

An Algorithm for Constructing Orthogonal and Nearly-Orthogonal Arrays With Mixed Levels and Small Runs

Hongquan Xu

Department of Statistics
University of California
Los Angeles, CA 90095
(hqxu@stat.ucla.edu)

Orthogonal arrays are used widely in manufacturing and high-technology industries for quality and productivity improvement experiments. For reasons of run size economy or flexibility, nearly-orthogonal arrays are also used. The construction of orthogonal or nearly-orthogonal arrays can be quite challenging. Most existing methods are complex and produce limited types of arrays. This article describes a simple and effective algorithm for constructing mixed-level orthogonal and nearly-orthogonal arrays that can construct a variety of small-run designs with good statistical properties efficiently.

KEY WORDS: J_2 -optimality; Exchange algorithm; Interchange algorithm; J_2 -optimality.

1. INTRODUCTION

Consider an experiment to screen factors that may influence the blood glucose readings of a clinical laboratory testing device. One two-level factor and eight three-level factors are included in the experiments. The nine factors (Wu and Hamada 2000, table 7.3) are (A) wash (no or yes), (B) microvial volume (2.0, 2.5, or 3.0 mL), (C) caras H₂O level (20, 28, or 35 mL), (D) centrifuge speed (2,100, 2,300, or 2,500 rpm), (E) centrifuge time (1.75, 3, or 4.5 minutes), (F) sensitivity (.10, .25, or .50), (G) temperature (25, 30, or 37°C), (H) dilution ratio (1:51, 1:101, or 1:151), and (I) absorption (2.5, 2, or 1.5). To ensure that all the main effects are estimated clearly from one another, it is desirable to use an *orthogonal array* (OA). The smallest OA found for one two-level factor and eight three-level factors requires 36 runs. However, the scientist wants to reduce the cost of this experiment and plans to use an 18-run design. A good solution then is to use an 18-run *nearly-orthogonal array* (NOA).

The concept of OA dates back to Rao (1947). OAs have been used widely in manufacturing and high-technology industries for quality and productivity improvement experiments, as evidenced by many industrial case studies and recent design textbooks (Myers and Montgomery 1995; Wu and Hamada 2000). Applications of NOAs have been described by Wang and Wu (1992), Nguyen (1996b), and the references cited therein.

Formally, an OA of strength two, denoted by $OA(N, s_1 \cdots s_n)$, is an $N \times n$ matrix of which the i th column has s_i levels and for any two columns all of their level combinations appear equally often. An OA is *mixed* if not all s_i 's are equal. An NOA, denoted by $OA'(N, s_1 \cdots s_n)$, is optimal under the J_2 criterion (defined in Sec. 2.1). From an estimation standpoint, all of the main effects of an OA are estimable and orthogonal to each other, whereas all of the main effects of an NOA are still estimable, but some are partially aliased with others. Because balance is a desired and important property in practice, in this article only balanced $OA'(N, s_1 \cdots s_n)$ are considered, in

which all levels appear equally often for any column. When an array is used as a factorial design, each column is assigned to a factor and each row corresponds to a run. Here the terms of "array" and "design," "row" and "run," and "column" and "factor" are freely exchanged.

The purpose of this article is to present a simple and effective algorithm for constructing OAs and NOAs with mixed levels and small runs. The algorithm can efficiently construct various designs with good statistical properties. Section 2 introduces the concept of J_2 -optimality and other optimality criteria. Section 3 describes an algorithm for constructing mixed-level OAs and NOAs. Section 4 gives the performance and compares the algorithm with others in terms of speed and efficiency. Section 5 revisits the blood glucose experiment, and Section 6 gives concluding remarks.

2. OPTIMALITY CRITERIA

A combinatorial criterion, J_2 -optimality, is introduced in Section 2.1. This criterion has the advantages of convenience for programming and efficiency for computation. The statistical justification of J_2 -optimality and other optimality criteria is given in Section 2.2.

2.1 The Concept of J_2 -Optimality

For an $N \times n$ matrix $\mathbf{d} = [x_{ik}]$, weight $w_k > 0$ is assigned for column k , which has s_k levels. For $1 \leq i, j \leq N$, let

$$\delta_{i,j}(\mathbf{d}) = \sum_{k=1}^n w_k \delta(x_{ik}, x_{jk}), \quad (1)$$

where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. The $\delta_{i,j}(\mathbf{d})$ value measures the similarity between the i th and j th rows of \mathbf{d} .

In particular, if $w_k = 1$ is chosen for all k , then $\delta_{i,j}(\mathbf{d})$ is the number of coincidences between the i th and j th rows. Define

$$J_2(\mathbf{d}) = \sum_{1 \leq i < j \leq N} [\delta_{i,j}(\mathbf{d})]^2.$$

A design is J_2 -optimal if it minimizes J_2 . Obviously, by minimizing $J_2(\mathbf{d})$, it is desired that the rows of \mathbf{d} be as dissimilar as possible. The following lemma shows an important lower bound of J_2 .

Lemma 1. For an $N \times n$ matrix \mathbf{d} whose k th column has s_k levels and weight w_k ,

$$J_2(\mathbf{d}) \geq L(n) = 2^{-1} \left[\left(\sum_{k=1}^n N s_k^{-1} w_k \right)^2 + \left(\sum_{k=1}^n (s_k - 1) (N s_k^{-1} w_k)^2 \right) - N \left(\sum_{k=1}^n w_k \right)^2 \right], \tag{2}$$

and the equality holds if and only if \mathbf{d} is an OA.

The proof is given in Appendix A. From Lemma 1, an OA is J_2 -optimal with any choice of weights if it exists, whereas an NOA under J_2 -optimality may depend on the choice of weights.

Example 1. Consider the 12×10 matrix given in Table 1. The first column has three levels, and the other nine columns have two levels each. For illustration, $w_k = 1$ is chosen for all k . First, consider a design comprising the first five columns.

Table 1. $OA(12, 3^1 2^9)$

Run	1	2	3	4	5	6	7	8	9	10
1	0	0	1	0	1	1	1	0	0	0
2	0	1	0	0	1	1	0	0	1	0
3	0	0	1	1	0	1	0	1	1	1
4	0	1	0	1	0	0	1	1	0	0
5	1	0	0	0	0	0	0	0	0	1
6	1	1	1	0	0	0	1	0	1	1
7	1	0	1	1	1	0	0	1	1	0
8	1	1	0	1	1	1	1	1	0	1
9	2	0	0	1	0	1	1	0	1	0
10	2	1	1	0	0	1	0	1	0	0
11	2	0	0	0	1	0	1	1	1	1
12	2	1	1	1	1	0	0	0	0	1

NOTE: The 'rst 've columns form an $OA(12, 3^1 2^4)$. The pairs of columns (1, 6) and (1, 10) are nonorthogonal and have an A_2 value of .167; the pairs of columns (2, 9), (3, 7), (4, 8), and (6, 10) are nonorthogonal and have an A_2 value of .111; and all other pairs of columns are orthogonal.

The coincidence matrix ($\delta_{i,j}(\mathbf{d})$) of the 12 rows is

5	3	3	1	2	2	3	1	1	2	3	2
3	5	1	3	2	2	1	3	1	2	3	2
3	1	5	3	2	2	3	1	3	2	1	2
1	3	3	5	2	2	1	3	3	2	1	2
2	2	2	2	5	3	2	2	3	2	3	0
2	2	2	2	3	5	2	2	1	4	1	2
3	1	3	1	2	2	5	3	2	1	2	3
1	3	1	3	2	2	3	5	2	1	2	3
1	1	3	3	3	1	2	2	5	2	3	2
2	2	2	2	2	4	1	1	2	5	2	3
3	3	1	1	3	1	2	2	3	2	5	2
2	2	2	2	0	2	3	3	2	3	2	5

and J_2 is the sum of squares of the elements above the diagonal. It is easy to verify that $J_2 = 330$ and that the lower bound in (2) is also 330 for one three-level and four two-level columns with $w_k = 1$. Therefore, the first five columns form an $OA(12, 3^1 2^4)$, because the J_2 value equals the lower bound. Next, consider the whole array, comprising all 10 columns. Simple calculation shows that $J_2 = 1,284$ and that the lower bound in (2) is 1,260. Therefore, the whole array is not an OA, because the J_2 value is greater than the lower bound.

