-1)E_{dk}^{-1} + (v-1) - r(k-1)} \tag{3.26}$$

For block designs with $rk > v$, E_{dk} is given by

$$E_{dk} = \frac{r(k-1)}{\sum_{i=1}^{v-1} e_i^{-1} + r(k-1) - (v-1)} \tag{3.27}$$

and for block designs with $rk \leq v$

$$E_{dk} = \frac{r(k-1)}{\sum_{i=1}^{r(k-1)} e_i^{-1}}$$

where e_i are the canonical efficiency factors of $\mathbf{A}_{dk}$; see Patterson and Williams (1976b) and John and Williams (1995, p40).

As stated in Section 3.6, it is often computationally less expensive to invert a matrix than to find the canonical efficiency factors of a matrix. For both cases presented above the average efficiency factor of the row component dual design can be given by

$$E_{dk} = \frac{r(k-1)}{\text{trace}(\mathbf{A}_{dk}^{+}) - r + 1} \tag{3.28}$$

where $\mathbf{A}_{dk}^{+}$ is the Moore–Penrose inverse of $\mathbf{A}_{dk}$.

Similarly, the dual of the column component design is given by deleting the first factor in the dual of the resolvable row–column design. The column component dual design is therefore a block design with rs treatments set out in v blocks of size r.

The canonical efficiency factors of the information matrix for the dual of the column component design are given by the non–zero eigenvalues of $\mathbf{A}_{ds}$, again excluding the $(r-1)$ eigenvalues equal to 1 which represent the between replicate contrasts. $\mathbf{A}_{ds}$ is the bottom right sub matrix of (3.3), namely

$$\mathbf{A}_{ds} = \mathbf{I} - \tfrac{1}{rk}\mathbf{N}_s'\mathbf{N}_s$$

From (3.6) it can be seen that $\mathbf{A}_{ds}$ can be expressed in terms of the column incidence matrices for the second factor in the contraction, that is, $\mathbf{C}_{2j}$ $(j = 1, 2, \ldots, h)$.

The average efficiency factor E_s for the column component design is given by

$$E_s = \frac{v-1}{r(s-1)E_{ds}^{-1} + (v-1) - r(s-1)} \tag{3.29}$$

where E_{ds} is the average efficiency factor for the column component dual design. For block designs with $rs > v$

$$E_{ds} = \frac{r(s-1)}{\sum_{i=1}^{v-1} e_i^{-1} + r(s-1) - (v-1)} \tag{3.30}$$

and for designs with $rs \leq v$

$$E_{ds} = \frac{r(s-1)}{\sum_{i=1}^{r(s-1)} e_i^{-1}}$$

where e_i are the canonical efficiency factors of $\mathbf{A}_{ds}$.

For both cases presented above, the average efficiency factor of the column component dual design can be expressed in terms of $\mathbf{A}_{ds}^{+}$, which is the Moore–Penrose inverse of $\mathbf{A}_{ds}$, namely

$$E_{ds} = \frac{r(s-1)}{\text{trace}(\mathbf{A}_{ds}^{+}) - r + 1} \tag{3.31}$$

The matrix $\mathbf{A}_d$ can be formed directly from the row and column incidence matrices of the two factors forming the contraction, that is, $\mathbf{R}_{1j}$, $\mathbf{R}_{2j}$, $\mathbf{C}_{1j}$ and $\mathbf{C}_{2j}$ $(j = 1, 2, \ldots, h)$. The three average efficiency factors of interest, E, E_k and E_s, can all be calculated from $\mathbf{A}_d$. The average efficiency factor E for the resolvable row–column design is calculated as a function of the harmonic mean of the canonical efficiency factors of $\mathbf{A}_d$. The average efficiency factors of the row and column component designs, E_k and E_s respectively, are calculated as a function of the harmonic mean of the canonical efficiency factors of two submatrices of $\mathbf{A}_d$, namely $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ respectively. This is an important feature when considering the computational effort required by search algorithms. By forming and updating only one matrix of order $r(k+s)$, considerable savings can be made in terms of time and computational effort.

Example (continued)

For the three–replicate resolvable row–column design given in Table 3.1, the average efficiency factors for the row and column component dual designs can be obtained using (3.27) and (3.30) respectively. The canonical efficiency factors of $\mathbf{A}_{dk}$ are

$$0.5000, \quad 0.5000, \quad 0.5833, \quad 0.7500, \quad 0.7500, \quad 0.9167$$

and those of $\mathbf{A}_{ds}$ are

$$\begin{array}{lllll} 0.4444, & 0.4444, & 0.5556, & 0.5556, & 0.7778, \\ 0.7778, & 0.7778, & 0.7778, & 0.8889 & \end{array}$$

Hence, $E_{dk} = 0.633455$ and $E_{ds} = 0.626398$. Substituting these values into (3.26) and (3.29) gives the average efficiency factors for the row and column component designs. Respectively these are

$$E_k = \frac{11}{6(0.633455)^{-1} + 11 - 6} = 0.760096$$

$$E_s = \frac{11}{9(0.626398)^{-1} + 11 - 9} = 0.672049$$

Chapter 4

Design Generation

4.1 Introduction

Since the general availability of computers there has been a number of computer algorithms developed to construct resolvable row–column designs; see John (1996). One of the most comprehensive design generation packages is CycDesigN version 2.0 (Whitaker *et al.*, 2002). CycDesigN offers three different methods of constructing resolvable row–column designs. For designs with less than 400 treatments CycDesigN uses an (M,S)–optimality interchange algorithm. To generate designs with 400 or more treatments the recursive method of John and Whitaker (2000) for updating the average efficiency factor after each interchange is used. This approach to generating resolvable row–column designs is discussed in Section 4.7.1. It is also possible to generate resolvable row–column designs in CycDesigN using a two stage algorithm. An optimal or near optimal column component design is constructed at the first stage, and the row component design and the overall row–column design are generated at the second stage. This two stage approach is recommended for large designs and is discussed in Section 4.7.2.

The (M,S)–optimality algorithm in CycDesigN has been amended to enable the construction of resolvable row–column designs with more than 400 treatments. This algorithm, referred to as Cyc($\mathbf{A}$), is discussed in detail in Section 4.2. Also based on the (M,S)–optimality algorithm in CycDesigN is a contraction algorithm Con($\mathbf{A}_d$), which uses the theory in Chapter 3 to generate resolvable row–column designs and is discussed in Section 4.3.

Two further algorithms, Cyc($\mathbf{A}_d$) and Con($\mathbf{A}$), evolved as a consequence of the number of calculations required at several key steps in Cyc($\mathbf{A}$) and Con($\mathbf{A}_d$). By combining features of Cyc($\mathbf{A}$) and Con($\mathbf{A}_d$) further improvements in the speed of design generation were possible. Cyc($\mathbf{A}_d$) and Con($\mathbf{A}$) are discussed in Section 4.4.

With the advances in computer technology, the focus is increasingly being placed on the speed at which these algorithms generate designs. A comparison in the performance of these four algorithms, Cyc($\mathbf{A}$), Cyc($\mathbf{A}_d$), Con($\mathbf{A}_d$) and Con($\mathbf{A}$), in generating resolvable row–column designs is outlined in Section 4.5. Each of the algorithms aims to find an optimal, or at least near optimal, resolvable row–column design. If an optimal design is not found the user needs to intervene and manually terminate the algorithm. Different approaches for choosing when to terminate the algorithm are discussed in Section 4.6.

4.2 Cyc(A) Algorithm

An interchange algorithm begins with the choosing of some arbitrary starting design. Two treatments are then selected from the same replicate and the effect on some objective function of interchanging the treatments is calculated. Based on the interchange decision rule these treatments are either swapped or the design remains unchanged. This process of selecting two treatments and calculating the effect is continued until the stopping criteria is satisfied, or the user intervenes.

The basic steps of the interchange algorithm used by Cyc($\mathbf{A}$) for constructing resolvable row–column designs are:

1. Choose a connected starting design.

   For each replicate the v treatments are randomly assigned to the ks plots. The random assignment is determined by a random number seed which can be chosen by the user or determined by the computer clock. Different random number seeds will lead to different starting designs. The starting design is stored as the best design available.

2. Calculate the row and column concurrence matrices, $\mathbf{N}_k\mathbf{N}_k'$ and $\mathbf{N}_s\mathbf{N}_s'$ respectively.

3. Calculate the objective function, O_1, and store as best O.

4. For the starting design, calculate the average efficiency factors for the row–column design E_1, the row component design E_{k1}, the column component design E_{s1}, and E_{w1}, which is a combined weighted average efficiency factor of E_1, E_{k1} and E_{s1}. Store E_1 and E_{w1} as the best average efficiency factors available.

5. Perturb the resolvable row–column design.

   A *random descent* method is first used where the replicate and treatments for interchange are selected in a systematic way such that all neighbourhood possibilities are considered. When no further improvement can be made to the objective function, *simulated annealing* is used where the replicates and pairs of treatments are selected at random. In simulated annealing it is possible that an interchange is accepted which does not improve the objective function.

6. Calculate the change in the objective function, $\triangle O$.

7. Using the decision rule, decide whether to accept the interchange.

   The decision rule states that an interchange is always accepted if it results in an improvement to the objective function. In the simulated annealing stage of the algorithm, there is a small decreasing probability that an interchange which does not improve the objective function can be accepted. This prevents the algorithm from becoming stuck at local optima; see Whitaker (1995).

8. If the interchange is accepted, swap the interchange treatments, update $\mathbf{N}_k\mathbf{N}_k'$, $\mathbf{N}_s\mathbf{N}_s'$ and the objective function $O_{i+1} = O_i + \triangle O$.

9. If the interchange is accepted and $O_{i+1} \leq O_i$, calculate E_{i+1} and update best O.

10. If $O_{i+1} \leq O_i$ and $E_{i+1} \geq$ best E, calculate $E_{k(i+1)}$, $E_{s(i+1)}$ and $E_{w(i+1)}$.

11. If $E_{i+1} >$ best E or $E_{w(i+1)} >$ best E_{wi}, update the best average efficiency factors and store the design as the best available.

12. Stop the algorithm if $E_{w(i+1)}$ equals the upper bound or the user intervenes, otherwise return to step 5.

Further details for calculating the objective function (step 3), the average efficiency factors (steps 4, 9 and 10), the change in the objective function (step 6) and updating the concurrence matrices (step 8) are given in the following subsections.

4.2.1 Objective Function (Step 3)

Cyc(**A**) aims to find resolvable row–column designs which maximise the average efficiency factor E. These designs are referred to as being A–optimal. It can be computationally expensive to calculate E, so an alternative is to use an (M,S)–optimality criterion which will filter out poor designs. As noted in Section 1.3, it has been conjectured that A–optimal designs are contained within the subclass of (M,S)–optimal designs. Also that a resolvable row–column design is (M,S)–optimal if it minimises trace($\mathbf{W}^2$) where

$$\mathbf{W} = \tfrac{1}{s}\mathbf{N}_k\mathbf{N}_k' + \tfrac{1}{k}\mathbf{N}_s\mathbf{N}_s' \qquad (4.1)$$

Now

$$\text{trace}(\mathbf{W}^2) = \text{trace}(\mathbf{W}_k^2) + \text{trace}(\mathbf{W}_s^2) + 2\,\text{trace}(\mathbf{W}_{ks})$$

where

$$\mathbf{W}_k = \tfrac{1}{s}\mathbf{N}_k\mathbf{N}_k' \qquad \mathbf{W}_s = \tfrac{1}{k}\mathbf{N}_s\mathbf{N}_s' \qquad \mathbf{W}_{ks} = \tfrac{1}{ks}\mathbf{N}_k\mathbf{N}_k'\mathbf{N}_s\mathbf{N}_s'$$

The (M,S)–optimality criterion used by Cyc(**A**) allows for consideration of the row and column component designs as well as the row–column design. This is the approach taken by John and Whitaker (1993). The objective functions for the row and column component designs are to minimise trace($\mathbf{W}_k^2$) and trace($\mathbf{W}_s^2$) respectively. The weighted linear combination of the objective functions for the row and column component designs and the row–column design used by Cyc(**A**) is

$$O_1 = \omega_1\,\text{trace}(\mathbf{W}_k^2) + \omega_2\,\text{trace}(\mathbf{W}_s^2) + \omega_3\,\text{trace}(\mathbf{W}^2) \qquad (4.2)$$

where ω_1, ω_2 and ω_3 are weights. The default weights in Cyc(**A**) are $\omega_1 = 1/k$, $\omega_2 = 1/s$ and $\omega_3 = 1$, giving most of the weight to the row–column design, but also some weight to the component designs. These default weights will be used throughout this chapter.

4.2.2 Average Efficiency Factors (Steps 4, 9 and 10)

The subclass of (M,S)–optimal designs will, in general, still contain a large number of designs. The approach taken by Cyc($\mathbf{A}$) to reduce the search space is to calculate the average efficiency factor only when an improvement is made to the (M,S)–optimality criterion. It is well known that the average efficiency factor can be calculated from the information matrix $\mathbf{A}$ given in (1.2). As given in (1.5), the average efficiency factor E is calculated as

$$E = \frac{v-1}{r\ \mathrm{trace}(\mathbf{A}^{+})}$$

where $\mathbf{A}^{+}$ is the Moore–Penrose inverse of $\mathbf{A}$. Since it is computationally faster to invert a matrix than to obtain eigenvalues and eigenvectors, $\mathbf{A}^{+}$ is calculated using

$$\mathbf{A}^{+} = (\mathbf{A}+\mathbf{J})^{-1} - \mathbf{J}$$

where $\mathbf{J}$ is a matrix of ones. This result follows from the fact that $\mathbf{1}$ is the eigenvector corresponding to the zero eigenvalue; see Section 1.2. Hence, to calculate E in Cyc($\mathbf{A}$) requires the inversion of a matrix of order v.

The matrix $(\mathbf{A}+\mathbf{J})$ is constructed from the row and column concurrence matrices as needed. It is not stored or updated at any stage throughout the algorithm. For simplicity the matrix $(\mathbf{A}+\mathbf{J})$ will be denoted $\mathbf{A}_J$.

If the average efficiency factor E_{i+1} is equal to or better than E_i, the average efficiency factors for the row and column component designs, E_k and E_s respectively, are then calculated. It is well known that E_k and E_s are given by the harmonic mean of the non–zero eigenvalues of the information matrices $\mathbf{A}_k/r$ and $\mathbf{A}_s/r$ respectively, where

$$\mathbf{A}_k = r\mathbf{I} - \tfrac{1}{s}\mathbf{N}_k\mathbf{N}_k' \qquad \mathbf{A}_s = r\mathbf{I} - \tfrac{1}{k}\mathbf{N}_s\mathbf{N}_s'$$

As with the matrix $\mathbf{A}_J$, Cyc($\mathbf{A}$) constructs $\mathbf{A}_k$ and $\mathbf{A}_s$ as needed. They are neither stored nor updated after a treatment interchange is accepted.

The average efficiency factors for the row and column component designs are calculated as

$$E_k = \frac{v-1}{r\ \mathrm{trace}(\mathbf{A}_k^{+})} \qquad E_s = \frac{v-1}{r\ \mathrm{trace}(\mathbf{A}_s^{+})}$$

where

$$\mathbf{A}_k^{+} = (\mathbf{A}_k+\mathbf{J})^{-1} - \mathbf{J} \qquad \mathbf{A}_s^{+} = (\mathbf{A}_s+\mathbf{J})^{-1} - \mathbf{J}$$

These calculations of $\mathbf{A}_k^+$ and $\mathbf{A}_s^+$ are possible as $\mathbf{1}$ is the eigenvector corresponding to the zero eigenvalue for both $\mathbf{A}_k$ and $\mathbf{A}_s$. Therefore, to calculate the average efficiency factors for the row and column component designs in Cyc($\mathbf{A}$) requires the inversion of matrices of order v. For simplicity $(\mathbf{A}_k+\mathbf{J})$ and $(\mathbf{A}_s+\mathbf{J})$ will be denoted $\mathbf{A}_{kJ}$ and $\mathbf{A}_{sJ}$ respectively.

It is possible for the average efficiency factor of the row–column design to be equal for two designs of the same size. In such a case, the row–component design average efficiency factors and/or the column component design average efficiency factors may be different. A weighted average efficiency factor E_w is calculated to determine the overall best design and is given by

$$E_w = \omega_1 E_k + \omega_2 E_s + \omega_3 E$$

where ω_1, ω_2 and ω_3 are the weights defined in Section 4.2.1.

4.2.3 Change in the Objective Function (Step 6)

The change in the objective function, $\triangle O$, is calculated after each interchange and is a computationally time consuming step. It is however, less intensive than recalculating the objective function using (4.2). In Cyc($\mathbf{A}$) the changes to the row and column treatment concurrences are identified so that the changes to trace($\mathbf{W}_k^2$), trace($\mathbf{W}_s^2$) and trace($\mathbf{W}^2$) can be evaluated; see Section 4.2.1 for definitions of $\mathbf{W}_k$, $\mathbf{W}_s$ and $\mathbf{W}$.

First consider the change to trace($\mathbf{W}_k^2$). If the two interchange treatments occur in the same row, the change to trace($\mathbf{W}_k^2$) will be zero. Let t_1 and t_2 in replicate j, say, be the two treatments considered for interchange where t_1 and t_2 occur in rows r_1 and r_2 respectively ($r_1 \neq r_2$). Suppose treatment t is in the same row as t_i in n_i replicates of the resolvable row–column design ($i = 1, 2$). In replicate j, suppose treatment t occurs in the same row as treatment t_1, that is, in row r_1. After the interchange, treatment t will occur in the same row as treatment t_1 in $(n_1 - 1)$ replicates and in the same row as t_2 in $(n_2 + 1)$ replicates. Hence, the change in trace($\mathbf{W}_k^2$) from treatment t is

$$[(n_1 - 1)^2 - n_1^2 + (n_2 + 1)^2 - n_2^2]/s^2 = 2(n_2 - n_1 + 1)/s^2$$

This type of calculation is carried out for each of the $(s - 1)$ treatments that occur in the same row as treatment t_i $(i = 1, 2)$, giving a total of $2(s-1)$ calculations to evaluate the change in trace$(\mathbf{W}_k^2)$.

As the row concurrence matrix $\mathbf{N}_k\mathbf{N}_k'$ is a symmetric matrix, the overall change to trace$(\mathbf{W}_k^2)$ needs to be doubled. This is due to the fact that any changes to the concurrence of treatments t and t_i will have an identical effect on the concurrence of treatments t_i and t $(i = 1, 2)$.

Similarly, the change to trace$(\mathbf{W}_s^2)$ is calculated by considering the treatment concurrences with respect to the columns. Assuming the two interchange treatments occur in different columns, the following type of calculation is performed for the $(k-1)$ treatments that occur in the same column as interchange treatment t_1

$$[(m_1-1)^2 - m_1^2 + (m_2+1)^2 - m_2^2]/k^2 \quad = \quad 2(m_2 - m_1 + 1)/k^2$$

where m_i is the number of replicates where the treatment of interest occurs in the same column as interchange treatment t_i $(i = 1, 2)$. To evaluate the change in trace$(\mathbf{W}_s^2)$, a total of $2(k-1)$ calculations are performed. The results of each of the $2(k-1)$ calculations need to be doubled to account for the symmetric property of the column concurrence matrix.

The change to trace$(\mathbf{W}^2)$ due to a treatment interchange is calculated in a similar way. From Section 4.2.1 trace$(\mathbf{W}^2)$ is given by

$$\text{trace}(\mathbf{W}^2) \quad = \quad \text{trace}(\mathbf{W}_k^2) + \text{trace}(\mathbf{W}_s^2) + 2\,\text{trace}(\mathbf{W}_{ks})$$

where

$$\mathbf{W}_{ks} \quad = \quad \tfrac{1}{ks}\mathbf{N}_k\mathbf{N}_k'\mathbf{N}_s\mathbf{N}_s'$$

It has previously been shown in this section how the changes in trace$(\mathbf{W}_k^2)$ and trace$(\mathbf{W}_s^2)$ are calculated. To calculate the change in trace$(\mathbf{W}_{ks})$ consider two treatments for interchange, say, t_1 and t_2 in replicate j. Let treatment t_1 occur in row r_1 column c_1 in replicate j, and treatment t_2 occur in row r_2 column c_2 in replicate j. Suppose treatment t is in the same row as treatment t_i in n_i replicates and in the same column as treatment t_i in m_i replicates $(i = 1, 2)$.

