Comments on algorithmic design

Robert E. Wheeler

Contents

1 Introduction

This paper has three goals:

1. To briefly describe algorithmic design in a way that will be helpful to those who primarily use other experimental design methodologies.
2. To provide examples illustrating the use of the functions in AlgDesign.
3. To document the mathematical expressions used in the programs and call attention to any differences from those available in the literature.

The general model used for experimental design is

$$Y = X\beta + \Sigma_{i=1}^{b} Z_i\gamma_i + \epsilon, \tag{1}$$

where Y is a vector of N observations, X is a $N \times k$ matrix, $Z_i, (i = 1 \ldots b)$ are $N \times m_i$ matrices. The vectors $\gamma_i, (i = 1 \ldots b)$ and the N vector ϵ, are random and independent of each other with variances $\sigma_i^2 I_{m_i}$ and $\sigma_\epsilon^2 I$, respectively.

The variance of Y is $V\sigma_\epsilon^2$ where $V = \Sigma_{i=1}^{b} X_i X_i^T \rho_i^2 + I$, and $\rho_i = \sigma_i^2/\sigma_\epsilon^2$. The generalized least squares estimator of β is $\hat{\beta} = (X^T V^{-1} X)^{-1} X^T V^{-1} Y$, and thus the covariance is $M^{-1}\sigma^2/N$, where $M = X^T V^{-1} X/N$.

From this it may be seen that the "best" estimates of the parameters are obtained when M^{-1} is made as small as possible in some sense.

Established, or classical, designs that are described in the various textbooks and papers on the subject, all fit within a larger framework wherein designs are constructed to meet various criteria for the information matrix, region of experimentation, and model. Algorithmic optimization of these criteria will produce the above mentioned designs. Simply replaying tables of such designs is not the goal of algorithmic design, but rather the goal is to extend these criteria to the often awkward conditions of practical experimentation which defy standard designs, and to produce highly efficient designs not in the canon.

For example, a 3^{6-1} factorial requires 243 observations to estimate 28 terms if the model is quadratic; thus, there are 215 degrees of freedom for error, which even to the most conservative experimenter, must seem to be a great many. Of course, power calculations may indicate that one needs many observations in order to estimate the coefficients with adequate precision. More commonly, however, one would be satisfied with something like 10 degrees of freedom for error and would like an experiment with about 40 observations. The canon contains no suitable design, but an algorithmic calculation will create a 40 run design with coefficient variances only 10% larger than those for the 243 observation fractional factorial.

The model described by equation (1) contains multiple error terms. This is quite unrealistic. To be sure, in the analysis, estimates of the several variance components can be obtained, but in the design phase, there is very seldom a rational basis for their specification, and the "best" design does depend on their values. Fortunately, a great many experimental problems involve only one or at the most two variance components. The problem of designing with multiple error terms is not one that has not been generally solved for established designs.

A simply written survey of some of the ideas involved is presented in Chapter 7 of Cox and Reid [2000], and a longer, but equally simply written, exposition may be found in the first part of Atkinson and Donev [1992].

2 The varieties of designs

2.1 One-way layouts

The very first statistical experiment Peirce and Jastrow [1884] was a simple one-way layout with equal numbers of observations at each level. The estimates were uncorrelated because randomization was employed (its first use in an experimental setting). This simple experimental arrangement may be represented by a table with observed values y_i in the cells:

level 1	y_1
level 2	y_2
level 3	y_3

For this, the structure of X could be

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which means that M is diagonal, and the estimates uncorrelated. If there are n_i replications at each level, then X comprises replications of the rows of I, and

$$M = \frac{1}{N} \begin{pmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \end{pmatrix},$$

where $N = \sum n_i$.

The estimates of the parameters are easy to compute, being the means of the cells, $\tilde{\beta}_i = \bar{y}_i$ for $i = 1, 2, 3$. Simplicity of computation was very important in the days before computers, and many standard designs require nothing more than simple arithmetic for their analysis. In order to preserve this simplicity, various schemes have been developed for "filling in" missing data to preserve the computational protocol: but this is another topic, and our concern is with design.

The only design question is the number of observations in each cell. If N is fixed, then it is easy to see[1]that the expected variance of any pairwise contrast of the parameter estimates is minimized by making the n_i equal. Almost all of the discussion of efficiency in the standard texts, such as Cochran and Cox [1950] is in terms of pairwise contrasts, and by and large this has been the major criterion in constructing designs. It is however a specialized instance of more general criteria. In particular, maximizing the determinant of M leads to the same design.

The form of X is not unique, since any non-singular linear transformation of the parameters, say from $X\beta$ to $XTT^{-1}\beta = Z\gamma$, will leave unchanged the statistical characteristics of the experiment. For example, the F-ratio for the single factor is unchanged by such a transformation.

A common transformation restates the parameters in terms of deviations from their mean, producing "main effects." For example, one might use

$$T = \begin{pmatrix} 1 & -1 & -1 \\ 1 & 1 & 0 \\ 1 & 0 & -1 \end{pmatrix},$$

with

$$T^{-1} = \frac{1}{3}\begin{pmatrix} 1 & 1 & 1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix}.$$

The first form has a parameter $(\beta_1, \beta_2, \beta_3)$ for each level, while the new one has parameters representing a mean $\beta_. = \frac{1}{3}(\beta_1 + \beta_2 + \beta_3)$, and the main effects $(\beta_2 - \beta_.)$, $(\beta_3 - \beta_.)$. Although apparently asymmetric, all main effects are estimated, since $(\beta_1 - \beta_.) = -(\beta_2 - \beta_.) - (\beta_3 - \beta_.)$. Each of these main effects has the same variance, but in contrast to the original parameters, they are correlated: the expected value of the correlation is 0.67 in this example.

Correlation among parameter estimates is common in experimental design, and can lead to misinterpretations, since the observed magnitude of an effect may be due to a substantial value for another parameter. The usual technique for dealing with this problem is to use an omnibus test for all parameters of interest and then to follow this up with multiple comparisons using contrasts among the parameters. That is to rely on ANOVA. The point is important especially in connection with response surface designs which can tempt the experimenter into the misinterpretation of individual parameters.

Another transformation that is frequently used is the orthogonal representation

$$T = \begin{pmatrix} 1 & -1 & -1 \\ 1 & 0 & 2 \\ 1 & 1 & -1 \end{pmatrix},$$

which produces uncorrelated estimates of the parameters $\beta_.$, $(\beta_3 - \beta_1)/2$, and $(\beta_2 - (\beta_1 +$

[1]The expected variance of a pairwise contrast is proportional to $\frac{1}{n+x} + \frac{1}{n-x} = \frac{2n^2}{n^2-x^2}$, where x is an integral increment or decrement. This is obviously minimized for $x = 0$.

$\beta_3)/2)/3$. These parameters are interpretable as linear and quadratic components if the levels of the variable are from a continuum such as time or temperature.

There are basically three types of variables in experimentation: (1) *categorical* variables, which assume discrete levels unrelated to each other, such as "old men," "young men," etc.; (2) *continuous* variables such as time or temperature; and (3) *random* variables such as "first sample," "second sample," etc. For the most part we will be concerned with categorical and continuous variables, but the distinction is not always clear. In the Peirce and Jastrow [1884] experiments, the levels were designated by the ratio of two weights (1.015, 1.030, and 1.060 in one experiment). These can be interpreted as three categories, "small," "medium," and "large." They could also be interpreted as continuous, and one might even prefer to take logarithms so that they represent an approximate doubling between steps. Some experimental designs assume that the variables are continuous, and some designs for categorical variables are seldom used for continuous variables, but in general, the distinction between these two types is not important as far as design construction is concerned.

For fixed N, minimizing the expected variance of pairwise contrasts leads to a design with equal sample sizes in each cell, but after transformation, optimization in terms of the parameters is no longer obvious, which serves to indicate that a better understanding of criteria is needed. At the moment it is useful to note that fixed linear transformations have no effect on the maximization of the determinant of M; and the maximization of the determinant, in this one-way example, leads to equal numbers of observations at each level regardless of the transformation chosen. The trace of M^{-1} is proportional to the average variance of the parameter estimates, and is also a criterion of interest. Its minimization is not invariant under linear transformation however, and the number of observations per level which minimize the trace can differ from transformation to transformation.

The original form of X was not particularly interesting, since the parameters represent expectations of cell means. When one observes several levels in for a factor, one naturally inquires about the differences between them, and the original parameters are not informative in this. The second form, however, with a grand mean and main effects as parameters is more informative, in that some of the most interesting contrasts are expressed as parameters. In times past this was more important than today, since in certain cases it saved computational labor. It is of little practical importance nowadays, since one may always compute any contrast of interest, together with its standard error, with modern computer software.

There are of course many other possible transformations. One could even write X as $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}$, which this has more parameters than data values, leading to multiple solutions. Except in very special cases, one may as well choose a non-singular transformation, as in the previous forms, since any function of the parameters is readily computed from such. For the most part, forms involving a grand mean are preferred. The general structure of X is then $X = [1, X_m]$, where the column rank of X_m is $k - 1$.

2.2 Higher way layouts

For two factors, one has a two way layout:

$y_{1,1}$	$y_{1,2}$	$y_{1,3}$
$y_{2,1}$	$y_{2,2}$	$y_{2,3}$
$y_{3,1}$	$y_{3,2}$	$y_{3,3}$

The basic parameterization of one parameter per cell is seldom used, rather, more informative parameterizations are common, such a main effect, interaction form such as

$$X = [1, X_{\text{meff}}, X_{\text{int}}] = \left(\begin{array}{c|cccc|cccc} 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & -1 & -1 & -1 & -1 & 0 & 0 \\ 1 & 0 & 1 & -1 & -1 & 0 & 0 & -1 & -1 \\ 1 & -1 & -1 & 1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & -1 & 0 & 1 & 0 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right),$$

where X_{int} is an interaction expansion of X_{meff} obtained by taking cross products. The main effect estimates for the first factor are $(y_{1.} - y_{..}, y_{2.} - y_{..}, y_{3.} - y_{..})$, where dots denote averages, and similar estimates for the other factor. An interaction is a product, a typical one is $(y_{2.} - y_{..})(y_{.3} - y_{..})$. All these are obtained by simple summation as indicated by the pattern in X above.

As before, the optimum allocation is equal numbers of observations in each cell of the two-way layout, which is also the allocation obtained by maximizing the determinant of M.

2.3 Factorial Designs

As Lorenzen and Anderson [1993] have shown, linear experimental designs may be expressed as factorial designs involving linear models which are variants of the fundamental model in equation (1): the usual enhancement is to include additional error terms.

When there are three factors, there will be columns in X representing three way interactions, and as the number of factors increase, so does the order of the maximum interaction. In practice, designs with several factors require substantial numbers of observations, and the question arrises as to their utility. One solution is to select rows from X which will enable the estimation of main effects and low order interactions. In doing this, information about the higher order interactions is confounded and they are no longer estimable. The complete layout is called a factorial layout, and the reduced layout obtained by selecting rows from X is called a fractional factorial. The idea was first addressed by Finney [1945].

Note that the first three rows for columns 2 and 3 is repeated in the next three rows and in the last three. Each of these repeats is associated with fixed values for the second factor, hence one can estimate the parameters for the first factor in each of the three repeats. Of course these estimates are conditional on the second factor; however, if it were possible to assume that the second factor had no effect, then one could use the estimates from the first repeat alone. One could even pool the estimates from the three repeats to obtain greater precision.

When there are several factors in an experiment, patterns enabling the estimation of low order effects will be found that are associated with fixed values of high order interactions, and the technique of fractional factorial design consists of selecting one of the several repeats associated with a fixed high order interaction. All this, of course, on the assumption that the high order interaction is negligible.

An easy illustration of this is the fractioning of a two-level experiment. Part of the pattern for four two level factors is

$$\begin{pmatrix}
1 & -1 & -1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 \\
1 & -1 & 1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 \\
1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1 & -1 \\
1 & -1 & -1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & -1 \\
1 & -1 & 1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 \\
1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1 & -1 & -1
\end{pmatrix},$$

where the two-level factor are coded -1 and 1, and only the mean, main effect and four-way interaction columns are shown. The rows have been rearranged according to the levels of the four-way interactions, and it is easy to see that if the experiment were divided in two according to the four-way levels, that each half would provide data for estimating the main effects of all four factors. Such a fraction is called a half fraction and denoted 2^{4-1}.

As before, the allocation that minimizes the variance of an estimate of pairwise contrasts is an equal allocation of observations to the cells, and this is also the allocation that maximizes the determinant of M for a fixed number of trials.

2.4 Blocked experiments

One of the first problems that arose in the early days of experimental design was in the practical allocation of experimental units. It often happens that the amount of material or time available is inadequate for all experimental trials. Many things happen from trial to trial which are unrelated to the treatment applied. For the most part these can be lumped together as "experimental error," but this becomes less tenable if the material or conditions of the experiment change. For example, Peirce and Jastrow [1884] found that their judgments improved as their experiments progressed, and they did not think it fair to directly compare results taken from a late series with those from an earlier one.

Early on, the idea of dividing the experiment into "blocks" arose. In this, one seeks to arrange the trials in such a way that the comparisons of interest may all be made in close proximity, so that disparities due to extraneous factors can be avoided. If it is possible to fraction an experiment so that each fraction fits into a homogeneous block, then one can estimate the parameters in each block and pool the estimates. To the first order, the differences between blocks will be captured by the constant terms which can be discarded. If it is not possible to fraction the experiment, then other methods must be used.

In general, for each block there is a constant column, so that $X = [I_b, \tilde{X}]$, where $\tilde{X}$ is the block centered[2] form of the expanded design matrix and I_b is a block diagonal matrix with columns of unities on the diagonal and zeros off-diagonal. Thus X for a blocked experiment has as many constant columns as there are blocks. Since the columns of $\tilde{X}$ and those of I_b are orthogonal, the least squares parameter estimates for the factors do not depend on the block parameters[3] when the number of blocks and their sizes are fixed.

