Title
An Algorithm for Constructing Orthogonal and Nearly Orthogonal Arrays with Mixed Levels and Small Runs

Author
Xu, Hongquan

Publication Date
2000

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

Hongquan Xu

Department of Statistics

University of Michigan

Ann Arbor, MI 48109-1285

(hqxu@umich.edu)

Orthogonal arrays are widely used in manufacturing and high-tech 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. It can construct a variety of small-run arrays with good statistical properties efficiently.

KEY WORDS: Exchange algorithm; Generalized minimum aberration; Interchange algorithm; Minimum moment aberration.

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 H_2O level (20, 28, or 35 ml), (D) centrifuge RPM (2100, 2300, 2500), (E) centrifuge time (1.75, 3, 4.5 min), (F) sensitivity (0.10, 0.25, 0.50), (G) temperature (25, 30, 37 ^{0}C), (H) dilution ratio (1:51, 1:101, 1:151), and (I) absorption (2.5, 2, 1.5). To ensure that all the main effects are estimated clearly from each other, 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 widely used in manufacturing and high-tech 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 can be found in Wang and Wu (1992), Nguyen (1996b) and the references therein.

Formally, an orthogonal array (OA) (of strength two), denoted by $OA(N, s_1 \cdots s_n)$, is an $N \times n$ matrix of which the ith column has s_i levels and for any two columns all of their level combinations appear equally often. It is said to be *mixed* if not all s_i's are equal. For convenience, an $OA(N, s_1^{k_1} \cdots s_r^{k_r})$ has k_i columns with s_i levels. A nearly orthogonal array (NOA), denoted by $OA'(N, s_1 \cdots s_n)$, is optimal under some optimality criterion (which to be defined later). From estimation point of view, all the main effects of an OA are estimable and orthogonal to each other while all the main effects of an NOA are still estimable but some of them are partially aliased with others. Because balance is a desired and important property in practice, we here only consider balanced $OA'(N, s_1 \cdots s_n)$ 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 we freely exchange the words of array and design, row and run, and column and factor.

The purpose of this paper is to present a unified and effective algorithm for constructing OAs and NOAs with mixed levels and small runs. The algorithm can construct a variety of arrays with good statistical properties efficiently. The paper is organized as follows. Section 1 introduces the concept of J_2-optimality and other optimality criteria. Section 2 describes an algorithm for constructing mixed-level OAs and NOAs. The performance and comparison of the algorithm are given in Section 3. The blood glucose experiment is revisited in Section 4 and concluding remarks are given in Section 5.

1 Optimality Criteria

A combinatorial criterion, J_2-optimality, is introduced in Section 1.1. It has the advantage of convenience for programming and efficiency for computation. The statistical justification of J_2-optimality and other optimality criteria are given in Section 1.2.

1.1 The Concept of J_2-Optimality

For an $N \times n$ matrix $\mathbf{d} = [x_{ik}]$, of which the kth column has s_k levels, assign weight $w_k > 0$ for column k. For $1 \le i, j \le N$, let $\delta_{i,j}(\mathbf{d}) = \sum_{k=1}^{n} w_k \delta(x_{ik}, x_{jk})$, where $\delta(x, y) = 1$ if $x = y$ and 0 otherwise. Define

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

A design is called J_2-*optimal* if it minimizes J_2. The J_2 criterion can be used to measure the nonorthogonality of a design because of the following inequality.

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

where the equality holds if and only if D is an OA. The proof is given in the Appendix. Therefore, an OA is J_2-optimal with any choice of weights if it exists while an NOA under J_2-optimality may depend on the choice of weights. The J_2-optimality is a special case of the *minimum moment aberration* proposed by the author for assessing nonregular designs and supersaturated designs. The concept and theory of minimum moment aberration will appear elsewhere.