First, consider the case where $r_1 \neq r_2$ and $c_1 \neq c_2$, and suppose treatment t occurs in row r_1 column c_2. After the interchange of treatments t_1 and t_2, treatment t will occur in the same row as treatment t_1 in $(n_1 - 1)$ replicates, and in the same column as treatment t_1 in $(m_1 + 1)$ replicates. With respect to treatment t_2, after the interchange, treatment t will occur in the same row as treatment t_2 in (n_2+1) replicates and in the same column as treatment t_2 in (m_2-1) replicates. The overall change in trace($\mathbf{W}_{ks}$) due to treatment t is

$$\begin{aligned}&[(n_1-1)(m_1+1)-n_1m_1+(n_2+1)(m_2-1)-n_2m_2]/ks\\&=(n_1-n_2+m_2-m_1-2)/ks\end{aligned}$$

A similar calculation is performed for the treatment which occurs in row r_2 column c_1. For the remaining $2(s-2)$ treatments which occur in rows r_1 and r_2 and the $2(k-2)$ treatments which occur in columns c_1 and c_2, similar calculations to those which follow are performed.

Consider the treatments which occur in columns c_1 and c_2, excluding the treatments in rows r_1 and r_2. Suppose treatment t occurs in the same column as treatment t_1, that is, column c_1. If $c_1=c_2$, the change in trace($\mathbf{W}_{ks}$) due to the column concurrences of treatment t is zero. If $c_1\neq c_2$, after the treatment interchange t will occur in the same column as t_1 in (m_1-1) replicates, and in the same column as t_2 in (m_2+1) replicates. Hence, the change in trace($\mathbf{W}_{ks}$) due to the column concurrence of treatment t is

$$[n_1(m_1-1)-n_1m_1+n_2(m_2+1)-n_2m_2]/ks \quad = \quad (n_2-n_1)/ks$$

Similarly, the change in trace($\mathbf{W}_{ks}$) due to the row concurrence of treatment t will be zero if $r_1=r_2$, else it is

$$[(n_1-1)m_1-n_1m_1+(n_2+1)m_2-n_2m_2]/ks \quad = \quad (m_2-m_1)/ks$$

Due to the symmetric property of the row and column concurrence matrices, the sum of the changes in trace($\mathbf{W}_{ks}$) needs to be multiplied by two. There is in total, $2(k+s-2)$ calculations carried out to determine the change in trace($\mathbf{W}_{ks}$).

4.2.4 Updating $\mathbf{N}_k\mathbf{N}_k'$ and $\mathbf{N}_s\mathbf{N}_s'$ (Step 8)

While the change in the objective function is calculated after each interchange, the row and column concurrence matrices are only updated when an interchange is accepted. If the two treatments being interchanged are from the same row, there is no change to the row concurrence matrix $\mathbf{N}_k\mathbf{N}_k'$. Similarly, if the two interchange treatments are from the same column, the column concurrence matrix $\mathbf{N}_s\mathbf{N}'_s$ does not require updating.

As explained in the previous section, if the interchange treatments are from different rows, and accounting for the symmetric property of $\mathbf{N}_k\mathbf{N}'_k$, $4(s-1)$ elements in the row concurrence matrix have to be changed. Suppose treatments t_1 and t_2 in replicate j are the treatments for interchange and they occur in rows r_1 and r_2 respectively. Say treatment t occurs in the same row as treatment t_i in n_i replicates, then, in terms of the row concurrence matrix, the entry in row t column t_i is n_i $(i = 1, 2)$. Let treatments t and t_1 occur in the same row in replicate j. After the interchange, treatment t will occur in the same row as treatment t_1 in $(n_1 - 1)$ replicates and in the same row as treatment t_2 in $(n_2 + 1)$ replicates. The affect on $\mathbf{N}_k\mathbf{N}'_k$ is for the entry in row t column t_1 to decrease by one to $(n_1 - 1)$, and the entry in row t column t_2 to increase to $(n_2 + 1)$. Similar changes are made for the remaining $(s - 2)$ treatments which occur in the same row as treatment t_1 and the $(s - 1)$ treatments which occur in the same row as treatment t_2. As $\mathbf{N}_k\mathbf{N}'_k$ is a symmetric matrix, any change to the entry in row r_1 column c_1, will result in the same change to row c_1 column r_1. Similarly, $4(k - 1)$ elements in the column concurrence matrix will require updating.

4.3 Con($\mathbf{A}_d$) Algorithm

A contraction algorithm Con($\mathbf{A}_d$) was developed from the (M,S)–optimality algorithm in CycDesigN to generate resolvable row–column designs and uses the theory in Chapter 3. Given the same random number seed, Cyc($\mathbf{A}$) and Con($\mathbf{A}_d$) will generate identical starting designs in step 1 of Section 4.2. The objective function value O_1 (step 3) and the average efficiency factors E_1, E_{k1}, E_{s1} and E_{w1} (step 4) are the same for both algorithms. The choice of treatments to interchange (step 5) and the decision rule for accepting the interchange (step 7) are also the same for the two algorithms. This results in Cyc($\mathbf{A}$) and Con($\mathbf{A}_d$) following the same interchange acceptance path for the same random number seed. Hence, both algorithms will generate identical resolvable row–column designs.

While both algorithms calculate the same objective function value and average efficiency factors, the method of calculating these values are different. Other differences between Con($\mathbf{A}_d$) and Cyc($\mathbf{A}$) are in the matrices formed and updated (steps 2 and 8) and the method of calculating the change to the objective function (step 6). The methods used by Con($\mathbf{A}_d$) are outlined in the following subsections.

4.3.1 Design Matrices (Step 2)

From the starting design Con($\mathbf{A}_d$) constructs two location arrays, the row and column incidence matrices of the contraction arrays and the secondary arrays, and $\mathbf{A}_d$ as given in (3.3). The location arrays $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are $v \times r$ matrices. $\mathbf{Z}_1$ gives the row position of each treatment within each replicate and $\mathbf{Z}_2$ gives the column position. Let

$$\mathbf{Z}_i = \begin{pmatrix} \mathbf{Z}_{i1} & \mathbf{Z}_{i2} & \ldots & \mathbf{Z}_{ir} \end{pmatrix}$$

where $i = 1, 2$ and $\mathbf{Z}_{1j}$ is the row location array for treatments in replicate j and $\mathbf{Z}_{2j}$ is the column location array for treatments in replicate j ($j = 1, 2, \ldots, r$).

Example

The three–replicate resolvable row–column design given in Table 3.1 is reproduced in Table 4.1.

Table 4.1: Resolvable row–column design for $v = 12$, $k = 3$, $s = 4$, $r = 3$

Replicate		1				2				3			
Column		1	2	3	4	1	2	3	4	1	2	3	4
Row	1	11	12	9	10	7	9	10	6	**5**	2	7	12
	2	5	6	4	**1**	12	8	5	11	9	6	3	8
	3	**2**	3	7	8	1	2	3	4	1	10	11	4

The location arrays for this design are

$$\mathbf{Z}_1 = \begin{pmatrix} 2 & 3 & 3 \\ 3 & 3 & 1 \\ 3 & 3 & 2 \\ 2 & 3 & 3 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 1 \\ 3 & 2 & 2 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 3 \\ 1 & 2 & 1 \end{pmatrix} \qquad \mathbf{Z}_2 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 3 & 3 \\ 3 & 4 & 4 \\ 1 & 3 & 1 \\ 2 & 4 & 2 \\ 3 & 1 & 3 \\ 4 & 2 & 4 \\ 3 & 2 & 1 \\ 4 & 3 & 2 \\ 1 & 4 & 3 \\ 2 & 1 & 4 \end{pmatrix}$$

For instance, the first column of $\mathbf{Z}_1$ specifies the rows that contain each treatment in replicate 1. For example, in replicate 1, treatment 1 occurs in row 2 and treatment 2 occurs in row 3. The first two elements in column 1 of $\mathbf{Z}_1$ are therefore 2 and 3 respectively. The first column of $\mathbf{Z}_2$ specifies the column positions of the treatments in replicate 1. Treatment 1 occurs in column 4 and treatment 2 occurs in column 1 in replicate 1.

As a further example, the third column of $\mathbf{Z}_1$ gives the row position of each treatment in replicate 3, and the third column of $\mathbf{Z}_2$ gives the column position of treatments in replicate 3. For example, treatment 5 in replicate 3 occurs in row 1 column 1, hence, row 5 column 3 of both $\mathbf{Z}_1$ and $\mathbf{Z}_2$ contain a 1.

The row and column incidence matrices for the contraction arrays and the secondary arrays can be readily obtained using pairs of columns from the location arrays, as follows

$$\begin{array}{llcll} \mathbf{R}_{1j} & \left\{\begin{pmatrix} \mathbf{Z}_{1i} & \mathbf{Z}_{1m} \end{pmatrix}\right\} & & \mathbf{R}_{2j} & : \left\{\begin{pmatrix} \mathbf{Z}_{2i} & \mathbf{Z}_{1m} \end{pmatrix}\right\} \\ \mathbf{C}_{1j} & \left\{\begin{pmatrix} \mathbf{Z}_{1i} & \mathbf{Z}_{2m} \end{pmatrix}\right\} & & \mathbf{C}_{2j} & : \left\{\begin{pmatrix} \mathbf{Z}_{2i} & \mathbf{Z}_{2m} \end{pmatrix}\right\} \end{array}$$

For the contraction arrays, $j < r$, $i = 1$ and $m = j + 1$. For instance, for designs with more than two replicates, $\mathbf{R}_{11}$ is obtained from columns $i = 1$ and $m = 2$ of $\mathbf{Z}_1$, while $\mathbf{C}_{12}$ is obtained from column $i = 1$ of $\mathbf{Z}_1$ and column $m = 3$ of $\mathbf{Z}_2$.

For the secondary arrays, $j \geq r$, and i is the smallest integer satisfying $i(2r - i - 1) \geq 2j$ and $m = [j - (i-1)r + i(i+1)/2]$. For instance, for a three–replicate resolvable row–column design, $\mathbf{R}_{13}$ is obtained from columns $i = 2$ and $m = 3$ of $\mathbf{Z}_1$, and $\mathbf{R}_{23}$ is obtained from column $i = 2$ of $\mathbf{Z}_2$ and column $m = 3$ of $\mathbf{Z}_1$.

The row and column incidence matrices are constructed from these column pairs. The (wx)th element of each incidence matrix is given by the number of rows containing both w and x in the first and second columns respectively of the pair. For instance, $\mathbf{R}_{11}$ is constructed from the elements in the first two columns of $\mathbf{Z}_1$, that is $\mathbf{Z}_{11}$ and $\mathbf{Z}_{12}$.

Example (continued)

Consider the column incidence matrix for factor 2 in array 1, $\mathbf{C}_{21}$, which is constructed from the elements in $\mathbf{Z}_{21}$ and $\mathbf{Z}_{22}$. The entry in row w column x of $\mathbf{C}_{21}$ is equal to the number of rows containing both w in $\mathbf{Z}_{21}$ and x in $\mathbf{Z}_{22}$. For instance, in the previous example the elements 1 and 2 occur together in $\mathbf{Z}_{21}$ and $\mathbf{Z}_{22}$ in one row, namely row 2. Hence, the entry in row 1 column 2 of $\mathbf{C}_{21}$ will be 1. Similarly, since 1 and 1 do not occur in the same rows of $\mathbf{Z}_{21}$ and $\mathbf{Z}_{22}$, the entry in row 1 column 1 of $\mathbf{C}_{21}$ will be 0.

Having calculated the row and column incidence matrices, these can be used to construct $\mathbf{A}_d$ in the form given by (3.3). $\mathbf{A}_d$ is used in the calculation of the average efficiency factors, but as will be shown in Section 4.3.3, to simplify calculations a matrix $\mathbf{MM}'$ is added to $\mathbf{A}_d$. It is $(\mathbf{A}_d + \mathbf{MM}')$ which is constructed, stored, and updated after an interchange is accepted.

4.3.2 Objective Function (Step 3)

Con($\mathbf{A}_d$) uses the same objective function as Cyc($\mathbf{A}$), namely (4.2), but it is calculated using the row and column incidence matrices of the contraction arrays and secondary arrays. The objective functions for the row and column component designs, trace($\mathbf{W}_k^2$) and trace($\mathbf{W}_s^2$) respectively, are

$$\text{trace}(\mathbf{W}_k^2) \quad = \quad 2\sum_{i=1}^{h} \text{ss}(\mathbf{R}_{1i})/s^2 + rk \qquad (4.3)$$

$$\text{trace}(\mathbf{W}_s^2) = 2\sum_{i=1}^{h} \text{ss}(\mathbf{C}_{2i})/k^2 + rs \quad (4.4)$$

where $h = r(r-1)/2$ and ss($\mathbf{B}$) represents the sum of the squares of the elements of some matrix $\mathbf{B}$.

Trace($\mathbf{W}^2$) is the objective function for the resolvable row–column design, where $\mathbf{W}$ is given by (4.1), and as previously given in (3.24), can be expressed as

$$\text{trace}(\mathbf{W}^2) = r(k+s+2) + 2\,\text{trace}(\mathbf{HH}') \quad (4.5)$$

where trace($\mathbf{HH}'$) is given in (3.25) as

$$\text{trace}(\mathbf{HH}') = \sum_{i=1}^{h}[\text{ss}(\mathbf{R}_{1i})/s^2 + \text{ss}(\mathbf{C}_{1i})/ks + \text{ss}(\mathbf{R}_{2i})/ks + \text{ss}(\mathbf{C}_{2i})/k^2]$$

Substituting (4.3), (4.4) and (4.5) into (4.2) produces the same objective function value for a given design as would be obtained by Cyc($\mathbf{A}$).

4.3.3 Average Efficiency Factors (Steps 4, 9 and 10)

Con($\mathbf{A}_d$) calculates the average efficiency factor E of the resolvable row–column design as a function of the average efficiency factor E_d of the dual design. E_d is calculated using the formula given by (3.22), namely

$$E_d = \frac{r(k+s-2)}{\text{trace}(\mathbf{A}_d^+) - (r-1)/2} \quad (4.6)$$

where $\mathbf{A}_d^+$ is the Moore–Penrose inverse of $\mathbf{A}_d$. The average efficiency factor E of the resolvable row–column design is then calculated using (3.21), which is

$$E = \frac{v-1}{r(k+s-2)E_d^{-1} + (v-1) - r(k+s-2)} \quad (4.7)$$

The Moore–Penrose inverse $\mathbf{A}_d^+$ can be calculated by finding the non–zero eigenvalues of $\mathbf{A}_d$ and the corresponding eigenvectors. Alternatively, a computationally faster method is to calculate

$$\mathbf{A}_d^+ = (\mathbf{A}_d + \mathbf{MM}')^{-1} - \mathbf{MM}'$$

which involves inverting a matrix of order $r(k+s)$. For simplicity, the matrix ($\mathbf{A}_d$+$\mathbf{MM}'$) will be denoted $\mathbf{A}_M$. The columns of $\mathbf{M}$ are the normalised eigenvectors of $\mathbf{A}_d$ corresponding to the $(r+1)$ zero eigenvalues, that is $\mathbf{A}_d\mathbf{M} = \mathbf{0}$. Following from Section 3.5, the normalised eigenvectors of $\mathbf{A}_d$ with zero eigenvalues are $\mathbf{x}$ and the columns of $\mathbf{X}_0$, where

$$\mathbf{x} = \frac{1}{\sqrt{2r}} \begin{pmatrix} q\mathbf{1}'_{rk} & p\mathbf{1}'_{rs} \end{pmatrix}' \tag{4.8}$$

and

$$\mathbf{X}_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} q\mathbf{I}_r \otimes \mathbf{1}_k \\ -p\mathbf{I}_r \otimes \mathbf{1}_s \end{pmatrix} \tag{4.9}$$

where $p^2 = 1/s$ and $q^2 = 1/k$. Therefore, using (4.8) and (4.9), $\mathbf{M} = (\mathbf{x} \;\; \mathbf{X}_0)$ and it follows that

$$\begin{aligned} \mathbf{MM}' &= \frac{1}{2r} \begin{pmatrix} q^2\mathbf{J}_{nn} + rq^2(\mathbf{I}_r \otimes \mathbf{J}_{kk}) & pq\mathbf{J}_{nm} - rpq(\mathbf{I}_r \otimes \mathbf{J}_{ks}) \\ pq\mathbf{J}_{mn} - rpq(\mathbf{I}_r \otimes \mathbf{J}_{sk}) & p^2\mathbf{J}_{mm} + rp^2(\mathbf{I}_r \otimes \mathbf{J}_{ss}) \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix} \end{aligned}$$

where $n = rk$ and $m = rs$. Hence

$$\begin{aligned} \text{trace}(\mathbf{A}_d^+) &= \text{trace}(\mathbf{A}_M^{-1}) - \text{trace}(\mathbf{MM}') \\ &= \text{trace}(\mathbf{A}_M^{-1}) - (r+1) \end{aligned} \tag{4.10}$$

Substituting (4.10) for $\text{trace}(\mathbf{A}_d^+)$ in (4.6), and then substituting (4.6) for E_d in (4.7), gives the average efficiency factor E of the resolvable row–column design.

As with Cyc($\mathbf{A}$), Con($\mathbf{A}_d$) only calculates the average efficiency factors for the row and column component designs if E_{i+1} is equal to, or an improvement on, the best average efficiency factor. The average efficiency factors for the component designs are calculated as a function of the average efficiency factors for the component dual designs; see Section 3.8. It is known that the average efficiency factors for the row and column component dual designs can be obtained from the non–zero, non–trivial eigenvalues of $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ respectively. $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ are submatrices of $\mathbf{A}_d$, where $\mathbf{A}_{dk}$ is the top left submatrix and $\mathbf{A}_{ds}$ is the bottom right submatrix.

The average efficiency factor of the row component dual design is calculated using the formula given by (3.28), namely

$$E_{dk} = \frac{r(k-1)}{\text{trace}(\mathbf{A}_{dk}^+) - r + 1}$$

where $\mathbf{A}_{dk}^+$ is the Moore–Penrose inverse of $\mathbf{A}_{dk}$. The average efficiency factor of the row component design is then calculated using (3.26), which is

$$E_k = \frac{v-1}{r(k-1)E_{dk}^{-1} + (v-1) - r(k-1)}$$

Con($\mathbf{A}_d$) stores the matrix $\mathbf{A}_M$ which makes it computationally inexpensive to calculate the trace of the Moore–Penrose inverses for the average efficiency factors. The trace of the Moore–Penrose inverse of $\mathbf{A}_{dk}$ can be calculated as

$$\text{trace}(\mathbf{A}_{dk}^{+}) = \text{trace}(\mathbf{A}_{Mk}^{-1}) + \tfrac{(r-4)}{3}$$

where $\mathbf{A}_{Mk} = \mathbf{A}_{dk} + \mathbf{M}_{11}$. Therefore, to calculate trace($\mathbf{A}_{dk}^{+}$) involves inverting a matrix of order rk and the proof of this result is given in Appendix B.