Now consider the change in the J_2 value if a column is added to \mathbf{d} or if two symbols are switched in a column. If a column $\mathbf{c} = (c_1, \dots, c_N)'$ is added to \mathbf{d} and \mathbf{d}_+ is the new $N \times (n + 1)$ design, and if \mathbf{c} has s_k levels and weight w_k , then for $1 \leq i, j \leq N$,

$$\delta_{i,j}(\mathbf{d}_+) = \delta_{i,j}(\mathbf{d}) + w_k \delta_{i,j}(\mathbf{c}), \tag{3}$$

where $\delta_{i,j}(\mathbf{c}) = \delta(c_i, c_j)$. In addition, if the added column \mathbf{c} is balanced, then it is easy to show that

$$J_2(\mathbf{d}_+) = J_2(\mathbf{d}) + 2w_k \sum_{1 \leq i < j \leq N} \delta_{i,j}(\mathbf{d}) \delta_{i,j}(\mathbf{c}) + 2^{-1} N w_k^2 (N s_k^{-1} - 1). \tag{4}$$

The summation in the second term on the right side of (4) does not involve any multiplication, because $\delta_{i,j}(\mathbf{c})$ is either 0 or 1. Therefore, calculating $J_2(\mathbf{d}_+)$ as in (4) is much faster than by taking the sum of squares of all $\delta_{i,j}(\mathbf{d}_+)$ in (3). Now suppose that two distinct symbols, $c_a \neq c_b$, in rows a and b in the added column are switched. Then all $\delta_{i,j}(\mathbf{c})$ are unchanged, except that $\delta_{a,j}(\mathbf{c}) = \delta_{j,a}(\mathbf{c})$ and $\delta_{b,j}(\mathbf{c}) = \delta_{j,b}(\mathbf{c})$ are switched for $j \neq a, b$. Hence $J_2(\mathbf{d}_+)$ is reduced by $2w_k \Delta(a, b)$, where

$$\Delta(a, b) = \sum_{1 \leq j \neq a, b \leq N} [\delta_{a,j}(\mathbf{d}) - \delta_{b,j}(\mathbf{d})][\delta_{a,j}(\mathbf{c}) - \delta_{b,j}(\mathbf{c})]. \tag{5}$$

Calculation of $\Delta(a, b)$ involves no multiplication, because both $\delta_{a,j}(\mathbf{c})$ and $\delta_{b,j}(\mathbf{c})$ are either 0 or 1. These formulas provide an efficient way to update the J_2 value and are used in the algorithm.

2.2 Other Optimality Criteria and Statistical Justification of J_2 -Optimality

To gain an understanding of the statistical justification of J_2 -optimality, recall other optimality criteria. It is well known that an s -level factor has $s - 1$ degrees of freedom. Commonly used contrasts are from orthogonal polynomials, especially for quantitative factors. For example, the orthogonal polynomials corresponding to levels 0 and 1 of a two-level factor are -1 and $+1$, and the orthogonal polynomials corresponding to levels 0, 1, and 2 of a three-level factor are -1 , 0, and $+1$ for linear effects and 1, -2 , and 1 for quadratic effects.

For an $N \times n$ matrix $\mathbf{d} = [x_{ik}]$, whose k th column has s_k levels, consider the main-effects model

$$Y = \beta_0 \mathbf{1} + X_1 \beta_1 + \varepsilon,$$

where Y is the vector of N observations, β_0 is the general mean, β_1 is the vector of all the main effects, $\mathbf{1}$ is the vector of 1s, X_1 is the matrix of contrast coefficients for β_1 , and ε is the vector of independent random errors. Let $X_1 = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ and $X = (\mathbf{x}_1/\|\mathbf{x}_1\|, \dots, \mathbf{x}_m/\|\mathbf{x}_m\|)$, where $m = \sum (s_i - 1)$. In the literature, \mathbf{d} is known as the design matrix and \mathbf{X}_1 is the model matrix (of the main-effects model). A design is D -optimal if it maximizes $|X'X|$. It is well known that $|X'X| \leq 1$ for any design and that $|X'X| = 1$ if and only if the original design \mathbf{d} is an OA. Wang and Wu (1992) proposed the D criterion

$$D = |X'X|^{1/m} \quad (6)$$

to measure the overall efficiency of an NOA. Note that $R = X'X$ is the correlation matrix of m columns of X_1 .

A good surrogate for the D criterion is the (M, S) criterion (Eccleston and Hedayat 1974). A design is (M, S) -optimal if it maximizes $\text{tr}(X'X)$ and minimizes $\text{tr}[(X'X)^2]$ among those designs that maximize $\text{tr}(X'X)$. The (M, S) criterion is cheaper to compute than the D criterion and has been used in the construction of computer-aided designs (see, e.g., Lin 1993; Nguyen 2001). Because all diagonal elements of $X'X$ are 1s, the (M, S) criterion reduces to the minimization of $\text{tr}[(X'X)^2]$, which is the sum of squares of elements of $X'X$, or, equivalently, to the minimization of the sum of squares of off-diagonal elements of $X'X$. This minimization leads to the following concept of A_2 -optimality.

Formally, if $X'X = [r_{ij}]$, let

$$A_2 = \sum_{i < j} r_{ij}^2,$$

which measures the overall aliasing (or nonorthogonality) between all possible pairs of columns. In particular, for a two-level design, A_2 equals the sum of squares of correlation between all possible pairs of columns, and therefore it is equivalent to the popular $\text{ave}(s^2)$ criterion in the context of two-level supersaturated designs. A design is A_2 -optimal if it minimizes A_2 . This is a good optimality criterion for NOAs because $A_2 = 0$ if and only if \mathbf{d} is an OA. Further, A_2 -optimality is a special case of the *generalized minimum aberration* criterion proposed by Xu and Wu (2001) for assessing nonregular designs.

The statistical justification for J_2 -optimality arises from the following lemma, which shows an important identity relating the J_2 and A_2 criteria.

Lemma 2. For a balanced design \mathbf{d} of N runs and n factors, if the weight equals the number of levels for each factor (i.e., $w_k = s_k$), then

$$J_2(\mathbf{d}) = N^2 A_2(\mathbf{d}) + 2^{-1} N \left[Nn(n-1) + N \sum s_k - (\sum s_k)^2 \right].$$

The proof is given in Appendix A. For convenience, the choice of $w_k = s_k$ is called *natural weights*. J_2 -optimality with natural weights is equivalent to A_2 -optimality and thus is a good surrogate for D -optimality.