The model for a blocked experiment is

$$\tilde{Y} = [I_b, \tilde{X}]\beta + \tilde{\epsilon}, \tag{2}$$

where $\tilde{Y}, \tilde{X}$ and $\tilde{\epsilon}$ correspond to values in equation (1) after centering by block means, and $\beta^T = (\beta_b^T, \beta_X^T)$. The uncentered observations have two error components, one within the block and one between the blocks. The centering cancels the between block error component. Maximizing the determinant of $\tilde{M} = \tilde{X}^T\tilde{X}$ will lead to the same design as minimizing the variances of pairwise differences within blocks.

In this formulation, it is assumed that the block parameters β_b, are nuisance parameters, of no particular interest. Indeed, they are often random, such as "oven load 1," "oven load 2," etc., and although one could treat them in the usual way and estimate variance components, the number of blocks involved is often so small that the estimates are hardly worth having.

An incomplete block experiment is one in which the trials are arranged so that contrasts of interest can be estimated within the several blocks. For example, suppose that one can complete only 3 trials per day, but that 7 treatments are to be compared. If 21 trials are arranged as in Table (1), then it may seen that each pair of treatments occurs together exactly once in the same block. Thus the differences between treatments may all be estimated without the error that might be caused by block effects.

[2]The block mean is subtracted from each value in the block.

[3]The least square parameter estimates are $(X^TX)^{-1}X^TY$, and $X^TX = \begin{pmatrix} \tilde{X}^T\tilde{X} & 0 \\ 0 & nI \end{pmatrix}$, when all blocks have n trials.

Table 1: A balanced incomplete block experiment

block 1	1	2	4
block 2	2	3	5
block 3	3	4	6
block 4	4	5	7
block 5	1	5	6
block 6	2	6	7
block 7	1	3	7

The X matrix for this experiment has 21 rows selected from a contrast matrix for a 7 level factor. Any 7×7 matrix of full column rank will do. The default contrast matrix built into **R** has a constant column followed by 6 mutually orthogonal columns:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The constant column is discarded, because each block has its own constant column. In this case, the design in Table (1) will be obtained when $\tilde{M}$ is maximized.

The blocks are not always nuisance parameters, and in some practical situations, such as split-plots, represent factors of interest. The model for this is $\tilde{Y} = \tilde{X}\beta + X_b\beta_b + Z\theta + \tilde{\epsilon}$, where X_b and β_b represent the whole plot factors, while Z and θ represent the random block components. This model is discussed in Appendix B.

There are other, more complicated blocking schemes involving blocks in a two dimensional layout, but these will not be discussed. Blocking of response surface, and factorial experiments will be discussed later.

2.5 Response surface experiments

2.5.1 The form of X

When the variables are continuous, it is natural to envision the experimental space as a multidimensional continuum – a multidimensional box or sphere. Predictions from the "model" may assume any value in the range of the response. Equation (1) still applies, but now $X = [1, F(x)]$, where $F(x) = \{f_1(x), \ldots, f_k(x)\}$ for vector valued functions $f_i(x), i = 1 \ldots k$, and x is a set of N design vectors $x = \{x_1, \ldots, x_N\}^T$, where $\{x_j = (x_{j,1}, \ldots x_{j,p})^T, j = 1 \ldots N\}$.

For example if $p = 2$, $N = 9$, and if both design variables range from -1 to 1, then x might look like

$$x = \begin{pmatrix} -1 & -1 \\ -1 & 0 \\ -1 & 1 \\ 0 & -1 \\ 0 & 0 \\ 0 & 1 \\ 1 & -1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

The $f_i()$ are terms in the model. Frequently, the $f_i()$ describe a polynomial, for example row j of $F(x)$ might be $\{x_{j,1}, x_{j,2}, x_{j,1}^2, x_{j,2}^2, x_{j,1}x_{j,2}\}$ with $k = 6$, and

$$F(x) = \begin{pmatrix} -1 & -1 & 1 & 1 & -1 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 1 & 1 & -1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

2.5.2 Design criteria

The design problem becomes one of selecting points in the multidimensional region that will "best" estimate some important function of the parameters. In this case, there is no obvious criterion as there was for one-way layouts. See Federov [1972] or Silvey [1980] for a discussion of this topic.

The one most generally used is the D criterion, which maximizes the determinant of $M = F(x)^T F(x)$ for a fixed number of design points. There are several rationales for using D. For example, a confidence ellipsoid for β is

$$(\hat{\beta} - \beta)^T M (\hat{\beta} - \beta) \leq \text{constant},$$

and since the volume of this ellipsoid is proportional to $|M|^{-\frac{1}{2}}$, maximizing M will make this volume as small as possible. Similarly, the numerator of the F-test when the errors are iid normal, is proportional to $\hat{\beta}^T M \hat{\beta}$, which by the same argument leads to the minimization of M for a fixed number of design points.

2.5.3 Design spaces

Although response surface problems are defined for continuous variables which can assume any real value in a region, only a subset of the points support the design. Any point not in this set of support points may be exchanged for a point in the set with an increase in the determinant. For quadratic models, this set of support points is the set of points of a three level factorial. In general, one can replace the continuous region with a set of discrete points, such as the points from a factorial with an appropriate number of levels. The actual support points are always in the neighborhood of the factorial points, and improvements due to their use are minor Donev and Atkinson [1988].

2.5.4 Goals

There are usually two goals of interest in a response surface experiment: parameter estimation, and prediction.

Although the parameters themselves are of interest, the fact that the model is an approximation, makes them less interesting than in other forms of experimentation, where the factors represent important attributes. Polynomial models act as mathematical French curves to graduate the response surface, and within a given region will usually mimic the actual underlying functionality, but by themselves have no meaning; and clearly are false outside the region. It is an error to think of them as Taylor series approximations, which is a siren that often tempts those new to the field. One can parcel out the terms in a polynomial model as if they were factors and interactions and perform ANOVA. For most response surface designs, the ANOVA sums of squares will not be independent, but they still provide likelihood ratio tests for the individual sources, and looking at such is a better practice than attempting to interpret the individual terms.

Prediction is a related, but independent goal. For approximate theory, where fractional points are allowed, the general equivalence theorem Atkinson and Donev [1992] says that a D-optimal design is also a G-optimal design, where G is the criterion that minimizes the maximum variance in the experimental region. Thus, for approximate theory, the maximum prediction variance will be minimized by maximizing[4] $|M|$. For exact theory, where one has a discrete set of N points in the design, the two criteria are not equivalent, although G can be used to bound D.

A criterion that deals specifically with prediction is the I criterion, which is defined as the average prediction variance in the region. In spite of the apparent difference between the two criteria, the points chosen by the D and I criterion are similar. The I criterion, by and large, tends to select its points from the support points for the D criterion.

2.5.5 Mixture experiments

Since the variables are continuous, they may represent mixtures, as for example, mixtures of components in a paint. In these, the variables are constrained to sum to a constant, usually unity. An example of a design for three mixture components is shown in Table (2).

Because of the constraint, ordinary polynomial models will have redundant terms.

[4]In approximate theory, the fractional weights add to unity, thus ensuring a maximum

Table 2: A mixture experiment

1.0	0.0	0.0
0.0	1.0	0.0
0.0	0.0	1.0
0.5	0.5	0.0
0.5	0.0	0.5
0.0	0.5	0.5

This may be dealt with by appending non-estimable constraints to the design, or by reformulating polynomial models to account for the constraint. The constraint method may be useful in the analysis in those cases where it is desired to interpret the coefficients. Cox [1971] treats this problem. For design, there is no advantage in carrying redundant terms through the calculation, and so reformulated models is appropriate. These have been given by H. [1958], and elaborated upon by Gorman and Hinman [1962]. The Scheffé models for three variables are shown in Table (3). Note that the constant term is omitted from these models, which among other things, means that they are unaffected by block effects.

Table 3: Scheffé models

linear	$X_1 + X_2 + X_3$
quadratic	$X_1 + X_2 + X_3 + X_1X_2 + X_1X_3 + X_2X_3$
cubic	$X_1 + X_2 + X_3 + X_1X_2 + X_1X_2 + X_2X_3 +$ $X_1X_2(X_1 - X_2) + X_1X_3(X_1 - X_3) + X_2X_3(X_2 - X_3)$

2.5.6 Blocking response surface experiments

Blocking is often required when using response surface designs which require too many trials for a single physical test. The division into blocks is seldom symmetrical, and instead of seeking to balance pairs in the blocks, one seeks to obtain parameter estimates orthogonal to block effects. In general, the D criterion is useful; however, when the blocks are of a size to allow estimation of the parameters within each block, the resulting designs may be D-optimal in toto, but are not necessarily D-optimum within blocks. An auxiliary criterion, D_p is then useful, which attempts to maximize the product of the determinants of the individual blocks. Table (4) shows the cross product matrix for one block of a 2^4, blocked into two 8 run blocks by using the D criterion. The model is linear.

Table 4: Cross product of an 8 run block of a 2^4

8	0	-4	0
0	8	0	0
-4	0	8	0
0	0	0	8

Using the D_p criterion, the crossproduct matrix becomes diagonal, and the block is shown in Table (5). Note that this is not the usual fractioning of a 2^{4-1} in that no higher order interaction is used to divide the trials. The usual fractioning results in interaction columns orthogonal to main effect columns. A D_p design will not usually have this property.

Table 5: One block from a blocked 2^4

1	-1	-1	-1
-1	1	-1	-1
-1	-1	1	-1
1	-1	1	-1
-1	1	1	-1
1	1	1	-1
-1	-1	1	1
1	1	1	1

3 Confounding

Confounding is a very important criterion: it is a dominant factor in the structure of established designs. The fact that a parameter estimate is significant, is always tempered by the degree to which the parameter may be confounded with other parameters in the design. Builders of established designs have taken pains to eliminate confounding; but this often has the side effect of producing oversize designs.

A measure of confounding it the degree of diagonality of a design. Since positive definite information matrices, M, have weighty diagonals[5], this may be measured by comparing $|M|$ with the product of the diagonal elements of M. Diagonality may thus be defined as $O = [|M|/\prod \text{diag}(M)]^{1/k}$, where k is the number of columns of M.

It is important to note that a design which maximized $|M|$ also seems to maximize O, which means that the confounding is minimized by designs that maximize $|M|$.

4 Examples using R

4.1 Approximate theory designs

Abstraction and simplification are often useful in understanding difficult problems, since the simpler structure often leeds to insights into the original problem. This is the case in experimental design, where the X in the basic model, $Y = X\beta + \epsilon$, can assume many forms, and almost always leads to combinatorial problems of one sort or another. The useful simplification, due I believe to Elfving [1952], is to write the observational weights as $p = 1/N$, where N is the number of observations, and then to substitute a probability. Thus one might write the matrix $M = X^T X/N = X^T PX$, where P is a diagonal matrix,

[5]The 2×2 principal minors are positive

all of whose diagonal values are p; and then to allow the elements of P to assume any real values, so long as the trace of P is unity. This is the germ of approximate theory, which by an large replaces consideration of special combinatorial cases with a unified analytical approach.

4.1.1 Support Points

One of the interesting results that flow from this simplification is that only a subset of the possible points in an experimental region are support points for an experimental design. Consider a simple quadratic model in three variables, and assume that the variables can assume seven levels $(-3, -2, -1, 0, 1, 2, 3)$. The candidate set has 3^7 possible points, and running optFedrov() with a quadratic model in the following fashion will produce a list of the support points for the D criterion:

```
dat<-gen.factorial(levels=7,nVars=3,center=TRUE,varNames=c("A","B","C"))
desD<-optFederov(~quad(.),dat,approximate=TRUE)
desD$design[c(1:5,23:27),]
    Proportion  A  B  C
1        0.069 -3 -3 -3
4        0.029  0 -3 -3
7        0.072  3 -3 -3
22       0.022 -3  0 -3
25       0.017  0  0 -3
...
319      0.024  0  0  3
322      0.020  3  0  3
337      0.071 -3  3  3
340      0.024  0  3  3
343      0.075  3  3  3
```

There are 27 support points in all, and each assumes only the three values $(-3, 0, 3)$. The support points for a quadratic model in fact correspond to the points of a 3^m factorial, where m is the number of variables. The support points for other criteria are different. For example, the support points for the I criterion in this example are as below, where it may be seen that points not on the 3^m grid are obtained.

```
desI<-optFederov(~quad(.),dat,approximate=TRUE,criterion="I")
desI$design[c(1:5,27:31),]
    Proportion  A  B  C
1        0.042 -3 -3 -3
4        0.030  0 -3 -3
7        0.043  3 -3 -3
22       0.034 -3  0 -3
25       0.021  0  0 -3
...
322      0.028  3  0  3
337      0.038 -3  3  3
```

```
340        0.035  0  3  3
341        0.002  1  3  3
343        0.040  3  3  3
```

This result indicates that experimental regions are not quite what they appear to be with respect to experimental designs. Even though one thinks of a variable as continuous between some limits, the points in this region are not all candidates for inclusion in an optimal experimental design. For example the support points for a quadratic polynomial on the real line are the two extreme points and the mid point. No other points are involved.