Similarly, using (3.31), the average efficiency factor of the column component dual design is given by

$$E_{ds} = \frac{r(s-1)}{\text{trace}(\mathbf{A}_{ds}^{+}) - r + 1}$$

where $\mathbf{A}_{ds}^{+}$ is the Moore–Penrose inverse of $\mathbf{A}_{ds}$ and

$$\text{trace}(\mathbf{A}_{ds}^{+}) = \text{trace}(\mathbf{A}_{Ms}^{-1}) - \tfrac{(r-4)}{3}$$

where $\mathbf{A}_{Ms} = \mathbf{A}_{ds} + \mathbf{M}_{22}$. Hence, calculating trace($\mathbf{A}_{ds}^{+}$) requires inverting a matrix of order rs. The average efficiency factor of the column component design is then given by (3.29), namely

$$E_s = \frac{v-1}{r(s-1)E_{ds}^{-1} + (v-1) - r(s-1)}$$

4.3.4 Change in the Objective Function (Step 6)

As stated in Section 4.2.3, the change in the objective function is calculated after each interchange and is computationally time consuming. In Con($\mathbf{A}_d$), calculating this change involves calculating the change in the sums of squares of the row and column incidence matrices of the contraction arrays and secondary arrays.

Throughout this thesis the contraction has been considered with respect to the first replicate of the resolvable row–column design. Any treatment interchanges in the first replicate will affect all arrays forming the contraction. That is, changes will occur to $(r-1)$ arrays and hence, will affect $(r-1)$ row and column incidence matrices for each factor. Since the contraction can be defined with respect to any replicate, it follows that interchanging two treatments in any replicate will affect $(r-1)$ row incidence matrices and $(r-1)$ column incidence matrices for each factor.

Recall from Section 4.3.1 that the row and column incidence matrices are obtained from pairs of columns in the location arrays $\mathbf{Z}_1$ and $\mathbf{Z}_2$. Suppose the two treatments for interchange are t_1 and t_2 in replicate j. The current row positions of the interchange treatments can be obtained from $\mathbf{Z}_1$ and the column positions from $\mathbf{Z}_2$. If t_1 and t_2 occur in the same row, no changes are made to the row and column incidence matrices obtained from column j of $\mathbf{Z}_1$, that is $\mathbf{Z}_{1j}$. Similarly, if t_1 and t_2 occur in the same column, the row and column incidence matrices formed from $\mathbf{Z}_{2j}$ are unaffected. Therefore, the change in the sum of squares for these matrices is equal to zero.

For the row and column incidence matrices which are affected, four changes will occur to each matrix. Two elements in each matrix will increase by one, and two will decrease by one. Suppose, for example, treatments 1 and 2 in replicate 2 are chosen for interchange, where treatment 1 occurs in row x and treatment 2 occurs in row z. Let the first two rows of $\mathbf{Z}_1$ before the interchange take the form on the left, and the corresponding rows after the interchange take the form on the right

$$\begin{pmatrix} w & x & \dots \\ y & z & \dots \end{pmatrix} \qquad \begin{pmatrix} w & z & \dots \\ y & x & \dots \end{pmatrix}$$

where w, x, y and z are all positive integers less than or equal to k. Consider $\mathbf{R}_{11}$ which is formed from the concurrences of elements in $\mathbf{Z}_{11}$ and $\mathbf{Z}_{12}$. Interchanging treatments 1 and 2 would result in the (wx)th and (yz)th elements of $\mathbf{R}_{11}$ decreasing by one and the (wz)th and (yx)th elements increasing by one.

Assuming that a particular incidence matrix is affected by the interchange, let a_1 and a_2 be the current values in the matrix which are increasing by one, and d_1 and d_2 be the values which are decreasing by one. The change in the sum of squares for the incidence matrix is given by

$$\begin{aligned} &(a_1+1)^2 - a_1^2 + (a_2+1)^2 - a_2^2 + (d_1-1)^2 - d_1^2 + (d_2-1)^2 - d_2^2 \\ &= 2(a_1 + a_2 - d_1 - d_2) + 4 \end{aligned} \tag{4.11}$$

This type of calculation is carried out for each of the $(r-1)$ row and column incidence matrices that may be affected by the interchange. Therefore, in order to calculate the change in the objective function, a maximum of $2(r-1)$ calculations need to be made and summed.

Example (continued)

Suppose in replicate 2 of the resolvable row–column design in Table 4.1, treatments 1 and 2 are selected for interchange. The first two rows of $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are

$$\mathbf{Z}_1 \rightarrow \begin{pmatrix} 2 & \mathbf{3} & 3 \\ 3 & \mathbf{3} & 1 \end{pmatrix} \qquad \mathbf{Z}_2 \rightarrow \begin{pmatrix} 4 & 1 & 1 \\ 1 & 2 & 2 \end{pmatrix}$$

After the interchange they become

$$\mathbf{Z}_1 \rightarrow \begin{pmatrix} 2 & \mathbf{3} & 3 \\ 3 & \mathbf{3} & 1 \end{pmatrix} \qquad \mathbf{Z}_2 \rightarrow \begin{pmatrix} 4 & \mathbf{2} & 1 \\ 1 & \mathbf{1} & 2 \end{pmatrix}$$

From $\mathbf{Z}_1$ it can be seen that treatments 1 and 2 in replicate 2 both occur in row 3. Therefore, there is no change to the sum of squares for the row and column incidence matrices constructed from $\mathbf{Z}_{12}$. These matrices are $\mathbf{R}_{11}$, $\mathbf{R}_{13}$, $\mathbf{R}_{21}$ and $\mathbf{C}_{13}$.

One incidence matrix which is affected by the interchange is $\mathbf{R}_{23}$ which is formed by considering $\mathbf{Z}_{22}$ and $\mathbf{Z}_{13}$. The concurrences in $\mathbf{Z}_{22}$ and $\mathbf{Z}_{13}$ which are no longer present after the interchange are (1,3) and (2,1). Therefore, the (13)th and (21)th elements of $\mathbf{R}_{23}$ will decrease by one. The new concurrences which occur after the interchange are (2,3) and (1,1). Hence the elements which increase by one are the (11)th and the (23)th. Given that $\mathbf{R}_{23}$ is currently equal to

$$\mathbf{R}_{23} = \begin{pmatrix} \mathbf{2} & 0 & \mathbf{1} \\ \mathbf{1} & 2 & \mathbf{0} \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

the change in the sum of squares is calculated by letting a_1 and a_2 be the values which are increasing by one, and d_1 and d_2 be the values decreasing by one. That is, $a_1 = 2$, $a_2 = 0$, $d_1 = 1$ and $d_2 = 1$. Using (4.11), the change in the sum of squares for $\mathbf{R}_{23}$ is $2(2 + 0 - 1 - 1) + 4 = 4$.

By calculating the change in the sum of squares for the affected matrices, namely $\mathbf{C}_{11}$, $\mathbf{C}_{21}$, $\mathbf{C}_{23}$ and $\mathbf{R}_{23}$, the overall change in the objective function is +8. This corresponds to an increase in the objective function.

4.3.5 Updating the Design Matrices (Step 8)

Each time an interchange is accepted, the location arrays, the row and column incidence matrices, and $\mathbf{A}_M$ are updated. Suppose that treatments t_1 and t_2 in replicate j are interchanged. The location arrays are updated by swapping the elements in rows t_1 and t_2 of column j.

As seen in the previous section, if the treatments being interchanged are from the same row or column, some incidence matrices will not be affected. The matrices that are affected are updated by adding or subtracting one from the four elements identified when calculating the change in the objective function; see Section 4.3.4. The updated incidence matrices are then stored for future use.

By expressing $\mathbf{A}_d$ in the form given by (3.3), the elements of $\mathbf{A}_M$ which need updating can be easily located. Any changes to the row and column incidence matrices will lead to a corresponding change in $\mathbf{A}_M$.

Example (continued)

The row and column incidence matrices for the contraction of the three–replicate resolvable row–column design in Table 4.1 are given in Section 3.2. Suppose treatments 1 and 2 in replicate 2 are accepted for interchange. Replicate 2 is used in the formation of arrays 1 and 3, therefore these are the $(r-1) = 2$ arrays which will require updating. In replicate 2, treatment 1 occurs in row 3 column 1 and treatment 2 occurs in row 3 column 2. As treatments 1 and 2 occur in the same row, there will be no change to the row and column incidence matrices which are constructed from $\mathbf{Z}_{12}$. The matrices which do not need updating are therefore $\mathbf{R}_{11}$, $\mathbf{R}_{13}$, $\mathbf{R}_{21}$ and $\mathbf{C}_{13}$. After the interchange, the row and column incidence matrices for arrays 1 and 3 which do require updating become

$$\mathbf{C}_{21} = \begin{pmatrix} \mathbf{1} & \mathbf{0} & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ \mathbf{0} & \mathbf{2} & 1 & 0 \end{pmatrix} \qquad \mathbf{C}_{23} = \begin{pmatrix} \mathbf{0} & \mathbf{1} & 1 & 1 \\ \mathbf{2} & \mathbf{0} & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{C}_{11} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ \mathbf{0} & \mathbf{1} & 1 & 2 \\ \mathbf{2} & \mathbf{1} & 1 & 0 \end{pmatrix} \qquad \mathbf{R}_{23} = \begin{pmatrix} \mathbf{3} & 0 & \mathbf{0} \\ \mathbf{0} & 2 & \mathbf{1} \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix}$$

4.4 Comparison of Cyc(A) and Con($\mathbf{A}_d$) Algorithms

Given the same random number seed, Cyc($\mathbf{A}$) and Con($\mathbf{A}_d$) will generate the same starting design and accept the same treatment interchanges. This results in the two algorithms generating identical resolvable row–column designs. The primary interest now is which algorithm is quicker at generating these designs.

The first four steps in each algorithm, as set out in Section 4.2, are only executed once and will have little impact on the speed of the algorithm. The steps that will have the most effect are steps 6, 8, 9 and 10, namely calculating the change in the objective function after each interchange, and when the interchange is accepted, updating the design matrices and recalculating the average efficiency factors.

The number of C++ computer statements executed by Cyc($\mathbf{A}$) to calculate the change in the objective function is of order $(k + s)$. When a treatment interchange is accepted, the row and column concurrence matrices of the row–column design are updated which requires the execution of computer statements also of order $(k + s)$. If the interchange is accepted and there is an improvement in the objective function, the average efficiency factor for the resolvable row–column design is calculated using the matrix $\mathbf{A}_J$. To calculate the average efficiency factor requires inverting this $v \times v$ matrix. Cyc($\mathbf{A}$) has to construct the $v \times v$ matrix $\mathbf{A}_J$ from the row and column concurrence matrices of the row–column design each time the average efficiency factor E is calculated. The average efficiency factors for the row and column component designs and the weighted average efficiency factor, E_k, E_s and E_w respectively, are calculated if the average efficiency factor of the resolvable row–column design equals, or betters, the best E. In order to calculate E_k and E_s, Cyc($\mathbf{A}$) has to construct the $v \times v$ matrices $\mathbf{A}_{kJ}$ and $\mathbf{A}_{sJ}$.

In Con($\mathbf{A}_d$) the number of C++ computer statements executed to calculate the change in the objective function is of order r. Each time an interchange is accepted, the design matrices are updated and this also requires computer statements of order r to be executed. If the interchange is accepted and there is an improvement in the objective function, the average efficiency factor E for the resolvable row–column design is calculated by inverting the matrix $\mathbf{A}_M$. $\mathbf{A}_M$ is a $r(k + s)$ square matrix which is stored and updated each time an interchange is accepted. If E is at least as good as the best E, the average efficiency factors E_k, E_s and E_w are calculated. The matrices inverted to calculate E_k and E_s are the $rk \times rk$ matrix $\mathbf{A}_{Mk}$ and the $rs \times rs$ matrix $\mathbf{A}_{Ms}$ respectively, which are extracted from $\mathbf{A}_M$.

In most, if not all practical cases, the number of replicates r is considerably less than $(k+s)$, especially for designs involving a large number of treatments. Therefore, Con($\mathbf{A}_d$) will be quicker at calculating the change in the objective function and updating the design matrices for most resolvable row–column designs.

To calculate the average efficiency factors for the resolvable row–column design and the row and column component designs requires inverting three matrices. The smaller the order of the matrix, the quicker the algorithm is able to perform the calculations. For designs where $v < r(k+s)$, it will be quicker to calculate the average efficiency factor for the resolvable row–column design using Cyc($\mathbf{A}$). However, Cyc($\mathbf{A}$) is slowed down by having to construct the matrix $\mathbf{A}_J$ each time E is to be calculated. The equivalent matrix in Con($\mathbf{A}_d$), $\mathbf{A}_M$, is stored and updated each time an interchange is accepted and is readily available. Therefore, if v is not much smaller than $r(k+s)$, the overhead in Cyc($\mathbf{A}$) may result in Con($\mathbf{A}_d$) calculating E quicker. For designs where $v \geq r(k+s)$, Con($\mathbf{A}_d$) will be able to calculate the average efficiency factor for the resolvable row–column design more efficiently.

The average efficiency factors for the row and column component designs are calculated by inverting matrices of order v using Cyc($\mathbf{A}$), and of order rk and rs respectively using Con($\mathbf{A}_d$). For designs with a large number of treatments and small replication, Con($\mathbf{A}_d$) will be more efficient at calculating these average efficiency factors.

The speed of generating designs with a large number of treatments could be improved using Cyc($\mathbf{A}$) by calculating the average efficiency factors from the matrix $\mathbf{A}_M$. The algorithm would be required to construct $\mathbf{A}_M$ each time the average efficiency factor E is calculated, which involves constructing the location arrays and the row and column incidence matrices of the contraction arrays and secondary arrays. For designs where v is only slightly larger than $r(k+s)$, these extra calculations may overshadow the benefits of inverting the smaller $r(k+s)$ square matrix to obtain E. If v is considerably greater than $r(k+s)$, then the expense of constructing $\mathbf{A}_M$ will be outweighed by the algorithm's ability to quickly invert a square matrix of order $r(k+s)$. In terms of calculating the row and column component average efficiency factors, a large saving will be made by using the submatrices of $\mathbf{A}_M$ for designs where $v > r(k+s)$. Rather than constructing and inverting $v \times v$ matrices, matrices of order rk and rs can be obtained from $\mathbf{A}_M$ to calculate E_k and E_s respectively. Cyc($\mathbf{A}$) was modified to allow for the average efficiency factors for the resolvable row–column design and the row and column component designs to be calculated via $\mathbf{A}_M$, $\mathbf{A}_{Mk}$ and $\mathbf{A}_{Ms}$ respectively, and this algorithm is called Cyc($\mathbf{A}_d$).

Con($\mathbf{A}_d$) will not be as efficient as Cyc($\mathbf{A}$) at calculating the average efficiency factor for resolvable row–column designs when $v < r(k+s)$. For such designs, the speed of calculating the average efficiency factor E in Con($\mathbf{A}_d$) can be improved by using the matrix $\mathbf{A}_J$, rather than $\mathbf{A}_M$. This would involve the algorithm constructing the row and column concurrence matrices in order to form $\mathbf{A}_J$ each time the average efficiency factor is to be calculated. Con($\mathbf{A}_d$) was amended such that E could be calculated using $\mathbf{A}_J$ and this fourth algorithm is called Con($\mathbf{A}$). For designs where v is considerably less than $r(k+s)$, the benefits of inverting the smaller $\mathbf{A}_J$ matrix should outweigh the overhead of constructing the required matrices. To improve the speed of calculating the average efficiency factors for the row and column component designs when $v < r(k+s)$, the matrices $\mathbf{A}_{kJ}$ and $\mathbf{A}_{sJ}$ should be used; see Section 4.2.2. As $\mathbf{A}_M$ is no longer required to calculate the average efficiency factors using Con($\mathbf{A}$), it does not feature at all in the modified algorithm. However, the row and column incidence matrices of the contraction arrays and secondary arrays will still need to be stored and updated, as they are required for calculating the change to the objective function.

4.5 Performance of the Algorithms

The primary interest in comparing the four algorithms is the speed in which optimal or near optimal resolvable row–column designs are generated. The same set of random number seeds were used for each algorithm to ensure they accepted the same interchanges and, hence, generated the same designs. This allows for a direct time comparison for generating a design with a given average efficiency factor. All runs of the four algorithms were carried out on a PC computer running at 1200MHz, with no other programs or software running simultaneously.

To compare the speed of design generation, Con($\mathbf{A}_d$) was run for approximately

60 seconds and the average efficiency factor for the best resolvable row–column design was recorded. Cyc($\mathbf{A}$), Cyc($\mathbf{A}_d$) and Con($\mathbf{A}$) were then run until this same design was generated and the time taken was noted. Although the effect of randomness is present due to the choice of the random number seed, multiple runs of each algorithm using different random number seeds produced similar results to those presented.

Table 4.2 gives the average efficiency factor E and the number of seconds taken for each algorithm to generate some designs with $v < r(k+s)$. In general, it can be seen that the contraction algorithms outperform the CycDesigN algorithms. When v is only marginally smaller than $r(k+s)$, the contraction algorithms are much quicker at generating resolvable row–column designs. It can be seen from Table 4.2 that Cyc($\mathbf{A}$) outperforms Con($\mathbf{A}_d$) when generating designs where v is greater than 100 and $r(k+s)$ is much larger than v. This can be explained by the method of calculating the average efficiency factors; see Section 4.4. The final design in Table 4.2, where $v = 250$, shows that for a large design, the overhead in Con($\mathbf{A}$) of constructing the row and column concurrence matrices of the row–column design each time E is calculated, has outweighed the benefits of the contraction algorithm. This has resulted in Cyc($\mathbf{A}$) outperforming the other three algorithms.

From Table 4.2 it can also be seen that when v is less than 100, there is little difference in the speed of design generation between the two CycDesigN algorithms. As the number of treatments increases, differences in the speed of the two algorithms become apparent. For designs where v is large, yet considerably smaller than $r(k+s)$, Cyc($\mathbf{A}$) is able to generate designs much faster than Cyc($\mathbf{A}_d$). However, for designs where v is large but only marginally smaller than $r(k+s)$, Cyc($\mathbf{A}_d$) is the faster of the two CycDesigN algorithms. An explanation for this is how the average efficiency factors for the row and column component designs are calculated; see Section 4.4.

Similar results are seen when comparing the contraction algorithms. For resolvable row–column designs with less than 100 treatments, the speed of design generation for Con($\mathbf{A}_d$) and Con($\mathbf{A}$) are similar. As the number of treatments increases, and $r(k+s)$ is much larger than v, Con($\mathbf{A}$) is quicker at generating designs. If v is only slightly smaller than $r(k+s)$, then Con($\mathbf{A}_d$) is the quicker of the two contraction algorithms.

Table 4.3 gives a time comparison of the four algorithms for some designs where

Table 4.2: Design generation times (seconds), $v < r(k+s)$

v	k	s	r	$r(k+s)$	E	Cyc(**A**)	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)	Con(**A**)
12	3	4	3	21	0.5076	68	69	60	**58**
20	4	5	4	36	0.6104	86	87	**70**	**70**
28	4	7	6	66	0.6546	64	65	58	**56**
35	7	5	5	60	0.6894	79	79	63	**58**
56	7	8	4	60	0.7385	84	83	**52**	**52**
65	5	13	4	72	0.7142	97	97	**54**	**54**
80	8	10	6	108	0.7835	97	98	63	**62**
100	10	10	8	160	0.8091	92	99	73	**65**
112	8	14	9	198	0.8120	54	76	60	**43**
120	15	8	10	230	0.8167	32	69	64	**27**
140	10	14	6	144	0.8272	131	121	**66**	79
171	9	19	7	196	0.8346	96	85	**60**	74
200	20	10	7	210	0.8474	127	110	**60**	80
250	10	25	10	350	0.8578	**31**	61	60	33

$v \geq r(k+s)$. In general, it can be seen that Con($\mathbf{A}_d$) generates resolvable row–column designs more efficiently than the other three algorithms. When generating designs with less than 300 treatments, both contraction algorithms perform better than the CycDesigN algorithms. Cyc($\mathbf{A}_d$) begins to produce resolvable row–column designs more efficiently than Con(**A**) for designs with more than 300 treatments. Again this can be explained by the methods of calculating the average efficiency factor. For designs with more than 300 treatments, it can be seen that as the number of treatments increases, the time taken to generate these designs increases dramatically for Cyc(**A**) and Con(**A**).

