

EQUIVALENCE OF FRACTIONAL FACTORIAL DESIGNS

James B. Clark and A. M. Dean

Aerospace Corporation and The Ohio State University

Abstract: Two designs for a fractional factorial experiment are equivalent if one can be obtained from the other by reordering the treatment combinations, relabeling the factors and relabeling the factor levels. Designs can be viewed as sets of points in p -dimensional space, where p is the number of factors. It is shown that, in this setting, two designs are equivalent if the Hamming distances between the points are the same in all possible dimensions. An algorithm is given, based on this representation, that can detect distinct designs for 2^p experiments without a complete search of all reorderings and relabelings in the fraction. In addition, if two designs are equivalent, the algorithm gives a set of permutations which map one design to the other.

Key words and phrases: Design equivalence, fractional factorial experiment, Hamming distance.

1. Introduction

We consider factorial experiments with p factors where the i th factor is observed at $m_i \geq 2$ levels, $i = 1, \dots, p$. A *fractional factorial design* is composed of n (not necessarily distinct) treatment combinations of the $v = m_1 m_2 \dots m_p$ possible treatment combinations. Those fractions defined through defining relations will be called *group-generated* fractions. Fractional factorial designs with differing properties have been discussed extensively in the literature. Some of these are group-generated fractions (for example, those discussed by Bailey (1977), Box, Hunter and Hunter (1978), Lewis (1982)), and many are not (for example, those discussed by Plackett and Burman (1946), Wang and Wu (1992), Lin (1993), Dean and Draper (1999)).

Several authors have provided computer algorithms for constructing fractions which are optimal, or close to optimal, under a particular optimality criterion (see, for example, Mitchell (1974), Welch (1982), Atkinson and Donev (1992), Nguyen (1996), Hardin and Sloane (1993)). The constructed designs may or may not be group-generated fractions. For a given experiment, an algorithm may produce several designs, all of which have the same value of the selected optimality criterion (see Example 3.1). This does not necessarily imply, however, that the designs are “equivalent” in the following sense.

Two designs are said to be *equivalent* if one can be obtained from the other by reordering the treatment combinations, relabeling factors having the same number of levels, and/or relabeling the levels of one or more factors. Equivalent designs can be transformed into each other by the usual randomization of factor labels and level labels, whereas this is not possible for non-equivalent designs. The enumeration of equivalence classes of designs is an important combinatorial problem (for example, Cohn (1994), Seiden (1999)). From a statistical point of view, the identification of non-equivalent designs not only enlarges the class of designs that can be obtained by randomization but also may enlarge the choice of value of various efficiency criteria such as ranges of p -efficiencies (Lin (1993)) as discussed in Example 3.1.

The equivalence of group-generated fractional factorial designs has been discussed in the literature by a number of different authors. A necessary condition for the equivalence of two such designs is that they have the same *wordlength pattern* (the number of words of length i in the defining relation). Draper and Mitchell (1968) give an example of two designs with identical wordlength patterns that are, nevertheless, not equivalent, since they differ in their *letter patterns*. The letter pattern matrix of design d is the matrix A_d whose (i, j) th element is the frequency of letter i in words of length j in the defining relation of d . A necessary condition for equivalence of two designs d_1 and d_2 is that A_{d_2} can be transformed into A_{d_1} by row permutations. Draper and Mitchell (1970) conjectured that this is also a sufficient condition for design equivalence, but Chen and Lin (1991) disproved this via a counter-example. Chen (1992) discussed equivalence of group-generated designs in terms of existence of relabeling maps between frequency vectors representing the designs. He showed that, in the special case of 2^{p-1} or 2^{p-2} group-generated fractions, the wordlength pattern does uniquely determine the design. The wordlength pattern also uniquely determines the design for 2^{p-3} and 2^{p-4} group-generated minimum aberration (Fries and Hunter (1980)) fractions.

The purpose of this paper is to present a method of determining equivalence of any two factorial designs (non-group-generated as well as group-generated). The method, which is based on a geometrical representation of a design, can be used for factors at any number of levels. In Section 2, we consider equivalence of designs with factors at two or more than two qualitative levels. We view a design as a set of n points in p -dimensional space. Then two designs are equivalent if the factors can be labeled so that the Hamming distance between a pair of corresponding points is the same for the two designs in all possible dimensions. We discuss the necessary modifications for quantitative factors at more than two levels in Section 4.