Even after accepting this fact, there are still surprises. The support points on the real line between 1 and 2 may be obtained as follows. These may be shown to be the support points on the continuous interval between 1 and 2.

```
desA<-optFederov(~quad(.),data.frame(A=1+((0:100)/100)),approximate=TRUE)
desA$design
    Proportion   A
1        0.333 1.0
51       0.333 1.5
101      0.333 2.0
```

The slight change caused by running the interval from 1.01 to 2 produces, the following, in which the proportions are quite different from the previous ones, although there still remain only three support points.

```
desB<-optFederov(~quad(.),data.frame(A=1+((1:100)/100)),approximate=TRUE)
desB$design
    Proportion    A
1        0.485 1.01
50       0.029 1.50
100      0.485 2.00
```

The difference between these examples is due to the fact that the precise midpoint is not included in the second set of candidate points. When it is, the optimum design agrees with the optimum design on the continuous interval, as the following shows:

```
desC<-optFederov(~quad(.),data.frame(A=c(1+((1:100)/100),1.505)),approximate=TRUE)
desC$design
    Proportion     A
1        0.333 1.010
100      0.333 2.000
101      0.333 1.505
```

The basic conclusion from these examples is that an optimal design is a function of the set of candidate points, and if these candidate points fail to include the support points of the underlying continuum, the design will differ from the optimal design on the continuum.

Atkinson and Donev [1992] give a useful table of support points and their weights for quadratic models in cubic regions on page 130.

4.1.2 Rounding Approximate Designs

The sample size for an experimental design must be integral, and thus the question of rounding arises. The proportions in the first example in the previous section may be rounded by specifying the `nTrials` parameter, as follows:

```
dat<-gen.factorial(levels=7,nVars=3,center=TRUE,varNames=c("A","B","C"))
desDR<-optFederov(~quad(.),dat,approximate=TRUE,nTrials=20)
desDR$design
    Rep..  A  B  C
1       1 -3 -3 -3
4       1  0 -3 -3
7       1  3 -3 -3
22      1 -3  0 -3
28      1  3  0 -3
43      1 -3  3 -3
49      1  3  3 -3
148     1 -3 -3  0
154     1  3 -3  0
175     1  3  0  0
190     1 -3  3  0
193     1  0  3  0
295     1 -3 -3  3
298     1  0 -3  3
301     1  3 -3  3
316     1 -3  0  3
319     1  0  0  3
337     1 -3  3  3
340     1  0  3  3
343     1  3  3  3
```

The unrounded design had 30 support points, but the rounding has discarded ten of them to produce a 20 run design. Had 40 trials been specified, all 30 support points would have been included and the result would have been:

```
desDR2<-optFederov(~quad(.),dat,approximate=TRUE,nTrials=40)
desDR2$design[c(1:5,23:27),]
    Rep..  A  B  C
1       2 -3 -3 -3
4       1  0 -3 -3
7       2  3 -3 -3
22      1 -3  0 -3
25      1  0  0 -3
...
```

```
319      1  0  0  3
322      1  3  0  3
337      2 -3  3  3
340      1  0  3  3
343      2  3  3  3
```

The rounding is done with an algorithm for efficient rounding of experimental designs by Pukelsheim and Rieder [1992] which produces the smallest loss in efficiency for several criteria. The `efficient.rounding()` function is included as part of the package.

4.2 Exact Designs

Away from the world of theory, experimental designs are composed mostly of unique points. Some replication is done to aid in the estimation of error, but for the most part, practical constraints dominate and limit the number of points available. The rounding of approximate theory designs has never been very satisfactory, because such designs can usually be improved upon when the sample size is fixed. In practice, then, one has to deal with the combinatorial problem; however, insights from approximate theory are very helpful.

The most successful algorithm for dealing with exact designs is due to Federov [1972]. It starts with a non-singular design matrix, and sequentially exchanges points until a local optima is found. Since local optimas abound, the process is usually repeated a number of times and the best design reported.

With the exception of Ge, the criteria efficiencies are not known for exact designs, and it is therefore not possible to tell from their values whether or not a design is globally optimal. The efficiencey Ge, is defined relative to the opimum approximate theory design and is thus a useful guide when judging an exact design. In particular, a Ge of unity not only indicates that the exact design is optimum, but that the optimum weights are all equal.

4.2.1 Classical Designs

Classical designs can be produced algorithmically, since they are almost invariably optimal designs. It is simply a matter of choosing the correct number of trials and repeating the algorithmic calculations until the global optimum is found. This section illustrates this. The advantage of algorithmic design lies, however, in its ability to find optimal or near optimal designs for situations in which classical designs do not or can not exist.

4.2.1.1 Fractional Factorial The following design is a one third fraction of a 3^3. The confounding definition is AB^2C^2.

```
dat<-gen.factorial(levels=3,nVars=3,varNames=c("A","B","C"))
desT<-optFederov(~.,dat,nTrials=9)
desT$design
```

```
   A B C
2  2 1 1
6  3 2 1
7  1 3 1
12 3 1 2
13 1 2 2
17 2 3 2
19 1 1 3
23 2 2 3
27 3 3 3
```

4.2.1.2 Orthogonal design An orthogonal design, similar to a 12 run Plackett-Burman design can be produced by the following. If you try this, you may need to set `nRepeats` to more than 20 to get an optimum design, which is marked by a Ge of unity.

```
dat<-gen.factorial(levels=2,nVars=11,center=TRUE)
desPB<-optFederov(~.,dat,12,nRepeats=20)
desPB$design
$design
     X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11
180   1  1 -1 -1  1  1 -1  1 -1  -1  -1
317  -1 -1  1  1  1  1 -1 -1  1  -1  -1
332   1  1 -1  1 -1 -1  1 -1  1  -1  -1
609  -1 -1 -1 -1 -1  1  1 -1 -1   1  -1
734   1 -1  1  1  1 -1  1  1 -1   1  -1
903  -1  1  1 -1 -1 -1 -1  1  1   1  -1
1111 -1  1  1 -1  1 -1  1 -1 -1  -1   1
1161 -1 -1 -1  1 -1 -1 -1  1 -1  -1   1
1510  1 -1  1 -1 -1  1  1  1  1  -1   1
1584  1  1  1  1 -1  1 -1 -1 -1   1   1
1810  1 -1 -1 -1  1 -1 -1 -1  1   1   1
2043 -1  1 -1  1  1  1  1  1  1   1   1
```

4.2.1.3 Central composite A central composite design in three variables may be obtained. This central composite is an optimal exact design. The D and G efficiencies of the following, relative to the optimum approximate theory design, are 98% and 89%.

```
dat<-gen.factorial(3,3,center=TRUE,varNames=c("A","B","C"))
desC<-optFederov(~quad(A,B,C),dat,nTrials=14,evaluateI=TRUE,nR=100)
desC$design
    A  B  C
1  -1 -1 -1
3   1 -1 -1
5   0  0 -1
7  -1  1 -1
```

```
9   1  1 -1
11  0 -1  0
13 -1  0  0
15  1  0  0
17  0  1  0
19 -1 -1  1
21  1 -1  1
23  0  0  1
25 -1  1  1
27  1  1  1
```

4.2.1.4 Latin square One can even find Latin squares; although this is more difficult, which is why it is repeated 1000 times, and that may not be enough if you were to try it.

```
dat<-gen.factorial(5,3)
desL<-optFederov(~.,dat,nTrials=25,nRepeats=1000)
cs<-xtabs(~.,desL$design)
{xx<-matrix(0,5,5); for (i in 1:5) xx=xx+cs[1:5,1:5,i]*i;xx}
   X2
X1   1 2 3 4 5
   1 1 3 4 5 2
   2 5 1 3 2 4
   3 4 2 1 3 5
   4 3 5 2 4 1
   5 2 4 5 1 3
```

The reason the above designs appear, is because most classic designs are D-optimal, and indeed the above designs are D-optimal.

4.2.2 Factorials

According to Lorenzen and Anderson [1993], all linear designs are factorial, and the various names such as Latin square, split-plot, incomplete block, etc. are due to historical processes and not to anything essentially different in their construction or analysis. There is considerable justification for this viewpoint, and it does eliminate the need to treat a host of special cases which in essence differ mostly in nomenclature. So let's consider a few factorial designs.

4.2.2.1 Two level Designs Two levels designs are very useful. There are statistical consultants that except for very occasional forays, use nothing else. Such designs certainly make possible the screening of many variables, as the 12 run orthogonal design, in the previous section, illustrates. A problem with factorial designs in general is that the number of experimental trials increases exponentially with the number of variables, while the number of terms of interest increases linearally. Thus the need to fraction such designs, but even this soon runs into difficulties.

Consider a 2^7 design. A full factorial requires 128 trials, while a half fraction 2^{7-1} requires 64. In both, all two-factor interactions are estimable; however, there are only $1+\binom{7}{2}$ such terms. The remaining 35 degrees of freedom are allocated to error. Surely this is wasteful, but standard methods of fractioning offer no solution, since a quarter fraction leaves some of the two-factor interactions unestimable. Instead, consider the following design in 34 trials:

```
dat<-gen.factorial(2,7,center=TRUE)
desF<-optFederov(~.^2,dat,nTrials=34,nRepeats=100)
desF
$D
[1] 0.9223281

$A
[1] 1.181451

$Ge
[1] 0.675

$Dea
[1] 0.617

$design
    X1 X2 X3 X4 X5 X6 X7
2    1 -1 -1 -1 -1 -1 -1
5   -1 -1  1 -1 -1 -1 -1
8    1  1  1 -1 -1 -1 -1
11  -1  1 -1  1 -1 -1 -1
19  -1  1 -1 -1  1 -1 -1
...
112  1  1  1  1 -1  1  1
113 -1 -1 -1 -1  1  1  1
116  1  1 -1 -1  1  1  1
119 -1  1  1 -1  1  1  1
126  1 -1  1  1  1  1  1
```

The design is not optimal, but nearly so, since the D value for the optimal design is unity, making this one 92% efficient. Moreover the confounding is minimal as indicated by the diagonality of 0.92 and the confounding matrix, shown below. The columns of the confounding matrix give the regression coefficeints of a variable regressed on the other variables. If X is a design matrix and C a confounding matrix $(-XC)$ is a matrix of residuals of each variable regressed on the other variables.

```
eval.design(~.^2,desF$design,confounding=TRUE)
```

```
$confounding

               [,1]    [,2]    [,3]  ...   [,27]   [,28]   [,29]
(Intercept) -1.0000  0.0000  0.0000  ...  0.0000 -0.0203  0.0000
x1           0.0000 -1.0000 -0.0323  ... -0.0323  0.0000  0.0323
x2           0.0000 -0.0323 -1.0000  ...  0.0323  0.0000 -0.0323
...            ...     ...     ...   ...    ...     ...     ...
x5:x6        0.0000 -0.0323  0.0323  ... -1.0000  0.0000 -0.0323
x5:x7       -0.0278  0.0000  0.0000  ...  0.0000 -1.0000  0.0000
x6:x7        0.0000  0.0323 -0.0323  ... -0.0323  0.0000 -1.0000

$determinant
[1] 0.9223281

$A
[1] 1.181451

$diagonality
[1] 0.92

$gmean.variances
[1] 1.179251
```

This design has 5 degrees of freedom for estimating error, which some may consider too small. If so, it is easy to create a larger design. However, the power function changes rapidly and the gains from larger error degrees of freedom are smaller than many suppose. Most of the power of a design comes not from a large error degrees of freedom, but from the averaging that occurs in the coefficient estimation; due to the fact that the variance of a coefficient is proportional to $1/N$, where N is the number of trials in the design.

4.2.2.2 Mixed level designs Mixed level factorial designs can be created in the same way with savings in the number of trials. For example a one half fraction of a $3^2 2^4$ requires 72 trials to estimate 35 terms. The following design does it in 40. Note: It is prefereable to switch to orthogonal contrasts when judgments about the orthogonality of designs are of interest.

```
options(contrasts=c("contr.sum","contr.poly"))

dat<-gen.factorial(c(3,3,2,2,2,2),factors=1:2)
desFM<-optFederov(~.^2,dat,nTrials=40)
desFM[1:4]
$D
[1] 0.5782264

$A
[1] 2.397925
```

```
$Ge
[1] 0.436

$Dea
[1] 0.274

eval.design(~.^2,desFM$design,confounding=TRUE)
$confounding

                [,1]    [,2]    [,3]  ...   [,33]   [,34]  [,35]
(Intercept) -1.0000 -0.0346  0.0009  ...  0.0177  0.0328  0.0247
X11         -0.0771 -1.0000  0.5154  ...  0.2885 -0.1956 -0.0467
X12          0.0021  0.5364 -1.0000  ... -0.1327  0.3419 -0.0075
...             ...     ...     ...  ...    ...     ...     ...
X4:X5        0.0186  0.1364 -0.0603  ... -1.0000  0.0828  0.0528
X4:X6        0.0412 -0.1102  0.1850  ...  0.0986 -1.0000 -0.0185
X5:X6        0.0252 -0.0214 -0.0033  ...  0.0511 -0.0150 -1.0000

$determinant
[1] 0.5782264

$A
[1] 2.397925

$diagonality
[1] 0.801

$gmean.variances
[1] 2.223677
```

4.2.3 Response surface designs

These designs treat the model as a mathematical French curve for graduating the response, which is visualized as a surface in multidimensional space. The model is invariably a low order polynomial, usually quadratic. The model has no validity outside the experimental region as decribed by the data, and only in exceptional cases can it be assumed to represent some underlying function. The variables are continuous, although it is reasonable to incorporate categorical variables into the model to allow for baseline changes.

For such designs, the orthogonality or lack of orthogonality of the design is of small importance; rather the prediction ability is paramount. However, designs which have good prediciton ability also have good parameter estimation ability, which leads to designs that tend to orthogonality. For approximate theory designs, the fundamental theorem says that the two problems are equivalent, in that a design with minimizes the maximum prediction variance in a region also maximize the determinant of the information matrix: thus the optimun prediction and parameter estimation designs are the same.