When $v = r(k+s)$ and v is small, the method of calculating the average efficiency factors has little effect on the speed of generating designs. That is, the two CycDesigN algorithms generate designs at approximately the same rate, as do the two contraction algorithms. For larger v, for example $v = 150$, from Table 4.3 it can be seen that this is no longer true. Cyc($\mathbf{A}_d$) and Con($\mathbf{A}_d$) are much quicker at generating designs than Cyc(**A**) and Con(**A**) respectively.

Based on the examples given in Tables 4.2 and 4.3 and others that have been

Table 4.3: Design generation times (seconds), $v \geq r(k+s)$

v	k	s	r	$r(k+s)$	E	Cyc(**A**)	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)	Con(**A**)
28	4	7	2	22	0.5547	28	23	**15**	21
32	4	8	2	24	0.5541	29	25	**15**	21
36	6	6	3	36	0.6811	102	102	**64**	**64**
40	5	8	3	39	0.6735	90	92	**54**	**54**
60	6	10	3	48	0.7139	138	135	**69**	70
64	8	8	4	64	0.7520	130	130	**74**	78
90	6	15	3	63	0.7300	117	91	**48**	77
130	10	13	5	115	0.8183	133	123	**65**	75
150	15	10	6	150	0.8311	124	107	**60**	80
240	20	12	3	96	0.8337	373	278	**65**	168
320	16	20	6	216	0.8804	400	160	**60**	305
400	20	20	4	160	0.8830	1082	224	**58**	943
600	15	40	4	220	0.8888	2021	230	**60**	1898
720	20	36	3	168	0.8954	6552	308	**59**	6327
1000	25	40	3	195	0.9114	9504	573	**67**	9016
1200	40	30	2	140	0.8964	10230	1391	**70**	8933

examined, it can be concluded that the contraction approach of updating the design matrices and calculating the change in the objective function is more efficient than the approach used by the CycDesigN algorithms. The quickest method of calculating the average efficiency factor, and hence which contraction algorithm is best, is dependent on the design parameters. The overall conclusion is that Con($\mathbf{A}_d$) is the best algorithm for generating designs where v is only slightly smaller than $r(k+s)$ and also for designs where $v \geq r(k+s)$. For the remaining cases where v is much smaller than $r(k+s)$, the best algorithm for generating resolvable row–column designs is Con($\mathbf{A}$).

4.6 Design Selection

If a design is found which has a weighted average efficiency factor equal to the calculated weighted upper bound, the algorithm is terminated and this optimal design is available for selection. Often an optimal design is not found and the user is required to choose when to the stop the algorithm. There are various approaches to deciding when to terminate the algorithm and the choice of approach will vary from user to user. Possible approaches include running the algorithm for a fixed length of time, terminating the algorithm when an improved design is not found after so many seconds, or alternatively at the end of the random descent phase. The end of the random descent phase could be considered a natural place to terminate the algorithm as there is often a delay before an improved design is found in the simulated annealing phase.

For small designs the best algorithm will find a near optimal or optimal design quickly, so the problem of when to terminate the algorithm is mainly of interest for large designs. From personal experience working at the Victorian Institute for Dryland Agriculture, agricultural experiments with 400 or more treatments generally have less than five replicates. Cyc($\mathbf{A}_d$) is the quickest of the two CycDesigN algorithms for generating these resolvable row–column designs, while Con($\mathbf{A}_d$) is the best overall.

Table 4.4 gives the number of seconds for the average efficiency factors to be achieved corresponding to an interchange acceptance path generated by Cyc($\mathbf{A}_d$) and Con($\mathbf{A}_d$). The resolvable row–column design has parameters $v = 1000$, $k = 25$, $s = 40$ and $r = 3$. The same random number seeds have been used for each algorithm to ensure the same interchange acceptance path is followed. As expected from the results in the previous section, Con($\mathbf{A}_d$) is the faster of the two algorithms at generating these designs. If the algorithms were run for a fixed length of time, Cyc($\mathbf{A}_d$) would not be able to produce a design with a better average efficiency factor than Con($\mathbf{A}_d$).

It can be seen from Table 4.4 that during the random descent phase both algorithms begin by generating designs at a steady rate and then slow down towards the end of the phase. The maximum delay between successive designs in the random decent phase for Cyc($\mathbf{A}_d$) is 60 seconds, compared to 4 seconds for Con($\mathbf{A}_d$). If the algorithms were stopped when no improved design is generated after, say, 10 seconds, Cyc($\mathbf{A}_d$) would not reach the end of the random descent phase.

Three interchange acceptance paths for different sized designs are given in Table 4.5. The average efficiency factor of the designs generated in the first few seconds of

Table 4.4: Times to generate designs for $v = 1000$, $k = 25$, $s = 40$, $r = 3$

E	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)	E	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)
0.905568	1	0	0.910734	25	15
0.905703	1	1	0.910794	27	15
0.905860	1	1	0.910853	29	16
0.906001	1	1	0.910906	31	16
0.906132	2	1	0.910961	33	16
0.906231	2	2	0.911017	35	17
0.906353	2	2	0.911070	39	17
0.906499	2	2	0.911116	44	18
0.906614	3	2	0.911162	50	18
0.906703	3	3	0.911205	59	19
0.906817	3	3	0.911243	71	20
0.906942	3	3	0.911281	80	21
0.907067	4	3	0.911313	104	23
0.907208	4	3	0.911348	164	27
0.907311	4	4	0.911360	211	31
0.907388	4	4	End of random descent		
0.907470	5	4	0.911363	548	72
0.907586	5	4	0.911363	550	73
0.907682	5	5	0.911364	551	73
0.907787	5	5	0.911364	561	74
0.907869	6	5	0.911365	565	75
0.907959	6	5	0.911366	565	75
0.908069	6	6	0.911367	572	76
0.908141	7	6	0.911367	624	83
0.908203	7	6	0.911368	627	84
0.908271	7	6	0.911368	631	85
0.908356	8	7	0.911368	633	85
0.908422	8	7	0.911370	634	86
0.908484	9	7	0.911371	635	86
0.908570	9	7	0.911372	635	86
0.908637	10	8	0.911373	636	86
0.908700	10	8	0.911374	636	87
0.908768	11	8	0.911375	638	87
0.908842	11	8	0.911375	638	87
0.908905	12	9	0.911375	640	88
0.908962	13	9	0.911376	641	88
0.909027	13	9	0.911376	641	88
0.909115	13	9	0.911376	643	89
0.909205	14	9	0.911377	646	89
0.909309	14	10	0.911378	646	89
0.909401	14	10	0.911379	648	90
0.909486	14	10	0.911380	651	90
0.909581	15	10	0.911380	651	90
0.909655	15	11	0.911381	651	91
0.909728	15	11	0.911382	652	91
0.909799	16	11	0.911383	653	91
0.909864	16	11	0.911384	653	91
0.909935	16	12	0.911386	655	92
0.910007	17	12	0.911387	659	93
0.910069	17	12	0.911387	660	93
0.910126	18	12	0.911387	666	94
0.910197	18	13	0.911389	666	94
0.910264	19	13	0.911389	667	94
0.910321	19	13	0.911389	667	95
0.910384	20	13	0.911390	667	95
0.910474	21	14	0.911390	670	95
0.910518	21	14	0.911392	671	96
0.910566	22	14	0.911393	677	96
0.910625	23	14	0.911394	678	97
0.910679	24	15	0.911395	680	97

Table 4.5: Times to follow interchange acceptance paths

$v=480,\ k=20,\ s=24,\ r=4$			$v=760,\ k=38,\ s=20,\ r=3$			$v=1400,\ k=40,\ s=35,\ r=2$		
E	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)	E	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)	E	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)
0.890126	7	5	0.895691	8	5	0.902674	7	3
0.890232	7	5	0.895809	8	5	0.902792	7	3
0.890318	7	5	0.895907	8	5	0.902910	8	3
0.890451	8	6	0.896015	9	6	0.903038	9	3
0.890582	8	6	0.896106	10	6	0.903162	9	3
0.890685	9	6	0.896195	10	6	0.903291	10	3
0.890791	9	6	0.896281	11	6	0.903388	11	4
0.890882	10	6	0.896360	12	6	0.903499	12	4
0.890978	11	7	0.896452	13	6	0.903622	14	4
0.891062	12	7	0.896530	15	7	0.903737	16	4
0.891131	13	7	0.896606	17	7	0.903853	18	4
0.891222	14	7	0.896684	21	7	0.903955	21	4
0.891309	17	8	0.896760	25	8	0.904065	25	4
0.891389	21	8	0.896833	37	9	0.904157	33	5
0.891458	25	9	0.896861	62	11	0.904259	47	5
0.891495	39	11	End of random descent			0.904317	128	8
End of random descent			0.896863	510	49	End of random descent		
0.891498	87	22	0.896864	510	49	0.904317	188	11
0.891500	87	22	0.896864	512	50	0.904318	188	11
0.891503	88	22	0.896866	512	50	0.904321	189	11
0.891504	88	22	0.896866	512	50	0.904321	189	11
0.891506	88	23	0.896868	512	50	0.904322	189	11
0.891507	89	23	0.896869	512	50	0.904325	190	11
0.891508	89	23	0.896872	513	50	0.904327	190	11
0.891511	89	23	0.896874	513	50	0.904328	192	12
0.891512	89	23	0.896875	514	51	0.904328	193	12
0.891514	90	23	0.896881	514	51	0.904331	193	12
0.891516	90	23	0.896882	514	51	0.904332	193	12
0.891516	90	24	0.896883	515	51	0.904332	194	12
0.891517	90	24	0.896885	515	51	0.904335	194	12
0.891519	91	24	0.896885	515	51	0.904338	195	12
0.891520	91	24	0.896886	515	51	0.904338	197	12
0.891522	91	24	0.896887	516	52	0.904339	198	12
0.891523	92	24	0.896887	517	52	0.904341	198	13
0.891523	92	24	0.896887	517	52	0.904342	198	13
0.891526	92	25	0.896890	517	52	0.904345	198	13
0.891526	95	25	0.896892	518	52	0.904348	199	13
0.891528	98	26	0.896893	519	53	0.904351	200	13
0.891531	99	27	0.896899	519	53	0.904357	201	13
0.891532	100	27	0.896900	519	53	0.904357	202	13
0.891534	100	27	0.896900	519	53	0.904361	202	13
0.891534	108	29	0.896903	520	53	0.904364	202	13
0.891535	110	29	0.896907	521	53	0.904366	203	13
0.891535	110	29	0.896908	521	54	0.904370	203	13
0.891536	111	30	0.896910	522	54	0.904372	204	14
0.891537	111	30	0.896911	523	54	0.904375	204	14
0.891539	111	30	0.896914	523	54	0.904378	206	14
0.891539	112	30	0.896914	523	54	0.904379	207	14
0.891541	112	30	0.896917	523	54	0.904379	209	14
0.891543	112	30	0.896918	524	55	0.904380	209	14
0.891545	113	31	0.896919	526	55	0.904382	209	14
0.891546	113	31	0.896919	528	56	0.904385	210	14
0.891547	113	31	0.896921	531	56	0.904388	210	14
0.891551	113	31	0.896925	1079	100	0.904391	210	15
0.891553	113	31	0.896926	1079	100	0.904392	210	15
0.891554	114	31	0.896927	1079	100	0.904395	211	15
0.891555	114	31	0.896930	1080	100	0.904398	212	15
0.891558	115	32	0.896931	1080	100	0.904401	212	15
0.891559	116	32	0.896932	1080	101	0.904404	213	15
0.891560	116	32	0.896933	1080	101	0.904407	213	15

each algorithm have not been presented. Similar results are seen for other random number seeds and designs of different sizes.

As Con($\mathbf{A}_d$) is able to generate resolvable row–column designs faster than Cyc($\mathbf{A}_d$), it is expected that Con($\mathbf{A}_d$) will reach the end of the random descent phase quicker. The number of seconds taken for Cyc($\mathbf{A}_d$) and Con($\mathbf{A}_d$) to reach the end of the random descent phase for several designs are given in Table 4.6. The effect of randomness is present due to the random number seed chosen, but multiple runs of each algorithm produced similar results to those presented. For the designs in Table 4.6, Con($\mathbf{A}_d$) reaches the end of the random descent phase in less than a third of the time taken by Cyc($\mathbf{A}_d$).

Table 4.6: Time to reach the end of random descent

v	k	s	r	Cyc($\mathbf{A}_d$)	Con($\mathbf{A}_d$)
420	21	20	4	26	8
480	20	24	4	39	11
620	20	31	3	41	7
800	20	40	2	18	2
1050	35	30	3	163	24
1400	40	35	2	122	8

Based on the results in Tables 4.4, 4.5 and 4.6, Con($\mathbf{A}_d$) is the preferred algorithm for the three design selection approaches considered. If the algorithms are run for a fixed length of time, Con($\mathbf{A}_d$) will always obtain a design as good as, if not better than, Cyc($\mathbf{A}_d$). The same conclusion is reached if the algorithms are terminated when no improved design is generated after a given length of time. If the two algorithms are given the same random number seed, and hence the same interchange acceptance path is followed, Con($\mathbf{A}_d$) will reach the end of the random descent phase much quicker than Cyc($\mathbf{A}_d$).

4.7 CycDesigN version 2.0

CycDesigN version 2.0 offers two methods of generating resolvable row–column designs when $v \geq 400$. The default method is the recursive update of the average efficiency factor (John and Whitaker, 2000) which is discussed in Section 4.7.1. An alternative method is a two stage procedure which is discussed in Section 4.7.2. Both these methods are compared to Con($\mathbf{A}_d$) which has been shown to be the best algorithm of the four considered for generating resolvable row–column designs with a large number of treatments and $v \geq r(k+s)$.

4.7.1 Updating the Average Efficiency Factor

Comparing designs of the same size using the average efficiency factor E as the objective function can be computationally expensive. However, John and Whitaker (2000) have developed a recursive formulae for updating E after an interchange is accepted. This approach is currently used in version 2.0 of CycDesigN for generating resolvable row–column designs with 400 or more treatments and will be referred to as Cyc(E). Con($\mathbf{A}_d$) and Cyc(E) will not follow the same interchange acceptance path for a given random number seed. This is due to Con($\mathbf{A}_d$) using (4.2) as the objective function, while Cyc(E) uses the average efficiency factor E.

A comparison was made in terms of the speed of generating resolvable row–column designs between Cyc(E) and Con($\mathbf{A}_d$). Table 4.7 shows the average efficiency factors corresponding to two interchange acceptance paths for each algorithm. The same random number seed has been used for each algorithm to ensure the same initial design is constructed. For a four–replicate resolvable row–column design with 400 treatments set out in 20 rows and 20 columns, designs are generated much faster using Con($\mathbf{A}_d$). If the algorithms are stopped at the end of the random descent phase, it can be seen from Table 4.7 that Cyc(E) would be terminated with an average efficiency factor of 0.882070 after 45 seconds, while Con($\mathbf{A}_d$) would be stopped after 7 seconds and the average efficiency factor would be 0.882782.

Similar results are evident for a two–replicate resolvable row–column design with 1000 treatments set out in 25 rows and 40 columns. From Table 4.7 it can be seen that if the algorithms are terminated at the end of the random descent phase, Cyc(E) would return a design with an average efficiency factor of 0.884980 after 45 seconds. Using Con($\mathbf{A}_d$), the average efficiency factor would be 0.885305 after just 5 seconds. It is apparent from these two results, and others which have been examined, that Con($\mathbf{A}_d$) is able to produce designs with a higher average efficiency factor quicker

Table 4.7: Interchange acceptance paths for Cyc(E) and Con($\mathbf{A}_d$)

$v=400, k=20, s=20, r=4$				$v=1000, k=25, s=40, r=2$			
Cyc(E)	Time	Con($\mathbf{A}_d$)	Time	Cyc(E)	Time	Con($\mathbf{A}_d$)	Time
0.872931	1	0.872931	0	0.874493	2	0.874493	0
0.873399	1	0.873415	1	0.874886	2	0.874927	0
0.873874	1	0.873795	1	0.875341	2	0.875393	0
0.874245	2	0.874196	1	0.875878	2	0.875869	0
0.874616	2	0.874585	1	0.876380	2	0.876304	1
0.874994	2	0.874934	1	0.876767	3	0.876709	1
0.875380	2	0.875395	1	0.877154	3	0.877266	1
0.875668	3	0.875745	1	0.877588	3	0.877766	1
0.875954	3	0.876008	1	0.878054	3	0.878265	1
0.876230	3	0.876358	1	0.878432	3	0.878674	1
0.876430	4	0.876617	1	0.878811	4	0.879059	1
0.876660	4	0.876791	2	0.879198	4	0.879368	1
0.876909	5	0.876933	2	0.879532	4	0.879761	1
0.877108	6	0.877152	2	0.879813	4	0.880187	1
0.877252	8	0.877390	2	0.880177	4	0.880466	1
0.877576	8	0.877695	2	0.880515	5	0.880830	1
0.877840	8	0.877966	2	0.880770	5	0.881153	1
0.878169	8	0.878235	2	0.880979	5	0.881473	1
0.878484	9	0.878517	2	0.881235	5	0.881745	2
0.878708	9	0.878752	2	0.881494	6	0.881996	2
0.878897	9	0.878962	2	0.881756	6	0.882268	2
0.879116	10	0.879149	3	0.881972	6	0.882527	2
0.879307	11	0.879383	3	0.882200	7	0.882740	2
0.879469	12	0.879563	3	0.882396	7	0.882996	2
0.879636	13	0.879778	3	0.882606	8	0.883267	2
0.879817	14	0.879987	3	0.882767	8	0.883508	2
0.880039	14	0.880219	3	0.882974	9	0.883756	2
0.880218	15	0.880405	3	0.883153	10	0.884005	2
0.880404	16	0.880604	3	0.883328	10	0.884232	2
0.880587	16	0.880769	3	0.883503	11	0.884432	2
0.880755	18	0.880954	3	0.883661	12	0.884692	3
0.880930	19	0.881141	4	0.883811	14	0.884928	3
0.881060	21	0.881344	4	0.883970	15	0.885143	3
0.881235	24	0.881520	4	0.884093	18	0.885305	5
0.881392	26	0.881661	4	0.884218	20	End of r.d.	
0.881556	28	0.881817	4	0.884358	22	0.885305	8
0.881672	31	0.881956	4	0.884493	24	0.885306	9
0.881782	34	0.882126	4	0.884597	27	0.885309	9
0.881930	39	0.882291	5	0.884739	31	0.885315	9
0.882070	45	0.882425	5	0.884837	38	0.885319	9
End of r.d.		0.882578	5	0.884970	44	0.885320	12
0.882070	94	0.882705	5	0.884980	45	0.885325	12
0.882073	94	0.882782	7	End of r.d.		0.885332	13
0.882074	95	End of r.d.		0.884983	88	0.885334	13
0.882074	95	0.882782	13	0.884983	88	0.885336	13
0.882074	95	0.882788	13	0.884985	88	0.885336	16
0.882077	95	0.882793	13	0.884992	89	0.885336	16
0.882085	95	0.882797	13	0.884998	89	0.885349	34
0.882096	95	0.882797	15	0.884999	89	0.885350	37
0.882098	96	0.882798	15	0.885007	89	0.885350	37
0.882101	96	0.882806	15	0.885007	89	0.885354	75
0.882101	96	0.882806	15	0.885013	90	0.885360	326
0.882105	96	0.882809	16	0.885013	90	0.885367	326
0.882106	96	0.882813	16	0.885013	91	0.885372	327
0.882107	97	0.882813	16	0.885017	91	0.885377	330
0.882110	98	0.882818	16	0.885017	91	0.885382	331
0.882110	98	0.882822	16	0.885020	91	0.885384	331
0.882113	98	0.882825	16	0.885026	92	0.885387	331
0.882119	99	0.882828	19	0.885027	92	0.885390	332
0.882120	99	0.882830	19	0.885034	92	0.885394	332

than the method of updating E in CycDesigN. This result is consistent for different random number seeds and different sized designs.