Advantages of the use of J_2 over D , (M, S) , A_2 and $\text{ave}(s^2)$ as an objective function include the following:

1. It is simple to program. J_2 works with the design matrix, whereas all other criteria work with the model matrix.
2. It is cheap to compute. Neither the calculation of $\delta_{i,j}(\mathbf{d})$ in (1) nor that of $\delta_{i,j}(\mathbf{d}_+)$ in (3) involves any multiplication, because both $\delta(x_{ik}, x_{jk})$ and $\delta_{i,j}(\mathbf{c})$ are either 0 or 1, and this speeds up the algorithm.
3. It works with columns of more than two levels. Note that the NOA algorithm of Nguyen (1996b) works only with two-level columns. To construct an $OA'(18, 2^1 3^8)$, for example, Nguyen has to use a separate blocking algorithm (see Nguyen 2001) to divide an $OA(18, 2^1 3^7)$ into three blocks.
4. It works with any choice of weights. By choosing proper weights, one can construct different types of NOAs with a single algorithm. Note that to construct two types of $OA'(12, 3^1 2^9)$'s, Nguyen (1996b) has to code the three-level column differently in his NOA algorithm. This advantage is discussed in more detail at the end of Section 4.2.
5. It is very efficient when the number of runs is less than the number of parameters, as in the case of supersaturated designs.

3. AN ALGORITHM

The basic idea of the algorithm is to add columns sequentially to an existing design. The sequential operation is adopted for speed and simplicity. This operation avoids an exhaustive search of columns for improvement, which could be complex and inefficient in computation. The two operations when adding a column are *interchange* and *exchange*. The interchange procedure, also called the pairwise switch, switches a pair of distinct symbols in a column. For each candidate column, the algorithm searches all possible interchanges and makes an interchange that reduces J_2 the most. The interchange procedure is repeated until a lower bound is achieved or until no further improvement is possible. The exchange procedure replaces the candidate column by a randomly selected column. This procedure is allowed to repeat at most T times if no lower bound is achieved. The value of T depends on the orthogonality of the previous design. If the previous design is an OA, then $T = T_1$; otherwise, $T = T_2$, where T_1 and T_2 are two constants controlled by the user. With any specified weights w_1, \dots, w_n , the algorithm constructs an $OA'(N, s_1 \dots s_n)$, in which the first n_0 columns form an $OA(N, s_1 \dots s_{n_0})$.

The algorithm proceeds as follows:

1. For $k = 1, \dots, n$, compute the lower bound $L(k)$ according to (2).

2. Specify an initial design \mathbf{d} with two columns, $(0, \dots, 0, 1, \dots, 1, \dots, s_1 - 1, \dots, s_1 - 1)$ and $(0, \dots, s_2 - 1, 0, \dots, s_2 - 1, \dots, 0, \dots, s_2 - 1)$. Compute $\delta_{i,j}(\mathbf{d})$ and $J_2(\mathbf{d})$ by definition. If $J_2(\mathbf{d}) = L(2)$, then let $n_0 = 2$ and $T = T_1$; otherwise, let $n_0 = 0$ and $T = T_2$.

3. For $k = 3, \dots, n$, do the following:

a. Randomly generate a balanced s_k -level column \mathbf{c} . Compute $J_2(\mathbf{d}_+)$ by (4). If $J_2(\mathbf{d}_+) = L(k)$, go to (d).

b. For all pairs of rows a and b with distinct symbols, compute $\Delta(a, b)$ as in (5). Choose a pair of rows with largest $\Delta(a, b)$ and exchange the symbols in rows a and b of column \mathbf{c} . Reduce $J_2(\mathbf{d}_+)$ by $2w_k\Delta(a, b)$. If $J_2(\mathbf{d}_+) = L(k)$, then go to (d); otherwise, repeat (b) until no further improvement is made.

c. Repeat (a) and (b) T times and choose a column \mathbf{c} that produces the smallest $J_2(\mathbf{d}_+)$.

d. Add column \mathbf{c} as the k th column of \mathbf{d} , let $J_2(\mathbf{d}) = J_2(\mathbf{d}_+)$, and update $\delta_{i,j}(\mathbf{d})$ by (3). If $J_2(\mathbf{d}) = L(k)$, then let $n_0 = k$; otherwise, let $T = T_2$.

4. Return the final $N \times n$ design \mathbf{d} , of which the first n_0 columns form an OA.

This is an example of a columnwise algorithm. As noted by Li and Wu (1997), the advantage of columnwise instead of rowwise operation is that the balance property of a design is retained at each iteration. A simple way of adding an s -level column is to choose a best column from all possible candidate columns. However, it is computationally impossible to enumerate all possible candidate columns if the run size N is not small: $\binom{N}{N/2}$ balanced columns for $s = 2$ and $\binom{N}{N/3}\binom{2N/3}{N/3}$ balanced columns for $s = 3$. The numbers grow exponentially with N ; for example, $\binom{24}{12} = 2,704,156$; $\binom{32}{16} = 601,080,390$; $\binom{18}{12}\binom{12}{6} = 17,153,136$; and $\binom{27}{18}\binom{18}{9} \approx 2.3 \times 10^{11}$. The interchange and exchange procedures used in the algorithm yield a feasible approach to minimizing J_2 for computational efficiency. An interchange operation searches $N^2/4$ columns for $s = 2$, $N^2/3$ columns for $s = 3$, and fewer than $N^2/2$ columns for any s . The interchange procedure usually involves a few (typically less than six) iterations. Compared with the size of all candidate columns, the interchange operation searches only a rather small portion of the whole space. Thus it is an efficient local learning procedure, but often ends up with a local minimum. For this reason, global exchange procedures are also incorporated into the algorithm to allow the search to move around the whole space and not be limited to a small neighborhood. As discussed later, the global exchange procedure with moderate T_1 and T_2 improves the performance of the algorithm tremendously.

The values of T_1 and T_2 determines the speed and performance of the algorithm. A large T_i value allows the algorithm to spend more effort in searching for a good column, which takes more time. The choice of T_1 and T_2 depends on the type of design to be constructed. For constructing OAs, a moderate T_1 , say 100, is recommended and T_2 can be 0; for constructing NOAs, moderate T_1 and T_2 are recommended. More details are given in the next section.

Remark 1. Both interchange and exchange algorithms have been proposed by a number of authors for various purposes. (See Nguyen 1996a and Li and Wu 1997 in the context of constructing supersaturated designs.)

Remark 2. The performance of the algorithm may depend on the order of levels. Experience suggests that it is most effective if all levels are arranged in a decreasing order (i.e., $s_1 \geq s_2 \geq \dots \geq s_n$), because the number of balanced columns is much larger for a higher level than a lower level.

Remark 3. The speed of the algorithm is maximized because only integer operations are required if integral weights are used. For efficiency and flexibility, the algorithm is implemented as a function in C and can be called from S. Both C and S source codes are available from the author on request.

4. PERFORMANCE AND COMPARISON

This section reports the performance and comparison of the algorithm with others for the construction of OAs and NOAs.

4.1 Orthogonal Arrays

In the construction of OAs, the weights can be fixed at $w_i = 1$, and T_2 should be 0 because it is unnecessary to continue adding columns if the current design is not orthogonal. Here the choice of T_1 is studied in more detail, because it determines the speed and performance of the algorithm.

The algorithm is tested with four choices of T_1 : 1, 10, 100 and 1,000. For each OA and T_1 , the algorithm is repeated 1,000 times with different random seeds on a Sun SPARC 400-MHz workstation. It either succeeds or fails in constructing an OA each time. Table 2 shows the success rate and the average time in seconds over 1,000 repetitions. In the construction of a mixed-level OA, as stated in Remark 2, the levels are arranged in a decreasing order. Table 2 shows clearly the trade-off between the success rate and speed, which depends on the choice of T_1 . The success rate increases and the speed decreases as T_1 increases. A good measure is the number of OAs constructed per CPU time. The algorithm is least efficient for $T_1 = 1$ and is more efficient for $T_1 = 10$ or 100 than $T_1 = 1,000$. Overall, the choice of $T_1 = 100$ balances success rate and speed and so is generally recommended.