4.2.3.1 Three variables in a cube

In this case the experimental region is a cube. For the D criterion, the support points are the corners and centers of the cube, as the following shows:

```
dat<-gen.factorial(7,3)
desCD=optFederov(~quad(.),dat,approximate=TRUE)
dim(desCD$design)
[1] 27  4
 apply(desCD$design[,-1],2,table)
   x1 x2 x3
-3  9  9  9
 0  9  9  9
 3  9  9  9
```

For the A criterion, there are 66 support points using all 5 levels of the variables. There is some assymetry in the design.

```
desCA<-optFederov(~quad(.),dat,approximate=TRUE,criterion="A")
dim(desCA$design)
[1] 66  4
apply(desCA$design[,-1],2,table)
   x1 x2 x3
-3 19 19 19
-1  7  7  7
 0 12 15 15
 1  8  5  7
 3 20 20 18
```

For the I criterion, there are 31 support points. There is considerable assymetry due to the discreteness of the candidate set. The I criterion is a measure most appropriatly visualized with respect to a continuous region. In such a region, the I optimal design is symmetric and concentrated on a 5^3 grid.

```
desCI<-optFederov(~quad(.),dat,approx=T,criterion="I")
dim(desCI$design)
[1] 31  4
apply(desCI$design[,-1],2,table)
$x1

-3 -1  0  1  3
 9  1  9  2 10

$x2
```

```
-3  0  3
 9 11 11

$x3

-3  0  1  3
10  9  1 11
```

Exact designs for three variables in a cube select points from a grid, which can be a 3^3 for D, but should be a 5^3 for A and I. The quadratic model has 10 terms, so 15 trial designs seem appropriate.

An exact design for D may be obtained as follows. (Using the 5^3 grid so that D and I statistics may be compared.) There is little to choose between these designs, except that the I design requires more levels.

```
dat<-gen.factorial(5,3)
desDE<-optFederov(~quad(.),dat,nTrials=15,evaluateI=TRUE)
desDE[1:5]
$D
[1] 3.675919

$A
[1] 1.255597

$I
[1] 8.848874

$Ge
[1] 0.775

$Dea
[1] 0.749

The optimum approximate theory D is 3.8, making this design about 97\% efficient.

eval.design(~quad(.),desDE$design)
$determinant
[1] 3.675919

$A
[1] 1.255597

$diagonality
[1] 0.755

$gmean.variances
```

```
[1] 0.2324225

apply(desDE$design,2,table)
   X1 X2 X3
-2  6  6  5
 0  3  4  3
 2  6  5  7
```

And one for I as follows:

```
> desIE<-optFederov(~quad(.),dat,nTrials=15,criterion="I")
> desIE[1:5]
$D
[1] 3.485428

$A
[1] 0.9161151

$I
[1] 8.096772

$Ge
[1] 0.582

$Dea
[1] 0.488

The optimum approximate theory I is 7.57, making this design about 93\% efficient.

> eval.design(~quad(.),desIE$design)
$determinant
[1] 3.485428

$A
[1] 0.9161151

$diagonality
[1] 0.809

$gmean.variances
[1] 0.2452361

> apply(desIE$design,2,table)
$X1

-2 -1  0  2
 5  1  4  5
```

```
$X2

-2  0  2
 5  5  5

$X3

-2  0  1  2
 5  4  1  5
```

4.2.3.2 Design augmentation Designs sometimes break because certain variable combinations are not taken, and sometimes the experimenter desires to include certain points in the design: both requre design agumentation. A boken design usually requires the region to be constrained: this is discussed in section (4.2.3.5).

Suppose one wants to include the followng trials in a design:

```
myData<-data.frame(X1=c(0.5,-0.5,-1.0),X2=c(-.05,0.5,-1.0),X3=c(1.5,-0.5,0.5))
myData
       X1   X2   X3
[1,]  0.5 -0.5  1.5
[2,] -0.5  0.5 -0.5
[3,] -1.0 -1.0  0.5
```

Simply add these to the candidate list, and run optFederov() as follows. This may be compared with the unaugmented design on page 26.

```
dat<-rbind(myData,gen.factorial(5,3))
desAG<-optFederov(~quad(.),dat,nTrials=15,rows=1:3,augment=TRUE)
desAG
$D
[1] 3.40889

$A
[1] 0.9248038

$Ge
[1] 0.564

$Dea
[1] 0.462

$design
      X1    X2   X3
1    0.5 -0.05  1.5
2   -0.5  0.50 -0.5
```

```
3   -1.0 -1.00  0.5
4   -2.0 -2.00 -2.0
7    1.0 -2.00 -2.0
18   2.0  0.00 -2.0
24  -2.0  2.00 -2.0
26   0.0  2.00 -2.0
28   2.0  2.00 -2.0
58   2.0 -2.00  0.0
78   2.0  2.00  0.0
104 -2.0 -2.00  2.0
108  2.0 -2.00  2.0
124 -2.0  2.00  2.0
128  2.0  2.00  2.0
```

4.2.3.3 Mixture designs Mixture variables sum to a constant, usually unity. This constraint confines the variables to a multidimensional simplex, and requires an adjustment in the model to accommodate the constraint. The following illustrates a mixture calculation for a quadratic model: note that the constant term is deleted from the model in order to remove the singularity caused by the mixture constraint. In general the confounding is substantial between linear and interaction terms.

```
dat<-expand.mixture(4,3)
desMix<-optFederov(~-1+.^2,dat,nTrials=8)
desMix
$D
[1] 0.03623366

$A
[1] 98.34085

$Ge
[1] 0.62

$Dea
[1] 0.541

$design
         X1        X2        X3
1  1.0000000 0.0000000 0.0000000
2  0.6666667 0.3333333 0.0000000
4  0.0000000 1.0000000 0.0000000
5  0.6666667 0.0000000 0.3333333
6  0.0000000 0.6666667 0.3333333
8  0.3333333 0.0000000 0.6666667
9  0.0000000 0.3333333 0.6666667
```

```
10 0.0000000 0.0000000 1.0000000

eval.design(~-1+.^2,desMix$design,confounding=TRUE)
$confounding

           [,1]    [,2]    [,3]    [,4]    [,5]    [,6]
  X1    -1.0000 -0.0026 -0.0526  0.0922  0.1122  0.0056
  X2    -0.0026 -1.0000 -0.0526  0.0463  0.0056  0.1122
  X3    -0.0501 -0.0501 -1.0000  0.0069  0.1072  0.1072
  X1:X2  3.0040  1.5079  0.2368 -1.0000 -0.3452 -0.1853
  X1:X3  2.3628  0.1187  2.3684 -0.2230 -1.0000 -0.2538
  X2:X3  0.1187  2.3628  2.3684 -0.1197 -0.2538 -1.0000

$determinant
[1] 0.03623366

$A
[1] 98.34085

$diagonality
[1] 0.748

$gmean.variances
[1] 37.19754
```

4.2.3.4 Large problems The implementation of algorithmic design in optFederov() is somewhat wasteful of memory. Nonetheless, the program can deal with substantial problems. Up to 11 variables in a quadratic model on my machine. A more careful structuring could no doubt push this up to 12 or 13 variables, but larger problems are not really feasible with a straightforward application of Federov's algorithms. To deal with larger problmes, `optMonteCarlo()` calls `optFederov()` with a reduced candidate list obtained by randomly sampling from the full candidate list. The full candidate list is never created, only the necessary trials are generated. The result, is in general, quite satisfactory.

As an example, the following may be compared with the design on page 26.

```
dat<-data.frame(var=paste("X",1:3,sep=""),low=-2,high=2,
     center=0,nLevels=5,round=1,factor=FALSE
dat
  var low high center nLevels round factor
1  X1  -2    2      0       5     1  FALSE
2  X2  -2    2      0       5     1  FALSE
3  X3  -2    2      0       5     1  FALSE

desMC<-optMonteCarlo(~quad(.),dat)
desMC
```

```
$D
[1] 3.192013

$A
[1] 1.173419

$Ge
[1] 0.741

$Dea
[1] 0.705

$design
   X1 X2 X3
1   0  0  0
2   0  0  2
3  -1 -2  0
4  -2 -2  2
5   2  2  0
6  -2  2  1
7   0 -2 -2
8   2  2  2
9   2  2 -2
10 -2 -1 -2
11 -1  2 -2
12  1 -2  2
13  2  0 -2
14  2 -2  0
15 -2  1 -2

> eval.design(~quad(.),desMC$design,confounding=TRUE)
$confounding

                  [,1]    [,2]    [,3]    [,4]  ...
  (Intercept) -1.0000  0.0557 -0.6953 -0.1220  ...
  X1           0.0028 -1.0000  0.1509  0.0301  ...
  X2          -0.0332  0.1455 -1.0000 -0.1231  ...
  X3          -0.0057  0.0283 -0.1198 -1.0000  ...
  I(X1^2)      0.0924 -0.0072  0.2939 -0.0973  ...
  I(X2^2)      0.1189  0.0835 -0.0418  0.1719  ...
  I(X3^2)      0.1204 -0.0712 -0.0099 -0.1240  ...
  X1:X2       -0.0237  0.0932  0.0385  0.0673  ...
  X1:X3       -0.0004 -0.1114  0.0912  0.0151  ...
  X2:X3        0.0181  0.0662 -0.0341 -0.0039  ...

$determinant
[1] 3.192013
```

```
$A
[1] 1.173419

$diagonality
[1] 0.78

$gmean.variances
[1] 0.2981729
```

Now for a really big design, a 20 variable quadratic. This takes some time to calculate. It's G efficiency is low, but this is typical of large designs. The diagonality could be better. The determinant of the optimal approximate design is 0.337, thus this design is about 53% efficient, which is acceptable given the difficulty of the problem. Increasing `nRepeats` may produce a better design.

```
dat<-data.frame(var=paste("X",1:20,sep=""),low=-1,high=1,
     center=0,nLevels=3,round=0,factor=FALSE)
dat
desBig[1:4]
$D
[1] 0.1785814

$A
[1] 27.70869

$Ge
[1] 0.046

$Dea
[1] 0

eval.design(~quad(.),desBig$design)
$determinant
[1] 0.1785814

$A
[1] 27.70869

$diagonality
[1] 0.455

$gmean.variances
[1] 24.51871
```

As a final point in this section, let me call your attention to the parameter *nCand* which controls the size of the randomly selected candidate list. It is by default set to

100 times the number of terms in the model. This may not be large enough for difficult problems, so try changing if the designs being produced are not satisfactory.

4.2.3.5 Constrained regions In practice most response surfaces are defined over cubical regions, even though quite a bit of optimal design theory is devoted to spherical regions. The limits of the variables are usually difficult enough to specify without the additonal complication of dealing with spheres; however, it is common to have inadmissible corners in which the mechanism underlying the process changes. One way to handle this is to edit the candidate list, deleting inadmissible points. A better way is to construct a boundary and allocate candidate points on this boundary. An efficient algorithm for doing this exists, but has not been implemented as yet. In the meantime, `optMonteCarlo()` will accept a constraint function that may be used to exclude points. The best way to use it is to create a dense gird of candidate points so that the boundary will be roughly marked by non-eliminated points near it. The following example cuts off the upper half of a cubical region.

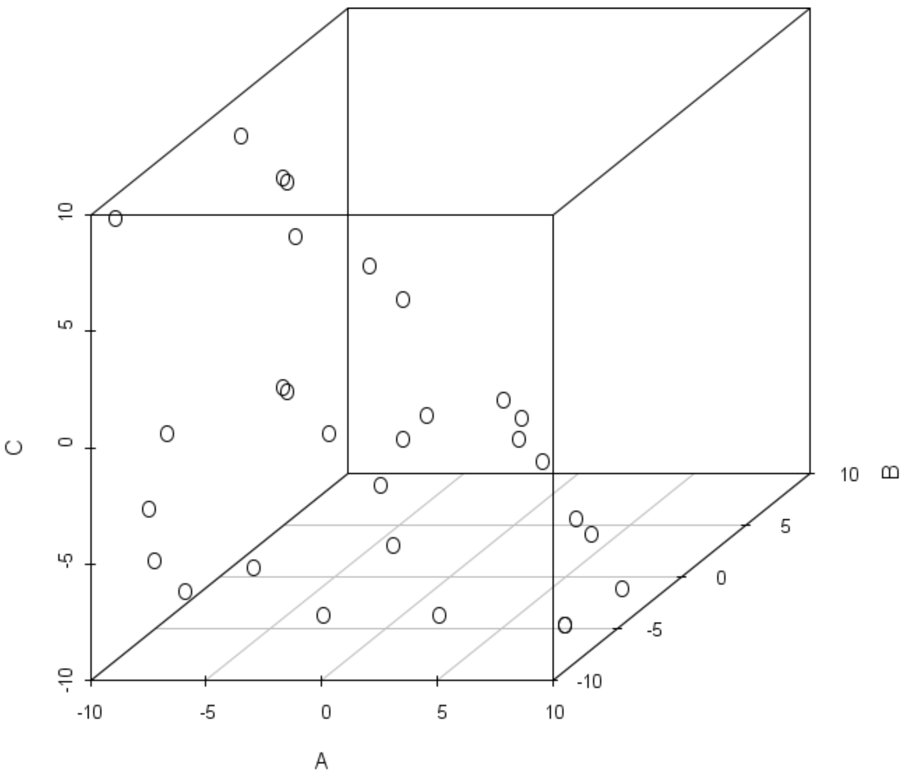
```
dat<-data.frame(vars=c("A","B","C"),low=-10,
     high=10,center=0,nLevels=21,round=1,factor=FALSE)
dat
  vars low high center nLevels round factor
1    A -10   10      0      21     1  FALSE
2    B -10   10      0      21     1  FALSE
3    C -10   10      0      21     1  FALSE

constFcn<-function(x){if (sum(x)<=0) return(TRUE);return(FALSE);}
desCon<-optMonteCarlo(~quad(.),dat,constraint=constFcn,nTrials=15)
desCon
$D
[1] 154.4033
$A
[1] 0.9057725
$Ge
[1] 0.456
$Dea
[1] 0.303
```

The design is no longer concentrated on the support points of the cube. A plot of the resulting design is shown in Figure (1). It is useful to note that `optMonteCarlo()` tends to underpopulate the corners of experimental regions because the probability of random points falling therein is low.