Further results for comparing the method of updating the average efficiency factor with Con($\mathbf{A}_d$) are presented in Table 4.8. Given the large number of treatments in each design, the two algorithms were run for three minutes and the average efficiency factor of the best resolvable row–column design was recorded. The results show that Con($\mathbf{A}_d$) is capable of generating designs within three minutes that have a better average efficiency factor than those obtained by Cyc(E).

Table 4.8: Cyc(E) and Con($\mathbf{A}_d$) average efficiency factors after 3 minutes

v	k	s	r	Cyc(E)	Con($\mathbf{A}_d$)
550	22	25	3	0.8874	**0.8877**
600	15	40	4	0.8884	**0.8889**
884	26	34	3	0.9076	**0.9084**
1000	25	40	3	0.9108	**0.9115**
1500	50	30	2	0.9036	**0.9040**
1800	40	45	4	0.9394	**0.9403**
3000	50	60	2	0.9317	**0.9323**

The average efficiency factor obtained at the end of the random descent phase, when there is a break in the algorithms, is also of interest. Table 4.9 gives the average efficiency factors obtained and the number of seconds taken to reach this point. The same parameter sets and random number seeds as those used to obtain the results in Table 4.8 have been used. The interchange acceptance paths for $v = 1800$ in Table 4.8 did not reach the end of the random descent phase within three minutes. This explains why the average efficiency factors given in Table 4.9 for this parameter set are higher than those given in Table 4.8. The interchange acceptance paths for all other parameter sets presented did reach the end of the random descent phase within three minutes. Con($\mathbf{A}_d$) is much quicker at reaching the end of the random descent phase and the average efficiency factors are also higher. Similar results to those presented are obtained with different random number seeds and different parameter sets.

The average efficiency factors presented in Table 4.8 are only marginally higher

Table 4.9: Cyc(E) and Con($\mathbf{A}_d$) times to the end of random descent

				Cyc(E)		Con($\mathbf{A}_d$)	
v	k	s	r	E	Time	E	Time
550	22	25	3	0.8867	43	0.8875	5
600	15	40	4	0.8884	132	0.8888	21
884	26	34	3	0.9074	75	0.9082	15
1000	25	40	3	0.9106	84	0.9114	23
1500	50	30	2	0.9034	68	0.9037	8
1800	40	45	4	0.9406	924	0.9410	343
3000	50	60	2	0.9317	175	0.9322	45

than those obtained at the end of the random descent phase and given in Table 4.9. For two of the designs presented, namely $v = 600$ and $v = 3000$, the best average efficiency factor obtained by Cyc(E) within three minutes was that found at the end of the random descent phase. Unless users are prepared to run the algorithms for an extended period of time, there is little lost in terms of the average efficiency factor by stopping the algorithms at the end of the random descent phase. Users who do choose to stop the algorithms at this point will obtain better average efficiency factors quicker by using Con($\mathbf{A}_d$). This issue of terminating the algorithm at the end of the random descent phase is investigated further in Chapter 5.

4.7.2 Two Stage Construction

The five algorithms discussed previously in this chapter, Cyc($\mathbf{A}$), Cyc($\mathbf{A}_d$), Con($\mathbf{A}$), Con($\mathbf{A}_d$) and Cyc(E), are all examples of one stage design generation algorithms. The one stage method constructs the resolvable row–column design as a single process. Whitaker, Williams and John (1999) recommend that large resolvable row–column designs should be constructed in two stages. In the first stage, an efficient resolvable block design is obtained which becomes the column component design of the row–column design. At the second stage, treatment interchanges are restricted to treatments within the same column. Hence, at this stage, the column component design is fixed and interchanges are made with the aim of finding an efficient row component design and overall row–column design. One reason for this strategy is that efficient resolvable block designs can be constructed quickly, especially if α–designs are used (Patterson and Williams, 1976a). Secondly, the number of interchanges at the second stage is restricted to those within the same column. For designs with a large number of treatments, this two stage approach often produces efficient designs quicker than constructing a design in a single process. However, the disadvantage is that it may not be possible to find a row component design that fits well with the fixed column component design to produce a good overall row–column design.

It was shown in Section 4.5 that for resolvable row–column designs with $v < r(k+s)$, Con($\mathbf{A}$) outperformed both Cyc($\mathbf{A}$) and Cyc($\mathbf{A}_d$), and when $v \geq r(k+s)$, Con($\mathbf{A}_d$) outperformed the CycDesigN algorithms. These results suggest that a two stage contraction algorithm would outperform a two stage CycDesigN algorithm. Therefore, a two stage contraction algorithm was proposed which chooses the method of calculating the average efficiency factor based on the design parameters. For designs where $v < r(k+s)$, the average efficiency factor E is calculated from the matrix $\mathbf{A}_J$ and the method discussed in Section 4.2.2. If $v \geq r(k+s)$, the method outlined in Section 4.3.3 would be used to calculate E via the average efficiency factor E_d of the dual design.

Of interest is the speed in generating resolvable row–column designs using the two stage contraction algorithm and the one stage contraction algorithms. As the two stage algorithm is recommended for designs with a large number of treatments, it is only tested on designs with $v \geq 100$. The first stage of the two stage algorithm is run for thirty seconds and the second stage for four and a half minutes. The first stage is run for a much shorter time as the column component design generated is an α–design and these can be generated quickly. If the optimal column component design was found in stage one before the thirty seconds had elapsed, the second stage was permitted to run longer, such that the algorithm was run for a total of five minutes.

The choice of which one stage contraction algorithm to use in the comparison is dependent on the design parameters. Based on the results from Section 4.5, if $v < r(k+s)$, then Con($\mathbf{A}$) is used to generate the designs, while if $v \geq r(k+s)$, Con($\mathbf{A}_d$) is used. The one stage algorithms were also run for a total of five minutes.

Table 4.10: One stage versus two stage contraction algorithms

v	k	s	r	$r(k+s)$	E_1 stage	E_2 stage
100	10	10	8	160	0.809448	**0.811224**
120	15	8	10	230	**0.817043**	0.816684
130	10	13	5	115	0.818331	**0.819973**
240	20	12	3	96	**0.833912**	0.833672
400	20	20	4	160	0.883190	**0.883868**
600	15	40	4	220	0.889035	**0.889210**
720	20	36	3	168	0.895531	**0.895680**
1000	25	40	3	195	0.911528	**0.911577**
1200	40	30	2	140	0.896512	**0.896584**

The average efficiency factors presented in Table 4.10 are the best obtained from five runs of each algorithm. For designs with $v \leq 400$, there appears to be no consistency as to which algorithm produced the design with the best average efficiency factor. The two stage algorithm seems most suitable for designs with a very large number of treatments. For designs with $v \geq 600$, the absolute difference between the two average efficiency factors given in Table 4.10 is less than 2×10^{-4}.

4.8 Conclusions

From the results presented in Sections 4.5, 4.6 and 4.7, it is apparent that the current method of constructing resolvable row–column designs using CycDesigN version 2.0 can be improved dramatically. By constructing resolvable row–column designs based on the row and column incidence matrices of the contraction arrays and secondary arrays, the performance in terms of speed is superior to the current approaches.

When comparing the one stage approaches of Cyc($\mathbf{A}$), Cyc($\mathbf{A}_d$), Con($\mathbf{A}_d$) and Con($\mathbf{A}$), the biggest improvement is evident when considering designs with a large number of treatments. For such designs, the forming of the matrix $\mathbf{A}_J$ and the subsequent inverting of this $v \times v$ matrix is computationally expensive. By selecting the smaller of $\mathbf{A}_J$ and $\mathbf{A}_M$ when calculating the average efficiency factor, large savings in computing effort can be made.

The recursive method of updating the average efficiency factor is no longer the best approach to constructing resolvable row–column designs with a large number of treatments where $v \geq r(k+s)$. The single stage contraction algorithm, Con($\mathbf{A}_d$), is capable of producing more efficient designs quicker than the recursive method of Cyc(E).

The two stage design construction method remains an important tool for generating resolvable row–column designs. For designs with $v < 600$, there is no clear set of parameter constraints for which the two stage contraction algorithm produces more efficient designs than the one stage contraction algorithms. The row–column design obtained using the two stage algorithm is restricted by the choice of α–design at the first stage. This suggests that several resolvable row–column designs should be constructed using both approaches and the design with the better average efficiency factor selected.

Chapter 5

Design Considerations for Large Treatment Numbers

5.1 Introduction

Existing optimisation algorithms are slow at constructing efficient resolvable row–column designs with thousands of treatments. The contraction algorithm Con($\mathbf{A}_d$), discussed in Chapter 4, is able to generate resolvable row–column designs quicker than CycDesigN version 2.0, but it still requires considerable time to generate efficient designs when v is very large.

From personal experience, it is uncommon in agricultural field trials to require a replicated design with more than 1000 treatments. Gene expression microarray experiments are one example where thousands of treatments, or genes, are replicated multiple times in the one design. Microarray technology provides a tool for exploring and understanding an organisms genetic material. As an example, microarrays allow comparison of gene expression between normal and diseased cells. A brief introduction to microarrays and an example are given in Section 5.2.

Calculating the average efficiency factor E requires the most computational effort in CycDesigN and Con($\mathbf{A}_d$). For designs with a large number of treatments, such as microarray experiments, calculating E is a time consuming process. The speed of the algorithms could be improved by not calculating E, or by estimating E. If E was not calculated, the sole objective function would be the (M,S)–optimality criterion. Both methods of speed improvement are discussed in Section 5.3.

When generating resolvable row–column designs suitable for experiments with a large number of treatments, numerous observations have shown that the best average efficiency factor does not change dramatically throughout the simulated annealing phase. This does not suggest that there is no improvement in the designs as the distributional properties of the pairwise efficiency factors may be improving. The distribution of the pairwise efficiency factors provide further information to the usefulness of a design and is discussed in Section 5.4.

5.2 Microarray Example

A microarray is usually a glass or nylon slide onto which DNA molecules are attached at fixed locations known as spots. On a single microarray there may be tens of thousands of spots. It is recommended that the DNA molecules are replicated on at least three spots on the microarray (Lee, Kuo, Whitmore and Sklar, 2000) and dispersed over the surface of the microarray (Simon, Radmacher and Dobbin, 2002).

In the most basic microarray experiment, two samples of mRNA are compared by first converting the samples to cDNA and then labelling one sample with a green fluorescent dye and the second sample with a red fluorescent dye. The two samples are then mixed and washed over the microarray. Using a laser, the spots are excited and the amount of each sample bound to a spot can be measured. For each spot, two intensities are measured which indicate the level of expression for the two samples. If the spot is green, then the sample dyed green is in abundance, while if the sample dyed red is in abundance, the spot will be red. If both samples are equally expressed, the spot will be yellow. In the situation where neither gene is expressed, the spot will appear black. Based on the colour of each spot, the relative expression levels of the two samples can be estimated.

In one such experiment it was required to set out 2000 genes in a four–replicate resolvable row–column design, where each replicate was laid out in 50 rows and 40 columns (personal communication). The replicates were to be in a two by two grid such that the replicates form two contiguous groups. To complicate the design, latinisation was added to the design structure. A resolvable row–column design can be latinised in the row and/or column direction; see John and Williams (1998). For example, a design is said to be t–latinised in the column direction if no treatment occurs more than once in each group of t long columns. It was desirable for this design to be latinised in both the row and column direction. The design needed to be 25–latinised in the row direction, and 20–latinised in the column direction. Figure 5.1 shows the layout of the microarray experiment.

The contraction algorithm Con($\mathbf{A}_d$) took 996 seconds (approximately 16 minutes) to find an initial solution with an average efficiency factor of 0.942987. This is a time consuming process if a scientist requires many such designs. However, to put things in perspective, CycDesigN took 241707 seconds (approximately 67 hours) to find the first design which had an average efficiency factor of 0.942963. As the number of treatments is greater than 400, CycDesigN uses the recursive average efficiency factor update method; see Section 4.7.1. Several runs of each algorithm with different random number seeds gave similar results.

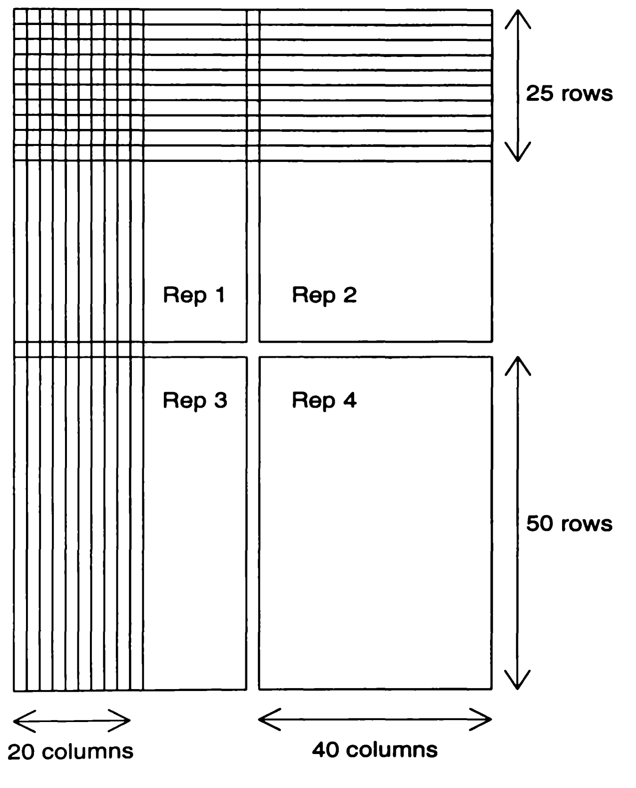

Figure 5.1: Microarray layout for $v = 2000$, $k = 50$, $s = 40$, $r = 4$

5.3 Objective Functions

The calculation of the average efficiency factor is computationally expensive for large designs and alternative objective functions are required which will improve the speed of Con($\mathbf{A}_d$). One possibility is the estimation of the average efficiency factor. The average efficiency factor for the resolvable row–column design can be calculated using (1.1), that is

$$E = \frac{v-1}{\sum_{i=1}^{v-1} e_i^{-1}}$$

where e_i are the canonical efficiency factors of the information matrix $\mathbf{A}$. By estimating $\sum_i e_i^{-1}$, an estimate of E is obtained.

The canonical efficiency factors of $\mathbf{A}$ are given by a subset of the non–zero eigenvalues of $\mathbf{A}/r$. From (1.2), $\mathbf{A}/r$ can be expressed as

$$\tfrac{1}{r}\mathbf{A} = (\mathbf{I} - \tfrac{1}{v}\mathbf{J}) - \mathbf{W}$$

where $\mathbf{W} = \frac{1}{rs}\mathbf{N}_k\mathbf{N}_k' + \frac{1}{rk}\mathbf{N}_s\mathbf{N}_s' - \frac{2}{v}\mathbf{J}$. Following John and Williams (1995, p35), a geometric expansion of $r\mathbf{A}^+$ is given by

$$r\mathbf{A}^+ = (\mathbf{I} - \tfrac{1}{v}\mathbf{J}) + \textstyle\sum_{i=1}^{\infty} \mathbf{W}^i$$

A proof is given in Appendix C. It then follows that

$$\sum_{i=1}^{v-1} e_i^{-1} = (v-1) + \sum_{i=1}^{\infty} \text{trace}(\mathbf{W}^i) \qquad (5.1)$$

The purpose of estimating E is to reduce the computational effort required. It therefore seems reasonable to estimate E using the first two terms of the geometric expansion given in (5.1) as these are easily computed, that is

$$\text{trace}(\mathbf{W}) = k+s-2 \qquad \text{trace}(\mathbf{W}^2) = \frac{k+s+2}{r} + \frac{2}{r^2}\text{trace}(\mathbf{HH}') - 4$$

where trace($\mathbf{HH}'$) is given in (3.25).

Table 5.1 gives the start of an interchange acceptance path for a three–replicate resolvable row–column design with 3000 treatments set out in 50 rows and 60 columns. The estimated average efficiency factor, E_{est}, is calculated using the first two terms of the geometric expansion. The time in seconds to generate each design is also presented.

Table 5.1: Actual and estimated E for $v = 3000$, $k = 50$, $s = 60$, $r = 3$ (T = time)

E_{actual}	T	E_{est}	T	E_{actual}	T	E_{est}	T	E_{actual}	T	E_{est}	T
0.945979	1	0.953396	1	0.946732	88	0.953600	2	0.947200	177	0.953731	3
0.945992	3	0.953400	1	0.946739	90	0.953602	2	0.947206	178	0.953732	3
0.946007	4	0.953404	1	0.946750	91	0.953605	2	0.947213	180	0.953734	3
0.946024	5	0.953408	1	0.946760	93	0.953608	2	0.947220	181	0.953736	3
0.946038	7	0.953412	1	0.946768	94	0.953610	2	0.947225	183	0.953738	3
0.946049	8	0.953415	1	0.946779	96	0.953613	2	0.947231	184	0.953739	4
0.946064	10	0.953419	1	0.946787	97	0.953615	2	0.947237	186	0.953741	4
0.946079	11	0.953423	1	0.946796	98	0.953618	2	0.947243	187	0.953743	4
0.946093	13	0.953427	1	0.946804	100	0.953620	2	0.947248	189	0.953745	4
0.946108	14	0.953431	1	0.946810	101	0.953622	2	0.947253	190	0.953746	4
0.946119	16	0.953434	1	0.946820	103	0.953624	2	0.947263	192	0.953749	4
0.946135	17	0.953438	1	0.946830	104	0.953627	2	0.947273	193	0.953752	4
0.946151	18	0.953442	2	0.946838	106	0.953629	2	0.947282	195	0.953754	4
0.946162	20	0.953445	2	0.946848	107	0.953632	2	0.947295	196	0.953758	4
0.946176	21	0.953449	2	0.946857	109	0.953635	2	0.947303	198	0.953760	4
0.946190	23	0.953452	2	0.946867	110	0.953637	2	0.947313	199	0.953763	4
0.946206	24	0.953457	2	0.946876	112	0.953640	2	0.947323	201	0.953766	4
0.946222	26	0.953461	2	0.946882	113	0.953642	2	0.947333	202	0.953769	4
0.946235	27	0.953465	2	0.946893	114	0.953645	2	0.947342	203	0.953771	4
0.946248	29	0.953468	2	0.946902	116	0.953647	2	0.947351	205	0.953774	4
0.946262	30	0.953472	2	0.946909	117	0.953649	2	0.947358	206	0.953776	4
0.946272	32	0.953475	2	0.946918	119	0.953652	2	0.947367	208	0.953778	4
0.946283	33	0.953478	2	0.946925	120	0.953654	2	0.947378	209	0.953782	4
0.946296	34	0.953481	2	0.946933	122	0.953656	2	0.947387	211	0.953784	4
0.946310	36	0.953485	2	0.946942	123	0.953658	2	0.947398	212	0.953787	4
0.946324	37	0.953489	2	0.946951	125	0.953661	2	0.947409	214	0.953790	4
0.946336	39	0.953492	2	0.946957	126	0.953662	2	0.947420	215	0.953794	4
0.946350	40	0.953496	2	0.946967	128	0.953665	2	0.947431	217	0.953797	4
0.946360	42	0.953499	2	0.946974	129	0.953667	2	0.947439	218	0.953799	4
0.946371	43	0.953502	2	0.946982	131	0.953669	2	0.947448	219	0.953802	4
0.946382	45	0.953505	2	0.946988	132	0.953671	2	0.947456	221	0.953804	4
0.946398	46	0.953509	2	0.946996	133	0.953674	2	0.947464	222	0.953806	4
0.946411	47	0.953513	2	0.947002	135	0.953675	2	0.947473	224	0.953809	4
0.946423	49	0.953516	2	0.947011	136	0.953677	2	0.947482	225	0.953811	4
0.946433	50	0.953519	2	0.947019	138	0.953680	2	0.947490	227	0.953814	4
0.946444	52	0.953522	2	0.947027	139	0.953682	2	0.947498	228	0.953816	4
0.946455	53	0.953525	2	0.947037	141	0.953685	2	0.947506	230	0.953818	4
0.946470	55	0.953529	2	0.947043	142	0.953686	2	0.947515	231	0.953821	4
0.946481	56	0.953532	2	0.947051	144	0.953688	2	0.947522	233	0.953823	4
0.946493	58	0.953535	2	0.947057	145	0.953690	2	0.947530	234	0.953825	4
0.946506	59	0.953539	2	0.947063	147	0.953692	2	0.947539	236	0.953828	4
0.946519	61	0.953542	2	0.947069	148	0.953694	2	0.947547	237	0.953830	4
0.946533	62	0.953546	2	0.947076	150	0.953696	2	0.947554	238	0.953832	4
0.946545	64	0.953549	2	0.947083	151	0.953698	2	0.947560	240	0.953834	4
0.946554	65	0.953552	2	0.947089	153	0.953699	2	0.947569	241	0.953836	4
0.946566	66	0.953555	2	0.947096	154	0.953701	2	0.947576	243	0.953839	4
0.946573	68	0.953557	2	0.947103	156	0.953703	2	0.947585	244	0.953841	4
0.946586	69	0.953561	2	0.947110	157	0.953705	2	0.947593	246	0.953843	4
0.946598	71	0.953564	2	0.947117	159	0.953707	3	0.947600	247	0.953845	4
0.946611	72	0.953567	2	0.947126	160	0.953709	3	0.947608	249	0.953848	4
0.946622	74	0.953570	2	0.947133	162	0.953711	3	0.947615	250	0.953850	4
0.946635	75	0.953574	2	0.947141	163	0.953713	3	0.947621	252	0.953851	4
0.946646	77	0.953577	2	0.947146	165	0.953715	3	0.947629	253	0.953854	4
0.946657	78	0.953580	2	0.947152	166	0.953717	3	0.947637	255	0.953856	4
0.946669	79	0.953583	2	0.947158	168	0.953719	3	0.947644	256	0.953858	4
0.946679	81	0.953586	2	0.947165	169	0.953720	3	0.947653	258	0.953861	4
0.946689	82	0.953589	2	0.947171	171	0.953722	3	0.947659	259	0.953862	4
0.946699	84	0.953591	2	0.947178	172	0.953724	3	0.947667	261	0.953865	4
0.946709	85	0.953594	2	0.947185	174	0.953727	3	0.947674	262	0.953866	4
0.946722	87	0.953597	2	0.947192	175	0.953729	3	0.947679	264	0.953868	5