In Section 3, we describe an algorithm for checking equivalence of fractional factorial designs when all factors have two levels. The algorithm generally avoids

a complete search of all $n!p!2^p$ reorderings and relabelings in the fraction and, in our experience, usually detects non-equivalent designs in just a few comparisons (often one). In theory, the worst possible case requires $p(p!)^2$ comparisons, but even this represents a considerable time saving for a design with at least $p + 1$ runs and large p . When two designs are equivalent, the algorithm gives a set of permutations which map one design to the other. The algorithm can be extended to deal with factors at more than two levels.

2. Qualitative Factors

Let \mathbf{T}_d be an $n \times p$ matrix whose (i, k) th element is the level at which the k th factor is observed in the i th run in the fractional factorial design d , $i = 1, \dots, n$, $k = 1, \dots, p$. Thus the rows of \mathbf{T}_d represent the n (not necessarily distinct) treatment combinations observed in design d and we call \mathbf{T}_d the *treatment matrix*.

The equivalence of designs can be expressed in terms of equivalence of treatment matrices as follows. Two designs d_1 and d_2 are equivalent if \mathbf{T}_{d_2} can be obtained from \mathbf{T}_{d_1} by permuting rows, permuting columns, and relabeling symbols within one or more columns. For example, the designs d_1 and d_2 corresponding to the treatment matrices \mathbf{T}_{d_1} and \mathbf{T}_{d_2} shown transposed below, are equivalent since \mathbf{T}_{d_2} can be obtained from \mathbf{T}_{d_1} by changing 1, 2 and 3 in column 2 to the symbols 3, 1 and 2 respectively, cycling rows and switching columns.

$$\mathbf{T}'_{d_1} = \begin{bmatrix} 1 & 2 & 3 & 3 & 3 \\ 1 & 1 & 2 & 2 & 3 \end{bmatrix}, \quad \mathbf{T}'_{d_2} = \begin{bmatrix} 1 & 2 & 3 & 3 & 1 \\ 3 & 3 & 1 & 2 & 3 \end{bmatrix}.$$

Let $\delta[\mathbf{T}_d]_{i,j}^k$ be equal to 1 if, in the k th column of \mathbf{T}_d , the symbols in the i th and j th rows are different, and equal to 0 if they are the same. If we depict a design as a set of n points in p -dimensional space, then $\sum_{k=1}^p \delta[\mathbf{T}_d]_{i,j}^k$ counts the number of dimensions in which the i th and j th points fail to coincide; that is, the Hamming distance between the two points. Thus, we may define the *distance matrix* H_d of design d to have (i, j) th element equal to

$$[H_d]_{i,j} = \sum_{k=1}^p \delta[\mathbf{T}_d]_{i,j}^k, \quad \text{for } i \neq j, \quad (2.1)$$

and equal to zero if $i = j$. The distance matrix H_d is invariant to permutations of columns and relabeling of symbols within columns of \mathbf{T}_d ; that is, it is invariant to relabeling of the factors and relabeling of the factor levels in d , but not to the re-ordering of treatment combinations. Consequently, a necessary condition for equivalence of designs d_1 and d_2 is that there exists a permutation matrix \mathbf{R} such that

$$H_{d_1} = \mathbf{R}H_{d_2}\mathbf{R}', \quad (2.2)$$

where $'$ denotes transpose. For example, the distance matrices H_{d_1} and H_{d_2} for the above two designs are as shown in Figure 1, and it can be verified that (2.2) holds when \mathbf{R} is a cyclic permutation matrix with first row $\{0\ 0\ 1\ 0\ 0\}$.

$$H_{d_1} = \begin{bmatrix} 0 & 1 & 2 & 2 & 2 \\ 1 & 0 & 2 & 2 & 2 \\ 2 & 2 & 0 & 0 & 1 \\ 2 & 2 & 0 & 0 & 1 \\ 2 & 2 & 1 & 1 & 0 \end{bmatrix}, \quad H_{d_2} = \begin{bmatrix} 0 & 1 & 2 & 2 & 0 \\ 1 & 0 & 2 & 2 & 1 \\ 2 & 2 & 0 & 1 & 2 \\ 2 & 2 & 1 & 0 & 2 \\ 0 & 1 & 2 & 2 & 0 \end{bmatrix}.$$

Figure 1. Distance matrices for designs d_1 and d_2 .