4.2.4 Starting designs

There are two ways to start the design calculation: at random or by using nullification. Nullification is essentially a Gram-Schmidt orthogonalization of the cadidate list, with at each step picks the longest vector in the null space. Randomization works well in most

Figure 1: Plot of a constrained design.



cases, but on occasion will be unable to find a starting design. Mixture problems are frequently of this nature. For example, the following will frequently produce singular designs for the default number of random starts,

```
dat<-gen.mixture(4,5)
optFederov(~-1+.^2,dat)
```

but one can be assured of obtaining a design with

```
optFederov(~-1+.^2,dat,nullify=TRUE)
```

or with

```
optFederov(~-1+.^2,dat,nullify=2)
```

where the second version introduces some randomness in the process. After a starting design is found with the number of trials equal to the number of terms, additonal trials will be selected at random to reach the value of `nTerms` selected.

4.3 Blocked designs

The blocking of expermental designs has always been important, and most tables of classical designs indicate ways in which this may be done. AlgDesign implements several types of blocking: (A) It will block an existing design, or construct a blocked design from a candidate set. (B) It will also block a design in which whole plot variables interact with within plot variables, or (C) it will block designs with multiple levels of blocking. Cook and Nachtsheim [1989] and Atkinson and Donev [1992] have developed methods for (A). Goos and Vandebroek [2003] and Trinca and Gilmour [2000] have investigated (B), and Trinca and Gilmour [2001] have shown how to do (C). The methodologies used in AlgDesign are different from these, although they make use of a fundamential idea from Cook and Nachsheim (*loc.cit.*). The methodologies are detailed in Appendix B.

4.3.1 Balanced and partially balanced designs

One can create a balanced incomplete block design for 7 treatments in 7 blocks of size 3 as may be seen from the level by level table produced by crossprod(). This is a permutation of Plan 11.7 in Cochran and Cox [1950]. Note how withinData is recycled to fill out the blocksize requirements.

```
BIB<-optBlock(~.,withinData=factor(1:7),blocksize=rep(3,7))
crossprod(table(c(rep(1:7, rep(3,7))),BIB$design[,1]))

    1 2 3 4 5 6 7
  1 3 1 1 1 1 1 1
  2 1 3 1 1 1 1 1
  3 1 1 3 1 1 1 1
  4 1 1 1 3 1 1 1
  5 1 1 1 1 3 1 1
  6 1 1 1 1 1 3 1
  7 1 1 1 1 1 1 3
```

A partially balanced incomplete block design with two associate classes:

```
tr<-factor(1:9)
PBIB<-optBlock(~.,withinData=tr,blocksizes=rep(3,9))
crossprod(table(c(rep(1:9, rep(3,9))),PBIB$rows))

    1 2 3 4 5 6 7 8 9
  1 3 0 1 0 1 1 1 1 1
  2 0 3 1 0 1 1 1 1 1
  3 1 1 3 1 1 1 0 1 0
  4 0 0 1 3 1 1 1 1 1
  5 1 1 1 1 3 0 1 0 1
  6 1 1 1 1 0 3 1 0 1
  7 1 1 0 1 1 1 3 1 0
  8 1 1 1 1 0 0 1 3 1
  9 1 1 0 1 1 1 0 1 3
```

4.3.2 Blocking illustration

A difficult two level factorial will be used to illustrate blocking. The usual methods for constructing half fractions of a 2^7 completely confound some second order effects. In the following design, all second order effects are estimable, but the design is not particulary orthogonal.

```
dat<-gen.factorial(2,7)
desF<-optFederov(~.^2,dat,nTrials=32,nRepeats=100)
desF[1:4]
$D
[1] 0.8868

$A
[1] 1.296784

$Ge
[1] 0.412

$Dea
[1] 0.241
```

The 32 trials of the design may be blocked into four blocks of eight, as follows:

```
desFBlk<-optBlock(~.^2,desF$design,rep(8,4),nRepeats=20)
desFBlk[1:3]
$D
[1] 0.8049815

$diagonality
[1] 0.842

$Blocks
$Blocks$B1
    X1 X2 X3 X4 X5 X6 X7
22   1 -1  1 -1  1 -1 -1
32   1  1  1  1  1 -1 -1
39  -1  1  1 -1 -1  1 -1
68   1  1 -1 -1 -1 -1  1
83  -1  1 -1 -1  1 -1  1
88   1  1  1 -1  1 -1  1
105 -1 -1 -1  1 -1  1  1
116  1  1 -1 -1  1  1  1
...
```

This is a good blocking with high diagonality. A further evaluation can be found from:

```
eval.blockdesign(~.^2,desFBlk$design,rep(8,4))
$determinant.all.terms.within.terms.centered
[1] 0.8868

$within.block.efficiencies

rho          1.000
lambda.det   0.926
lambda.trace 0.901

$comment
[1] "Too few blocks to recover interblock information."

$block.centered.properties
                   constant    whole   within
df                        1        1       28
determinant                          0.804982
gmean.variance     1.000000 1.000000 1.476160
gmean.efficiencies 1.000000    1.000    0.875
```

The most important measure of the blocking is the within blocking efficiency, which shows that the blocking is quite successful in decoupling the whole and within block effects. The `within.blocking.efficiencies` table shows the efficiency for rho=1, when the whole and within block errors are equal. Other choices for rho may be made when running the `eval.blockdesign()` program. It is good that there is little interblock information to recover, since the number of blocks is too small to enable such an analysis.

The `gmean.efficiencies` for the within terms measures the ratio of variances of centered to block centered designs, and indicates that there is some loss due to blocking.

One can input the entire candidate set to `optBlock()`, but sometimes, as in the present case, the optimum blocked design is hard to find, and the results will be disappointing as the following illustrates.

```
desFBlkD<-optBlock(~.^2,dat,rep(8,4),nRepeats=20)
desFBlkD[1:2]
$D
[1] 0.7619454

$diagonality
[1] 0.777
```

In this case it is better to start `optBlock()` with a good existing design, as follows.

```
rr<-as.numeric(rownames(desF$design))
desFBlkD<-optBlock(~.^2,dat,rows=rr,rep(8,4),nRepeats=20)
desFBlkD[1:2]
```

```
$D
[1] 0.8049815

$diagonality
[1] 0.838
```

4.3.3 Improving individual blocks with the D_p criterion

Two fractions of a 2^4 will be obtained from

```
dat<-gen.factorial(2,4)
od<-optBlock(~.,dat,c(8,8))
od

$D
[1] 1

$diagonality
[1] 1

$Blocks
$Blocks$B1
   X1 X2 X3 X4
1  -1 -1 -1 -1
2   1 -1 -1 -1
7  -1  1  1 -1
8   1  1  1 -1
9  -1 -1 -1  1
12  1  1 -1  1
14  1 -1  1  1
15 -1  1  1  1

$Blocks$B2
   X1 X2 X3 X4
3  -1  1 -1 -1
4   1  1 -1 -1
5  -1 -1  1 -1
6   1 -1  1 -1
10  1 -1 -1  1
11 -1  1 -1  1
13 -1 -1  1  1
16  1  1  1  1
```

This is an optimal design. It is orthogonal:

```
bk<-data.matrix(od$design)
t(bk)%*%bk
```

```
   X1 X2 X3 X4
X1 16  0  0  0
X2  0 16  0  0
X3  0  0 16  0
X4  0  0  0 16
```

However the individual blocks are not orthogonal:

```
bk<-data.matrix(od$Blocks$B1)
t(bk)%*%bk
```

```
   X1 X2 X3 X4
X1  8  0  0  0
X2  0  8  4  0
X3  0  4  8  0
X4  0  0  0  8
```

One can optimize with the Dp criterion to improve this situation. Either criterion="Dp" or criterion="Dpc" may be used.

```
od1<-optBlock(~.,dat,c(8,8),criterion="Dpc")
bk<-data.matrix(od1$Blocks$B1)
t(bk)%*%bk
```

```
   X1 X2 X3 X4
X1  8  0  0  0
X2  0  8  0  0
X3  0  0  8  0
X4  0  0  0  8
```

The D_p criterion may be used to find standard fractions; however, there are many optimal designs in additon to the standard fractions, and these are the most likely result. For example, the following will sometimes produce half fractions with the defining contrast -X2X3X4, but more often the cross-product matrix will be cluttered, although still representing an optimum design. Note, the model now contains second order terms.

```
od2<-optBlock(~.^2,dat,c(8,8),criterion="Dpc",nR=1000)
od2[1:2]
$D
[1] 1

$Dpc
[1] 1

dt<-model.matrix(~.^2,od2$Blocks$B1)
t(dt)%*%dt
```

```
            (Intercept) X1 X2 X3 X4 X1:X2 X1:X3 X1:X4 X2:X3 X2:X4 X3:X4
(Intercept)           8  0  0  0  0     0     0     0     0     0     0
X1                    0  8  0  0  0     0     0     0     0     0     0
X2                    0  0  8  0  0     0     0     0     0     0    -8
X3                    0  0  0  8  0     0     0     0     0    -8     0
X4                    0  0  0  0  8     0     0     0    -8     0     0
X1:X2                 0  0  0  0  0     8     0     0     0     0     0
X1:X3                 0  0  0  0  0     0     8     0     0     0     0
X1:X4                 0  0  0  0  0     0     0     8     0     0     0
X2:X3                 0  0  0  0 -8     0     0     0     8     0     0
X2:X4                 0  0  0 -8  0     0     0     0     0     8     0
X3:X4                 0  0 -8  0  0     0     0     0     0     0     8
```

4.3.4 Orthogonal Blocking – Nguyen's ideas

Orthogonal blocking occurs when the means of the variables within the blocks equal the means of the variables over all blocks, or what is the same that the means of the variables are the same within each block. This is "orthogonal" because if one subtracts the means from the variables, then their sums within each block will be zero. The Nguyen [2001] algorithm successively switches trials between blocks until their means become equal. His criterion is the sum of squares of the centered data. His idea has been extended in `optBlock()` to allow for interactions between whole and split block factors and to allow new points from a candidate list to be exchanged with those in the design – Nguyen considered only rearranging the trials in a fixed design.

A simple example will illustrate the method. The blocks are orthogonal in this example, but as in previous examples the variables are not mutually orthogonal within blocks. Note that the minimim value of the sums of squares of the S matrix is reported.

```
dat<-gen.factorial(2,4)
ob<-optBlock(~.,dat,c(8,8),criterion="OB")
ob[1:3]
$D
[1] 1

$SS
[1] 0

$Blocks
$Blocks$B1
   X1 X2 X3 X4
1  -1 -1 -1 -1
4   1  1 -1 -1
5  -1 -1  1 -1
8   1  1  1 -1
10  1 -1 -1  1
12  1  1 -1  1
```

```
13 -1 -1  1  1
15 -1  1  1  1
...

> bk<-data.matrix(ob$Blocks$B1)
> t(bk)%*%bk

     X1 X2 X3 X4
  X1  8  4 -4  0
  X2  4  8  0  0
  X3 -4  0  8  0
  X4  0  0  0  8
```

A more instructive example follows. In this example, a standard fractional factorial with defining contrast $-X1X3X4$ is found.

```
ob2<-optBlock(~.^2,dat,c(8,8),crit="OB")
dt<-model.matrix(~.^2,ob2$Blocks$B1)
t(dt)%*%dt

              (Intercept) X1 X2 X3 X4 X1:X2 X1:X3 X1:X4 X2:X3 X2:X4 X3:X4
  (Intercept)           8  0  0  0  0     0     0     0     0     0     0
  X1                    0  8  0  0  0     0     0     0     0     0    -8
  X2                    0  0  8  0  0     0     0     0     0     0     0
  X3                    0  0  0  8  0     0     0    -8     0     0     0
  X4                    0  0  0  0  8     0    -8     0     0     0     0
  X1:X2                 0  0  0  0  0     8     0     0     0     0     0
  X1:X3                 0  0  0  0 -8     0     8     0     0     0     0
  X1:X4                 0  0  0 -8  0     0     0     8     0     0     0
  X2:X3                 0  0  0  0  0     0     0     0     8     0     0
  X2:X4                 0  0  0  0  0     0     0     0     0     8     0
  X3:X4                 0 -8  0  0  0     0     0     0     0     0     8
```

4.3.5 Blocking with whole plot factors (split plots)

The blocks are often interesting, representing transitory manufacturing or other constraint conditions on the experiment. If for example, one must block an experiment because only a subset of the trials may be performed each day, then there is little reason to be interested in the average response on a particular day. Sometimes, however, blocks are more interesting, and can comprise variables of importance in a study. A common problem occurs when a variable is difficult or expensive to change, such as a temperature setting that requires hours to come to equilibrium. In such cases, the experiment can be quite lengthy if one insists on a full randomization; and it is usual to compromise and repeat a substantial fraction of the experiment without changing the setting of the “difficult” variable. Split plot experiments represent a similar structure, where the experimental factors are divided into "whole plot" and "within plot" groups. In these cases, the whole plot variables are in and of themselves interesting and must be studied with the rest.