For the interchange acceptance path given in Table 5.1, the two algorithms have accepted the same treatment interchanges. This allows for a direct comparison of the actual average efficiency factor and the estimated average efficiency factor. It can clearly be seen that estimating the average efficiency factor is much quicker than calculating the actual value. However, the estimated E are much higher than the actual E. This result is consistent with different random number seeds and different parameter sets. The estimated E could be improved by including more terms in the geometric expansion but this requires considerable computational effort, and hence, defeats the purpose of estimating E.

The average efficiency factor is maximised when $\sum e_i^{-1}$ is minimised. Based on (5.1) and the first two terms of the geometric expansion, $\sum e_i^{-1}$ is minimised when trace($\mathbf{HH}'$) is minimised. This is equivalent to the (M,S)–optimality criterion which selects designs that minimise trace($\mathbf{HH}'$). Therefore, an alternative approach for improving the speed of the algorithm is to accept interchanges based solely on the (M,S)–optimality objective function. An algorithm was developed which is based on Con($\mathbf{A}_d$) but no average efficiency factors are calculated. This algorithm is called Con(M,S). An interchange will always be accepted if the value of the (M,S)–optimality objective function is decreased, and during the simulated annealing stage there is a small probability that an interchange will be accepted that does not improve the objective function value.

This approach is not suitable for designs with a small number of treatments as the algorithm will quickly reduce the search space to the designs which are (M,S)–optimal. It is conjectured that A–optimal designs are contained within the subclass of (M,S)–optimal designs, therefore, it is likely that several designs with the same (M,S)–optimality objective function value will have different average efficiency factors. To differentiate between these designs the average efficiency factor must be calculated which Con(M,S) does not do. An example highlighting this constraint with Con(M,S) is for three–replicate resolvable row–column designs with 12 treatments set out in 3 rows and 4 columns. The best (M,S)–optimality objective function value found by Con($\mathbf{A}_d$) is 58.5000. Within the group of designs with this value, are designs with an average efficiency factor of 0.4901 and 0.5001. For designs with a large number of treatments, the algorithm is more likely to find a near optimal design, rather than an optimal design, so this problem should not be encountered

(Whitaker, 1995).

If an improvement is made to the value of the (M,S)–optimality objective function, both Con($\mathbf{A}_d$) and Con(M,S) will accept the treatment interchange. However, the method of determining the best design is different for the two algorithms. The design obtained from an interchange which results in the (M,S)–optimality objective function value improving is classed as the best design for Con(M,S). Con($\mathbf{A}_d$) only updates the best design when the value of the (M,S)–optimality objective function is improved and E, or E_w, is better than the best E, or best E_w. Venables and Eccleston (1993) found that a decrease in the value of the (M,S)–optimality objective function can correspond to a decrease, rather than an increase, in the average efficiency factor. This is a rare event, but would result in the best design being updated by Con(M,S) but not by Con($\mathbf{A}_d$). The average efficiency factor for the row–column design is only calculated by Con(M,S) when the algorithm is stopped. It is possible for the best design found by Con($\mathbf{A}_d$) and Con(M,S) to be different when the algorithms are terminated.

To compare the speed and the quality of design generation, Con($\mathbf{A}_d$) was run for approximately 60 seconds and the best value of the (M,S)–optimality objective function was recorded. Con(M,S) was then run using the same random number seed and the time to find a design with an equal or better (M,S)–optimality objective function value was recorded. The results for a selection of parameter sets is given in Table 5.2. Similar results were obtained for different random number seeds and different parameter sets.

For all the designs in Table 5.2, Con($\mathbf{A}_d$) and Con(M,S) made the same treatment interchanges for the period of the interchange acceptance path followed. The average efficiency factor presented in Table 5.2 corresponds to the given (M,S)–optimality objective function value which was obtained by both algorithms.

These results show for designs with a large number of treatments, selecting designs based solely on the (M,S)–optimality criterion, and not calculating E, is much quicker than the approach of Con($\mathbf{A}_d$). Therefore, running both algorithms for the same length of time will result in more efficient designs being generated by Con(M,S) than Con($\mathbf{A}_d$).

Table 5.2: Speed comparison of Con($\mathbf{A}_d$) and Con(M,S)

v	k	s	r	(M,S)–objective value	E	Time (seconds) Con($\mathbf{A}_d$)	Con(M,S)
1000	25	40	6	629.425	0.9249	62	0
2000	50	40	4	456.725	0.9417	60	0
2025	45	45	4	456.089	0.9425	62	0
3000	50	60	4	540.671	0.9521	62	1
3200	40	80	6	981.222	0.9558	63	1
4000	80	50	3	442.663	0.9522	62	0
6000	100	60	2	338.836	0.9478	60	0
10000	100	100	3	653.364	0.9703	63	1

5.4 Pairwise Efficiency Factors

The average efficiency factor is one possible objective function to help the user decide when to terminate the algorithm when generating resolvable row–column designs. Based on numerous observations, the average efficiency factor does not increase remarkably during the simulated annealing stage for designs with a large number of treatments. Figure 5.2 shows how the best average efficiency factor changes as interchanges are accepted for four resolvable row–column designs generated using Con($\mathbf{A}_d$). The x axis is the number of seconds taken to generate the designs and the vertical line signifies where the random descent phase ends and simulated annealing begins. It can be seen from Figure 5.2 that the main improvement in the average efficiency factor occurs during the random descent phase. Once the algorithm enters the simulated annealing phase there is little improvement in the best average efficiency factor found.

Table 5.3 gives the average efficiency factors for three designs generated during each of the interchange acceptance paths presented in Figure 5.2. These three designs are the first design generated D_1, the best design at the end of the random descent phase D_{rd} and the best design found when Con($\mathbf{A}_d$) is terminated D_{final}. The best design is defined as the design with the highest average efficiency factor at a given time. The time in seconds to reach the end of the random descent phase and the time when Con($\mathbf{A}_d$) is stopped are also presented. For each parameter set,

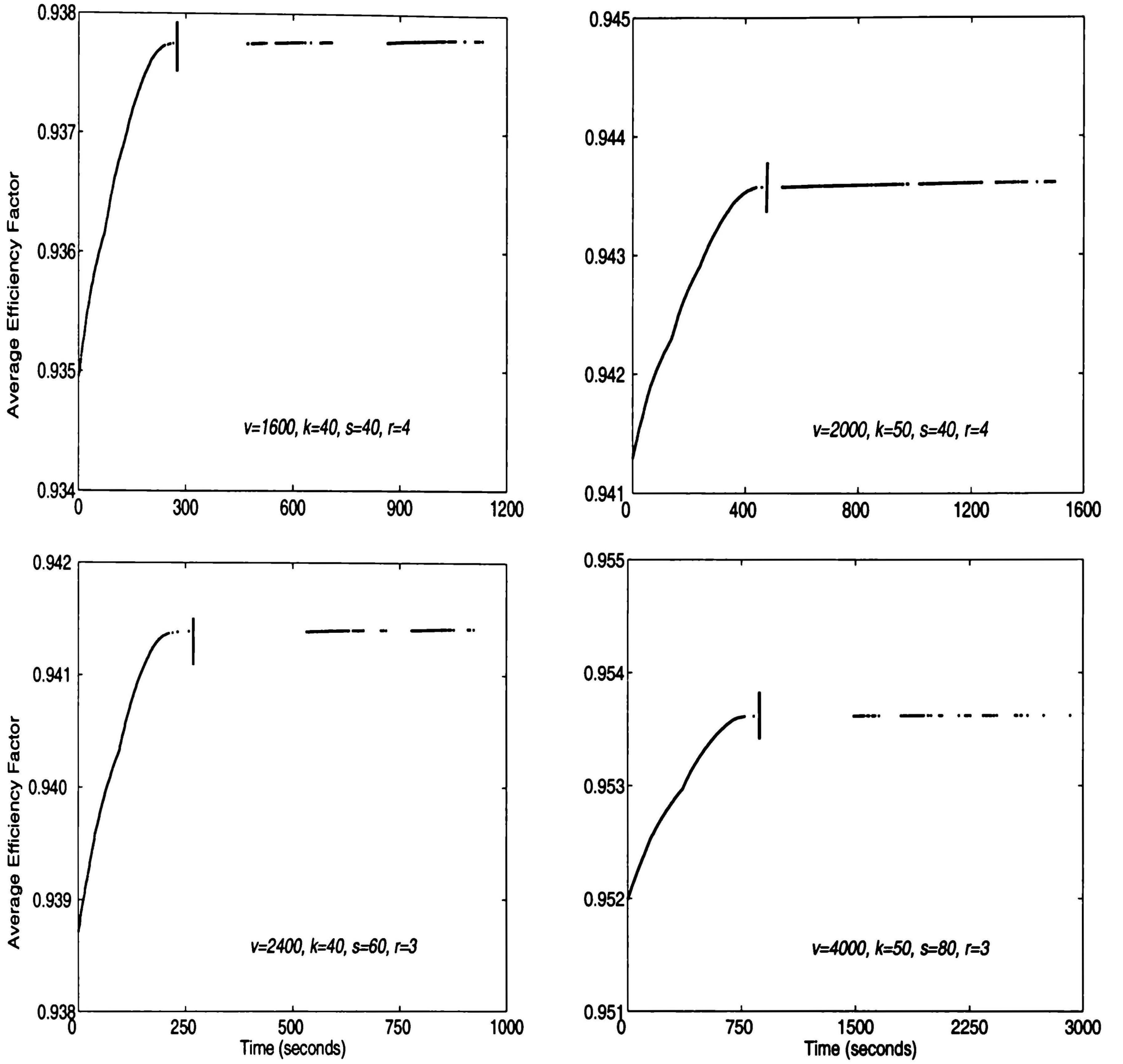

Figure 5.2: Change in the best E

$\mathrm{Con}(\mathbf{A}_d)$ was allowed to run at least twice as long in the simulated annealing phase as in the random descent phase.

It could be argued that as there is little overall change in the average efficiency factor during the simulated annealing phase, the best design at the end of the random descent phase would be a suitable design to select. This would save the user time waiting for the algorithm to enter the simulated annealing phase to generate further designs. However, even if E does not change much, there may be changes in the distribution of the efficiency factors e_{ij} for the pairwise comparisons. The distribution of the e_{ij} values provide further information about the suitability of the design and E can be expressed as the harmonic mean of the e_{ij} values.

To calculate the pairwise efficiency factors e_{ij}, consider the pairwise comparison for treatment i and j $(i \neq j)$ which is estimable with estimators $\hat{\tau}_i - \hat{\tau}_j$, where τ_i is

Table 5.3: Average efficiency factors

				D_1	D_{rd}		D_{final}	
v	k	s	r	E_1	E_{rd}	Time	E_{final}	Time
1600	40	40	4	0.9347	0.9377	265	0.9378	1132
2000	50	40	4	0.9413	0.9436	458	0.9436	1498
2400	40	60	3	0.9387	0.9414	258	0.9414	924
4000	50	80	3	0.9520	0.9536	831	0.9536	2915

the effect of treatment i. The variance of the pairwise comparison is calculated as

$$\text{var}(\hat{\tau}_i - \hat{\tau}_j) = (\omega_{ii} + \omega_{jj} - 2\omega_{ij})\sigma^2 = \nu_{ij}\sigma^2, \text{ say,}$$

where $i \neq j$ and ω_{ij} is the (ij)th element of $\mathbf{A}^+$, where $\mathbf{A}$ is given in (1.2). The pairwise variances for a complete block design are $2\sigma^2/r$ for all $i \neq j$ (John and Williams, 1995, p28), and therefore, e_{ij} for the pairwise comparison $\tau_i - \tau_j$ is given by

$$e_{ij} = \frac{2\sigma^2/r}{\nu_{ij}\sigma^2} = \frac{2}{r\nu_{ij}}$$

To compare the distributions of the pairwise efficiency factors, the e_{ij} values were calculated for two designs from the same interchange acceptance path. The two designs chosen were the best design obtained at the end of the random descent phase D_{rd}, and the best design found when $\text{Con}(\mathbf{A}_d)$ was stopped D_{final}. The e_{ij} values were calculated for these two designs from the interchange acceptance paths given in Figure 5.2 for the four–replicate resolvable row–column designs with 1600 and 2000 treatments. The corresponding distributions of the e_{ij} values are given in Figure 5.3.

The distributions of the pairwise efficiency factors are clearly positively skewed and multi–modal. The peaks in the distributions correspond to treatments with the same row and column concurrences. It is recommended that a design is chosen which has as few distinct concurrence groups as possible, and that the concurrences in rows and columns are as equal as possible (Whitaker *et al.*, 1999). The highest peak at the very left of the distributions in Figure 5.3 correspond to treatment pairs that do not occur in the same row or column in any of the four replicates. For designs with a large number of treatments, this peak will be the most prominent,

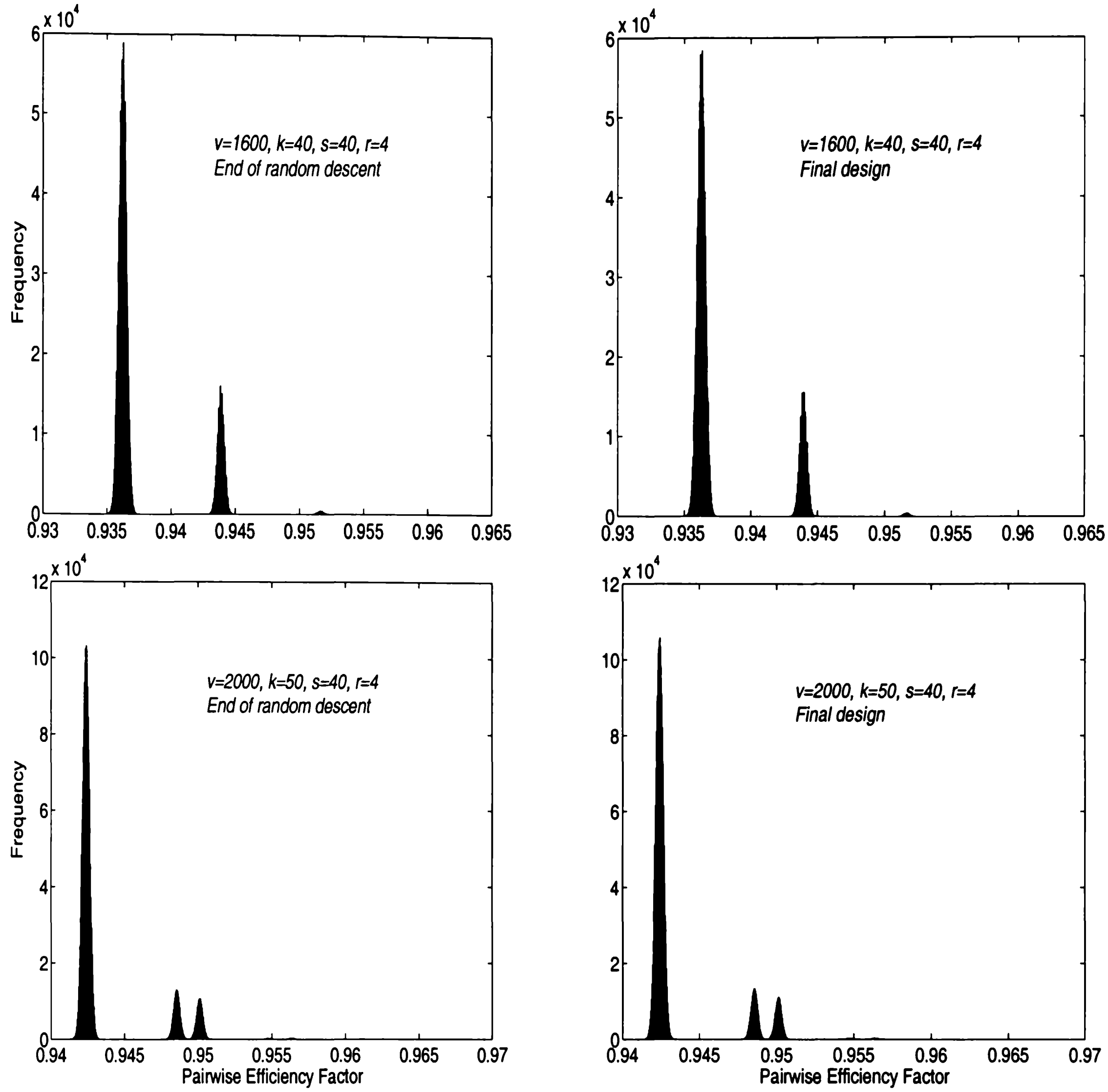

Figure 5.3: Distributions of pairwise efficiency factors

but it is desirable to keep the number of treatment pairs occurring in this group as few as possible. The histograms for the two designs with 1600 treatments has a second prominent peak to the right. These pairwise efficiency factors correspond to treatment pairs which occur in exactly one row or one column together in the four replicates. The row and column concurrences equal to one are grouped together for this design as the number of rows is equal to the number of columns in each replicate. In the histograms for the two designs with 2000 treatments, the next two visible peaks to the right correspond respectively, to treatment pairs which only occur in one row together in the design, and treatment pairs which only occur in one column together. As the number of row and/or column concurrences increases, the pairwise efficiency factor of the treatment pair also increases.