In Theorem 2.1, we show that, in order to obtain a sufficient condition for equivalence of d_1 and d_2 , we need to check that, for the projection of d_1 into the q -dimensional subspace defined by factors $1, \dots, q$, there exists a projection of d_2 into some q -dimensional subspace such that (2.2) holds, for each $q = 1, \dots, p - 1$. Let $\{c_1, \dots, c_p\}$ be any permutation of the integers $\{1, \dots, p\}$ and let $H_d^{\{c_1, \dots, c_q\}}$ be the distance matrix corresponding to columns $\{c_1, c_2, \dots, c_q\}$ of \mathbf{T}_d . In order to prove Theorem 2.1, we need the following lemma.

Lemma 2.1. *For any given permutation $\{c_1, c_2, \dots, c_p\}$ of $\{1, 2, \dots, p\}$ and any given $n \times n$ permutation matrix \mathbf{R} , the sequence of matrices $\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_q\}})\mathbf{R}'$, $q = 1, 2, \dots, p$, uniquely determine the treatment matrix \mathbf{T}_d up to design-equivalence.*

Proof. Let $\{c_1, c_2, \dots, c_p\}$ be a fixed permutation of the integers $\{1, 2, \dots, p\}$ and let $H_d^{\{c_1, c_2, \dots, c_q\}}$ be the distance matrix corresponding to columns $\{c_1, c_2, \dots, c_q\}$ of \mathbf{T}_d . Then $\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_q\}})\mathbf{R}'$ is the distance matrix corresponding to columns $\{c_1, c_2, \dots, c_q\}$ of $\mathbf{R}\mathbf{T}_d$ for some given permutation matrix \mathbf{R} .

For a given \mathbf{R} let the sequence of matrices $\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_q\}})\mathbf{R}'$, $q = 1, 2, \dots, p$, corresponding to a fixed but unknown design d , be fixed. Note that, for any $q \leq p$,

$$\begin{aligned} [\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_q\}})\mathbf{R}']_{i,j} &= \sum_{k=1}^{q-1} \delta[\mathbf{R}\mathbf{T}_d]_{i,j}^{c_k} + \delta[\mathbf{R}\mathbf{T}_d]_{i,j}^{c_q} \\ &= [\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_{q-1}\}})\mathbf{R}']_{i,j} + [\mathbf{R}(H_d^{\{c_q\}})\mathbf{R}']_{i,j}. \end{aligned}$$

Thus, a fixed sequence of distance matrices $\mathbf{R}(H_d^{\{c_1, c_2, \dots, c_q\}})\mathbf{R}'$, $q = 1, 2, \dots, p$, implies a fixed sequence $\mathbf{R}(H_d^{\{c_q\}})\mathbf{R}'$, $q = 1, 2, \dots, p$, and we may investigate each column of the treatment matrix separately. Let \mathbf{T}_{d^*} be an $n \times p$ treatment matrix with first row $[1, 1, \dots, 1]$. For each $q \in \{1, 2, \dots, p\}$, we construct the q th column of \mathbf{T}_{d^*} as follows. For $i = 2, 3, \dots, n$ in turn, if $[\mathbf{R}(H_d^{\{c_q\}})\mathbf{R}']_{i,j} = 0$

for some $j = 1, 2, \dots, i - 1$, then the symbol in the i th row of column q of \mathbf{T}_{d^*} is identical to the symbol in the j th row, so set $[\mathbf{T}_{d^*}]_{i,q} = [\mathbf{T}_{d^*}]_{j,q}$. Otherwise, set $[\mathbf{T}_{d^*}]_{i,q}$ equal to an unused symbol. The q th column of \mathbf{T}_{d^*} is then identical to the c_q th column of \mathbf{RT}_d , up to a relabeling of the symbols in the column. Hence \mathbf{T}_{d^*} is identical to \mathbf{T}_d up to a permutation of rows, columns and a relabeling of symbols within columns. Since row and column permutations and symbol relabelings do not affect design-equivalence, d^* is equivalent to d , and the statement of the lemma follows.