The construction of OAs continues to be an active research topic since Rao (1947) introduced the concept. Construction methods include combinatorial, geometrical, algebraic, coding theoretic, and algorithmic approaches. State-of-the-art construction of OAs has been described by Hedayat, Sloane, and Stufken (1999). Here the focus is on algorithmic construction and comparison with existing algorithms.

Many exchange algorithms have been proposed for constructing exact D -optimal designs. (For reviews, see Cook and Nachtsheim 1980 and Nguyen and Miller 1992.) These algorithms can be used to construct OAs; however, they are inefficient, and the largest OA constructed and published so far is $OA(12, 2^{11})$ (Galil and Kiefer 1980). By modifying the exchange procedure, Nguyen (1996a) proposed an interchange algorithm for constructing supersaturated designs. His program can be used to construct two-level OAs; the largest OA constructed and published is $OA(20, 2^{19})$.

Global optimization algorithms, including simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983), thresholding accepting (Dueck and Scheuer 1990), and genetic algorithms

Table 2. Performance in the Construction of OAs

Array	$T_1 = 1$		$T_1 = 10$		$T_1 = 100$		$T_1 = 1,000$	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time
OA(9, 3 ⁴)	100.0%	.001	100.0%	.001	100.0%	.001	100.0%	.001
OA(12, 2 ¹¹)	93.6%	.002	93.5%	.002	95.9%	.002	94.6%	.010
OA(16, 8 ¹ 2 ⁹)	2.2%	.002	96.8%	.006	100.0%	.007	100.0%	.007
OA(16, 2 ¹⁵)	7.2%	.002	99.9%	.009	100.0%	.008	100.0%	.008
OA(16, 4 ⁵)	5.6%	.003	15.4%	.013	15.7%	.107	17.5%	.889
OA(18, 3 ⁷ 2 ¹)	.3%	.003	42.9%	.022	82.7%	.051	81.8%	.260
OA(18, 6 ¹ 3 ⁹)	.9%	.004	16.4%	.017	18.6%	.115	16.2%	1.110
OA(20, 2 ¹⁹)	0%	.005	54.9%	.029	63.4%	.090	63.8%	.444
OA(20, 5 ¹ 2 ⁹)	0%	.003	4.4%	.020	32.2%	.107	33.5%	.782
OA(24, 2 ²³)	0%	.009	9.1%	.055	30.4%	.370	28.4%	1.903
OA(24, 4 ¹ 2 ²⁰)	0%	.008	16.5%	.056	45.5%	.309	43.4%	1.609
OA(24, 3 ¹ 2 ¹⁶)	0%	.008	.4%	.049	3.5%	.398	3.3%	2.568
OA(24, 12 ¹ 2 ¹²)	0%	.006	28.1%	.059	98.8%	.104	98.5%	.129
OA(24, 4 ¹ 3 ¹ 2 ¹³)	0%	.007	.4%	.045	5.6%	.383	5.3%	2.507
OA(24, 6 ¹ 4 ¹ 2 ¹¹)	0%	.006	1.2%	.045	10.1%	.327	8.9%	2.339
OA(25, 5 ⁶)	.5%	.009	9.3%	.077	12.0%	.608	10.7%	5.989
OA(27, 9 ¹ 3 ⁹)	0%	.012	10.4%	.106	97.0%	.433	100.0%	.450
OA(27, 3 ¹³)	0%	.013	0%	.091	.2%	.878	.3%	7.782
OA(28, 2 ²⁷)	0%	.015	0%	.078	1.4%	.764	.8%	5.544
OA(32, 16 ¹ 2 ¹⁶)	0%	.017	0%	.144	88.1%	1.079	100.0%	1.158
OA(32, 8 ¹ 4 ² 2 ¹⁸)	0%	.018	1.3%	.134	38.1%	1.364	40.0%	5.644
OA(40, 20 ¹ 2 ²⁰)	0%	.037	0%	.258	8.1%	3.146	68.9%	13.972

NOTE: The entries in the columns are the success rate of constructing an OA and the average time in seconds per repetition.

(Goldberg 1989), may be used to construct OAs. These algorithms often involve a large number of iterations and are very slow to converge. These algorithms have been applied to many hard problems and are documented to be powerful. However, they are not effective in the construction of OAs (Hedayat et al. 1999, p. 337); for example, thresholding accepting algorithms of Fang, Lin, Winker, and Zhang (2000) and Ma, Fang, and Liski (2000) failed to produce any $OA(27, 3^{13})$ or $OA(28, 2^{27})$.

DeCock and Stufken (2000) proposed an algorithm for constructing mixed-level OAs via searching some existing two-level OAs. Their purpose is to construct mixed-level OAs with as many two-level columns as possible, and their algorithm succeeded in constructing several new large mixed-level OAs. In contrast, the purpose in the present article is to construct as many nonisomorphic mixed-level OAs (with small runs) as possible, for which the proposed algorithm is more flexible and effective. For example, the proposed algorithm is quite effective in constructing an $OA(20, 5^{12}8)$ that is known to have maximal two-level columns whereas the algorithm of DeCock and Stufken fails to produce any $OA(20, 5^{12}7)$. Furthermore, the proposed algorithm successfully constructs several new 36-run OAs not constructed by their algorithm. Appendix B lists nine new 36-run OAs.

It is interesting to have some head-to-head timing comparisons between this and other algorithms. A Fedorov exchange algorithm and Nguyen's NOA algorithm are chosen for comparison. Cook and Nachtsheim (1980) reported that the Fedorov exchange algorithm produces the best result but takes the longest CPU time among several D -optimal exchange algorithms. The Fortran source code due to Miller and Nguyen (1994) is used for the Fedorov algorithm, downloaded from StatLib (<http://www.lib.stat.cmu.edu>). The Nguyen algorithm is implemented by replacing $ave(s^2)$ with J_2 for convenience because no source code is available. This modification will

affect the speed, but not the efficiency in constructing OAs. Table 3 shows the comparisons of the algorithms in terms of speed and efficiency. All algorithms were compiled and run on an iMac PowerPC G4 computer for a fair comparison. The iMac computer has a 867-MHz CPU, about three times faster than the Sun workstation described earlier. In the simulation, the Fedorov algorithm was repeated 1,000 times for $OA(12, 2^{11})$ and $OA(16, 2^{15})$ because it is very slow, and other algorithms were repeated 10,000 times for all arrays. Table 3 clearly shows that the proposed algorithm performs the best and the Fedorov algorithm performs the worst in terms of both speed and efficiency. The Fedorov algorithm is slow because it uses an exhaustive search of points for improvement and uses D -optimality as the objective function, which involves real-valued matrix operations. The (modified) Nguyen algorithm is slower than the proposed algorithm, because the former uses a nonsequential approach and the latter uses a sequential approach. The nonsequential approach stops only when no swap is made on any column for consecutive n times (where n is the number of columns), whereas the sequential approach stops when it fails to add an orthogonal column for consecutive T_1 times. When T_1 is less than n , the sequential approach stops earlier in the case of failure. This explains why the sequential approach is faster. Furthermore, the high success rate of the proposed algorithm shows that the sequential approach is more efficient than the nonsequential approach. Note that with the increased computer power, the Fedorov algorithm succeeds in generating some $OA(16, 2^{15})$'s, whereas the Nguyen algorithm still fails to generate any $OA(24, 2^{23})$ in 10,000 repetitions.