Now consider the change in the J_2 value if a column is added to $\mathbf{d}$ or a pair of symbols are switched in a column. First consider adding a column to $\mathbf{d}$. Suppose column $\mathbf{c} = (c_1, \ldots, c_N)'$ is added to $\mathbf{d}$ and let $\mathbf{d}_+$ be the new $N \times (n+1)$ design. If $\mathbf{c}$ has s_p levels and weight w_p, then

$$\delta_{i,j}(\mathbf{d}_+) = \delta_{i,j}(\mathbf{d}) + \delta_{i,j}(\mathbf{c}) \text{ for } 1 \le i, j \le N, \quad (2)$$

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

$$J_2(\mathbf{d}_+) = J_2(\mathbf{d}) + 2 \sum_{1 \le i < j \le N} \delta_{i,j}(\mathbf{d})\delta_{i,j}(\mathbf{c}) + 2^{-1} N w_p^2 (N s_p^{-1} - 1). \quad (3)$$

Next consider switching a pair of symbols in the added column. Suppose the symbols in rows a and b of the added column are distinct, i.e., $c_a \ne c_b$. If these two symbols are exchanged, 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 $j \ne a, b$. Hence, $J_2(\mathbf{d}_+)$ is reduced by $2S(a,b)$, where

$$S(a,b) = -\sum_{1 \le j \ne a,b \le N} [\delta_{a,j}(\mathbf{d}) - \delta_{b,j}(\mathbf{d})][\delta_{a,j}(\mathbf{c}) - \delta_{b,j}(\mathbf{c})]. \quad (4)$$

These formulas provide an efficient way of updating the J_2 value and will be used in the algorithm.

1.2 Other Optimality Criteria and Statistical Justification of J_2-Optimality

In order to state the statistical justification of J_2-optimality, it is necessary to introduce 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 $+1$; 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, respectively.

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

$$Y = \beta_0 I + X_1 \beta_1 + \varepsilon,$$

where Y is the vector of N observations, β_0 is the general mean, β_1 is the vector of all the main effects, I is the vector of ones, X_1 is the matrix of contrast coefficients for β_1, and ε is the vector of independent random errors. Let $X_1 = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ and $X = (\mathbf{x}_1/\|\mathbf{x}_1\|, \ldots, \mathbf{x}_m/\|\mathbf{x}_m\|)$, where $m = \sum(s_i - 1)$. A design is said to be *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 following D criterion

$$D = |X'X|^{1/m} \tag{5}$$

to measure the overall efficiency of an NOA.

It is well known in the optimum design theory that a good surrogate of the D criterion is the (M,S) criterion (Eccleston and Hedayat 1974). A design is called (M,S)*-optimal* if it maximizes $\operatorname{tr}(X'X)$ and minimizes $\operatorname{tr}[(X'X)^2]$ among those which maximize $\operatorname{tr}(X'X)$. The (M,S) criterion is cheaper in computation than the D criterion. In addition, the (M,S) criterion is independent of the coding of the treatment contrast while the D criterion does. It is easy to verify that for a balanced design, all diagonal elements of $X'X$ are ones. Then the (M,S) criterion reduces to the minimization of $\operatorname{tr}[(X'X)^2]$, 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 leads to the following concept of A_2-optimality.

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

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

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. A design is called A_2-*optimal* if it minimizes A_2. It is a good optimality criterion of NOAs because $A_2 = 0$ if and only if $\mathbf{d}$ is an OA. It is worth to note that the A_2-optimality is a special case of the *generalized minimum aberration* criterion proposed by Xu and Wu (2001) for assessing nonregular designs.

Now we can state the statistical justification of the J_2-optimality. The following identity relates the J_2 criterion to the A_2 criterion. If $w_k = s_k$ for all k, then

$$J_2(\mathbf{d}) = N^2 A_2(\mathbf{d}) + 2^{-1} N \left[Nn(n-1) + N \sum s_k - \left(\sum s_k \right)^2 \right]. \tag{6}$$

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

An obvious advantage of the J_2 criterion over A_2, (M, S) and D criterion is that J_2 is cheaper in computation and easier for programming because it avoids the coding of treatment contrasts. An additional advantage of the J_2 criterion is that it is more flexible because it uses weights to control the structure of an NOA. Illustration of this advantage is given in the end of Section 3.2.

