AUSTRALASIAN JOURNAL OF COMBINATORICS
Volume **71(3)** (2018), Pages 336–350

Complete enumeration of all geometrically non-isomorphic three-level orthogonal arrays in 18 runs

E. M. Bird Deborah J. Street

University of Technology Sydney
NSW 2007
Australia

Dedicated to the memory of Anne Penfold Street:
a mother, a mentor, a role model, a friend
and a fighter until the end.

Abstract

A method is described for the construction of all geometrically non-isomorphic ternary OAs in 18 runs. One representative from each geometric isomorphism class is provided in the electronic appendix.

1 Introduction

A symmetric *orthogonal array*, OA$[N, s^m]$, of strength 2 is an array of order $N \times m$ where the set of m columns contain elements from a distinct set of s symbols arranged such that, for any pair of columns, every pair of level combinations appears equally often. Orthogonal arrays are a subset of the class of fractional factorial designs and are frequently used as designs in various areas (Prvan and Street [10]). When choosing a design to use in practice, it is important to confirm that the design chosen can be used to fit the model being proposed. Identifying equivalent designs depends on whether the factors are qualitative or quantitative. If all of the factors are qualitative then two designs are said to be equivalent (*combinatorially isomorphic*) if one can be constructed from the other through any combination of row permutations, column permutations or level permutations of one or more factors. Quantitative factors, however, have an inherent ordering in the levels, and any permutation of levels that disrupts this order may result in a design that does not retain the same statistical properties. Thus, two designs are said to be *geometrically isomorphic* if one can be constructed from the other through any combination of row or column permutations or the reversal of the levels in one, or more, factors.

For binary factors the two sorts of isomorphism are the same and there are many results available on non-isomorphic binary factorial designs; see, for instance, Katsaounis et al. [6] and references cited therein for a summary of the results. In view of our approach below, it is interesting to observe that Shrivastava and Ding [14] used a graphical representation of two-level regular fractional factorial designs to determine the isomorphism properties of the designs.

For three or more levels the two equivalences are clearly different. In this paper we focus on the determination of all the geometrically non-isomorphic symmetric ternary designs in 18 runs, as having such a complete enumeration is the only way to choose the best design for a given situation. For 18 runs, the combinatorially non-isomorphic designs with 8 or fewer factors were enumerated by Evangelaras et al. [4] and independently by Schoen [11] using a classification method based on two different definitions of the generalised word length pattern of the original arrays, and of their projections into fewer factors. Schoen and Nguyen [13] and Schoen et al. [12] give an algorithm for complete enumeration of pure-level and mixed level combinatorially inequivalent orthogonal arrays of given strength t, and run-size N.

There have been three earlier papers that have considered the determination of the geometrically non-isomorphic symmetric ternary designs in 18 runs, and three different numbers of designs have been obtained; see Table 1.1. All three methods are based on a columnwise extension procedure, where the set of geometrically non-isomorphic designs with $m-1$ factors are extended to designs with m factors. These are checked for geometrically isomorphic designs and only the geometrically non-isomorphic designs are retained for use in the next step of the procedure.

Table 1.1: Number of Geometrically Non-Isomorphic OA$[18, 3^m]$s

	m				
Authors	**3**	**4**	**5**	**6**	**7**
Tsai et al. [16]	13	129	320	440	253
Tsai et al. [15]	13	133	332	478	284
Pang and Liu [9]	13	137	333	485	291
This paper	13	137	333	485	291

In Tsai et al. [16] two designs are said to be in the same "design family" (that is, the same geometric isomorphism class) if a design criterion based on an approximation to the average A_s efficiency of the models considered, $Q(\Gamma^{(m)})$, is the same for both designs. They state that "Although it is possible that some designs will be missed, there is no certain way to avoid this without looking at every possible design, when the problem soon becomes unmanageable". The number of designs that they found is shown in Table 1.1. The same authors (Tsai et al. [17]) studied the statistical properties of the geometrically non-isomorphic three-level designs with three factors in 18 runs that can be obtained from Latin Squares.

The other two approaches are based on the use of indicator functions. These functions were introduced by Cheng and Ye [3], who show that a design is uniquely represented by its indicator function. In an unpublished manuscript, Tsai et al. [15] claim to have enumerated all geometrically non-isomorphic 18-run three-level OAs using an algorithm for checking geometric isomorphism based on the indicator function calculated using the orthogonal polynomial basis. Pang and Liu [8], however, show that an indicator function based on orthogonal complex contrasts may retain more information about the projection properties of the design. Pang and Liu [9] give an algorithm for checking geometric isomorphism based on an indicator function calculated using the orthogonal complex contrasts given by Bailey [1]. The number of designs said to have been found in each of these papers is shown in Table 1.1. Unfortunately the actual OAs are not available.