Theorem 2.1. *Designs d_1 and d_2 are equivalent if and only if there exists an $n \times n$ permutation matrix \mathbf{R} and a permutation $\{c_1, c_2, \dots, c_p\}$ of $\{1, 2, \dots, p\}$ such that, for $q = 1, 2, \dots, p$, $H_{d_1}^{\{1,2,\dots,q\}} = \mathbf{R}(H_{d_2}^{\{c_1,c_2,\dots,c_q\}})\mathbf{R}'$.*

Proof. Suppose that designs d_1 and d_2 are equivalent. The distance matrix H_{d_2} is invariant to symbol relabelings in any columns of \mathbf{T}_{d_2} . Consequently, we can assume without loss of generality that the factors in d_1 and d_2 have the same level labeling. Then we can write $\mathbf{T}_{d_1} = \mathbf{RT}_{d_2}\mathbf{C}$, where \mathbf{C} is the permutation matrix corresponding to the permutation $\{c_1, c_2, \dots, c_p\}$ that maps the factor labels of d_2 to those of d_1 , and \mathbf{R} is the permutation matrix that reorders the treatment combinations in d_2 into the same order as those in d_1 . Then, for $1 \leq k \leq p$, $[H_{d_1}^k]_{i,j} = \delta[\mathbf{RT}_{d_2}\mathbf{C}]_{i,j}^k = \delta[\mathbf{RT}_{d_2}]_{i,j}^{c_k} = [H_{d_2}^{c_k}]_{r_i,r_j}$. Necessity follows from the fact that, for each $q = 1, \dots, p$, $H_{d_1}^{\{1,2,\dots,q\}} = \sum_{k=1}^q [H_{d_1}^k]_{i,j} = \sum_{k=1}^q [H_{d_2}^{c_k}]_{r_i,r_j} = \sum_{k=1}^q \mathbf{R}(H_{d_2}^{c_k})\mathbf{R}' = \mathbf{R}(H_{d_2}^{\{c_1,\dots,c_q\}})\mathbf{R}'$. Sufficiency follows from Lemma 2.1.

Corollary 2.1. *Designs d_1 and d_2 are equivalent if and only if there exists an $n \times n$ permutation matrix \mathbf{R} and a permutation $\{c_1, c_2, \dots, c_p\}$ of $\{1, 2, \dots, p\}$ such that, for $q = 1, 2, \dots, p$, $H_{d_1}^{\{q\}} = \mathbf{R}(H_{d_2}^{\{c_q\}})\mathbf{R}'$.*

Example 2.1. Cohn (1994) constructs three distinct 18×18 binary arrays (with symbols ± 1), which can be transformed into D-optimal designs. Cohn refers to the arrays as the “case-9”, the “case (6, 3)”, and the “case (3, 3, 3)” arrays. We create a design from each of these arrays by interchanging all the symbols in any column whose first element is -1 , transposing the array, and deleting the first column (which is now a column of $+1$'s). This results in treatment matrices for three D-optimal designs with 17 factors and 18 runs. We will refer to the designs as d_9 , $d_{6,3}$ and $d_{3,3,3}$.

The design d_9 is non-equivalent to the other two since there is no subset $\{c_1, c_2, \dots, c_{15}\}$ of $\{1, 2, \dots, 18\}$ for which there exists a permutation matrix \mathbf{R} satisfying $H_{d_9}^{\{1,2,\dots,15\}} = \mathbf{R}(H_{d_{6,3}}^{\{c_1,c_2,\dots,c_{15}\}})\mathbf{R}'$ or $H_{d_9}^{\{1,2,\dots,15\}} = \mathbf{R}(H_{d_{3,3,3}}^{\{c_1,c_2,\dots,c_{15}\}})\mathbf{R}'$. Similarly, there is no subset $\{c_1, c_2, \dots, c_{12}\}$ of $\{1, 2, \dots, 18\}$ for which there exists

a permutation matrix \mathbf{R} satisfying $H_{d_{6,3}}^{\{1,2,\dots,12\}} = \mathbf{R}(H_{d_{3,3,3}}^{\{c_1,c_2,\dots,c_{12}\}})\mathbf{R}'$. The non-equivalence can be verified using the algorithm discussed in Section 3 — see Example 3.3.