2 An Algorithm

The basic idea of the algorithm is to sequentially add columns to an existing design. The sequential operation is adopted for speed and simplicity. It avoids an exhaustive search of columns for improvement, which could be complex and inefficient in computation. There are two types of operations when adding a column: *interchange* and *exchange*. The interchange procedure, also called 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 which reduces J_2 most. The interchange procedure is repeated until a lower bound is achieved or no further improvement is possible. On the other hand, the exchange procedure replaces an existing 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 current design. If the current design is an OA, $T = T_1$; otherwise, $T = T_2$, where T_1 and T_2 are two constants controlled by the user. With any specified weights $w_1, \ldots, w_n$, the algorithm constructs an $OA'(N, s_1 \cdots s_n)$, of which the first n_0 columns form an $OA(N, s_1 \ldots s_{n_0})$.

The algorithm is given as follows:

1. For $p = 1, \ldots, n$, compute the lower bound $L(p)$ according to (1).

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

3. For $p = 3, \ldots, n$, do the following:

   (a) Randomly generate a balanced s_p-level column $\mathbf{c}$. Compute $J_2(\mathbf{d}_+)$ by (3). If $J_2(\mathbf{d}_+) = L(p)$, go to (d).

   (b) For all pairs of rows a and b with distinct symbols, compute $S(a, b)$ as in (4). Choose a pair of rows with largest $S(a, b)$ and exchange the symbols in rows a and b of column $\mathbf{c}$. Reduce $J_2(\mathbf{d}_+)$ by $2S(a, b)$. If $J_2(\mathbf{d}_+) = L(p)$, go to (d); otherwise, repeat this procedure 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 pth column of $\mathbf{d}$, let $J_2(\mathbf{d}) = J_2(\mathbf{d}_+)$ and update $\delta_{i,j}(\mathbf{d})$ by (2). If $J_2(\mathbf{d}) = L(p)$, let $n_0 = p$; 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 column-wise algorithm. As noted by Li and Wu (1997), the advantage of column-wise instead of row-wise 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. There are in total $\binom{N}{N/2}$ balanced columns for $s = 2$ and $\binom{N}{N/3}\binom{2N/3}{N/3}$ balanced columns for $s = 3$. The number grows 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 $\binom{27}{18}\binom{18}{9} \approx 2.8 \times 10^{11}$. The interchange and exchange procedures used in the algorithm are a feasible approach for both computation and efficiency. An interchange operation searches $N^2/4$ columns for $s = 2$, $N^2/3$ columns for $s = 3$ and less than $N^2/2$ columns for any s. The interchange procedure usually involves a few (typically less than 6) iterations. Compared to the size of all candidate columns, the interchange operation only searches 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 particular reason, global exchange procedures are also incorporated in the algorithm. It allows the search to move around the whole space and not limited to a small neighborhood. As will be seen later, the global exchange procedure with moderate T_1 and T_2 improves the performance of the algorithm tremendously.

The value of T_1 and T_2 controls the speed and performance of the algorithm. A large T_i value allows the algorithm to spend more effort in searching 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 zero; 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. Our experience suggests that it is most effective if all levels are arranged in a decreasing order, that is, $s_1 \geq s_2 \geq \ldots \geq s_n$. The reason is that 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 language and can be called in S language. Both C and S source codes are available from the author.

3 Performance and Comparison

The performance and comparison of the algorithm are reported in two parts for the construction of OAs and NOAs.

3.1 Orthogonal Arrays

In the construction of OAs, the weights can be fixed at $w_i = 1$, and T_2 should be zero because it is unnecessary to continue adding columns if the current design is not orthogonal. Here we study the choice of T_1 in more details since it controls the performance of the algorithm.

The algorithm is tested with four choices of T_1: $1, 10, 100$ and 1000. For each specified OA and T_1, the algorithm is repeated for 1000 times. It either succeeds or fails in constructing an OA each time. Table 1 shows the total number of OAs constructed and the average time in seconds per repetition. The test was run on a Sun sparc workstation of CPU 400MHz. In the construction of a mixed-level OA, as stated in Remark 2, the levels are arranged in a decreasing order. Table 1 clearly shows that there is a tradeoff 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 more efficient for $T_1 = 10$ or 100 than $T_1 = 1000$. Overall, the choice of $T_1 = 100$ gives a good tradeoff between success rate and speed; therefore, it is generally recommended.