Chapter 10 of Kaski and Östergård [5] stresses the importance of using multiple methods to address enumeration problems. They suggest systems for checking the validity of computational results such as "consistency checking" in which "some relationships among the results—preferably involving the whole computation—are checked" (p. 299). As three different enumerations have given inconsistent results, it is prudent to either confirm or otherwise the reported counts of geometric isomorphism classes with an independent enumeration. In this paper, we also use a columnwise extension procedure (although we have halved the number of columns that need to be considered at each step compared to the papers mentioned above), but we use a graph-based method, implemented in the software package Nauty (McKay and Piperno [7]), to determine if two designs are geometrically non-isomorphic. We show that our method is exhaustive, and hence we have enumerated all geometrically non-isomorphic $\mathrm{OA}[18, 3^m]$s. We provide one representative from each class in the electronic appendix (Bird and Street [2]).

A ternary array in $N = 18$ runs, with equal replication of levels in each column, will satisfy the properties of an OA of strength 2 if the $3^2 = 9$ ordered pairs of level combinations appear exactly $18/9 = 2$ times in every pair of columns. When $3^m \leq 18$, there is only one design to consider, that is, all 3^m runs in the full factorial, each repeated $18/3^m$ times. Thus there is a unique O[18,3^2]. In the following section we will begin by constructing all non-isomorphic designs with $m = 3$ factors. In Section 3 we extend this process to designs with $m = 4$ factors and in Section 4, we generalise the process to m factors. We close with some further remarks.

2 Three Factors

As geometric isomorphism is invariant to row permutations, we always order the rows of an OA lexicographically. In the case of $\mathrm{OA}[18, 3^m]$s, this means we can assume without loss of generality, that the first two columns of the OA are:

$$\mathbf{c}_1' = [\ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 1\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ \ 2\ ]$$

$$\mathbf{c}_2' = [\ 0\ \ 0\ \ 1\ \ 1\ \ 2\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\ \ 2\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\ \ 2\ \ 2\ ]$$

With these two columns fixed, the task of enumerating all OAs with $m = 3$ factors has two stages:

1. Enumerate all possible third columns, which we denote by $\mathbf{c}_3$, such that all 9 pairs of treatment combinations appear exactly twice in each of the pairs of columns $[\mathbf{c}_1\ \mathbf{c}_3]$ and $[\mathbf{c}_2\ \mathbf{c}_3]$.
2. Compare all of the OAs defined by $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3]$, and discard any geometrically isomorphic designs.

We will now outline a method for enumerating all possible third columns, $\mathbf{c}_3$, and we will then use the software package Nauty (McKay and Piperno [7]) to discard isomorphic designs.

2.1 All Possible Third Columns

We can represent the entries in $\mathbf{c}_3$ as an incidence matrix of each level of $\mathbf{c}_3$ with each level of $\mathbf{c}_2$ for each level of $\mathbf{c}_1$. The following example illustrates this.

Example 2.1. *Consider the following potential third column:*

$$\mathbf{c}_3' = [\ 0\ \ 1\ \ 1\ \ 2\ \ 0\ \ 2\ \ 2\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ \ 2\ \ 0\ \ 2\ ]$$

For each level of $\mathbf{c}_1$, the incidence of each level of $\mathbf{c}_3$ appearing in the same row as each level of $\mathbf{c}_2$ is given in the Table 2.1 below. □

Table 2.1: Incidence matrix of levels of $\mathbf{c}_3$ vs. levels of $\mathbf{c}_2$ for each level of $\mathbf{c}_1$

$\mathbf{c}_1 = 0$					$\mathbf{c}_1 = 1$					$\mathbf{c}_1 = 2$				
	$\mathbf{c}_2$					$\mathbf{c}_2$					$\mathbf{c}_2$			
$\mathbf{c}_3$	**0**	**1**	**2**	Total	$\mathbf{c}_3$	**0**	**1**	**2**	Total	$\mathbf{c}_3$	**0**	**1**	**2**	Total
0	1	0	1	2	**0**	0	2	0	2	**0**	1	0	1	2
1	1	1	0	2	**1**	0	0	2	2	**1**	1	1	0	2
2	0	1	1	2	**2**	2	0	0	2	**2**	0	1	1	2
Total	2	2	2	6	Total	2	2	2	6	Total	2	2	2	6

Notice that the marginal totals for each square incidence matrix in Table 2.1 is equal to 2 for every row and column. The row totals for each matrix record the number of times that each level of $\mathbf{c}_1$ appears with each level of $\mathbf{c}_3$ (here exactly twice), and the column totals for each square record the number of times that each level of $\mathbf{c}_1$ appears with each level of $\mathbf{c}_2$ (again exactly twice, which is implicit given the specification of $\mathbf{c}_1$ and $\mathbf{c}_2$).

Finally, for the columns $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3]$ to be an OA, we must ensure that each level of $\mathbf{c}_2$ appears with each level of $\mathbf{c}_3$ exactly twice. We check this by superimposing all three incidence matrices and calculating the sum of the entries in each of the cells,

Table 2.2: Matrices within Table 2.1 superimposed

	$\mathbf{c}_2$		
$\mathbf{c}_3$	**0**	**1**	**2**
0	1+0+1=2	0+2+0=2	1+0+1=2
1	1+0+1=2	1+0+1=2	0+2+0=2
2	0+2+0=2	1+0+1=2	1+0+1=2

as illustrated in Table 2.2 below. As all counts in this superimposed matrix equal 2, each level of $\mathbf{c}_2$ appears with each level of $\mathbf{c}_3$ exactly twice, as required.