4.2 Nearly-Orthogonal Arrays

Wang and Wu (1992) systematically studied NOAs and proposed some general combinatorial construction methods. Nguyen (1996b) proposed an algorithm for constructing

Table 3. Comparison of Algorithms in Terms of Speed and Efficiency

Array	Fedorov		Nguyen		Author ($T_1 = 1$)		Author ($T_1 = 10$)	
	Rate	Time	Rate	Time	Rate	Time	Rate	Time
OA(12, 2 ¹¹)	12.7%	.071	55.76%	.0012	93.52%	.0005	94.88%	.0005
OA(16, 2 ¹⁵)	2.1%	10.189	5.15%	.0054	7.42%	.0008	99.84%	.0026
OA(20, 2 ¹⁹)			.03%	.0151	.05%	.0015	54.26%	.0096
OA(24, 2 ²³)			0%	.0361	0%	.0028	9.59%	.0189

NOTE: The entries in the columns are the success rate of constructing an OA and the average time in seconds per repetition. The Fedorov algorithm is due to Miller and Nguyen (1994), and the Nguyen algorithm is implemented with J_2 -optimality.

NOAs by adding two-level columns to existing OAs. Ma et al. (2000) proposed two algorithms for constructing NOAs by minimizing some combinatorial criteria via the thresholding accepting technique. Here the proposed algorithm is used to construct J_2 -optimal mixed-level NOAs and compare them with others. Table 4 shows the comparison of NOAs in terms of A_2 and D optimality. The arrays are chosen according to A_2 -optimality, that is, J_2 -optimality with natural weights. Of the designs with the same A_2 value, the one with the highest D efficiency is chosen. Orthogonal polynomial contrasts are used to calculate the D efficiency as in (6). The number of nonorthogonal pairs, Np , is also reported for reference. In the construction of an $OA'(N, s_1^{n_1} s_2^{n_2})$, all s_1 -level columns with weight s_1 are entered ahead of s_2 -level columns with weight s_2 . The algorithm is very efficient; most arrays can be obtained within seconds with the choice of $T_1 = T_2 = 100$.

The advantage of the proposed algorithm is clear from Table 4. Among all cases, the arrays from this algorithm have the smallest A_2 value and the largest D efficiency. Among the algorithms, the thresholding accepting algorithms of Ma et al. are most complicated but perform the worst. The poor performance of these algorithms again suggests that global optimization algorithms are not effective in the construction

of OAs and NOAs. The Nguyen arrays are competitive in terms of D and Np ; however, Nguyen's algorithm can construct only a small number of arrays, because it works only with two-level columns. Among the designs listed in Table 4 is a new $OA'(20, 5^1 2^{15})$ that is better than Nguyen's in terms of both A_2 and D . That array has 19 nonorthogonal pairs of columns, and his has 25 nonorthogonal pairs; on the other hand, that array has 7 orthogonal columns (the first 7) and his has 8 orthogonal columns (the first 8). Moreover, two arrays, $OA'(12, 3^1 2^9)$ and $OA'(24, 3^1 2^{21})$, are better than Nguyen's in terms of A_2 . In terms of Np , the arrays $OA'(12, 3^1 2^9)$ and $OA'(20, 5^1 2^{15})$ are better and $OA'(18, 2^1 3^8)$ and $OA'(24, 3^1 2^{21})$ are worse than his. For reference, these arrays are listed in Tables 1, 5, 6, and 7.

The proposed algorithm has an additional feature: Weights can be used to control the structure of NOAs. For example, consider the construction of an $OA'(12, 3^1 2^9)$. If the practitioner is more concerned with a three-level factor, then it is desirable to have an NOA in which the three-level column is orthogonal to all two-level columns. If the practitioner is more concerned with the two-level factors, it is desirable to have an NOA in which all two-level columns are orthogonal to each other. Wang and Wu (1992) referred to such designs as type

Table 4. Comparison of NOAs With Run Size ≤ 24

Array	Wang and Wu			Nguyen			Ma et al.			Author		
	A_2	D	Np	A_2	D	Np	A_2	D	Np	A_2	D	Np
OA'(6, 3 ¹ 2 ³)	.333	.901	3	.333	.901	3	.333	.901	3	.333	.901	3
OA'(10, 5 ¹ 2 ⁵)	.720	.883	10	.400	.967	10	.400	.967	10	.400	.967	10
OA'(12, 4 ¹ 3 ⁴)	.750	.946	6				.750	.946	6	.750	.946	6
OA'(12, 2 ³ 3 ⁴)	.750	.946	6				.750	.946	6	.750	.946	6
OA'(12, 6 ¹ 2 ⁵)	.667	.911	6	.444	.959	4	.778	.911	3	.444	.959	4
OA'(12, 6 ¹ 2 ⁶)				.667	.947	6	.889	.909	4	.667	.947	6
OA'(12, 3 ¹ 2 ⁹)	1.00	.867	9	.889	.933	8	.833	.905	5	.778	.933	6
OA'(12, 2 ¹ 3 ⁵)	1.25	.877	10							1.25	.877	10
OA'(12, 2 ² 3 ²)							.917	.899	6	.861	.909	6
OA'(12, 2 ⁵ 3 ³)							.875	.877	6	.875	.877	6
OA'(15, 5 ¹ 3 ⁵)	.800	.882	10				.800	.882	10	.800	.882	10
OA'(18, 2 ¹ 3 ⁸)				.500	.967	1				.500	.967	3
OA'(18, 3 ² 2 ³)	.333	.970	3	.333	.970	3	.432	.970	7	.333	.970	3
OA'(18, 9 ¹ 2 ⁸)	.346	.985	28	.346	.985	28	.346	.985	28	.346	.985	28
OA'(20, 5 ¹ 2 ¹⁵)	2.16	.838	30	1.00	.922	25	?	.623	14	.760	.925	19
OA'(24, 8 ¹ 3 ⁸)	.875	.897	28				2.24	.845	31	.875	.897	28
OA'(24, 3 ¹ 2 ²¹)	2.33	.853	21	.889	.968	8	.833	.953	14	.722	.968	23
OA'(24, 6 ¹ 2 ¹⁵)	2.00	.870	18	.111	.994	1	1.17	.934	12	.111	.994	1
OA'(24, 6 ² 2 ¹⁸)				.667	.974	6	2.50	.761	18	.667	.974	6
OA'(24, 2 ¹ 3 ¹¹)	2.75	.871	55							2.01	.895	56
OA'(24, 3 ¹ 4 ⁷)	5.44	.594	21							2.56	.858	21

NOTE: Np is the number of pairs of nonorthogonal columns in the design matrix. Wang and Wu arrays are based on the construction method specified in their Section 6.

* The value can not be determined because no design was given by Ma et al.

Table 5. $OA'(20, 5^1 2^{15})$

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0
2	0	1	0	1	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0	1
4	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1
5	1	0	0	1	0	1	1	1	1	1	0	0	0	0	1	1
6	1	1	0	0	0	1	0	1	1	0	1	0	1	1	0	0
7	1	0	1	1	1	0	1	0	0	1	0	1	0	1	0	0
8	1	1	1	0	1	0	0	0	0	0	1	1	1	0	1	1
9	2	0	1	0	0	0	0	0	1	1	0	0	1	0	1	0
10	2	1	0	1	0	0	1	0	1	0	1	1	0	1	1	1
11	2	0	0	0	1	1	1	0	0	0	1	1	0	0	0	0
12	2	1	1	1	1	1	0	1	0	1	1	0	0	1	0	1
13	3	0	1	1	1	1	0	1	1	0	1	1	0	0	1	0
14	3	1	1	1	0	0	1	1	0	0	0	0	1	1	1	0
15	3	0	0	0	1	0	1	0	1	1	0	1	1	0	0	1
16	3	1	0	0	0	1	0	0	0	1	0	1	0	0	0	1
17	4	0	1	1	0	1	0	0	1	0	0	1	1	1	0	1
18	4	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
19	4	0	0	0	1	0	0	1	0	0	0	0	0	1	1	1
20	4	1	0	1	1	1	1	0	0	1	1	0	1	0	1	0

NOTE: The first seven columns form an $OA(20, 5^1 2^6)$. The pairs of columns (2, 11), (3, 12), (4, 13), (4, 14), (4, 15), (4, 16), (5, 9), (6, 8), (6, 14), (6, 15), (6, 16), (7, 10), (8, 14), (8, 15), (8, 16), (13, 14), (13, 15), (13, 16), and (14, 16) are nonorthogonal and have an A_2 value of .04; all other pairs of columns are orthogonal.