In the special case where all factors have two levels, coded -1 and $+1$, the distance matrix can be written as $H_d = (p\mathbf{J}_p - \mathbf{T}_d\mathbf{T}_d')/2$, where \mathbf{J}_p is a $p \times p$ matrix of unit elements. Let $\mathbf{T}_d^{\{c_1,\dots,c_q\}}$ denote a matrix consisting of columns c_1, \dots, c_q of \mathbf{T}_d . We then have the following corollary to Theorem 2.1, which forms the basis of the algorithm discussed in Section 3.

Corollary 2.2. *Designs d_1 and d_2 are equivalent if and only if there exists an $n \times n$ permutation matrix \mathbf{R} and a permutation $\{c_1, c_2, \dots, c_p\}$ of $\{1, 2, \dots, p\}$ such that, for $q = 1, 2, \dots, p$,*

$$\mathbf{T}_{d_1}^{\{1,2,\dots,q\}}\mathbf{T}_{d_1}^{\{1,2,\dots,q\}'} = \mathbf{R}(\mathbf{T}_{d_2}^{\{c_1,c_2,\dots,c_q\}}\mathbf{T}_{d_2}^{\{c_1,c_2,\dots,c_q\}'})\mathbf{R}'. \quad (2.3)$$

3. An Algorithm for the Equivalence of Fractions of 2^p Experiments

Suppose all factors have two levels, coded -1 and $+1$. Then two treatment matrices \mathbf{T}_{d_1} and \mathbf{T}_{d_2} are equivalent if there exist permutation matrices \mathbf{R} and \mathbf{C} , and a diagonal matrix \mathbf{L} with $\mathbf{L}^2 = \mathbf{I}$, such that $\mathbf{T}_{d_1} = \mathbf{R}\mathbf{T}_{d_2}\mathbf{C}\mathbf{L}$. In order to establish equivalence, we need to determine \mathbf{R} , \mathbf{C} and \mathbf{L} .

Once \mathbf{R} and \mathbf{C} are known, \mathbf{L} is determined, since the negative entries on the diagonal of \mathbf{L} correspond to the elements in the first row of $\mathbf{R}\mathbf{T}_{d_2}\mathbf{C}$ that differ in sign from the corresponding elements in the first row of \mathbf{T}_{d_1} . The permutation matrices \mathbf{R} and \mathbf{C} are determined using Corollary 2.2.

A Fortran program for checking equivalence of two designs is obtainable from the authors. It has two parts: the first algorithm, called *Deseq1*, does an initial check for non-equivalence, while the second algorithm, *Deseq2*, looks for the permutations that transform one design to the other, if they exist. Specifically, *Deseq1* checks that, for each $q = p, p-1, 1, p-2, 2, \dots, p-[p/2], [p/2]$, there is some subset $\{c_1, \dots, c_q\}$ of $\{c_1, \dots, c_p\}$ so that the rows of $H_{d_1}^{1,\dots,q}$ and $H_{d_2}^{c_1,\dots,c_q}$ contain the same set of distances with the same multiplicity. (Here, $[.]$ denotes integer part). The algorithm calls the check at each value of q a different “stage” of check — the stage number being $p - q$. If one of the checks fails, the algorithm stops and indicates at which stage the condition failed to be satisfied. Geometrically, what these initial tests do is compare the plots for the two designs. If the designs are equivalent, the plots appear similar, and the distances between all pairs of points are the same for the two designs. Likewise if one design is collapsed into a q -dimensional space, equivalent to deleting $p - q$ columns from the design matrix, there must be some q -dimensional subspace onto which one can collapse the other

design to yield similar inter-point distances. This must be true for each value of q .