The first step in enumerating all possible vectors of $\mathbf{c}_3$ is to enumerate all possible incidence matrices. As the required marginal totals are 2, this gives us an upper bound for the entries in these matrices. Thus we can restrict our search to matrices in which in each cell is one of 0, 1 and 2. After considering each of the $3^9 = 19,683$ potential matrices and discarding any that do not have the required marginal totals, we are left with 21 valid incidence matrices, and these are given in Table 2.3.

Table 2.3: All valid incidence matrices

2 0 0	0 0 2	0 2 0	0 2 0	2 0 0	0 0 2	1 1 0
0 0 2	0 2 0	0 0 2	2 0 0	0 2 0	2 0 0	0 0 2
0 2 0	2 0 0	2 0 0	0 0 2	0 0 2	0 2 0	1 1 0
1 0 1	0 2 0	1 1 0	0 0 2	2 0 0	0 1 1	0 1 1
1 0 1	1 0 1	1 1 0	1 1 0	0 1 1	2 0 0	0 1 1
0 2 0	1 0 1	0 0 2	1 1 0	0 1 1	0 1 1	2 0 0
1 0 1	1 1 0	1 1 0	0 1 1	0 1 1	1 0 1	1 0 1
0 2 0	0 1 1	1 0 1	1 0 1	1 1 0	1 1 0	0 1 1
1 0 1	1 0 1	0 1 1	1 1 0	1 0 1	0 1 1	1 1 0

We now need to consider all $21^3 = 9,261$ sets of three of these matrices to check whether the sum in each entry of the superimposed matrix is equal to 2. Doing that, we find that there are 132 valid sets of three matrices. Each of the 132 sets is associated with a column vector, $\mathbf{c}_3$. Of these 132 column vectors, 6 remain unchanged when their levels are reversed and the resulting rows of $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3]$ are ordered lexicographically. The remaining 126 columns can be split into two groups with a one-to-one mapping when the levels are reversed. The next two examples illustrate these two types of columns.

Example 2.2. *Consider the column $\mathbf{c}_3$ in OA_1 in Table 2.4. If we reverse the levels of this column we obtain OA_2. We then re-order the rows of OA_2 lexicographically to obtain OA_3, which is exactly the same as OA_1. Hence, this OA remains unchanged when the levels of $\mathbf{c}_3$ are reversed.* □

Table 2.4: OA which remains unchanged after reversing levels of $\mathbf{c}_3$

OA_1			$\rightarrow$	OA_2			$\rightarrow$	OA_3		
$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$		$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$		$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$
0	0	1		0	0	1		0	0	1
0	0	1		0	0	1		0	0	1
0	1	0		0	1	2		0	1	0
0	1	2		0	1	0		0	1	2
0	2	0		0	2	2		0	2	0
0	2	2		0	2	0		0	2	2
1	0	0		1	0	2		1	0	0
1	0	2		1	0	0		1	0	2
1	1	0		1	1	2		1	1	0
1	1	2		1	1	0		1	1	2
1	2	1		1	2	1		1	2	1
1	2	1		1	2	1		1	2	1
2	0	0		2	0	2		2	0	0
2	0	2		2	0	0		2	0	2
2	1	1		2	1	1		2	1	1
2	1	1		2	1	1		2	1	1
2	2	0		2	2	2		2	2	0
2	2	2		2	2	0		2	2	2

Example 2.3. *Consider the column $\mathbf{c}_3$ in OA_1 in Table 2.5. If we reverse the levels of this column we obtain OA_2. We then re-order the rows of OA_2 lexicographically to obtain OA_3, which is not exactly the same as OA_1. Since the operations performed to construct OA_3 from OA_1 are allowed under geometric isomorphism, these two designs are in the same geometric isomorphism class. Hence, the two columns labelled $\mathbf{c}_3$ in each of OA_1 and OA_3 give rise to equivalent arrays and so we say that the columns are isomorphic.* □

We require only one representative from each geometric isomorphism class, so we can discard the $126/2 = 63$ columns that can be generated by reversing the levels of another column. Hence we are left with $63 + 6 = 69$ potential vectors for $\mathbf{c}_3$. These 69 columns are given in Table A1 of the Appendix.

2.2 Geometric Isomorphism

Although we were able to discard some vectors in the enumeration of all potential columns of $\mathbf{c}_3$ above, this process alone is not enough to guarantee that the remaining 69 columns will each belong to their own isomorphic class when appended to $[\mathbf{c}_1\ \ \mathbf{c}_2]$.