I and type II. Using this algorithm, one can easily construct both types of NOAs. For instance, a type I array is obtained if weight 10 is assigned to a three-level column and weight 1 to a two-level column, and a type II array is obtained if the weight assignment is reversed.

5. BLOOD GLUCOSE EXPERIMENT

Consider the blood glucose experiment described in Section 1. The original experiment used an $OA(18, 2^1 3^7)$ by combining two factors, sensitivity (F) and absorption (I), into

one factor. The disadvantage of this plan is obvious, in that the original factor cannot be distinguished if the combined factor is identified as significant. Unfortunately, data analysis of the original experiment suggested that the combined factor was significant. The details of the design matrix, response data, and data analysis have been given by Hamada and Wu (1992) and Wu and Hamada (2000, chaps. 7–8).

An alternative of the previous plan, as mentioned in Section 1, is to use an $OA'(18, 2^1 3^8)$. Table 7 lists two $OA'(18, 2^1 3^8)$'s. The first array, given by Nguyen (1996b),

Table 6. $OA'(24, 3^1 2^{21})$

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	1	1	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	0	1	1
2	0	1	0	0	1	1	1	0	1	0	1	0	0	0	1	0	1	1	1	1	0	1
3	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	1	1
4	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	1	0	1	1	1	0	0
5	0	0	1	1	0	1	0	0	1	0	1	0	0	1	0	1	1	0	0	1	0	0
6	0	1	1	1	0	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	1	1
7	0	0	1	0	1	1	1	0	0	0	0	1	1	0	1	1	0	1	1	1	0	0
8	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	0	0	1	0	1	0	0
9	1	0	0	1	1	1	0	0	0	1	1	1	0	1	1	0	0	1	0	0	1	1
10	1	1	0	1	1	0	0	0	1	0	0	1	0	1	0	1	0	1	1	1	1	0
11	1	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	1	1	0	0	0	0
12	1	1	1	0	0	1	0	1	0	1	1	0	0	0	1	1	0	1	0	1	1	1
13	1	0	0	0	0	1	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0
14	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1	0
15	1	0	1	0	1	0	1	1	1	0	1	0	1	1	0	0	0	0	1	1	0	1
16	1	1	0	1	0	0	1	0	0	0	0	0	1	0	1	1	1	0	0	1	0	1
17	2	0	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	1	1	1
18	2	1	1	0	0	0	1	0	1	1	0	0	1	1	1	0	0	1	0	0	0	0
19	2	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0	0	0	1
20	2	1	1	1	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0
21	2	0	0	1	0	1	0	1	1	0	0	0	1	0	1	0	0	0	1	1	1	0
22	2	1	0	0	0	1	0	1	0	0	1	1	1	1	0	1	1	1	1	0	0	1
23	2	0	0	0	1	0	1	1	0	0	1	1	0	1	1	1	1	0	0	1	1	0
24	2	1	1	0	1	1	0	0	1	1	0	1	1	1	1	0	1	0	0	1	1	1

NOTE: The first 11 columns form an $OA(24, 3^1 2^{10})$. The pair of columns (5, 19) is nonorthogonal and has an A_2 value of .111; the pairs of columns (4, 15), (4, 17), (5, 19), (6, 20), (6, 21), (6, 22), (9, 17), (9, 18), (9, 22), (10, 20), (10, 21), (10, 22), (11, 12), (11, 15), (11, 22), (12, 14), (12, 21), (14, 18), (15, 20), (17, 21), (18, 20), (20, 22), and (21, 22) are nonorthogonal and have an A_2 value of .028; and all other pairs of columns are orthogonal.

Table 7. $OA'(18, 2^1 3^8)$

Nguyen										Author									
Run	1	2	3	4	5	6	7	8	9	Run	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	2	1	0	2	0
2	0	0	1	1	1	1	1	1	0	2	0	1	0	1	2	0	0	1	1
3	0	0	2	2	2	2	2	2	0	3	0	2	2	2	0	1	0	1	2
4	0	1	0	0	1	1	2	2	1	4	0	0	0	2	1	1	1	0	1
5	0	1	1	1	2	2	0	0	1	5	0	1	1	2	0	0	2	0	0
6	0	1	2	2	0	0	1	1	1	6	0	2	0	1	0	2	1	2	0
7	0	2	0	1	0	2	1	2	2	7	0	0	2	1	1	0	2	2	2
8	0	2	1	2	1	0	2	0	2	8	0	1	1	0	1	2	1	1	2
9	0	2	2	0	2	1	0	1	2	9	0	2	2	0	2	2	2	0	1
10	1	0	0	2	2	1	1	0	1	10	1	0	0	2	2	2	2	1	2
11	1	0	1	0	0	2	2	1	1	11	1	1	0	0	0	1	2	2	1
12	1	0	2	1	1	0	0	2	1	12	1	2	1	1	1	1	2	1	0
13	1	1	0	1	2	0	2	1	2	13	1	0	1	1	0	2	0	0	1
14	1	1	1	2	0	1	0	2	2	14	1	1	2	2	1	2	0	2	0
15	1	1	2	0	1	2	1	0	2	15	1	2	1	2	2	0	1	2	1
16	1	2	0	2	1	2	0	1	0	16	1	0	2	0	0	0	1	1	0
17	1	2	1	0	2	0	1	2	0	17	1	1	2	1	2	1	1	0	2
18	1	2	2	1	0	1	2	0	0	18	1	2	0	0	1	0	0	0	2

NOTE: The first eight columns of each design form an $OA(18, 2^1 3^7)$. For Nguyen's array the pair of columns (2, 9) is nonorthogonal and has an A_2 value of .5. For the author's array the pairs of columns (3, 9), (5, 9), and (8, 9) are nonorthogonal and have an A_2 value of .167.

is obtained by adding one column to an $OA(18, 2^1 3^7)$. The second array is constructed by the algorithm proposed here. The comparison of these two NOAs is given in Table 8, along with the original plan of combined factors. Both NOAs are superior to the original plan. Either NOA has a D efficiency of .967, which implies that all main effects can be estimated efficiently. Both NOAs have the same overall nonorthogonality, that is, $A_2 = .5$. The Nguyen array has one pair of nonorthogonal columns, whereas that of the proposed algorithm has three pairs of nonorthogonal columns. However, the nonorthogonal pair of Nguyen's array has only six (among nine) different level combinations, whereas each nonorthogonal pair of the proposed array has all nine level combinations. Consequently, the aliasing between any nonorthogonal pair of the proposed array is one-third of the aliasing between the nonorthogonal pair of Nguyen's array (see the a_2 values in Table 8). Because the experimenter does not know in advance which factors will turn out to be significant, it is important to minimize the maximum aliasing of any two factors. The array of the proposed algorithm has the desired property that the nonorthogonality is uniformly spread among three pairs so that the nonorthogonality of each pair is minimized.