If there is no change in the distribution of the pairwise efficiency factors during the simulated annealing phase, there is no loss in accepting the best design found at the end of the random descent phase. If the distributional properties of the e_{ij} values do continue to improve, such as an overall increase in the e_{ij} values, a reduction in the range, or an increase in the minimum e_{ij}, then it will be worthwhile allowing the algorithm to run into the simulated annealing phase.

Table 5.4 gives some summary statistics of the e_{ij} values from the distributions presented in Figure 5.3. For the resolvable row–column design with 1600 treatments, the minimum, median and mean of the e_{ij} values increase for D_{final} in comparison to D_{rd}. Increases in these summary statistics are considered an improvement in the distributional properties of the pairwise efficiency factors. A further improvement can be noted from Table 5.4 in that the range of the e_{ij} values decreases. The magnitude of these improvements is in the order of the fifth decimal place.

Table 5.4: Summary statistics for e_{ij}

	v=1600, k=40, s=40, r=4		v=2000, k=50, s=40, r=4	
	D_{rd}	D_{final}	D_{rd}	D_{final}
minimum	0.93475	0.93477	0.94123	0.94119
median	0.93632	0.93637	0.94244	0.94246
mean	0.93772	0.93777	0.94358	0.94360
maximum	0.95988	0.95987	0.96396	0.96449
range	0.02513	0.02510	0.02273	0.02330
standard deviation	0.00313	0.00313	0.00271	0.00270

From Table 5.4 it can be seen that the median and mean of the e_{ij} values both increase from D_{rd} to D_{final} for the resolvable row–column design with 2000 treatments. For this interchange acceptance path, the minimum e_{ij} values decreases and the maximum increases. This is not consistent with improvements to the distribution of the e_{ij} values. Very small changes in the summary statistics were found for different random number seeds and different parameter sets.

There are no obvious differences in the distributions of the pairwise efficiency factors for the designs in Figure 5.3. The summary statistics of the pairwise efficiency factors given in Table 5.4 show there are improvements in the distribution of the e_{ij} values between D_{rd} and D_{final} within each parameter set. The magnitude of these improvements is, however, very small. This suggests there is no loss by stopping the algorithm at the end of the random descent phase. If the user does have the resources to allow the algorithm to run into the simulated annealing stage, minor improvements to the distribution of the e_{ij} values are possible.

5.5 Conclusions

When constructing resolvable row–column designs with a large number of treatments, Con($\mathbf{A}_d$) is the best algorithm, but further improvements are still possible. By not calculating the average efficiency factor, as in Con(M,S), the speed of generating designs can be dramatically improved. A feature of Con(M,S) is that any treatment interchange which decreases the value of the (M,S)–optimality objective function becomes the best design. It was shown in Section 5.3 that this has no adverse effects on the best design found. Therefore, it is recommended that Con(M,S) is used to generate resolvable row–column designs which have a large number of treatments.

For large designs, the average efficiency factor does not improve dramatically beyond the end of the random descent phase. Results have shown that only minor improvements to the distributional properties of the pairwise efficiency factors occur during the simulated annealing phase. For designs with a large number of treatments, the user should run Con(M,S) to at least the end of the random descent phase as this would result in an efficient design being generated quickly.

Chapter 6

Upper Bounds for the Average Efficiency Factor

6.1 Introduction

Upper bounds for the average efficiency factor E are a useful tool for providing the extent to which further improvement to any given design is possible. A computer search for good designs could be stopped when a design is found where E is considered to be sufficiently close to the upper bound. Often an optimum design is not achievable, so tight upper bounds are required to reduce the difference between the average efficiency factor of the best design possible and the upper bound.

Numerous upper bounds for block designs have been developed and for a given set of design parameters the best upper bound is the minimum available; see for example Williams and Patterson (1977), Jarrett (1977, 1983, 1989), Paterson and Wild (1986) and Tjur (1990). These methods for calculating block upper bounds can be applied to the resolvable block designs given by the row and column component designs and then combined using the method of Eccleston and McGilchrist (1985) to give an upper bound for the resolvable row–column design. The most recent development in upper bounds for resolvable row–column designs was the method of John and Street (1992).

Upper bounds for the dual of a resolvable block design can also be used to derive an upper bound for the block design; see for example Patterson and Williams (1976b), Williams and Patterson (1977) and Jarrett (1989). This approach often leads to better upper bounds for resolvable block designs where $v > b$ by exploiting properties of the block concurrence matrix (John and Williams, 1995, p81). For a discussion on upper bounds see John and Williams (1995) and Jarrett (1989).

Table 6.1 gives examples of resolvable row–column designs where the best average efficiency factor found is not close to the upper bound. The upper bound U is the minimum of several bounds calculated by CycDesigN version 2.0, including John and Street (1992) and the combining of resolvable block bounds using Eccleston and McGilchrist (1985). The average efficiency factor E given in Table 6.1 is the maximum CycDesigN was able to achieve from numerous runs. It is possible that CycDesigN has been unsuccessful at finding an optimum design, however, it is more likely that a design with E sufficiently close to the upper bound is simply not achievable. If a computer algorithm is set to stop when the average efficiency factor is sufficiently close to the upper bound, the algorithm will not be terminated for the designs in Table 6.1.

Table 6.1: Weak upper bounds

v	k	r	E	U	% of U
12	3	3	0.5076	0.5329	95.25
15	3	3	0.5371	0.5604	95.84
20	4	3	0.6047	0.6239	96.92
42	3	6	0.6098	0.6264	97.35

A new upper bound for resolvable row–column designs is developed in Section 6.2 which is based on the third order bound for block designs from Jarrett (1983), the corrected second moment bound for block designs from Jarrett (1989) and the corrected third moment bound for block designs from Williams and Patterson (1977). This new upper bound is compared with the best upper bound provided by CycDesigN version 2.0 and the results are presented in Section 6.3.

6.2 New Upper Bound

There are two basic approaches to calculating upper bounds for the average efficiency factor of resolvable row–column designs. The upper bound for E can be calculated directly, or it can be calculated via an upper bound for the average efficiency factor E_d of the dual of the resolvable row–column design.

The upper bound for the average efficiency factor E developed in this section uses the second approach given above. It is based on the third order bound given by Jarrett (1983) for a block design with v treatments in rs blocks of size k where each treatment occurs exactly r times. Let $e_1, e_2, \ldots, e_{v-1}$ be the canonical efficiency factors of the resolvable block design, then

$$U_b = \bar{e} - \frac{S_{2b}^2}{(v-1)(S_{3b} + \bar{e}S_{2b})}$$

where $\bar{e} = v(k-1)/k(v-1)$, $S_{2b} = \sum_i (e_i - \bar{e})^2$ and $S_{3b} = \sum_i (e_i - \bar{e})^3$. Applying this third moment bound to E_d and defining $e_{d1}, e_{d2}, \ldots, e_{dn}$ as the $n = r(s+k-2)$ canonical efficiency factors for the dual of the row–column design, gives

$$U_d = \bar{e}_d - \frac{S_{2d}^2}{n(S_{3d} + \bar{e}_d S_{2d})} \tag{6.1}$$

where $\bar{e}_d = (r-1)/r$ (Patterson and Williams, 1976b), $S_{2d} = \sum_i (e_{di} - \bar{e}_d)^2$ and $S_{3d} = \sum_i (e_{di} - \bar{e}_d)^3$.

From the upper bound U_d on the average efficiency factor of the dual of the row–column design, it is possible to calculate an upper bound U_r on the average efficiency factor of the resolvable row–column design. Using the relationship between the average efficiency factors of the dual design and the row–column design given in (3.21), it follows that

$$U_r = \frac{v-1}{r(k+s-2)U_d^{-1} + (v-1) - r(k+s-2)} \tag{6.2}$$

By substituting (6.1) into (6.2) an upper bound U_r on E is obtained.

To calculate the third order bound given by (6.1), lower bounds for the corrected second moment S_{2d} and the corrected third moment S_{3d} are required. It will be shown that the lower bounds for S_{2d} and S_{3d} for the dual of the resolvable row–column design are simple extensions of the corrected second and third moment bounds for the dual of a resolvable block design given by Jarrett (1989) and Williams and Patterson (1977) respectively.

The lower bound for the corrected second moment for the dual of a resolvable block design is given by Jarrett (1989) as

$$S_{2bL} = r(r-1)s^2\beta(1-\beta)/(rk)^2 \tag{6.3}$$

where β is the fractional part of k/s. Williams and Patterson (1977) give the corrected third moment bound for the dual of a resolvable block design as

$$S_{3bL} = r(r-1)(r-2)s^2\beta x/(rk)^3$$

where

$$x = \begin{cases} s\beta^2 & \text{if } \beta \leq 1/2 \text{ and } k \geq s \\ s\beta^2 - 1 & \text{if } \beta \leq 1/2 \text{ and } k < s \\ s(1-\beta)^2 & \text{if } \beta > 1/2 \end{cases} \tag{6.4}$$

In order to show that a lower bound for S_{2d} is an extension of (6.3), S_{2d} is first expressed in terms of the row and column incidence matrices for the contraction arrays and secondary arrays, that is

$$S_{2d} = \frac{2}{r^2}\text{trace}(\mathbf{HH}') - \frac{4(r-1)}{r} \tag{6.5}$$

where trace($\mathbf{HH}'$) is given in (3.25) as

$$\text{trace}(\mathbf{HH}') = \sum_{i=1}^{h}[\text{ss}(\mathbf{R}_{1i})/s^2 + \text{ss}(\mathbf{C}_{1i})/ks + \text{ss}(\mathbf{R}_{2i})/ks + \text{ss}(\mathbf{C}_{2i})/k^2]$$

A lower bound for S_{2d} can be calculated by finding a lower bound for trace($\mathbf{HH}'$). The theoretical minimum of trace($\mathbf{HH}'$) can be established by considering the counting rules given in (3.1) and (3.2), that is, for $j = 1, 2, \ldots, h$

$$\mathbf{C}_{1j}\mathbf{1} = s\mathbf{1}, \quad \mathbf{C}'_{1j}\mathbf{1} = k\mathbf{1}, \quad \mathbf{C}_{2j}\mathbf{1} = \mathbf{C}'_{2j}\mathbf{1} = k\mathbf{1}$$

$$\mathbf{R}_{1j}\mathbf{1} = \mathbf{R}'_{1j}\mathbf{1} = s\mathbf{1}, \quad \mathbf{R}_{2j}\mathbf{1} = k\mathbf{1}, \quad \mathbf{R}'_{2j}\mathbf{1} = s\mathbf{1}$$

The minimum sum of squares of $\mathbf{C}_{1j}$ and $\mathbf{R}_{2j}$ occurs when all elements equal 1. Therefore, the minimum sum of squares for these incidence matrices is sk. Designs with this property are said to be adjusted orthogonal.

For $\mathbf{C}_{2j}$ and $\mathbf{R}_{1j}$, the minimum sum of squares for each matrix will occur when the elements within each matrix are as uniform as possible. Assuming the elements in $\mathbf{C}_{2j}$ differ by at most 1, the minimum entry will be $(k/s - \beta)$ and this value will occur $s^2(1-\beta)$ times. The maximum entry is therefore $(k/s - \beta + 1)$, which will occur βs^2 times. The minimum sum of squares for each $\mathbf{C}_{2j}$ matrix is then given by $[k^2 + \beta s^2(1-\beta)]$. Similarly, the minimum sum of squares for each $\mathbf{R}_{1j}$ matrix is $[s^2 + \alpha k^2(1-\alpha)]$, where α is the fractional part of s/k.

Therefore, the theoretical minimum for trace($\mathbf{HH}'$) is given by

$$\text{trace}(\mathbf{HH}')_L = \frac{r(r-1)}{2}\left(4+\beta\frac{s^2}{k^2}(1-\beta)+\alpha\frac{k^2}{s^2}(1-\alpha)\right) \qquad (6.6)$$

Substituting (6.6) for trace($\mathbf{HH}'$) in (6.5), gives the lower bound for the corrected second moment S_{2d} as

$$S_{2dL} = \frac{r(r-1)s^2\beta(1-\beta)}{(rk)^2}+\frac{r(r-1)k^2\alpha(1-\alpha)}{(rs)^2} \qquad (6.7)$$

It can be seen that this lower bound is simply the sum of the lower bounds of the corrected second moments for the dual designs of the row component and the column component designs, as given in (6.3).

It can also be shown, if adjusted orthogonality is assumed, that S_{3d} can be seperated into two terms representing the dual of the row component design and the dual of the column component design. Hence, a lower bound for S_{3d} is the sum of the lower bounds for the dual designs of the row and column component designs. With the property of adjusted orthogonality, $\mathbf{N}_k'\mathbf{N}_s = \mathbf{J}$ and $\mathbf{A}_d$, as given in (3.3), can be expressed as

$$\mathbf{A}_d = \mathbf{V}_0 + \begin{pmatrix} \mathbf{A}_{dk} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{ds} \end{pmatrix}$$

where

$$\mathbf{V}_0 = \begin{pmatrix} \mathbf{0} & pq(\mathbf{I}_r \otimes \mathbf{J}_{ks}) - \frac{pq}{r}\mathbf{J} \\ pq(\mathbf{I}_r \otimes \mathbf{J}_{sk}) - \frac{pq}{r}\mathbf{J} & \mathbf{0} \end{pmatrix}$$

and $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ are the information matrices for the dual designs of the row and column component designs respectively; see Section 3.8. $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ are defined as

$$\mathbf{A}_{dk} = \mathbf{I} - \tfrac{1}{rs}\mathbf{N}_k'\mathbf{N}_k \qquad \mathbf{A}_{ds} = \mathbf{I} - \tfrac{1}{rk}\mathbf{N}_s'\mathbf{N}_s$$

It is known from Section 3.5 that the canonical efficiency factors of $\mathbf{A}_d$ correspond to eigenvectors in the form $\mathbf{z} = (\mathbf{w}' \ \ \mathbf{y}')'$ where $\mathbf{w} = (\mathbf{w}_1' \ \ \mathbf{w}_2' \ldots \mathbf{w}_r')'$ and $\mathbf{y} = (\mathbf{y}_1' \ \ \mathbf{y}_2' \ldots \mathbf{y}_r')'$ and $\mathbf{w}_i$ are vectors of length k and $\mathbf{y}_i$ are vectors of length s $(i = 1, 2, \ldots, r)$. It is shown in (3.12) that $\mathbf{w}_i'\mathbf{1}_k = \mathbf{y}_i'\mathbf{1}_s = 0$, therefore it follows that $\mathbf{V}_0\mathbf{z} = \mathbf{0}$.

For any $n \times n$ block diagonal matrix $\mathbf{B}$ with square diagonal blocks $\mathbf{B}_{11}$, $\mathbf{B}_{22}, \ldots, \mathbf{B}_{rr}$, it is known that the eigenvalues of $\mathbf{B}$ are given by the eigenvalues of the matrices $\mathbf{B}_{11}, \mathbf{B}_{22}, \ldots, \mathbf{B}_{rr}$. This result can be found in matrix algebra texts including Harville (1997, p523). The $r(k+s-2)$ canonical efficiency factors of $\mathbf{A}_d$ are therefore given by the $r(k-1)$ canonical efficiency factors of $\mathbf{A}_{dk}$ and the $r(s-1)$ canonical efficiency factors of $\mathbf{A}_{ds}$.

The arithmetic means of the canonical efficiency factors for the dual designs of the resolvable row–column design and the row and column component designs are all equal to $\bar{e_d} = (r-1)/r$; see Patterson and Williams (1976b) and John and Williams (1995, p82). It then follows for any $m > 1$, that

$$\sum_{i=1}^{r(k+s-2)} (e_{di} - \bar{e}_d)^m = \sum_{i=1}^{r(k-1)} (\lambda_i - \bar{e}_d)^m + \sum_{i=1}^{r(s-1)} (\mu_i - \bar{e}_d)^m$$

where e_{di}, λ_i and μ_i are the canonical efficiency factors of $\mathbf{A}_d$, $\mathbf{A}_{dk}$ and $\mathbf{A}_{ds}$ respectively. The corrected mth moment can therefore be expressed in terms of the dual of the row component design and the dual of the column component design. Hence, the corrected third moment bound for the dual of a resolvable block design can be extended to apply to the dual of a resolvable row–column design. A lower bound for S_{3d} is therefore given by

$$S_{3dL} = \frac{r(r-1)(r-2)s^2\beta x}{(rk)^3} + \frac{r(r-1)(r-2)k^2\alpha z}{(rs)^3} \tag{6.8}$$

where x is defined in (6.4) and

$$z = \begin{cases} k\alpha^2 & \text{if } \alpha \le 1/2 \text{ and } s \ge k \\ k\alpha^2 - 1 & \text{if } \alpha \le 1/2 \text{ and } s < k \\ k(1-\alpha)^2 & \text{if } \alpha > 1/2 \end{cases}$$

Substituting (6.7) for S_{2d} and (6.8) for S_{3d} in (6.1) gives an upper bound on E_d. This bound is then substituted for U_d in (6.2) to give an upper bound U_r on E.

An alternative approach is based on the research of John and Street (1992) where the corrected second moment can be expressed in terms of trace($\mathbf{W}^2$). They also derive a closed form expression for trace($\mathbf{W}^2$) based on an integer programming formulation of the problem. This derivation is not a true bound due to the assumption that the minimum occurs when the concurrences in $\mathbf{N}_k\mathbf{N}_k'$ and $\mathbf{N}_s\mathbf{N}_s'$ differ by at one and two respectively. John and Street (1993) show that a lower bound is possible when a wider range of concurrences is considered.

When calculating the corrected second moment bound to be used in the second order bound of Jarrett (1989), the assumption is made that the concurrences in $\mathbf{NN}'$

differ by at most one. The lower bound for the corrected second moment based on John and Street (1993) does not make this assumption. Using this corrected second moment bound in (6.1) and substituting U_d in (6.2), does not produce a true upper bound for E. As an example, using the above approach to calculate an upper bound for $v = 16$, $k = 4$, $s = 4$ and $r = 3$ produces an apparent upper bound of 0.5670. However, there exists a design with an average efficiency factor of 0.5696 which is higher than the upper bound.

To obtain a true upper bound using the second order bound of Jarrett (1989), the corrected second and third moment bounds must be based on the assumption that the concurrences differ by at most one.

6.3 Comparison of Upper Bounds

CycDesigN calculates several upper bounds and reports the tightest of these. The bounds calculated by CycDesigN include the row–column bound of John and Street (1992), and the combining of various block bounds using Eccleston and McGilchrist (1985). Included in the block bounds are those of Patterson and Williams (1976b), Jarrett (1983), Tjur (1990) and the arithmetic mean bound.

Upper bounds were calculated for the 864 resolvable row–column designs with parameters $12 \leq v \leq 100$, $3 \leq k \leq s$ and $2 \leq r \leq 10$. Of these parameter sets, no tighter bounds were produced by the new bound. However, in 12 cases the best upper bound calculated using the new approach was equal to the upper bound given by CycDesigN.

Appendix A

Estimable Functions

Theorem A.1 The $(r-1)$ eigenvalues of $\mathbf{A}_d$ equal to 2 are not canonical efficiency factors as the corresponding eigenvectors are not members of the treatment contrast space.

Proof. $\mathbf{A}_d$ is a square matrix of order $r(k+s)$ and $\text{rank}(\mathbf{A}_d) = [r(k+s)-r-1] = n$. In canonical form $\mathbf{A}_d$ can be written as

$$\mathbf{A}_d = \sum_{i=1}^{n} \lambda_i \mathbf{x}_i \mathbf{x}_i' \qquad (A.1)$$

where λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{x}_i$ satisfying

$$\mathbf{x}_i'\mathbf{x}_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{r-1}$ be the $(r-1)$ eigenvectors of $\mathbf{A}_d$ with eigenvalue 2 which are given by the columns of $\mathbf{X}_2$, where

$$\mathbf{X}_2 = \begin{pmatrix} q\mathbf{T} \otimes \mathbf{1}_k \\ p\mathbf{T} \otimes \mathbf{1}_s \end{pmatrix}$$

where $\mathbf{T}$ is an $r \times (r-1)$ matrix whose columns comprise a set of orthonormal contrasts, that is, $\mathbf{T}'\mathbf{T} = \mathbf{I}$ and $\mathbf{T}'\mathbf{1} = \mathbf{0}$.