Suppose we intend to establish whether a pair of designs, OA_1 and OA_2, are non-isomorphic via exhaustive checking of all allowable operations under geometric isomorphism. We must first generate all designs in the geometric isomorphism class of

Table 2.5: OA which changes after reversing levels of $\mathbf{c}_3$

OA_1			$\rightarrow$	OA_2			$\rightarrow$	OA_3		
$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$		$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$		$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$
0	0	0		0	0	2		0	0	0
0	0	2		0	0	0		0	0	2
0	1	1		0	1	1		0	1	0
0	1	2		0	1	0		0	1	1
0	2	0		0	2	2		0	2	1
0	2	1		0	2	1		0	2	2
1	0	1		1	0	1		1	0	0
1	0	2		1	0	0		1	0	1
1	1	0		1	1	2		1	1	0
1	1	2		1	1	0		1	1	2
1	2	0		1	2	2		1	2	1
1	2	1		1	2	1		1	2	2
2	0	0		2	0	2		2	0	1
2	0	1		2	0	1		2	0	2
2	1	0		2	1	2		2	1	1
2	1	1		2	1	1		2	1	2
2	2	2		2	2	0		2	2	0
2	2	2		2	2	0		2	2	0

OA_1, say, and then compare each of these designs to OA_2. If none of the generated designs match OA_2, only then can we conclude that OA_1 and OA_2 are geometrically non-isomorphic. For a single design, the construction of all designs in the same isomorphism class would require reversing the levels of all $\sum_{i=0}^{3}\binom{3}{i} = 2^3 = 8$ possible subsets of columns for each of the $3! = 6$ permutations of columns. Hence, there are $8 \times 6 = 48$ designs in a geometric isomorphism class for a three-factor design, assuming the rows of each design are ordered lexicographically. Since we have narrowed our search down to 69 OAs, there are potentially $\binom{69}{2} = 2,346$ pairs of designs to consider, each of which requires the construction of 48 designs. While this task may be manageable for three-factor designs, this naive approach quickly becomes computationally prohibitive when $m > 3$.

Instead we use nauty (McKay and Piperno [7]), a program for computing automorphism groups of graphs and digraphs, to determine whether a pair of designs are isomorphic. Algorithm 1 below gives details on how we use nauty to filter out any isomorphic OAs within a set.

A Python implementation of Algorithm 1, run on a x64 PC with Intel Core i5, on the set of 69 three-factor OAs identified previously took approximately 35 seconds[1].

We determined that there are 13 geometric isomorphism classes, which is consistent

[1]We note that nauty is written in C, hence the efficiency of this algorithm could be improved with an implementation in C that calls the nauty functions directly.

Algorithm 1: Keep one representative from each geometric isomorphism class from a set of OAs

Input: Set of i OAs, $A = (OA_1, OA_2, \ldots, OA_i)$

Output: Set of j non-isomorphic OAs, $B = (OA_1, OA_2, \ldots, OA_j)$, $j \leq i$

```
Initialise B ← empty set
for each OA_A in A do
    matched ← FALSE
    for each OA_B in B do
        Execute Nauty to determine isomorhpism between OA_A and OA_B
        if OA_A and OA_B are isomorphic then
            matched ← TRUE
            Exit inner loop
    if matched = FALSE then
        Append OA_A to B
Output B
```

with earlier work. Both Tsai et al. [16] and Pang and Liu [9] give one representative from each of the 13 geometric isomorphism classes that they found, and we have confirmed that exactly one of our designs is geometrically isomorphic to exactly one of each of these sets of designs. For convenience, we have arbitrarily labelled these classes from 1 to 13, as given in Table A1 of the Appendix.

We note that despite there being 48 allowable permutations under geometric isomorphism for three-factor designs, the resulting designs will not necessarily be distinct, as illustrated in Example 2.2. This explains why the complete enumeration of all possible $\mathbf{c}_3$ in Section 2.1 only uncovered 132 columns rather than $13 \times 48 = 624$ which we would expect if all 48 designs within each isomorphism class were distinct. To confirm this finding, we have exhaustively generated all 48 designs for each of the 13 isomorphism classes and confirmed that when duplicates are removed only 132 remain across all classes.

3 Four Factors

In this section we will enumerate all four-factor OAs by appending a column to one representative from each of the 13 three-factor classes. The following Lemma shows that this will be sufficient for finding all geometrically non-isomorphic four-factor designs in the design space provided that the set of potential columns we consider is complete.

Lemma 3.1. *Let $\{\mathbf{c}_m\}$ be a complete set of all possible columns to append to an OA with $m-1$ factors. Let OA_1 and OA_2 be geometrically isomorphic OAs with $m-1$ factors. Then every m-factor OA constructed by appending a column from $\{\mathbf{c}_m\}$ to*

OA_1 will be geometrically isomorphic to an m-factor OA that can be constructed by appending a column from $\{\mathbf{c}_m\}$ to OA_2.

Proof. Let $P(\cdot)$ denote the geometric operations that transforms OA_1 to OA_2. Suppose we append a particular column from $\{\mathbf{c}_m\}$, $\mathbf{c}_{m_1}$, to OA_1 and then perform $P(\cdot)$ on the resulting m-factor design so that the first $m-1$ columns now resemble OA_2. If there are any row permutations in $P(\cdot)$, then the corresponding entries in $\mathbf{c}_{m_1}$ will also be permuted. We will label this permuted column $\mathbf{c}_{m_2}$. Since $\{\mathbf{c}_m\}$ is a complete set of all possible columns, $\mathbf{c}_{m_2}$ must be in $\{\mathbf{c}_m\}$. Hence, the OA constructed by appending $\mathbf{c}_{m_1}$ to OA_1 is geometrically isomorphic to the OA constructed by appending $\mathbf{c}_{m_2}$ to OA_2. □

Following Lemma 3.1, the first step in enumerating all geometric isomorphism classes for four-factor OAs is the generation of a complete set of fourth columns, $\{\mathbf{c}_4\}$. We can use the results from Section 2.1, where we enumerated all possible vectors of $\mathbf{c}_3$, as a starting point for finding all possible vectors $\mathbf{c}_4$. However we will also need to consider some additional permutations to ensure that this set is complete. The following example illustrates why this is the case.