Unfortunately, whole plot main effects seldom have sufficient replication and thus have low power, but this is another problem. The problem of interest is the structure of the within plot factors and their interactions with the whole plot factors. A design for a quadratic model is given in Goos and Vandebroek [2003] for a problem of this nature.

```
data(GVTable1)
GVTable1
       w s1 s2 s3 s4
 [1,] -1  1 -1 -1  1
 [2,] -1  1  1  1 -1
 [3,] -1 -1  0 -1 -1
 [4,] -1  1  1 -1  1
 [5,] -1 -1  1  1  1
 ...
[38,]  1  1 -1 -1  1
[39,]  1 -1  1  1 -1
[40,]  1  1 -1  1 -1
[41,]  1 -1  1 -1  1
[42,]  1  1 -1 -1 -1
```

An evaluation of this design reveals the following. There is one whole plot factor, "w," and four within plot factors "s1" through "s4." Thus there are three df for whole plots and 18 for within plots.

```
eval.blockdesign(~quad(.),GVTable1,rep(2,21))
$determinant.all.terms.within.terms.centered
[1] 0.4814902

$within.block.efficiencies

rho          1.000
lambda.det   0.732
lambda.trace 0.745

$block.centered.properties
                   constant    whole   within
df                        1        2       18
determinant                           0.279146
gmean.variance     7.000000 3.086710 4.351714
gmean.efficiencies 1.347614    1.234    0.500
```

This design has a non-uniform allocation for the whole plot factor as shown below. This is due to the fact that the algorithm used, optimized the D criterion for both the whole and within factors. The whole plot allocation is not optimal for a single factor, which means that a tradeoff in efficiencies has occurred between the whole and within parts in the design construction. In view of the replication difficulty for whole plot factors, mentioned above, one wonders about the wisdom of this.

```
table(GVTable1[,1])

-1  0  1
18  6 18
```

The `optBlock()` design for this problem may be obtained as follows. Note: the whole plot design used is the optimum design for a single quadratic factor. It may be seen that the various properties for the two designs are quite similar: the centered determinant for the GVTable1 design is larger, but the variances are a bit better for the `optBlock()` design.

```
within<-gen.factorial(3,4,varNames=c("s1","s2","s3","s4"))
whole<-data.frame(w=rep(c(-1,0,1),rep(7,3)))
desProt<-optBlock(~quad(.),withinData=within,
     wholeBlockData=whole,blocksizes=rep(2,21),nR=100)
eval.blockdesign(~quad(.),desProt$design,rep(2,21))
$determinant.all.terms.within.terms.centered
[1] 0.4559886

$within.block.efficiencies

rho          1.000
lambda.det   0.738
lambda.trace 0.734

$block.centered.properties
                   constant    whole   within
df                        1        2       18
determinant                          0.268965
gmean.variance     3.000000 2.598076 4.055509
gmean.efficiencies 1.107215    1.209    0.581
```

One can of course, optimize the within design before blocking, and as may be seen from the following, it does not do quite as well in this case as the above.

```
dat<-gen.factorial(3,5,varNames=c("w","s1","s2","s3","s4"))
des<-optFederov(~quad(.),dat,nT=42,nR=100)
od<-optBlock(~quad(.),with=des$design,who=whole,rep(2,21),nR=100)
eval.blockdesign(~quad(.),od$design,rep(2,21))
$determinant.all.terms.within.terms.centered
[1] 0.4660674

$within.block.efficiencies

rho          1.000
```

```
lambda.det   0.718
lambda.trace 0.704

$block.centered.properties
                    constant    whole   within
df                         1        2       18
determinant                              0.267203
gmean.variance      3.000000 2.598076 4.306051
gmean.efficiencies 1.450530    1.434    0.532
```

A second example from Goos and Vandebroek [2003] is available as "GVTable3," as well as a design, produced by the Trinca and Gilmour Trinca and Gilmour [2001] methodology for this problem, in the file"TGTable5" The reader may like to compare the GVTable3 design with one generated by `optBlock()`.

4.3.6 Multistratum blocking

Most tabled designs are provided with more than one level of blocking. For example the incomplete block designs given in Chapter 11 of Cochran and Cox [1950] have both blocks and replicates, with a set of blocks being allocated to each replicate. It is not obvious how this may be achieved algorithmically without considerable complications. Trinca and Gilmour [2000] have shown, however, that it is quite easy with the repeated running of standard algorithms. In essence, they observed that a multidimensional blocking problem may be viewed in a sequential fashion in which at each step in the sequence one need only consider the blocking resulting from all previous steps and the current step. Subsequent steps can be ignored, and the current problem requires only a specification of whole and within block factors using existing algorithms.

This is perhaps best explained by giving an example. The partially balanced incomplete block design on page 453 of Cochran and Cox (*loc cit*) is as follows. This is available as data set `CCTable11.1a`.

Table 6: A 3 × 3 Tripple Lattice

Block	Rep. I				Rep. II				Rep. III		
(1)	1	2	3	(4)	1	4	7	(7)	1	6	9
(2)	4	5	6	(5)	2	5	8	(8)	7	2	6
(3)	7	8	9	(6)	3	6	9	(9)	4	8	3

There are two levels of blocking, Rep, and Block, and the trials are nested in both. The idea is to construct a blocked design using the first two variables, Rep and Block, and then construct the final design using this to describe the whole block structure. Thus

```
Rep<-gen.factorial(3,1,factor=1,varName="Rep")
Block<-gen.factorial(3,1,factor=1,varName="Block")
firstDesign<-optBlock(~.,within=Block,whole=Rep,blocks=rep(3,3))
firstDesign$Blocks
```

```
$B1
    Rep Block
1     1     1
1.1   1     2
1.2   1     3

$B2
    Rep Block
2     2     1
2.1   2     2
2.2   2     3

$B3
    Rep Block
3     3     1
3.1   3     2
3.2   3     3
```

Of course we could have written this down at once, but the above illustrates the method. Now a design is created using the firstDesign to describe the whole blocks structure.

```
Runs<-gen.factorial(9,1,factor=1,varName="Runs")
finalDesign<-optBlock(~.,within=Runs,whole=firstDesign$design,rep(3,9))
```

I have tabled the final design in the same form as in Table 6. The finalDesign is not as symmetrical as the one from Cochran and Cox, in which there are no duplicate runs in a replicate, but that aside, its properties seem as good.

Table 7: A 3×3 Algorithmic blocking

Block	Rep. I				Rep. II				Rep. III	
(1)	5	6	8	(4)	2	4	7	(7)	1	2	6
(2)	3	4	6	(5)	1	5	7	(8)	4	5	9
(3)	1	3	9	(6)	2	8	9	(9)	3	7	8

For the Cochran and Cox design, one has:

```
data(CCTable11a.1a)
eval.blockdesign(~.,CCTable11.1a,rep(3,9),rho=c(0.5,1,10))
$determinant.all.terms.within.terms.centered
[1] 0.2430429

$within.block.efficiencies

rho          0.500 1.000 10.000
lambda.det   0.808 0.783  0.744
lambda.trace 0.800 0.774  0.734
```

```
$comment
[1] "Too few blocks to recover interblock information."

$block.centered.properties
                   constant    whole    within
df                        1        4         8
determinant                           0.107887
gmean.variance     1.000000 2.000000 11.000000
gmean.efficiencies 1.000000    1.225     0.815
```

Both designs are partially balanced, but it is not possible to recover interblock information on all terms.

```
eval.blockdesign(~.,finalDesign$design,rep(3,9),rho=c(.5,1,10))
$determinant.all.terms.within.terms.centered
[1] 0.2400636

$within.block.efficiencies

rho          0.500 1.000 10.000
lambda.det   0.800 0.778  0.743
lambda.trace 0.791 0.768  0.733

$comment
[1] "Too few blocks to recover interblock information."

$block.centered.properties
                   constant    whole    within
df                        1        4         8
determinant                           0.107887
gmean.variance     1.000000 2.000000 11.000000
gmean.efficiencies 1.000000    1.315     0.834
```

Trinca and Gilmore (*loc.cit.*) give more substantial examples. An evaluation of their example given in the data set TGTable3 follows. In this problem there are four three level variables, and three levels of blocking. The first blocking is into five blocks. The second introduces two of the variables with a quadratic model and creates three sub-blocks in each of the five blocks. The final blocking introduces two more variables in a quadratic model interacting with the first two and themselves in three sub-sub blocks, for a 45 trial design.

```
data(TGTable3)
eval.blockdesign(~Block+quad(X1,X2,X3,X4),TGTable3,rep(3,15))
$determinant.all.terms.within.terms.centered
[1] 0.3808839

$within.block.efficiencies
```

```
rho          1.000
lambda.det   0.944
lambda.trace 0.939

$comment
[1] "Too few blocks to recover interblock information."

$block.centered.properties
                   constant    whole   within
df                        1        9        9
determinant                          0.385155
gmean.variance     3.857143 3.906399 2.602162
gmean.efficiencies 1.000000    1.018    0.938
```

which may be compared with the following produced by `optBlock`

```
Blocks<-data.frame(Block=factor(1:5))
firstVars<-gen.factorial(3,2)
secondVars<-gen.factorial(3,2,varNames=c("X3","X4"))
firstDesign<-optBlock(~Block+quad(X1,X2),
     within=firstVars,whole=Blocks,block=rep(3,5),nR=100)
secondDesign<-optBlock(~Block+quad(X1,X2,X3,X4),
     within=secondVars,whole=firstDesign$design,rep(3,15),nR=100)
eval.blockdesign(~Block+quad(X1,X2,X3,X4),
     secondDesign$design,rep(3,15))
$determinant.all.terms.within.terms.centered
[1] 0.3951278

$within.block.efficiencies

rho          1.000
lambda.det   0.924
lambda.trace 0.907

$comment
[1] "Too few blocks to recover interblock information."

$block.centered.properties
                   constant    whole   within
df                        1        9        9
determinant                          0.408619
gmean.variance     7.145839 3.426712 2.457743
gmean.efficiencies 1.126339    1.061    0.970
```

5 Appendix A: Federov's algorithm

This is a rephrasing of the material inFederov [1972] in the form used by `optFederov()`. Its primary use is as an aid in maintaining the package. Those interested in the general ideas will be better served by reading Federov's book.

Let M be the $k \times k$ dispersion matrix of a design, and consider the update involving removing the k vector y and replacing it with the k vector x. This can be represented as $M + FF^T$, where $F = (x, iy)$, giving rise to the usual representations:

$$\begin{aligned} M_1^{-1} &= (M + FF^T)^{-1}, \\ M_1^{-1} &= M^{-1} - M^{-1}F(I_2 + F^T M^{-1} F)^{-1} F^T M^{-1}, \\ |M + FF^T| &= |M||I_2 + F^T M^{-1} F|. \end{aligned}$$

If $d_{uv} = u^T M^{-1} v$, and $d_u \equiv d_{uu}$, then $|I_2 + F^T M^{-1} F| = (1 + d_x)(1 - d_y) + d_{xy}^2 = 1 + [d_x - (d_x d_y - d_{xy}^2) - d_y] = 1 + \Delta_D$. For the D criterion, $max\Delta_D$ over the candidate vectors is used in choosing points to add in the sequential algorithm.

For linear criteria, things are more complicated. A linear criterion, $L()$, is a function such that for two dispersion matrices A and B one has $L(A+B) = L(A) + L(B)$. The A and I criteria are linear. For such criteria, one has

$$L(M_1^{-1}) = L(M^{-1}) - L(M^{-1}F(I_2 + F^T M^{-1} F)^{-1} F^T M^{-1}),$$

or

$$L(M^{-1}) - L(M_1^{-1}) = \Delta_L = L(M^{-1}F(I_2 + F^T M^{-1} F)^{-1} F^T M^{-1}),$$

and the sequential algorithm uses Δ_L in choosing points to add in the sequential algorithm. This representation assumes that it is desired to decrease the criterion value by the addition of new points.

First note that

$$(I_2 + F^T M^{-1} F)^{-1} = \begin{pmatrix} 1 - d_y & -id_{xy} \\ -id_{xy} & 1 + d_x \end{pmatrix} / (1 + \Delta_D).$$

Then let $\phi_{uv} = L(M^{-1} uv^T M^{-1})$, and $\phi_u \equiv \phi_{uu}$, and from this one can see that

$$\Delta_L = \{(1 - d_y)\phi_x + d_{xy}[\phi_{xy} + \phi_{yx}] - (1 + d_x)\phi_y\} / (1 + \Delta_D).$$

The linear algorithm chooses points to maximize Δ_L, and thereby decreases the criterion.

For the A criterion, one has $\phi_{uv} = trace(M^{-1}uv^T M^{-1}) = v^T M^{-2} u$, and for the I criterion, one has $\phi_{uv} = trace(BM^{-1}uv^T M^{-1}) = v^T M^{-1} B M^{-1} u$, where $B = X^T X / N$, and the $N \times k$ matrix X is the matrix of candidate vectors.

6 Appendix B: Blocking designs

6.1 Background

6.1.1 Blocking models

One starts with the usual model:

$$Y = X\beta + \epsilon,$$

where Y is an $N \times 1$ vector of observations, X is an $N \times k$ design matrix, ϵ is an $N \times 1$ vector of iid errors with common variance σ^2, and β is a $k \times 1$ parameter vector.

In general the rows of X are randomized. Blocking occurs when this randomization is restricted in a way that causes disjoint sets of the rows of X to be randomized within sets. As Lorenzen and Anderson [1993] show, this has the effect of changing the model to

$$Y = X\beta + Z\theta + \epsilon, \tag{3}$$

where Z is an $N \times b$ matrix of block indicator variables, each column containing 1 or 0 as the corresponding observation appears or not in the block represented by the column, and θ a b element random vector whose elements are uncorrelated with ϵ, and are mutually uncorrelated and identically distributed, with common variance σ_b^2.

For this model, the variance of Y is $V = \sigma^2(I + \rho ZZ^T)$, where $\rho = \sigma_b^2/\sigma^2$, and thus the generalized least squares estimate of β is $\hat{\beta} = (X^TV^{-1}X)^{-1}X^TV^{-1}Y$, and the covariance matrix of the estimates is

$$\text{var}(\hat{\beta}) = (X^TV^{-1}X)^{-1}, \tag{4}$$

which unfortunately depends on σ_b^2 in an essential way.

The aim of algorithmic design is the optimization of some aspect of $\text{var}(\hat{\beta})$, which becomes difficult when $\text{var}(\hat{\beta})$ depends on an unknown parameter. A straightforward approach to this problem in which the designs are dependent on the unknown variance parameter, may be found in Goos and Vandebroek [2003], where the block factors are "slow" factors which may not easily be changed, and thus in a sense, force the design process into a split-plot type model.