To be members of the contrast space $\mathbf{X}_2'\boldsymbol{\tau}$ must be estimable, where $\boldsymbol{\tau} = (\boldsymbol{\alpha}'\ \boldsymbol{\gamma}')'$ is a function of the treatment parameters. For $\mathbf{X}_2'\boldsymbol{\tau}$ to be estimable, it is a necessary and sufficient condition that

$$\mathbf{X}_2' = \mathbf{X}_2'\mathbf{A}_d^+\mathbf{A}_d$$

(Searle, 1997, p185) where $\mathbf{A}_d^+$ is the Moore–Penrose inverse of $\mathbf{A}_d$ given by

$$\mathbf{A}_d^+ = \sum_{i=1}^{n} \lambda_i^{-1} \mathbf{x}_i \mathbf{x}_i'$$

Therefore

$$\begin{aligned} \mathbf{X}_2' = \mathbf{X}_2' \mathbf{A}_d^+ \mathbf{A}_d &= \mathbf{X}_2' (\sum_{i=1}^{n} \lambda_i^{-1} \mathbf{x}_i \mathbf{x}_i')(\sum_{i=1}^{n} \lambda_i \mathbf{x}_i \mathbf{x}_i') \\ &= \mathbf{X}_2' \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i' \end{aligned}$$

It follows from (A.1) that the full set of eigenvectors $\mathbf{x}_i$ $(i = 1, 2, \ldots, r(k+s))$ satisfy

$$\sum_{i=1}^{r(k+s)} \mathbf{x}_i \mathbf{x}_i' = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i' + \sum_{i=n+1}^{r(k+s)} \mathbf{x}_i \mathbf{x}_i' = \mathbf{I}$$

where $\mathbf{x}_{n+1}, \mathbf{x}_{n+2}, \ldots, \mathbf{x}_{r(k+s)}$ are the eigenvectors corresponding to zero eigenvalues.

If $\mathbf{X}_2' \boldsymbol{\tau}$ is estimable, then $\sum_{i=n+1}^{r(k+s)} \mathbf{x}_i \mathbf{x}_i' = \mathbf{0}$. For a r–replicate resolvable row–column design, the $(r+1)$ eigenvectors of $\mathbf{A}_d$ with zero eigenvalues are

$$\mathbf{x}_{n+1} = \begin{pmatrix} q\mathbf{1}_{rk} \\ p\mathbf{1}_{rs} \end{pmatrix} \qquad \begin{pmatrix} \mathbf{x}_{n+2} & \mathbf{x}_{n+3} & \ldots & \mathbf{x}_{r(k+s)} \end{pmatrix} = \begin{pmatrix} q\mathbf{I}_r \otimes \mathbf{1}_k \\ -p\mathbf{I}_r \otimes \mathbf{1}_s \end{pmatrix}$$

It can be shown that

$$\sum_{i=n+1}^{r(k+s)} \mathbf{x}_i \mathbf{x}_i' = \begin{pmatrix} q^2(\mathbf{I}_r \otimes \mathbf{J}_{kk}) + q^2\mathbf{J} & -pq(\mathbf{I}_r \otimes \mathbf{J}_{ks}) + pq\mathbf{J} \\ -pq(\mathbf{I}_r \otimes \mathbf{J}_{sk}) + pq\mathbf{J} & p^2(\mathbf{I}_r \otimes \mathbf{J}_{ss}) + p^2\mathbf{J} \end{pmatrix} \neq \mathbf{0}$$

Therefore, the $(r-1)$ eigenvectors of $\mathbf{A}_d$ equal to 2 are not estimable functions as

$$\mathbf{X}_2' \neq \mathbf{X}_2' \mathbf{A}_d^+ \mathbf{A}_d$$

Appendix B

Trace of the Moore–Penrose Inverse

Theorem B.1 Trace($\mathbf{A}_{dk}^{+}$) can be calculated as a function of trace($\mathbf{A}_{Mk}^{-1}$), that is

$$\text{trace}(\mathbf{A}_{dk}^{+}) = \text{trace}(\mathbf{A}_{Mk}^{-1}) + \tfrac{(r-4)}{3}$$

where

$$\begin{aligned}
\mathbf{A}_{dk} &= \mathbf{I} - \tfrac{p^2}{r}\mathbf{N}_k'\mathbf{N}_k \\
\mathbf{M}_{11} &= \tfrac{q^2}{2r}\mathbf{J} + \tfrac{q^2}{2}(\mathbf{I}_r \otimes \mathbf{J}_{kk}) \\
\mathbf{A}_{Mk} &= \mathbf{A}_{dk} + \mathbf{M}_{11}
\end{aligned}$$

and $p^2 = 1/s$ and $q^2 = 1/k$.

Proof. Using (3.1) and (3.2) it can been seen that $\mathbf{x}_1 = (q\mathbf{1}_{rk}/\sqrt{r})$ is an eigenvector of $\mathbf{A}_{dk}$ with eigenvalue 0 and an eigenvector of $\mathbf{M}_{11}$ with eigenvalue 1.

Let

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_2 & \mathbf{x}_3 & \dots & \mathbf{x}_r \end{pmatrix} = (q\mathbf{T} \otimes \mathbf{1}_k)$$

where $\mathbf{T}$ is an $r \times (r-1)$ matrix whose columns comprise a set of orthonormal contrasts, that is, $\mathbf{T}'\mathbf{T} = \mathbf{I}$ and $\mathbf{T}'\mathbf{1} = \mathbf{0}$. The columns of $\mathbf{X}$ are orthogonal to $\mathbf{x}_1$, as $\mathbf{X}'\mathbf{x}_1 = \mathbf{0}$. Using (3.1) and (3.2) it can be shown that the columns of $\mathbf{X}$ are eigenvectors of $\mathbf{A}_{dk}$ with eigenvalues 1 as $\mathbf{A}_{dk}\mathbf{X} = \mathbf{X}$, and are eigenvectors of $\mathbf{M}_{11}$ with eigenvalues 1/2 as $\mathbf{M}_{11}\mathbf{X} = \mathbf{X}/2$.

Any other eigenvector of $\mathbf{A}_{dk}$ and $\mathbf{M}_{11}$ must be orthogonal to $\mathbf{x}_i$ $(i = 1, 2, \ldots, r)$. Let $\mathbf{w} = (\mathbf{w}_1 \;\; \mathbf{w}_2 \ldots \mathbf{w}_r)$ where $\mathbf{w}_i$ are vectors of length k $(i = 1, 2, \ldots r)$. If $\mathbf{w}$ is an eigenvector of $\mathbf{A}_{dk}$ and $\mathbf{M}_{11}$, then

$$\begin{aligned} \mathbf{w}'\mathbf{x}_1 &= \tfrac{q}{\sqrt{r}}\mathbf{w}'\mathbf{1}_{rk} = 0 \\ \mathbf{w}'\mathbf{X} &= q\mathbf{w}'(\mathbf{T} \otimes \mathbf{1}_k) = \mathbf{0} \end{aligned}$$

which implies that

$$\mathbf{w}_i'\mathbf{1}_k = 0 \tag{B.1}$$

It then follows using (B.1) that $\mathbf{M}_{11}\mathbf{w} = \mathbf{0}$, hence the remaining $r(k-1)$ eigenvalues of $\mathbf{M}_{11}$ are equal to 0. The remaining $r(k-1)$ eigenvalues of $\mathbf{A}_{dk}$ are non–trivial and will be denoted by λ_i $(i = r+1, r+2, \ldots, rk)$.

$\mathbf{A}_{dk}$ and $\mathbf{M}_{11}$ can be expressed in canonical form as

$$\mathbf{A}_{dk} = \textstyle\sum_{i=2}^{rk} \lambda_i \mathbf{x}_i \mathbf{x}_i' \qquad \mathbf{M}_{11} = \sum_{i=1}^{r} \mu_i \mathbf{x}_i \mathbf{x}_i' \tag{B.2}$$

where λ_i and μ_i are the non–zero eigenvalues of $\mathbf{A}_{dk}$ and $\mathbf{M}_{11}$ respectively, corresponding to eigenvector $\mathbf{x}_i$. The Moore–Penrose inverse of $\mathbf{A}_{dk}$ is then given by

$$\mathbf{A}_{dk}^+ = \sum_{i=2}^{rk} \lambda_i^{-1} \mathbf{x}_i \mathbf{x}_i'$$

From (B.2), $\mathbf{A}_{Mk} = (\mathbf{A}_{dk} + \mathbf{M}_{11})$ can be expressed as

$$\mathbf{A}_{Mk} = \mathbf{x}_1\mathbf{x}_1' + \sum_{i=2}^{r}(\lambda_i + \mu_i)\mathbf{x}_i\mathbf{x}_i' + \sum_{i=r+1}^{rk} \lambda_i \mathbf{x}_i\mathbf{x}_i'$$

Therefore

$$\mathbf{A}_{Mk}^{-1} = \mathbf{x}_1\mathbf{x}_1' + \sum_{i=2}^{r}(\lambda_i + \mu_i)^{-1}\mathbf{x}_i\mathbf{x}_i' + \sum_{i=r+1}^{rk} \lambda_i^{-1} \mathbf{x}_i\mathbf{x}_i'$$

$\mathbf{A}_{dk}^+$ can be expressed in terms of $\mathbf{A}_{Mk}^{-1}$, namely

$$\begin{aligned} \mathbf{A}_{dk}^+ &= \mathbf{A}_{Mk}^{-1} - \mathbf{x}_1\mathbf{x}_1' + \sum_{i=2}^{r} \frac{\mu_i}{\lambda_i(\lambda_i + \mu_i)}\mathbf{x}_i\mathbf{x}_i' \\ &= \mathbf{A}_{Mk}^{-1} - \tfrac{q^2}{r}\mathbf{J} + \tfrac{(r-1)}{3}[q\mathbf{T} \otimes \mathbf{1}_k][q\mathbf{T} \otimes \mathbf{1}_k]' \end{aligned}$$

Hence, $\text{trace}(\mathbf{A}_{dk}^+)$ can be expressed as

$$\begin{aligned} \text{trace}(\mathbf{A}_{dk}^+) &= \text{trace}(\mathbf{A}_{Mk}^{-1}) - \text{trace}(\tfrac{q^2}{r}\mathbf{J}) + \text{trace}(\tfrac{(r-1)}{3}[q\mathbf{T} \otimes \mathbf{1}_k][q\mathbf{T} \otimes \mathbf{1}_k]') \\ &= \text{trace}(\mathbf{A}_{Mk}^{-1}) - 1 + \tfrac{(r-1)}{3} \\ &= \text{trace}(\mathbf{A}_{Mk}^{-1}) + \tfrac{(r-4)}{3} \end{aligned}$$

Appendix C

Geometric Expansion

Theorem C.1 If $\frac{1}{r}\mathbf{A} = \mathbf{I} - \frac{1}{v}\mathbf{J} - \mathbf{W}$ where $\mathbf{W} = \frac{1}{rs}\mathbf{N}_k\mathbf{N}_k' + \frac{1}{rk}\mathbf{N}_s\mathbf{N}_s' - \frac{2}{v}\mathbf{J}$, then a geometric expansion of $r\mathbf{A}^+$ is given by

$$r\mathbf{A}^+ = \mathbf{I} - \frac{1}{v}\mathbf{J} + \sum_{i=1}^{\infty}\mathbf{W}^i \tag{C.1}$$

Proof. Let $e_1, e_2, \ldots, e_v$ be the v eigenvalues of $\mathbf{A}/r$ with corresponding orthonormal eigenvectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_v$. It is known that a connected row–column design has one eigenvalue, e_v say, equal to 0, with corresponding eigenvector $\mathbf{x}_v = \mathbf{1}_v/\sqrt{v}$. The remaining $(v-1)$ eigenvalues of $\mathbf{A}/r$, for a connected design, are $0 < e_i \leq 1$ $(i = 1, 2, \ldots, v-1)$. If

$$\mathbf{W}_m = \frac{1}{rs}\mathbf{N}_k\mathbf{N}_k' + \frac{1}{rk}\mathbf{N}_s\mathbf{N}_s' - \frac{m}{v}\mathbf{J}$$

then it follows that $\mathbf{I} - \mathbf{W}_m = \frac{1}{r}\mathbf{A} + \frac{(m-1)}{v}\mathbf{J}$ is non–singular if $m \neq 1$. Hence

$$(\mathbf{I} - \mathbf{W}_m)^{-1} = \mathbf{I} + \sum_{i=1}^{\infty}\mathbf{W}_m^i$$

if and only if $\sum_{i=1}^{\infty}\mathbf{W}_m^i$ converges. It will converge if the absolute value of the maximum eigenvalue of $\mathbf{W}_m$ is less than 1 (Cullen, 1972, p256). The maximum eigenvalue of $\mathbf{W}_m$ corresponds to the eigenvector $\mathbf{x}_v$ and has the value $2-m$. In order for convergence to occur $|2-m| < 1$, which implies that $1 < m < 3$.

For any $n \times n$ symmetric matrices $\mathbf{X}$ and $\mathbf{Y}$, such that $\mathbf{XY} = \mathbf{0}$, $(\mathbf{X}+\mathbf{Y})^+ = \mathbf{X}^+ + \mathbf{Y}^+$ (Harville, 1997, p513). It can be shown that $\mathbf{AJ} = \mathbf{0}$, therefore

$$\left(\frac{1}{r}\mathbf{A} + \frac{(m-1)}{v}\mathbf{J}\right)^+ = r\mathbf{A}^+ + \left(\frac{(m-1)}{v}\mathbf{J}\right)^+$$

For any $n \times n$ non–singular matrix $\mathbf{Z}$, $\mathbf{Z}^{-1} = \mathbf{Z}^{+}$, therefore

$$r\mathbf{A}^{+} = \mathbf{I} - \left(\tfrac{(m-1)}{v}\mathbf{J}\right)^{+} + \sum_{i=1}^{\infty} \mathbf{W}_m^i \tag{C.2}$$

In order for (C.2) to be simplified to (C.1), $\frac{(m-1)}{v}\mathbf{J}$ must be a symmetric idempotent matrix (Harville, 1997, p495). An idempotent matrix $\mathbf{B}$ is one where $\mathbf{B}^2 = \mathbf{B}$. It can be shown that $\frac{(m-1)}{v}\mathbf{J}$ is an idempotent matrix for m such that $(m-1)(m-2) = 0$, therefore m must be equal to 2, giving (C.1).

References

Bailey RA and Patterson HD. (1991) A note on the construction of row–and–column designs with two replicates. *Journal of the Royal Statistical Society*, B, **53**, 645–648.

Ceranka B and Mejza S. (1979) On the efficiency factor for a contrast of treatment parameters. *Biometrical Journal*, **21**, 99–102.

Cullen CG. (1972) *Matrices and Linear Transformations.* Second edition. Addison–Wesley Publishing Company, Reading, Massachusetts.

Eccleston JA and Hedayat A. (1974) On the theory of connected designs: characterization and optimality. *Annals of Statistics*, **2**, 1238–1255.

Eccleston JA and McGilchrist CA. (1985) Algebra of a row column design. *Journal of Statistical Planning and Inference*, **12**, 305–310.

Eccleston JA and Russell KG. (1975) Connectedness and orthogonality in multi–factor designs. *Biometrika*, **62**, 341–345.

Harville DA. (1997) *Matrix Algebra from a Statistician's Perspective.* Springer, New York.

Jarrett RG. (1977) Bounds for the efficiency factor of block designs. *Biometrika*, **64**, 67–72.

Jarrett RG. (1983) Definitions and properties for m–concurrence designs. *Journal of the Royal Statistical Society*, B, **45**, 1–10.

Jarrett RG. (1989) A review of bounds for the efficiency factor of block designs. *The Australian Journal of Statistics*, **31**, 118–129.

Jarrett RG, Piper FC and Wild PR. (1997) Efficient two–replicate resolvable row–column designs. *Journal of Statistical Planning and Inference*, **58**, 65–77.

John JA. (1996) Interchange algorithms for constructing designs with complex blocking structures. *Statistical Theory and Applications*, Nagaraja HN, Prnab K S, Morrison DF (ed.), 233-246, Springer–Verlag, New York.

John JA and Eccleston JA. (1986) Row–column α–designs. *Biometrika*, **73**, 301–306.

John JA, Ruggiero K and Williams ER. (2002) α_n–Designs. *Australian & New Zealand Journal of Statistics*, **44**, 457–465.

John JA and Street DJ. (1992) Bounds for the efficiency factor of row–column designs. *Biometrika*, **79**, 658–661.

John JA and Street DJ. (1993) Bounds for the efficiency factor of row–column designs (amendment). *Biometrika*, **80**, 712–713.

John JA and Whitaker D. (1993) Construction of resolvable row–column designs using simulated annealing. *The Australian Journal of Statistics*, **35**, 237–245.

John JA and Whitaker D. (2000) Recursive formulae for the average efficiency factor in block and row–column designs. *Journal of the Royal Statistical Society*, B, **62**, 575–583.

John JA and Williams ER. (1982) Conjectures for optimal block designs. *Journal of the Royal Statistical Society*, B, **44**, 221–225.

John JA and Williams ER. (1995) *Cyclic and Computer Generated Designs.* Second edition. Chapman and Hall, London.

John JA and Williams ER. (1998) t–Latinized Designs. *Australian & New Zealand Journal of Statistics*, **40**, 111–118.

Kiefer J. (1959) Optimum Experimental Designs. *Journal of the Royal Statistical Society*, B, **21**, 272–319.

Lee MLT, Kuo FC, Whitmore GA and Sklar J. (2000) Importance of replication in microarray gene expression studies: Statistical methods and evidence from repetitive cDNA hybridizations. *Proceedings of the National Academy of Sciences, USA*, **97**, 9834–9839.

Paterson LJ and Wild P. (1986) Triangles and efficiency factors. *Biometrika*, **73**, 289–299.

Patterson HD and Williams ER. (1976a) A new class of resolvable incomplete block designs. *Biometrika*, **63**, 83–92.

Patterson HD and Williams ER. (1976b) Some theoretical results on general block designs. *Proceedings of the Fifth British Combinatorial Conference, Congressus Numerantium* **XV**, 489–496. Utilitas Mathematica, Winnipeg.

Pearce SC, Calinski T and Marshall TF de C. (1974) The basic contrasts of an experimental design with special reference to the analysis of data. *Biometrika*, **61**, 449–460.

Shah KR. (1960) Optimality criteria for incomplete block designs. *Annals of Mathematical Statistics*, **31**, 791–794.

Simon R, Radmacher MD and Dobbin K. (2002) Design of studies using DNA microarrays. *Genetic Epidemiology*, **23**, 21–26.

Searle SR. (1997) *Linear Models.* John Wiley & Sons, New York.

Tjur T. (1990) A new upper bound for the efficiency factor of a block design. *The Australian Journal of Statistics*, **32**, 231–237.

Venables WN and Eccleston JA. (1993) Randomized search strategies for finding optimal or near optimal block and row–column designs. *The Australian Journal of Statistics*, **35**, 371–382.

Whitaker D. (1995) A nested simulated annealing algorithm. *Journal of Statistical Computation and Simulation*, **53**, 233–241.

Whitaker D, Williams ER and John JA. (1999) *CycDesigN: A Package for the Computer Generation of Experimental Designs.* Version 1.2. CSIRO Forestry and Forest Products, Canberra.

Whitaker D, Williams ER and John JA. (2002) *CycDesigN: A Package for the Computer Generation of Experimental Designs.* Version 2.0. CSIRO Forestry and Forest Products, Canberra.

Williams ER and Patterson HD. (1977) Upper bounds for efficiency factors in block designs. *The Australian Journal of Statistics*, **19**, 194–201.