Example 3.1. *The three-factor OA comprised of columns $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3]$ in Table 3.1 below is a representative of the first three-factor isomorphism class, where $\mathbf{c}_3$ is column #7 from Table A1 of the Appendix. The two potential columns to be appended, $\mathbf{c}_{4_1}$ and $\mathbf{c}_{4_2}$, are identical except for the pairs of rows (9, 10) and (17, 18) in which the order of the entries differs between $\mathbf{c}_{4_1}$ and $\mathbf{c}_{4_2}$. It can be shown that the pair of four-factor OAs $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3\ \mathbf{c}_{4_1}]$ and $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3\ \mathbf{c}_{4_2}]$ are geometrically non-isomorphic, yet $\mathbf{c}_{4_1}$ and $\mathbf{c}_{4_2}$ have the same incidence matrices versus the levels in $\mathbf{c}_2$ for each level of $\mathbf{c}_1$, as shown in Table 3.2.* □

As the previous example illustrates, it is not sufficient to consider a single column associated with each of the 132 sets of incidence matrices identified in Section 2.1 since two non-isomorphic designs can be constructed from the same set of matrices. This was not an issue when $m = 3$ since the order of the levels in $\mathbf{c}_3$ within each pair of $[\mathbf{c}_1\ \mathbf{c}_2]$ is irrelevant. For example, note that rows 9 and 10 of Table 3.1 are both associated with the pair of levels (1, 1) in $[\mathbf{c}_1\ \mathbf{c}_2]$, and the levels of $\mathbf{c}_3$ in these rows are 0 and 2 respectively. Suppose we switch the order of the levels in $\mathbf{c}_3$. When the rows of the entire design are ordered lexicographically, these two rows will be swapped, hence the initial order does not influence the final design from the perspective of $\mathbf{c}_3$. The same logic cannot be applied to pairs of entries in $\mathbf{c}_4$, however, as this column is not necessarily used in the lexicographical ordering of the rows[2]. Hence, for each of the 132 sets of matrices enumerated in Section 2.1, we need to consider all potential permutations within each pair of $[\mathbf{c}_1\ \mathbf{c}_2]$. In a worse-case scenario, this means we will need to consider $2^9 = 512$ representations for a single incidence matrix. However

[2] $\mathbf{c}_4$ may be used in the lexicographical ordering of the rows if two triples in $[\mathbf{c}_1\ \mathbf{c}_2\ \mathbf{c}_3]$ happen to be the same. In that case, the method we describe in this section will produce two columns that are essentially the same, and we will let Algorithm 1 filter them out

Table 3.1: Append $\mathbf{c}_4$ to a three-factor OA

	$\mathbf{c}_1$	$\mathbf{c}_2$	$\mathbf{c}_3$	$\mathbf{c}_{4_1}$	$\mathbf{c}_{4_2}$
1	0	0	1	0	0
2	0	0	1	1	1
3	0	1	0	0	0
4	0	1	2	2	2
5	0	2	0	1	1
6	0	2	2	2	2
7	1	0	0	2	2
8	1	0	2	1	1
9	1	1	0	0	1
10	1	1	2	1	0
11	1	2	1	0	0
12	1	2	1	2	2
13	2	0	0	2	2
14	2	0	2	0	0
15	2	1	1	1	1
16	2	1	1	2	2
17	2	2	0	1	0
18	2	2	2	0	1

Table 3.2: Incidence matrix of levels of $\mathbf{c}_4$ (either $\mathbf{c}_{4_1}$ or $\mathbf{c}_{4_2}$) vs. levels of $\mathbf{c}_2$ for each level of $\mathbf{c}_1$

$\mathbf{c}_1 = 0$				
	$\mathbf{c}_2$			
$\mathbf{c}_4$	**0**	**1**	**2**	Total
0	1	1	0	2
1	1	0	1	2
2	0	1	1	2
Total	2	2	2	6

$\mathbf{c}_1 = 1$				
	$\mathbf{c}_2$			
$\mathbf{c}_4$	**0**	**1**	**2**	Total
0	0	1	1	2
1	1	1	0	2
2	1	0	1	2
Total	2	2	2	6

$\mathbf{c}_1 = 2$				
	$\mathbf{c}_2$			
$\mathbf{c}_4$	**0**	**1**	**2**	Total
0	1	0	1	2
1	0	1	1	2
2	1	1	0	2
Total	2	2	2	6

in some cases some pairs of entries are equal, and permuting within these pairs will result in a duplicated column.