6. CONCLUDING REMARKS

This article proposes an efficient algorithm for constructing mixed-level OAs and NOAs. The basic idea is to add columns sequentially such that the resultant array is optimal

Table 8. Comparison of $OA'(18, 2^1 3^8)$

Array	D	A_2	Np	a_2
Original plan	.000	2.0	1	2.000
Nguyen	.967	.5	1	.500
Author	.967	.5	3	.167

NOTE: Np is the number of pairs of nonorthogonal columns in the design matrix, and a_2 is the maximum aliasing among pairs of nonorthogonal columns.

with respect to some optimality criteria. Here the J_2 -optimality is used for simplicity in programming and efficiency in computation. The algorithm has the following advantages: (a) easy to use for practitioners, (b) flexible for constructing various mixed-level designs, (c) outperforms existing algorithms in both speed and efficiency, and (d) generates several new OAs not found by other algorithms.

The sequential procedure avoids an exhaustive search of columns for improvement and is computationally efficient. As with all other algorithms, this program may be trapped in a local minimum. To overcome this problem, the program should be rerun M times, where M could range from a few to the thousands. Table 2 provides a guideline on how large M should be in the construction of OAs.

Traditionally, OAs are used for estimating main effects only. All OAs are equally good as main-effects plans. For example, Cheng (1980) showed that OAs are universally optimal. Nevertheless, recent advances in analysis strategies show that OAs may entertain some interactions besides the main effects under the effect sparsity principle (Hamada and Wu 1992); hence, all OAs are no longer equivalent. (See Lin and Draper 1992, Wang and Wu 1995, Cheng 1995, Box and Tyssedal 1996, Deng and Tang 1999, Tang and Deng 1999, and Xu and Wu 2001 for classification or discrimination of OAs.) Various purposes of the experiments require different choices of OAs with different projective and other statistical properties. However, few standard OAs are listed in most textbooks. The proposed algorithm is important in this regard, because it can efficiently construct many new OAs with good projective properties. Appendix C lists six 27-run OAs with 5–10 columns, which are new in the sense that they are not equivalent to any existing OAs. Lam and Tonchev (1996) showed that there are exactly 68 nonisomorphic $OA(27, 3^{13})$'s. By considering projections onto three columns, it is shown that all six OAs given in Appendix C are not equivalent to any subdesigns from the 68 $OA(27, 3^{13})$'s. Furthermore, these new OAs have good projective properties. For example, all projected three-column designs from them have at least 18 distinct runs. In particular,

these OAs are better than all designs studied by Cheng and Wu (2001) for their dual purposes of factor screening and response surface exploration.

Finally, data from an experiment using OAs and NOAs can be analyzed by stepwise selection or Bayesian variable selection procedure. Details and examples have been given by Hamada and Wu (1992), Chipman, Hamada, and Wu (1997), and Wu and Hamada (2000, chap. 8).

ACKNOWLEDGMENTS

This article forms part of the author’s Ph.D. thesis at the University of Michigan under the supervision of C. F. Jeff Wu. The author thanks the editor, the associate editor, and the referees for their valuable comments and suggestions. This research was supported by National Science Foundation grant DMS-0072489.

APPENDIX A: PROOFS

Proof of Lemma 1

For $\mathbf{d} = [x_{ik}]_{N \times n}$, let $n_{kl}(a, b) = |\{i : x_{ik} = a, x_{il} = b\}|$. It is easy to verify that

$$\sum_{i=1}^N \sum_{j=1}^N \delta(x_{ik}, x_{jk}) \delta(x_{il}, x_{jl}) = \sum_{a=0}^{s_k-1} \sum_{b=0}^{s_l-1} n_{kl}(a, b)^2.$$

Then

$$\begin{aligned} 2J_2(\mathbf{d}) &= \sum_{i=1}^N \sum_{j=1}^N \left[\sum_{k=1}^n w_k \delta(x_{ik}, x_{jk}) \right]^2 - N \left(\sum_{k=1}^n w_k \right)^2 \\ &= \sum_{i=1}^N \sum_{j=1}^N \left[\sum_{k=1}^n \sum_{l=1}^n w_k \delta(x_{ik}, x_{jk}) w_l \delta(x_{il}, x_{jl}) \right] \\ &\quad - N \left(\sum_{k=1}^n w_k \right)^2 \\ &= \sum_{k=1}^n \sum_{l=1}^n w_k w_l \left[\sum_{i=1}^N \sum_{j=1}^N \delta(x_{ik}, x_{jk}) \delta(x_{il}, x_{jl}) \right] \\ &\quad - N \left(\sum_{k=1}^n w_k \right)^2 \\ &= \sum_{k=1}^n \sum_{l=1}^n w_k w_l \left[\sum_{a=0}^{s_k-1} \sum_{b=0}^{s_l-1} n_{kl}(a, b)^2 \right] - N \left(\sum_{k=1}^n w_k \right)^2 \\ &= \sum_{k=1}^n w_k^2 \left[\sum_{a=0}^{s_k-1} n_{kk}(a, a)^2 \right] \\ &\quad + \sum_{1 \leq k \neq l \leq n} w_k w_l \left[\sum_{a=0}^{s_k-1} \sum_{b=0}^{s_l-1} n_{kl}(a, b)^2 \right] - N \left(\sum_{k=1}^n w_k \right)^2 \\ &\geq \sum_{k=1}^n w_k^2 N^2 / s_k + \sum_{1 \leq k \neq l \leq n} w_k w_l [N^2 / (s_k s_l)] - N \left(\sum_{k=1}^n w_k \right)^2 \\ &= \left(\sum_{k=1}^n N w_k / s_k \right)^2 + \left(\sum_{k=1}^n (s_k - 1) (N w_k / s_k)^2 \right) \\ &\quad - N \left(\sum_{k=1}^n w_k \right)^2, \end{aligned}$$

where the inequality follows from the Cauchy–Schwartz inequality. In particular, the equality holds if and only if all level combinations appear equally often for any pair of columns; that is, \mathbf{d} is an OA.

Proof of Lemma 2

Because A_2 is invariant with respect to the choice of treatment contrasts, it is convenient, as done by Xu and Wu (2001), to use the *complex* contrasts. For $k = 1, \dots, n$, let $[z_{ip}^{(k)}]$ be the standardized contrast coefficients of the k th factor such that

$$\sum_{i=1}^N z_{ip}^{(k)} = 0 \quad \text{and} \quad \sum_{i=1}^N z_{ip}^{(k)} \overline{z_{iq}^{(k)}} = \delta(p, q)$$

for any $p, q = 1, \dots, s_k - 1$, where \bar{z} is the complex conjugate of z . In particular, the complex contrasts

$$z_{ip}^{(k)} = \xi_k^{px_{ik}} / N$$

are used, where $\xi_k = \exp(\sqrt{-1}2\pi / s_k)$ is a primitive s_k th root of unity in \mathbb{C} . Then