Deseq2 searches for row and column permutations which transform d_2 into d_1 . The steps in searching for permutation matrices \mathbf{R} are as follows. Initially, an $n \times n$ matrix \mathbf{P} of potential row permutations is constructed, where $[\mathbf{P}]_{ij} = 1$ if the i th row of $\mathbf{T}_{d_1} \mathbf{T}'_{d_1}$ has the same multiplicity of distinct elements as the j th row of $\mathbf{T}_{d_2} \mathbf{T}'_{d_2}$, and $[\mathbf{P}]_{ij} = 0$ otherwise. Next, the smallest m such that $[\mathbf{P}]_{1m} = 1$ is chosen and the algorithm sets $[\mathbf{R}]_{1m} = 1$. This corresponds to the first possible permutation of a row of \mathbf{T}_{d_2} to the first row of \mathbf{T}_{d_1} . The $[\mathbf{R}]_{1i}$ for $i \neq m$ are set to zero, $[\mathbf{R}]_{im}$ for $i \neq 1$ are set to zero, and for $j = 1, 2, \dots, n$, $[\mathbf{R}]_{jl}$ are set to zero for all l for which $[\mathbf{T}_{d_1} \mathbf{T}'_{d_1}]_{1j} \neq [\mathbf{T}_{d_2} \mathbf{T}'_{d_2}]_{ml}$. These same steps are carried out on succeeding rows of \mathbf{R} . If, at any step, a row of \mathbf{R} appears with no non-zero entries, the process backs up to the previous row, using the next available non-zero entry in the corresponding row of \mathbf{P} and the remainder of \mathbf{R} is simplified. Once the last row is reached, \mathbf{R} is a row permutation matrix. If no \mathbf{C} exists satisfying Corollary 2.2, another row permutation matrix is constructed by moving up one row of \mathbf{P} in the search, taking the next available non-zero entry in that row, making the simplifications, etc. This procedure is capable of finding the set \mathbf{R} of all the matrices \mathbf{R} and corresponding matrices \mathbf{C} and \mathbf{L} that satisfy $\mathbf{T}_{d_1} = \mathbf{R} \mathbf{T}_{d_2} \mathbf{C} \mathbf{L}$, without necessarily examining the entire set of permutation matrices. If \mathbf{R} is empty then d_1 and d_2 are non-equivalent at step 0. If \mathbf{R} is non-empty then *Deseq2* checks to see whether there exists at least one $\mathbf{R} \in \mathbf{R}$ satisfying the conditions of Corollary 2.2.

The algorithms can be adapted to handle qualitative factors with more than two levels by using Theorem 2.1 instead of Corollary 2.2 to calculate the distance matrices.

Example 3.1. For a 2^5 design with $n = 12$ runs, the ACED algorithm (Welch (1982)) was asked to return the ten best designs found under an excursion algorithm, the D-optimality criterion and a first order model. Two of the designs returned had $\det(n^{-1} X' X)^{-1/6} = 1.040$. The two designs had different multiplicities of runs and different values of $\text{trace}(X' X)^{-}$ (the A-criterion), and were therefore clearly not equivalent. Three other designs, d_1, d_2 and d_3 , have similar information matrices with $\det(n^{-1} X' X)^{-1/6} = 1.020$. If we list the 2^5 treatment combinations in lexicographical order from 1 to 32, then these three 12-run designs have the following treatment combinations: $d_1 : \{2, 3, 9, 13, 16, 16, 22, 23, 26, 27, 28, 29\}$; $d_2 : \{4, 6, 7, 9, 10, 16, 18, 19, 22, 28, 29, 32\}$; $d_3 : \{4, 6, 7, 10, 11, 13, 17, 20, 22, 25, 31, 32\}$. Examination of the treatment combinations reveals that design d_1 cannot be equivalent to designs d_2 and d_3 since d_1 has a repeated run and the other designs do not. Designs d_2 and d_3 are non-equivalent at stage 0, by *Deseq1*. Designs d_1 and d_2 perform similarly under the p -efficiency criterion defined by

Lin (1993), whereas design d_3 differs. Projecting into the space of $p = 2$ factors, nine of the possible subdesigns from d_3 are orthogonal arrays, but only six of the possible subdesigns from each of d_1 and d_2 are orthogonal arrays. Similarly, projecting into the space of $p = 3$ factors, seven of the possible subdesigns from d_3 are orthogonal arrays, whereas only one subdesign from each of d_1 and d_2 is an orthogonal array. Under factor sparsity, design d_3 would probably be preferred to design d_1 or d_2 .