We exhaustively constructed all permutations within the 9 pairs for each of the 132 sets of matrices and this resulted in 23,436 plausible columns for $\mathbf{c}_4$ after duplicates were removed. We note that this matches the number of columns quoted in Tsai et al. [16] for their columnwise procedure, as well as the number of "balanced columns" used in Pang and Liu [9]'s algorithm. We can discard exactly half of these columns as they have a one-to-one mapping with the retained columns when the levels within the column are reversed. Hence, we are left with 11,718 columns in the complete set of $\{\mathbf{c}_4\}$ to append, in turn, to one representative from each of the 13 three-factor OA isomorphism classes. Before we use Algorithm 1 to determine the non-isomorphic arrays, we first need to determine whether each array is an OA.

Since all of the 11,718 columns in the complete set $\{\mathbf{c}_4\}$ are representatives of one of the sets of squares enumerated in the Section 2.1, we know that all pairs of levels will appear equally often in the pairs of columns $[\mathbf{c}_1\ \mathbf{c}_4]$ and $[\mathbf{c}_2\ \mathbf{c}_4]$. What remains to be checked is whether all pairs of levels appear equally often in the pair of columns $[\mathbf{c}_3\ \mathbf{c}_4]$. Hence, the list of arrays to be inputed into Algorithm 1 may be smaller than $13 \times 11,718 = 152,334$ once invalid arrays have been discarded. In fact, the number of potential OAs to be fed into Algorithm 1 was reduced to 1,944, and took less than an hour to run. This resulted in 137 geometric isomorphism classes, which matches the number quoted in Pang and Liu [9]. Although Pang and Liu [9] do not provide representatives of the classes they found when $m \geq 4$, Tsai et al. [16] give electronic copies of one representative from each of the OA$[18, 3^m]$ geometric isomorphism classes that they found for all m. When $m = 4$ they have 129 classes and we have confirmed each of these is geometrically isomorphic to exactly one of our 137 designs.

4 m Factors

The process described in the previous section for generating all four-factor isomorphism classes is a specific example of the method we have applied for all m-factor isomorphism classes when $m > 3$. That is, we can view the 11,718 columns in the complete set of $\{\mathbf{c}_4\}$ identified in the previous section as 11,718 columns in the complete set of $\{\mathbf{c}_m\}$, which can be appended to one representative from each of the geometric isomorphism classes with $m-1$ factors. When we generate an m-factor design by appending an additional column from $\{\mathbf{c}_m\}$ to an OA with $m-1$ factors, we only need to check the $m-3$ pairs of columns $[\mathbf{c}_3\ \mathbf{c}_m]$, $[\mathbf{c}_4\ \mathbf{c}_m]$, $\ldots$, $[\mathbf{c}_{m-1}\ \mathbf{c}_m]$.

We summarise the process of generating a single representative from each of the geometric isomorphism classes of m-factor OAs, $m > 3$, below.

1. Find all geometric isomorphism classes for $m-1$ factors.
2. Generate an initial set of OAs:
   (a) Append each of the 11,718 columns in the complete set $\{\mathbf{c}_m\}$ to one representative from each of the classes for $m-1$ factors.
   (b) Discard any designs that do not have all pairs of level combinations appearing exactly twice in each of the pairs of columns $[\mathbf{c}_3\ \mathbf{c}_m]$, $[\mathbf{c}_4\ \mathbf{c}_m]$, $\ldots$, $[\mathbf{c}_{m-1}\ \mathbf{c}_m]$.
3. Input the set of OAs into Algorithm 1 to filter out isomorphic designs.

We found the same number of geometrically non-isomorphic designs as reported in Pang and Liu [9] for all m. Table 1.1 summarises our findings for all OA$[18, 3^m]$ geometric isomorphism classes compared to other authors. Since we used a different method to those authors, we have successfully carried out "consistency checking" as defined by Kaski and Östergård [5].

One representative from each geometric isomorphism class is given in the electronic appendix (Bird and Street [2]). We have confirmed for all m that each of the designs given by Tsai et al. [16] is geometrically isomorphic to exactly one of our designs.

5 Concluding Remarks

The approach that we have taken in this paper can be extended in principle but the problem rapidly becomes very large. For instance, when increasing the number of levels from 3 to 4, and considering $2 \times 4^2 = 32$ runs, the number of incidence matrices for two factors that needs to be considered to determine all possible valid third columns is 282, and the number of valid third columns is 22,695, none of which are geometrically isomorphic. If we keep ternary factors but instead increase the number of runs to 27 then the number of incidence matrices is 55, giving 847 valid triples of incidence matrices, and 424 valid third columns. To extend this to approach to larger m would require the consideration of $6^9 \times 847$ potential fourth columns. For $N = 36$ there are 120 incidence matrices, 3921 valid triples corresponding to 1971 possible third columns. The number of combinatorially non-isomorphic designs for these cases are given at the website based on Schoen et al. [12] and located at `http://pietereendebak.nl/oapage/`.