$$\begin{aligned} 2N^2 A_2(\mathbf{d}) &= 2N^2 \sum_{1 \leq k < l \leq n} \sum_{p=1}^{s_k-1} \sum_{q=1}^{s_l-1} \left| \sum_{i=1}^N z_{ip}^{(k)} \overline{z_{iq}^{(l)}} \right|^2 \\ &= \sum_{1 \leq k \neq l \leq n} \sum_{p=1}^{s_k-1} \sum_{q=1}^{s_l-1} \left| \sum_{i=1}^N \xi_k^{px_{ik}} \xi_l^{qx_{il}} \right|^2 \\ &= \sum_{1 \leq k \neq l \leq n} \sum_{p=1}^{s_k-1} \sum_{q=1}^{s_l-1} \sum_{i=1}^N \xi_k^{px_{ik}} \xi_l^{qx_{il}} \sum_{j=1}^N \overline{\xi_k^{px_{jk}}} \overline{\xi_l^{qx_{jl}}} \\ &= \sum_{i=1}^N \sum_{j=1}^N \sum_{1 \leq k \neq l \leq n} \sum_{p=1}^{s_k-1} \xi_k^{p(x_{ik} - x_{jk})} \sum_{q=1}^{s_l-1} \xi_l^{q(x_{il} - x_{jl})} \\ &= \sum_{i=1}^N \sum_{j=1}^N \sum_{1 \leq k \neq l \leq n} [s_k \delta(x_{ik}, x_{jk}) - 1] [s_l \delta(x_{il}, x_{jl}) - 1] \\ &= \sum_{i=1}^N \sum_{j=1}^N \left[\left(\sum_{k=1}^n s_k \delta(x_{ik}, x_{jk}) \right)^2 - \sum_{k=1}^n [s_k \delta(x_{ik}, x_{jk})]^2 \right. \\ &\quad \left. - 2(n-1) \sum_{k=1}^n s_k \delta(x_{ik}, x_{jk}) + n(n-1) \right] \\ &= \sum_{1 \leq i \neq j \leq N} \left(\sum_{k=1}^n s_k \delta(x_{ik}, x_{jk}) \right)^2 + N \left(\sum_{k=1}^n s_k \right)^2 \\ &\quad - \sum_{k=1}^n [s_k^2 + 2(n-1)s_k] \sum_{i=1}^N \sum_{j=1}^N \delta(x_{ik}, x_{jk}) \\ &\quad + N^2 n(n-1). \end{aligned}$$

Note that $\sum_{i=1}^N \sum_{j=1}^N \delta(x_{ik}, x_{jk}) = N^2 / s_k$, because \mathbf{d} is balanced. Then

$$\begin{aligned} 2N^2 A_2(\mathbf{d}) &= 2J_2(\mathbf{d}) + N \left(\sum_{k=1}^n s_k \right)^2 \\ &\quad - \sum_{k=1}^n [s_k^2 + 2(n-1)s_k] N^2 / s_k + N^2 n(n-1) \\ &= 2J_2(\mathbf{d}) + N \left(\sum_{k=1}^n s_k \right)^2 - N^2 \sum_{k=1}^n s_k - N^2 n(n-1). \end{aligned}$$

[Received August 2000. Revised March 2002.]

REFERENCES

- Box, G. E. P., and Tyssedal, J. (1996), "Projective Properties of Certain Orthogonal Arrays," *Biometrika*, 83, 950–955.
- Cheng, C. S. (1980), "Orthogonal Arrays With Variable Numbers of Symbols," *The Annals of Statistics*, 8, 447–453.
- (1995), "Some Projection Properties of Orthogonal Arrays," *The Annals of Statistics*, 23, 1223–1233.
- Cheng, S.-W., and Wu, C. F. J. (2001), "Factor Screening and Response Surface Exploration" (with discussion), *Statistica Sinica*, 11, 553–604.
- Chipman, H., Hamada, M., and Wu, C. F. J. (1997), "A Bayesian Variable-Selection Approach for Analyzing Designed Experiments With Complex Aliasing," *Technometrics*, 39, 372–381.
- Cook, R. D., and Nachtsheim, C. J. (1980), "A Comparison of Algorithms for Constructing Exact D -Optimal Designs," *Technometrics*, 22, 315–324.
- DeCook, D., and Stufken, J. (2000), "On Finding Mixed Orthogonal Arrays of Strength 2 With Many 2-Level Factors," *Statistics and Probability Letters*, 50, 383–388.
- Deng, L. Y., and Tang, B. (1999), "Generalized Resolution and Minimum Aberration Criteria for Plackett–Burman and Other Nonregular Factorial Designs," *Statistica Sinica*, 9, 1071–1082.
- Dueck, G., and Scheuer, T. (1990), "Threshold Accepting: A General Purpose Algorithm Appearing Superior to Simulated Annealing," *Journal of Computational Physics*, 90, 161–175.
- Eccleston, J. A., and Hedayat, A. (1974), "On the Theory of Connected Designs: Characterization and Optimality," *The Annals of Statistics*, 2, 1238–1255.
- Fang, K.-T., Lin, D. K. J., Winker, P., and Zhang, Y. (2000), "Uniform Design: Theory and Application," *Technometrics*, 42, 237–248.
- Galil, Z., and Kiefer, J. (1980), "Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D -Optimum Designs," *Technometrics*, 22, 301–313.
- Goldberg, D. E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading, MA: Addison-Wesley.
- Hamada, M. S., and Wu, C. F. J. (1992), "Analysis of Designed Experiments With Complex Aliasing," *Journal of Quality Technology*, 24, 130–137.
- Hedayat, A. S., Sloane, N. J. A., and Stufken, J. (1999), *Orthogonal Arrays: Theory and Applications*, New York: Springer-Verlag.
- Kirkpatrick, S., Gelatt, C. D., Jr., and Vecchi, M. P. (1983), "Optimization by Simulated Annealing," *Science*, 220, 671–680.
- Lam, C., and Tonchev, V. D. (1996), "Classification of Affine Resolvable 2-(27,9,4) Designs," *Journal of Statistical Planning and Inference*, 56, 187–202.
- Li, W. W., and Wu, C. F. J. (1997), "Columnwise-Pairwise Algorithms With Applications to the Construction of Supersaturated Designs," *Technometrics*, 39, 171–179.
- Lin, D. K. J. (1993), "Another Look at First-Order Saturated Designs: The p -Efficient Designs," *Technometrics*, 35, 284–292.
- Lin, D. K. J., and Draper, N. R. (1992), "Projection Properties of Plackett and Burman Designs," *Technometrics*, 34, 423–428.
- Ma, C.-X., Fang, K.-T., and Liski, E. (2000), "A New Approach in Constructing Orthogonal and Nearly Orthogonal Arrays," *Metrika*, 50, 255–268.
- Miller, A. J., and Nguyen, N.-K. (1994), "A Fedorov Exchange Algorithm for D -Optimal Design," *Applied Statistics*, 43, 669–677.
- Myers, R. H., and Montgomery, D. C. (1995), *Response Surface Methodology: Process and Product in Optimization Using Designed Experiments*, New York: Wiley.
- Nguyen, N.-K. (1996a), "An Algorithmic Approach to Constructing Supersaturated Designs," *Technometrics*, 38, 69–73.
- (1996b), "A Note on the Construction of Near-Orthogonal Arrays With Mixed Levels and Economic Run Size," *Technometrics*, 38, 279–283.
- (2001), "Cutting Experimental Designs into Blocks," *Australian & New Zealand Journal of Statistics*, 43, 367–374.
- Nguyen, N.-K., and Miller, A. J. (1992), "A Review of Some Exchange Algorithms for Constructing Discrete D -Optimal Designs," *Computational Statistics and Data Analysis*, 14, 489–498.
- Rao, C. R. (1947), "Factorial Experiments Derivable from Combinatorial Arrangements of Arrays," *Journal of the Royal Statistical Society, Supplement*, 9, 128–139.
- Tang, B., and Deng, L. Y. (1999), "Minimum G_2 -Aberration for Non-regular Fractional Factorial Designs," *The Annals of Statistics*, 27, 1914–1926.
- Wang, J. C., and Wu, C. F. J. (1992), "Nearly Orthogonal Arrays With Mixed Levels and Small Runs," *Technometrics*, 34, 409–422.
- (1995), "A Hidden Projection Property of Plackett–Burman and Related Designs," *Statistica Sinica*, 5, 235–250.
- Wu, C. F. J., and Hamada, M. S. (2000), *Experiments: Planning, Analysis and Parameter Design Optimization*, New York: Wiley.
- Xu, H., and Wu, C. F. J. (2001), "Generalized Minimum Aberration for Asymmetrical Fractional Factorial Designs," *The Annals of Statistics*, 29, 549–560.