The best five designs returned by the algorithm are orthogonal arrays (which have identical values under the D-, A- and E- criteria in all dimensions). None of these designs have repeated treatment combinations. The treatment matrices of the first three of these designs, d_4 , d_5 and d_6 , are shown below.

$$\begin{array}{ccc}
 \mathbf{T}_{d_4} & \mathbf{T}_{d_5} & \mathbf{T}_{d_6} \\
 \left[\begin{array}{cccc} 1 & 1 & 1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 \end{array} \right] & \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & 1 \end{array} \right] & \left[\begin{array}{ccccc} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ -1 & 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{array} \right]
 \end{array}$$

When *Deseq1* is used to compare pairs of these three designs, it immediately indicates that the initial checks are all passed and, therefore, the designs are possibly equivalent. The equivalence is confirmed by *Deseq2* which quickly returns the permutations for transforming d_5 into d_4 . The columns of d_5 need to be arranged in the order (1, 4, 3, 2, 5) and the rows in the order (1, 3, 5, 2, 4, 6, 11, 10, 8, 7, 9, 12). After applying these permutations to the rows and columns of d_5 , a sign change in the last column is necessary to obtain d_4 , since the first element of the last columns of T_{d_4} and T_{d_5} differ in sign. The column and row permutations for transforming d_6 into d_4 are given by *Deseq2* as (2, 1, 4, 5, 3) and (2, 3, 1, 7, 9, 8, 6, 4, 5, 12, 11, 10), respectively. Sign interchanges are necessary in the last three columns of the permuted T_{d_6} .

Example 3.2. Abraham, Chipman and Vijayan (1999) discuss supersaturated designs. For illustration, they take eight subarrays ($d_1 - d_8$) with 14 rows and 24 columns from a 28×28 Plackett and Burman orthogonal array, including

a constant column. Comparing all 28 pairs of these eight designs, *Deseq1* detects at stage zero that all but four design pairs are non-equivalent. The pairs (d_3, d_6) , (d_3, d_8) , (d_5, d_7) and (d_6, d_8) pass the checks of *Deseq1*. *Deseq2* shows that the pairs (d_3, d_6) and (d_3, d_8) contain non-equivalent designs. The design pairs (d_5, d_7) and (d_6, d_8) , however, are equivalent, and *Deseq2* returns the column and row permutations for transforming one design into the other.

Example 3.3. The three designs d_9 , $d_{6,3}$ and $d_{3,3,3}$ of Example 2.1 have the same information matrices and, consequently the same statistical properties. *Deseq1* can be used to verify that they are, nevertheless, non-equivalent. The design d_9 is found to differ from the other two designs at stage 3 ($q = 15$). The designs $d_{6,3}$ and $d_{3,3,3}$ differ at stage 6 ($q = 12$).

4. Quantitative Factors with More Than Two Levels

Quantitative factors differ from categorical factors in that the levels of the factors have a numerical meaning. The only level labels of a quantitative factor that result in equivalent designs are those that preserve the order and the relative magnitudes of the levels. We transform the levels of each factor so that the range for each is $[-1, 1]$. For a factor with a true range of $[l, u]$, this transformation is accomplished by mapping the true value x of the factor level to $[(2x - u - l)/(u - l)]$. For example, if a four-run experiment is performed with the levels of one factor set to $[100, 200, 200, 300]$, the corresponding transformed values are $[-1, 0, 0, 1]$. Any equivalent design would include two experimental runs with this factor at its middle level so, for example, a design with this factor set to $[100, 100, 200, 300]$, or $[-1, -1, 0, 1]$ when transformed, would not be equivalent. The only permissible level label change is obtained from multiplying the transformed labels of a factor by -1 .

Necessary and sufficient conditions for two designs with quantitative factors to be equivalent are similar to those for the case of qualitative factors. However, unlike the designs with qualitative factors or two-level factors, we cannot construct an equivalent design by arbitrarily setting the first row of its treatment matrix to be a row of all 1's. This means that the proof of Lemma 2.1 needs to be modified. The simplest solution is to modify \mathbf{T}_d by including an extra row of 1's as its first row, and defining the $(n + 1) \times (n + 1)$ distance matrix H_d as follows:

$$[H_d]_{ij} = \sum_{k=1}^p \left| [\mathbf{T}_d]_{ik} - [\mathbf{T}_d]_{jk} \right|, \quad \text{for } i, j \in 1, 2, \dots, n + 1.$$

Every permutation matrix \mathbf{R} is an $(n + 1) \times (n + 1)$ matrix with $\mathbf{R}_{11} = 1$, since both designs for which we are testing equivalence have a row of 1's added to

the first row of the treatment matrix. The proof of Lemma 2.1 is valid after modifying the construction of the q th column of \mathbf{T}_{d^*} as follows.