Appendix

Table A1: All possible $\mathbf{c}_3$

Column	Class	$\mathbf{c}_3'$																	
1	1	0	1	1	2	0	2	2	2	0	0	1	1	0	1	1	2	0	2	
2	1	0	2	0	2	1	1	1	1	0	2	0	2	0	2	1	1	0	2	*
3	1	0	2	1	1	0	2	0	1	2	2	0	1	1	2	0	0	1	2	
4	1	0	2	1	1	0	2	0	2	0	2	1	1	1	1	0	2	0	2	*
5	1	0	2	1	1	0	2	1	1	0	2	0	2	0	2	0	2	1	1	*
6	1	0	2	1	2	0	1	1	1	0	0	2	2	0	2	1	2	0	1	
7	1	1	1	0	2	0	2	0	2	0	2	1	1	0	2	1	1	0	2	*
8	1	1	2	0	0	1	2	0	1	2	2	0	1	0	2	1	1	0	2	
9	2	0	1	0	2	1	2	1	2	1	2	0	0	0	2	0	1	1	2	
10	2	0	1	1	2	0	2	0	2	1	2	0	1	1	2	0	0	1	2	
11	2	0	1	1	2	0	2	1	2	0	1	0	2	0	2	0	2	1	1	
12	2	0	1	1	2	0	2	2	2	0	1	0	1	0	1	0	2	1	2	
13	2	0	1	2	2	0	1	1	2	0	1	0	2	0	2	0	1	1	2	
14	2	0	2	0	1	1	2	0	2	1	2	0	1	1	1	0	2	0	2	
15	2	0	2	0	1	1	2	1	2	0	1	0	2	0	1	2	2	0	1	
16	2	0	2	0	1	1	2	1	2	1	2	0	0	0	1	0	2	1	2	
17	2	0	2	0	2	1	1	0	1	1	2	0	2	1	2	0	1	0	2	
18	2	1	1	0	2	0	2	0	2	1	2	0	1	0	2	0	1	1	2	
19	2	1	2	0	0	1	2	0	2	1	2	0	1	0	1	1	2	0	2	
20	2	1	2	0	2	0	1	0	0	1	2	1	2	1	2	0	1	0	2	
21	3	0	0	2	2	1	1	1	2	0	1	0	2	1	2	0	1	0	2	
22	3	0	1	0	1	2	2	1	2	1	2	0	0	0	2	0	2	1	1	
23	3	0	1	0	1	2	2	2	2	0	1	0	1	0	1	2	2	0	1	
24	3	0	1	2	2	0	1	2	2	0	1	0	1	0	1	0	1	2	2	
25	3	0	2	0	2	1	1	1	2	1	2	0	0	0	1	0	1	2	2	
26	3	0	2	1	2	0	1	0	2	1	2	0	1	1	1	0	0	2	2	
27	3	1	1	0	2	0	2	2	2	0	1	0	1	0	0	1	2	1	2	
28	3	1	1	2	2	0	0	0	2	0	1	1	2	0	2	0	1	1	2	
29	3	1	2	0	0	1	2	1	2	1	2	0	0	0	0	1	2	1	2	
30	3	1	2	0	1	0	2	1	2	0	1	0	2	0	0	2	2	1	1	
31	3	2	2	0	1	0	1	0	0	1	2	1	2	1	1	0	2	0	2	
32	3	2	2	0	1	0	1	0	1	0	1	2	2	0	1	2	2	0	1	
33	4	0	1	2	2	0	1	0	2	1	1	0	2	1	2	0	0	1	2	
34	4	0	2	0	2	1	1	0	2	1	1	0	2	1	1	0	2	0	2	*
35	4	1	1	0	2	0	2	0	2	1	1	0	2	0	2	0	2	1	1	*
36	4	1	2	0	2	0	1	0	0	1	1	2	2	1	2	0	2	0	1	

continued...