In order to deal with the blocking problem it is necessary to inquire more closely into the structure of the model. The key to the problem is the treatment of intrablock and interblock information. Intrablock information involves σ^2, while interblock information involves both σ^2 and σ_b^2. Indeed, as will be seen, choosing a design that maximizes intrablock information will in many cases solve the problem. A design in which the blocks are orthogonal cleanly separates the two types of information

An attractive formulation for this problem uses a rewritten version of (3) which decouples the within and between block variables. The idea is due to Cook and Nachtsheim [1989].

The columns of X may be divided into three types, $X = [X_b, X_{wb}, X_w]$. The rows in X_b are constant within blocks and represent block factors. The rows in X_w vary within blocks, and the rows in X_{wb} represent interactions between the block factors and the within block factors. It is convenient to rewrite the model (3) as

$$Y = X_a\beta_a + X_b\beta_c + Z\theta + \epsilon,$$

where $X_a = [X_w, X_{wb}]$, and the parameters are conformably partitioned as β_a and β_c. The matrix X_b often contains only the constant column.

A better representation of the model for our purposes, is the block centered form:

$$Y = \tilde{X}_a\beta_a + X_b\beta_b + Z\theta + \epsilon. \tag{5}$$

Here $\tilde{X}_a$ is the block centered form of X_a obtained by subtracting block means from each row of X_a. In this, the parameter β_a is unchanged, but the second parameter changes to β_b, representing the change due to absorbing the block means from X_a: the space spanned by the columns of X is unchanged. It is assumed that no columns of $\tilde{X}_a$ are identically zero. In this form $\tilde{X}_a^T Z \equiv 0$, and $\tilde{X_a}^T X_b \equiv 0$, and it is easy to see that

$$\text{var}(\hat{\beta_a}) = (\tilde{X_a}^T\tilde{X_a})^{-1}\sigma^2,$$

because $\sigma^2 V^{-1} = I - Z(I/\rho + Z^T Z)^{-1}Z^T$.

Thus the estimates $\tilde{\beta_a}$ and their covariances do not depend on ρ.

The interesting thing, is that something similar occurs for β_b. Let $\sigma^2 V = I - ZGZ^T$, where $G = (I/\rho + D)^{-1}$, and $D = Z^T Z$. Assume that the block sizes are all of size n, then G is a constant times the identity, and $\sigma^2 V^{-1} = I - \frac{\rho}{1+\rho n}ZZ^T$. Noting that $nX_b^T X_b = X_b^T ZZ^T X_b$, one has

$$\text{var}(\hat{\beta_b}) = \sigma^2(X_b^T X_b - \frac{n\rho}{1+\rho n}X_b^T ZZ^T X_b)^{-1} = \sigma^2(1+\rho n)(X_b^T X_b)^{-1}. \tag{6}$$

Thus, designs which optimize a criterion for both $\text{var}(\hat{\beta_a})$ and $\text{var}(\hat{\beta_b})$ may be found without reference to ρ. When the block sizes are not constant, this conclusion is no longer precise, but only approximate. There is considerable merit in keeping the block sizes constant, since in the analysis the usual residual mean squares may be used to estimate both variances.

Goos and Vandebroek choose the D criterion, and maximized $|X^T V^{-1} X|$. In their formulation the maximizing design depends on ρ. In the model (5) the determinant is $|\tilde{X}_a^T \tilde{X}_a||X_b^T X_b|/\sigma^4(1+\rho n)$, and the maximizing design does not depend on ρ.

The value of $|\tilde{X}_a^T \tilde{X}_a|$ depends on X_b, which means that simply choosing X_b to maximize $|X_b^T X_b|$, may not lead to a maximal product, and indeed it does not; however, numerical comparisons show that the effect is small, and a good strategy would seem to be to use an X_b that maximizes $|X_b^T X_b|$. An additional point worth noting, is that the variance for the whole block factors is larger than for the within block factors, and that a design which does not make $|X_b^T X_b|$ as large as possible represents a questionable tradeoff.

6.1.2 Interblock, intrablock information and optimal blocking.

The matrix $\tilde{X}_a^T \tilde{X}_a$ represents what has traditionally been called, intrablock information. It is maximized when all the block means are equal. Let $\check{X}_a$ be X_a centered about the grand mean vector g, then $\check{X}_a^T \check{X}_a = X_a^T X_a - Ngg^T$, where $N = \Sigma n_i$ and the n_i are the block sizes. For m_i the block means, this gives

$$\tilde{X}_a^T \tilde{X}_a = X_a^T X_a - \Sigma(n_i m_i m_i^T) = \check{X}_a^T \check{X}_a - \Sigma[n_i(m_i - g)(m_i - g)^T].$$

If g is fixed, then clearly the trace of $\tilde{X}_a^T \tilde{X}_a$ is maximized when all m_i are equal to g. A similar argument can be made about the determinant since $|\tilde{X}_a^T \tilde{X}_a| > 0$, $|\check{X}_a^T \check{X}_a| > 0$ and $\Sigma[n_i(m_i - g)(m_i - g)^T]$ is non-negative definite.

When the block means are all equal to the grand mean, $\check{X}_a$ is orthogonal to Z, and all information about β_a is intrablock.

The distinction between interblock and intrablock is best understood in terms of the analysis. When the block means differ, it is sometimes possible to use the block means to form an additional estimate of β_a, say $\hat{\beta}_a'$. Such an estimate is referred to as an interblock estimate, see [Scheffé, 1959, p170ff]. The intrablock estimate is $\hat{\beta}_a = (\tilde{X}_a^T \tilde{X}_a)^{-1} \tilde{X}_a^T Y$, with the estimate $\hat{\sigma}^2$ obtained from the residual variance. The interblock estimate is formed from the block means of the several matrices, $\hat{X}_a, \hat{X}_b$, and $\hat{Y}$ in a similar fashion, to obtain $(\hat{\beta}_a', \hat{\beta}_b)$ and the estimate $\hat{\eta}^2$ from the residual variance: equation (6) applies when the block sizes are equal, and of course, there must be at least as many blocks as parameters. These estimates are independent when Y is normally distributed. The final estimate is $(\hat{\beta}_a \hat{\eta}^2 + \hat{\beta}_a' \hat{\sigma}^2)/(\hat{\eta}^2 + \hat{\sigma}^2)$, where $E(\hat{\eta}^2) = \sigma^2(1 + \rho n)$. Scheffé suggests that approximate tests may be made by substituting the estimates for their expectations. Since the variance of the final estimate, using this approximation, is $\sigma^2\eta^2/(\sigma^2 + \eta^2)$, the precision of the combined estimate is always smaller than the variance of the intrablock estimate alone.

6.1.3 Measuring intrablock information

The quality of a blocking design may be measured by the amount of interblock information absorbed by the block means. Measures of this quality have been used to develop algorithms for orthogonal blocking,Nguyen [2001].

Consider the model

$$Y = \check{X}_a\beta_a + X_b\beta_b + Z\theta + \epsilon. \tag{7}$$

which is similar to (5) except that $\check{X}_a$ is X_a centered about the grand mean instead of being block centered. We will use the same symbol β_b for the block parameter, since in the case of orthogonal blocking, it is the same as in equation (5). If the block means contain no information about β_a, then residuals of $\check{X}_a$ regressed on X_b will be equal to $\check{X}_a$, and thus a measure of the magnitude of these residuals can be used to assess the quality of the design. The following makes this explicit.

First observe that for non-singular C, the upper left partition of the inverse appears as follows:

$$\begin{pmatrix} A & B^T \\ B & C \end{pmatrix}^{-1} = \begin{pmatrix} (A - B^TC^{-1}B)^{-1} & . \\ . & . \end{pmatrix} \tag{8}$$

Noting that $V^{-1} = I - Z(I/\rho + D)^{-1}Z^T = I - ZGZ^T$ with $G = (I/\rho + D)^{-1}$ and $D = Z^TZ$, one can write the following expressions for the covariance matrix of the estimates:

$$C = \sigma^2 \begin{pmatrix} \check{X}_a^TV^{-1}\check{X}_a & \check{X}_a^TV^{-1}X_b \\ X_b^TV^{-1}\check{X}_a & X_b^TV^{-1}X_b \end{pmatrix}^{-1} = \sigma^2 \begin{pmatrix} M - S^TGS & H - S^TGT \\ H^T - T^TGS & N - T^TGT \end{pmatrix}^{-1},$$

where $M = \check{X}_a^T\check{X}_a$, $N = X_b^TX_b$, $S = Z^T\check{X}_a$, $T = Z^TX_b$, $H = \check{X}_a^TX_b$.

For orthogonal blocking $S \equiv 0$ and $H \equiv 0$, and thus[6]

$$C_0 = \sigma^2 \begin{pmatrix} M & 0 \\ 0 & N - T^TGT \end{pmatrix}^{-1}.$$

A measure of the degree of orthogonal blocking is the ratio of criterion values for the within block portions of C and C_0. Using equation (8) one has the following ratio λ_D, for the determinant criterion. (It may help to note that the quantity in the numerator of λ_D represents a measurement of the residuals of $\check{X}_a$ regressed on X_b.)

$$\lambda_D = (|M - S^TGS - (H - S^TGT)(N - T^TGT)^{-1}(H^T - T^TGS)|/|M|)^{1/k}$$

Here k is the dimension of M.

When $X_b \equiv Z$, this simplifies to

$$\lambda_D = (|M - 2S^TD^{-1}S + S^TD^{-1}GDS|/|M|)^{1/k},$$

[6]It is useful to note, from equation (6), that when all blocks have the same size, n, that $N - T^TGT = \frac{N}{(1+\rho n)}$.

and to

$$\lambda_D = (|M - S^T D^{-1} S|/|M|)^{1/k},$$

when ρ is infinite; that is when the whole blocks are uninformative.

If one interprets λ_D as a measure of intrablock information, then $1 - \lambda_D$ is a measure of the interblock information that remains to be recovered by a interblock analysis.

It may be seen that λ_D is maximized by minimizing S. This is the essence of the orthogonal blocking algorithm of Nguyen [2001], which he does by minimizing the sum of squares of S. The idea is easily extended to the case where whole block factors are present.

6.1.4 Analysis of blocked experiments

The analysis of a blocked experiment is complicated by the fact that there are several variance components. The usual procedure is to estimate these by REML ,Corbeil and Searle [1976], which produces a maximum likelihood estimate using the residuals from a fixed effect analysis. There is in general no closed form and the solution is numerical. When the block sizes are equal, REML is not needed, because the usual residual estimates of variance are correct. In many unbalanced cases the error committed by ignoring the difference in block sizes is small: it is sometimes less than the error in the numerical solution involved in the maximum likelihood calculations.

The parameter σ^2 may be estimated from the residual variance when estimating β_a in model (5). An estimate of $\sigma_b^2 + \sigma^2/n$, where n is the common block size, may be obtianed from the residual variance of the model

$$\hat{Y} = \hat{X}_b \beta_b + I\theta + \hat{\epsilon},$$

where the hat's denote averages over the blocks, and I is the identity matrix.

6.1.5 Multistratum models

The model (5) may be extended to allow for several strata of blocking: see Trinca and Gilmour [2001]. For two strata, the model is

$$Y = \tilde{\tilde{X}}_a \beta_a + \tilde{X}_b \beta_b + X_c \beta_c + Z_b \theta_b + Z_c \theta_c + \epsilon,$$

where Z_b and Z_c are matrices of block indicator variables, and $\tilde{X}_b$ and X_c are, respectively, their matrices of block factors. Here $\tilde{X}_b$ is centered with respect to Z_c so that $\tilde{X}_b^T Z_c \equiv 0$. In additon $\tilde{\tilde{X}}_a$ is centered for both blocks so that $\tilde{X}_a^T Z_b \equiv 0$ and $\tilde{X}_a^T Z_c \equiv 0$. The algorithmic blocking proceeds sequentially: first blocking X_a with respect to Z_b, and then $(X_a|X_b)$ with respect to Z_c.

6.2 Alternative criteria

The D criterion minimizes the determinant of $\text{var}(\hat{\beta}_a)$ or equivalently, maximizes $|\tilde{M}_a|$, where $\tilde{M}_a = \tilde{X}_a^T \tilde{X}_a$. This is an attractive criterion for blocking. Among other attributes is the fact that D-optimum blocking also tends make the design as orthogonal as possible. The D criterion minimizes the overall variance of the estimates, but this does not guarantee that the individual blocks will be maximal in any sense. In situations where only a subset of the blocks are to be run, the parameters should be well estimated from the individual blocks. This suggests a criterion like

$$D_{pc} = \left(\prod_i^b |\tilde{M}_i/n_i|^{1/k} \right)^{1/b},$$

with $\tilde{M}_i = \tilde{X}_i^T \tilde{X}_i$, where $(\tilde{X}_i | i = 1 \ldots b)$ are the $n_i \times k$ sub-matrices of $\tilde{X}_a$ corresponding to the b blocks of size $(n_i, i = 1 \ldots b)$. One can also consider the alternative criterion:

$$D_p = \left(\prod_i^b |M_i/n_i|^{1/k} \right)^{1/b},$$

where $M_i = X_i^T X_i$, and where $(X_i | i = 1 \ldots b)$ are the $n_i \times k$ sub-matrices of X_a corresponding to the b blocks of size $(n_i, i = 1 \ldots b)$. Depending on one's viewpoint, this may or may not seem as attractive from a theoretical point of view.

Of course singularity can occur, but this may be dealt with by taking the determinants of the largest leading, non-singular submatrices. This will usually mean the leading $(n_i \times n_i)$ submatrix of M_i and the leading $(n_i - 1 \times n_i - 1)$ submatrix of $\tilde{M}_i$ when $n_i \leq k$.