Set $[\mathbf{T}_{d^*}]_{1,q} = 1$. For $k = 2, 3, \dots, n + 1$, since $[\mathbf{R}(H_d^{\{c_q\}})\mathbf{R}']_{k,1}$ indicates the distance from the element in the first row of \mathbf{RT}_d , which is a 1, to that in the k th row, set $[\mathbf{T}_{d^*}]_{k,q} = 1 - [\mathbf{R}(H_d^{\{c_q\}})\mathbf{R}']_{k,1}$. The first column of \mathbf{T}_{d^*} will be identical to the c_q th column of \mathbf{RT}_d with a possible sign change.

Theorem 2.1 and Corollary 2.2 are also valid using the new definition of H_d . For further details, see Clark (1997).

Acknowledgements

We would like to thank Rosemary Bailey for pointing out some errors in our original examples, and the Associate Editor and referees for helpful comments.

References

- Abraham, B., Chipman, H. and Vijayan, K. (1999). Some risks in the construction and analysis of supersaturated designs. *Technometrics* **41**, 135-141.
- Atkinson, A. C. and Donev, A. N. (1992). *Optimum Experimental Designs*. Oxford Science Publications.
- Bailey, R. A. (1977). Patterns of confounding in factorial designs. *Biometrika* **64**, 597-603.
- Box, G. E. P., Hunter, W. G. and Hunter J. S. (1978). *Statistics for Experimenters*. Wiley, New York.
- Chen, J., (1992). Some results on 2^{n-k} fractional factorial designs and search for minimum aberration designs. *Ann. Statist.* **20**, 2124-2141.
- Chen, J. and Lin, D. K. J. (1991). On the identity relationship of 2^{k-p} designs. *J. Statist. Plann. Inference* **28**, 95-98.
- Clark, J. B. (1997). *Equivalence and Augmenting of Fractional Factorial Designs*. Ph.D. dissertation, The Ohio State University.
- Cohn, J. H. E. (1994). On the number of d -optimal designs. *J. Comb. Theory Ser. A* **66**, 214-225.
- Dean, A. M. and Draper, N. R. (1999). Saturated main-effect designs for factorial experiments. *Statist. Comput.* **9**, 179-185.
- Draper, N. R. and Mitchell, T. J. (1968). Construction of the set of 256-run designs of resolution ≥ 5 and the set of even 512-run designs of resolution ≥ 6 with special reference to the unique saturated designs. *Ann. Math. Statist.* **39**, 246-255.
- Draper, N. R. and Mitchell, T. J. (1970). Construction of a set of 512-run designs of resolution ≥ 5 and the set of even 1024-run designs of resolution ≥ 6 . *Ann. Math. Statist.* **41**, 876-887.
- Fries, A. and Hunter, W. G. (1980). Minimum aberration 2^{k-p} designs. *Technometrics* **22**, 601-608.
- Hardin, R. H. and Sloane, N. J. A. (1993). A new approach to the construction of optimal designs. *J. Statist. Plann. Inference* **37**, 339-369.
- Lewis, S. M. (1982). Generators for asymmetrical factorial experiments. *J. Statist. Plann. Inference* **6**, 59-64.
- Lin, D. K. J. (1993). Another look at first order saturated designs: the p -efficient designs. *Technometrics* **35**, 284-292.

- Mitchell, T. J. (1974). Computer construction of 'D-optimal' first order designs. *Technometrics* **16**, 211-220.
- Nguyen, N-K. (1996). An algorithmic approach to constructing supersaturated designs. *Technometrics* **38**, 69-73.
- Plackett, R. L. and Burman, J. P. (1946). The design of optimum multifactorial experiments. *Biometrika* **33**, 305-325.
- Seiden, E. (1999). A remark on two level orthogonal arrays of order 12. Submitted for publication.
- Wang, J. C. and Wu, C. F. J. (1992). Nearly orthogonal arrays with mixed levels and small runs. *Technometrics* **34**, 409-422.
- Welch, W. J. (1982). Branch and bound search for experimental designs based on D-optimality and other criteria. *Technometrics* **24**, 41-48.

Aerospace Corporation, P.O. Box 92957, Los Angeles, California 90009-2957, U.S.A.

E-mail: James.B.Clark@aero.org

Statistics Department, The Ohio State University, Columbus, Ohio 43210, U.S.A.

E-mail: amd@stat.ohio-state.edu

(Received June 1999; accepted September 2000)