... Table A1 continued

Column	Class	$\mathbf{c}_3'$																	
37	5	0	0	1	2	1	2	1	2	0	2	0	1	1	2	0	1	0	2
38	5	0	1	1	2	0	2	0	1	0	2	1	2	2	2	0	1	0	1
39	5	0	2	1	2	0	1	1	2	0	2	0	1	0	1	0	1	2	2
40	5	1	2	1	2	0	0	0	1	0	2	1	2	0	2	0	1	1	2
41	6	0	0	1	2	1	2	1	2	0	0	1	2	1	2	1	2	0	0
42	6	0	1	0	1	2	2	0	1	2	2	0	1	2	2	0	1	0	1
43	6	0	1	0	1	2	2	0	2	0	2	1	1	1	2	1	2	0	0
44	6	0	1	0	2	1	2	0	1	0	2	1	2	2	2	1	1	0	0
45	6	2	2	0	1	0	1	1	1	0	2	0	2	0	0	1	2	1	2
46	6	2	2	1	1	0	0	0	1	0	2	1	2	0	1	0	2	1	2
47	7	0	1	0	2	1	2	0	2	0	1	1	2	1	2	1	2	0	0
48	7	1	2	0	2	0	1	1	2	0	1	0	2	0	0	1	2	1	2
49	7	1	2	1	2	0	0	0	2	0	1	1	2	0	1	0	2	1	2
50	7	2	2	0	1	0	1	0	1	1	2	0	2	0	1	0	2	1	2
51	8	0	1	0	2	1	2	0	2	1	2	0	1	1	2	0	1	0	2
52	8	0	1	1	2	0	2	0	2	0	1	1	2	1	2	0	2	0	1
53	8	0	2	0	1	1	2	0	1	1	2	0	2	1	2	0	2	0	1
54	8	1	2	0	2	0	1	0	1	1	2	0	2	0	2	0	1	1	2
55	9	0	1	1	2	0	2	0	2	0	2	1	1	1	2	0	1	0	2
56	9	0	1	1	2	0	2	1	2	0	0	1	2	0	2	1	2	0	1
57	9	0	2	0	1	1	2	0	1	2	2	0	1	1	2	0	1	0	2
58	9	0	2	1	1	0	2	1	2	0	2	0	1	0	1	0	2	1	2
59	9	0	2	1	2	0	1	1	1	0	2	0	2	0	2	0	1	1	2
60	9	1	2	0	2	0	1	0	1	0	2	1	2	0	2	1	1	0	2
61	10	1	2	0	2	0	1	0	2	1	1	0	2	0	1	0	2	1	2
62	11	0	2	1	2	0	1	0	1	0	2	1	2	1	2	0	1	0	2
63	11	1	2	0	1	0	2	0	1	0	2	1	2	0	2	1	2	0	1
64	12	0	0	1	1	2	2	1	1	2	2	0	0	2	2	0	0	1	1
65	12	0	0	2	2	1	1	1	1	0	0	2	2	2	2	1	1	0	0
66	12	1	1	2	2	0	0	2	2	0	0	1	1	0	0	1	1	2	2
67	12	2	2	1	1	0	0	0	0	2	2	1	1	1	1	0	0	2	2
68	13	0	0	2	2	1	1	2	2	1	1	0	0	1	1	0	0	2	2
69	13	1	1	0	0	2	2	2	2	1	1	0	0	0	0	2	2	1	1

*Columns marked with * remain unchanged after the levels have been reversed*

References

[1] R. Bailey, The decomposition of treatment degrees of freedom in quantitative factorial experiments, *J. R. Stat. Soc. Ser. B Stat. Methodol.* 44 (1982), 63–70.

[2] E. Bird and D. J. Street, *Table of ternary 18-run orthogonal arrays*, `http://preview.tinyurl.com/yb3onyxy` (2017).

[3] S.-W. Cheng and K. Q. Ye, Geometric isomorphism and minimum aberration for factorial designs with quantitative factors, *Annals of Statistics* 32(5) (2004), 2168–2185.

[4] H. Evangelaras, C. Koukouvinos and E. Lappas, 18-run nonisomorphic three level orthogonal arrays, *Metrika* 66 (2007), 31–37.

[5] P. Kaski and P. R. Östergård, *Classification Algorithms for Codes and Designs*, Springer-Verlag, Berlin Heidelberg, 2006.

[6] T. Katsaounis, A. Dean and B. Jones, On equivalence of fractional factorial designs based on singular value decomposition, *J. Statist. Plann. Inf.* 143 (2013), 1950–1953.

[7] B. D. McKay and A. Piperno, Practical graph isomorphism, II, *J. Symbolic Computation* 60 (2014), 94–112.

[8] F. Pang and M.-Q. Liu, Indicator function based on complex contrasts and its application in general factorial designs, *J. Statist. Plann. Inf.* 140 (2010), 189–197.

[9] F. Pang and M.-Q. Liu, Geometric isomorphism check for symmetric factorial designs, *J. Complexity* 27(5) (2011), 441–448.

[10] T. Prvan and D. J. Street, An annotated bibliography of application papers using certain classes of fractional factorial and related designs, *J. Statist. Plann. Inf.* 106 (2002), 245–269.

[11] E. D. Schoen, All orthogonal arrays with 18 runs, *Quality and Reliability Eng. Int.* 25(4) (2009), 467–480.

[12] E. D. Schoen, P. T. Eendebak and M. V. Nguyen, Complete enumeration of pure-level and mixed-level orthogonal arrays, *J. Combin. Des.* 18(2) (2010), 123–140.

[13] E. D. Schoen and M. V. Nguyen, *Enumeration and Classification of Orthogonal Arrays*, University of Antwerp, Faculty of Applied Economics, 2007.

[14] A. K. Shrivastava and Y. Ding, Graph based isomorph-free generation of two-level regular fractional factorial designs, *J. Statist. Plann. Inf.* 140 (2010), 169–179.

[15] K. Tsai, K. Ye and W. Li, A complete catalog of geometrically non-isomorphic 18-run orthogonal arrays, *Preprint.* `http://202.113.29.3/~stat/doe.files/CP/Ye,Kenny.pdf` (2006).

[16] P. W. Tsai, S. G. Gilmour and R. Mead, Projective three-level main effects designs robust to model uncertainty, *Biometrika* 87(2) (2000), 467–475.

[17] P. W. Tsai, S. G. Gilmour and R. Mead, Statistical isomorphism of three-level fractional factorial designs, *Utilitas Mathematica* 70 (2006), 3–9.

(Received 16 Jan 2017; revised 31 July 2017)