6.3 Updating equations for the D_{pc} criterion.

In this section, it is assumed that $X \equiv X_w$. Let X be the unadjusted matrix from which $\tilde{X}$ is obtained by subtracting block means. Let the means of the columns of X_i be $\bar{x}_i = \frac{1}{n_i} X_i^T 1_{n_i}$, where 1_{n_i} is a column vector of n_i unities, then $\tilde{X}_i = X_i - 1_{n_i} \bar{x}_i^T$.

Let $x_{r_i}^T$ be a row of X_i and $x_{r_j}^T$ be a row in some other block. I will develop equations for updating $\tilde{X}_i$ and the corresponding information matrix, when x_{r_i} is swapped with x_{r_j}.

Let $X_{i,r_i r_j}$ be X_i after the swap, then $X_{i,r_i r_j} = X_i + \alpha_{r_i}(x_{r_j} - x_{r_i})^T$, where α_{r_i} is a column vector of n_i elements all of which are zero except the one corresponding to x_{r_i}, which is unity. Similarly, the new block mean is $\bar{x}_{i,r_i r_j} = \frac{1}{n_i} X_{i,r_i r_j}^T 1_{n_i} = \bar{x}_i + \frac{1}{n_i}(x_{r_j} - x_{r_i})$.

Combining these expressions gives an expression for the updated centered matrix:

$$\tilde{X}_{i,r_i r_j} = \tilde{X}_i + (\alpha_{r_i} - \frac{1}{n_i} 1_{n_i})(x_{r_j} - x_{r_i})^T. \tag{9}$$

From this one obtains,

$$\begin{aligned}\tilde{X}^T_{i,r_ir_j}\tilde{X}_{i,r_ir_j} &= \tilde{X}^T_i\tilde{X}_i + (x_{r_i}-\bar{x}_i)(x_{r_j}-x_{r_i})^T + (x_{r_j}-x_{r_i})(x_{r_i}-\bar{x}_i)^T + \\ & (\frac{n_i-1}{n_i})(x_{r_j}-x_{r_i})(x_{r_j}-x_{r_i})^T, \qquad (10)\end{aligned}$$

or

$$\tilde{X}^T_{i,r_ir_j}\tilde{X}_{i,r_ir_j} = \tilde{X}^T_i\tilde{X}_i + (x_{r_j}-\bar{x}_i)^{[2]} - (x_{r_i}-\bar{x}_i)^{[2]} - \frac{1}{n_i}(x_{r_j}-x_{r_i})^{[2]}, \qquad (11)$$

where $a^{[2]} \equiv aa^T$.

In an obvious fashion we can write the right hand side as

$$\tilde{X}^T_i\tilde{X}_i + V_iB_iV_i^T, \qquad (12)$$

where V_i is a k by 3 matrix and B_i is diagonal. Let $S_i = V_iB_iV_i^T$ be the adjustment, and note that the columns of V_i are linearly dependent, and can be written as $V_i = W_iH_i$ where $W_i = [(x_{r_j}-\bar{x}_i),(x_{r_i}-\bar{x}_i)]$, and

$$H_i = \left(\begin{array}{ccc} 1 & 0 & 1 \\ 0 & 1 & -1 \end{array}\right),$$

giving $S_i = W_i(H_iB_iH_i^T)W_i^T = W_iG_iW_i^T$, with G_i a non-singular matrix with determinant -1:

$$G_i = \frac{1}{n_i}\left(\begin{array}{cc} (n_i-1) & 1 \\ 1 & -(n_i+1) \end{array}\right), \qquad (13)$$

and of course,

$$G_i^{-1} = \frac{-1}{n_i}\left(\begin{array}{cc} -(n_i+1) & -1 \\ -1 & (n_i-1) \end{array}\right).$$

Letting $\tilde{M}_i = \tilde{X}^T_i\tilde{X}_i$, and $\tilde{M}_{i,r_ir_j} = \tilde{X}^T_{i,r_ir_j}\tilde{X}_{i,r_ir_j}$, one has

$$\tilde{M}_{i,r_ir_j} = \tilde{M}_i + W_iG_iW_i^T, \qquad (14)$$

and from this (see [Federov, 1972, p99]), assuming $\tilde{M}_i$ is non-singular,

$$\tilde{M}^{-1}_{i,r_ir_j} = \tilde{M}^{-1}_i - \tilde{M}^{-1}_iW_i(G_i^{-1} + W_i^T\tilde{M}^{-1}_iW_i)^{-1}W_i^T\tilde{M}^{-1}_i), \qquad (15)$$

and

$$|\tilde{M}_{i,r_ir_j}| = -|\tilde{M}_i||G_i^{-1} + W_i^T\tilde{M}_i^{-1}W_i| = |\tilde{M}_i||\Delta_i(r_i,r_j)|, \tag{16}$$

which is a trivial computation since $|\Delta_i(r_i,r_j)|$ is the determinant of a 2 by 2 matrix.

One thus has $\prod_1^b |\tilde{M}_{i,r_ir_j}| = \left(\prod_1^b |\tilde{M}_i|\right) |\Delta_i(r_i,r_j)||\Delta_j(r_j,r_i)|$ for an exchange involving blocks i and j.

6.4 Updating equations for the D criterion.

In this section, it is assumed that $X \equiv X_w$. In a similar fashion to the above, one can treat the D criterion. If $\tilde{X}_j$ is the other block involved in the swap, the adjustment S_{ij} is, from equation (10),

$$[(x_{r_i} - \bar{x}_i) - (x_{r_j} - \bar{x}_j)](x_{r_j} - x_{r_i})^T + (x_{r_j} - x_{r_i})[(x_{r_i} - \bar{x}_i) - (x_{r_j} - \bar{x}_j)]^T + \\ (x_{r_j} - x_{r_i})(x_{r_j} - x_{r_i})^T(\frac{n_i - 1}{n_i} + \frac{n_j - 1}{n_j}),$$

which can be made symmetric by completing the square, giving:

$$S_{ij} = (\bar{x}_j - \bar{x}_i)^{[2]} - [(\bar{x}_j - \bar{x}_i) - (x_{r_j} - x_{r_i})]^{[2]} + (1 - \frac{n_i + n_j}{n_in_j})(x_{r_j} - x_{r_i})^{[2]},$$

or

$$S_{ij} = V_{ij}B_{ij}V_{ij}^T, \tag{17}$$

where V_{ij} is a k by 3 matrix and B_{ij} is diagonal and note that the columns of V_{ij} are linearly dependent, and can be written as $V_{ij} = W_{ij}H_{ij}$ where $W_{ij} = [(\bar{x}_j - \bar{x}_i), (x_{r_j} - x_{r_i})]$, and

$$H_{ij} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix},$$

giving $S_{ij} = W_{ij}(H_{ij}B_{ij}H_{ij}^T)W_{ij}^T = W_{ij}G_{ij}W_{ij}^T$, with G_{ij} a non-singular matrix with determinant -1:

$$G_{ij} = \begin{pmatrix} 0 & 1 \\ 1 & -C \end{pmatrix},$$

where $C = \frac{n_i+n_j}{n_in_j}$.

Letting $\tilde{M} = \tilde{X}^T\tilde{X}$, one has after the swap, $\tilde{M}_{ij,r_ir_j} = \tilde{M} + S_{ij} = \tilde{M} + W_{ij}G_{ij}W_{ij}^T$, and from this

$$\tilde{M}^{-1}_{ij,r_ir_j} = \tilde{M}^{-1} - \tilde{M}^{-1}W_{ij}(G^{-1}_{ij} + W^T_{ij}\tilde{M}^{-1}W_{ij})^{-1}W^T_{ij}\tilde{M}^{-1}), \tag{18}$$

and

$$|\tilde{M}_{ij,r_ir_j}| = -|\tilde{M}||G^{-1}_{ij} + W^T_{ij}\tilde{M}^{-1}W_{ij}| = |\tilde{M}||\Delta_{ij}(r_i, r_j)|. \tag{19}$$

One thus has $D = |\tilde{M}_{ij,r_ir_j}| = |\tilde{M}||\Delta_{ij}(r_i, r_j)|$ for an exchange involving blocks i and j.

6.5 Updating equations for the D criterion when there are interactions between block factors and within block factors.

In this case, the row elements of $X = [X_{wb}, X_{r_j}]$ must be modified during exchanges, since X_{wb} depends on the block factors. One may represent row r of X in the ith block as $x_{r,i} = \delta_i x_r$, where x_r involves only within block factors, and δ_i is a diagonal matrix involving only block factors. For example, if u_i is a factor in block i, and v, w are within block factors, one might have,

$$\begin{aligned} x^T_{r,i} &= (v, w, u_iv, u_iw, vw), \\ &= (v, w, v, w, vw)\text{diag}(1, 1, u_i, u_i, 1) \\ &= x^T_r\delta_i. \end{aligned}$$

Splitting $x_{r,i}$ into two multiplicative factors is simple if the factors are continuous variables. Setting the whole block factors to unity will produce x_r, and setting the within block factors to unity will produce δ_i. For a categorical factor, the same may be done by coding the first level as a vector of unities. Nesting presents a problem, since the contrast matrix for a nesting factor is the identity matrix, thus it is not possible to code a level as a vector of unities; however, if X_n is X with nesting factors, and X_m is X in which the same factors are multiplied, then $X_n = X_mT$, where T is a non-singular matrix with first column $(1, 0, \ldots, 0)^T$. Thus $|\tilde{X_n}^T\tilde{X_n}| = |\tilde{X_m}^T\tilde{X_m}| \times$ **constant**, which means that the optimal design is the same for both codings.

The arguments in section (6.3) leading to equation (14) may be used. The updating equation for exchanging rows r_1 and r_2 between blocks i and j is

$$\tilde{M}_{r_1r_2} = \tilde{M} + W_iG_iW^T_i + W_jG_jW^T_j, \tag{20}$$

where for l=i,j, $W_l = \delta_l[(x_{r_2} - \bar{x}_l), (x_{r_1} - \bar{x}_l)]$, with $\bar{x}_l$ being the means of the x_l's in each block, and

$$G_i = \frac{1}{n_i}\begin{pmatrix} (n_i - 1) & 1 \\ 1 & -(n_i + 1) \end{pmatrix},$$

$$G_j = \frac{1}{n_j}\begin{pmatrix} -(n_j+1) & 1 \\ 1 & (n_j-1) \end{pmatrix}.$$

Applying equations (15) and (16), and using $d_{i,j} = Wi^T\tilde{M}^{-1}Wj$, one has

$$|\tilde{M}_{r_2,r_1}| = |\tilde{M}||\Delta_i||\Delta_j - d_{j,i}\Delta_i^{-1}d_{i,j}|,$$

where $\Delta_l = G_l^{-1} + d_{l,l}$ for $l = i, j$.

6.6 Computational note

A useful numerical method for dealing with matrix problems is to reduce the cross product matrices to products of triangular matrices. Thus $\tilde{M} = \tilde{X}^T\tilde{X} = T^TT$, where T is upper triangular. The inverse $\tilde{M}^{-1} = T^{-1}(T^T)^{-1}$ is also the product of triangular matrices. For such, products of the form $v^T\tilde{M}^{-1}v$, where v is a conformable vector, become $(v^TT^{-1})^{[2]}$ with some saving in computation. Moreover, the algorithm that reduces $\tilde{X}$ to T is such that additional weighted rows may be included or removed with small effort.

It follows that one can update either with equations like (12) or with equations like (14).

References

A.C. Atkinson and A.N. Donev. *Optimum experimental designs.* Oxford, New York, 1992.

W.G. Cochran and G.M. Cox. *Experimental designs.* Wiley, New York, 1950.

R.D. Cook and C. Nachtsheim. Computer-aided blocking of factorial and response-surface designs. *Technometrics*, 31(3):339–346, 1989.

R.R. Corbeil and R. Searle. Restricted maximum likelihood (reml) estimation of variance components in the mixed model. *Technometrics*, 18(1):31–38, 1976.

D.R. Cox. A note on polynomial response functions for mixtures. *Biometrika*, 58(1): 155–159, 1971.

D.R. Cox and N. Reid. *The theory of the design of experiments.* Chapman & Hall, New York, 2000.

A.N. Donev and A.C. Atkinson. An adjustment algorithm for the construction of exact d-optimum experimental designs. *Technometrics*, 30:429–33, 1988.

G. Elfving. Optimum allocation in linear regression theory. *Ann. Math. Stat.*, 23:255–262, 1952.

V.V. Federov. *Theory of optimal experiments.* Academic Press, New York, 1972.

D.J. Finney. The fractional replication of factorial arrangements. *Ann. Eugen.*, 12:291–301, 1945.

P. Goos and M. Vandebroek. D-optimal split-plot designs with given numbers and sizes of whole plots. *Technometrics*, 45(3):235–245, 2003.

J.W. Gorman and J.E. Hinman. Simplex lattice designs for multicomponent systems. *Technometrics*, 4(4):463–487, 1962.

Scheffé H. Experiments with mixtures. *Jour. Roy. Statist. Soc (B)*, 20:344–360, 1958.

T.J. Lorenzen and V.L. Anderson. *Design of experiments, a no-name approach.* Dekker, New York, 1993.

Nam-Ky Nguyen. Cutting experimental designs into blocks. *AusNZJSt*, 43(3):367–374, 2001.

C.S. Peirce and J. Jastrow. On small differences of sensation. *Memoirs of the National Academy of Sciences*, 3:75–83, 1884.

F. Pukelsheim and Sabine. Rieder. Efficient rounding of approximate designs. *Biometrika*, 79(4):763–770, 1992.

H. Scheffé. *The Analysis of Variance.* Wiley, New York, 1959.

S.D. Silvey. *Optimal Design.* Chapman and Hall, New York, 1980.

L.A. Trinca and S.G. Gilmour. An algorithm for arranging response surface designs in small blocks. *Computational Statistics and Data Analysis*, 33:25–43, 2000.

L.A. Trinca and S.G. Gilmour. Multistratum response surface designs. *Technometrics*, 43(1):25–33, 2001.