The construction of OAs has been and will continue to be an active research topic since Rao (1947) introduced the concept. The construction methods include combinatorial, geometrical, algebraic, coding theoretic, and algorithmic approaches. For the state-of-art in the construction of OAs, see Hedayat, Sloane, and Stufken (1999). Here we focus on algorithmic construction and compare existing algorithms with ours.

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

Global optimization algorithms, such as simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983), thresholding accepting (Dueck and Scheuer 1990), and genetic algorithms (Goldberg 1989), may be used for constructing OAs. These algorithms often involve a large number of iterations and are very slow in convergence. These algorithms have been applied to many hard problems and are documented to be powerful. However, in the construction of OAs, they are not effective (Hedayat et al. 1999, p. 337). For example, those thresholding accepting algorithms in 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. Our purpose differs from theirs and is to construct all possible mixed-level OAs (with small runs). In terms of constructing small-run OAs, our algorithm is more flexible and effective than theirs. For example, our algorithm is quite effective in constructing an $OA(20, 5^1 2^8)$ which is known to have maximal two-level columns while their algorithm failed to construct an $OA(20, 5^1 2^7)$. In addition, our algorithm has succeeded in constructing several new 36-run OAs, which are not constructed by their algorithm. For instance, $OA(36, 6^2 3^2 2^6)$, $OA(36, 6^2 3^3 2^4)$, $OA(36, 6^1 3^3 2^8)$, and $OA(36, 6^1 3^4 2^6)$ appear to be new and are given in Tables 2 and 3.

3.2 Nearly Orthogonal Arrays

Wang and Wu (1992) systematically studied NOAs and proposed some general combinatorial construction methods. Nguyen (1996b) proposed an algorithm for constructing 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. Our algorithm is compared with Wang and Wu's combinatorial methods and Nguyen and Ma et al.'s algorithms. The comparison is summarized in Table 4. All arrays are chosen according to the A_2-optimality, i.e., the J_2-optimality with natural weights. Among 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 (5). The number of nonorthogonal pairs 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. Our algorithm is very effective and most arrays can be obtained within seconds with the choice of $T_1 = T_2 = 100$.

The advantage of our algorithm is clear from Table 4. Among all cases, our arrays have the smallest A_2 value and the largest D efficiency, in addition to that our algorithm can construct more arrays. Among the algorithms, the thresholding accepting algorithms of Ma et al. are most complicated but perform the worst. The poor performance of their algorithms again suggests that global optimization algorithms are not effective in the construction of OAs and NOAs. Nguyen's algorithm can only construct a small number of arrays due to its limitation. Among his 12 arrays listed in Table 4, two arrays, $OA'(12, 3^1 2^9)$ and $OA'(24, 3^1 2^{21})$, are worse than ours in terms of A_2;

and one array, $OA'(20, 5^1 2^{15})$, is worse than ours in terms of both A_2 and D. For reference, our arrays are listed in Tables 5, 6, and 7, respectively.

Our algorithm has an additional feature, that is, 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 the three-level factor, it is desirable to construct an NOA of which the three-level column is orthogonal to any two-level column. On the other hand, if the practitioner is more concerned with the two-level factors, it is desirable to construct an NOA of which all two-level columns are orthogonal to each other. Such designs are referred to type I and type II, respectively, by Wang and Wu (1992). Both types of NOAs can be easily constructed by the algorithm with a proper choice of weights. For instance, a type I array can be obtained by assigning weight 10 to a three-level column and weight 1 to a two-level column and a type II array can be obtained by reversing the weight assignment.

4 Blood Glucose Experiment

Consider the blood glucose experiment described in the preface. The original experiment used an $OA(18, 2^1 3^7)$ by combining two factors (F) sensitivity and (I) absorption into one factor. The disadvantage of this plan is obvious, that is, there is no way to distinguish which original factor is significant if the combined factor is identified to be 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 are given in Hamada and Wu (1992) and Wu and Hamada (2000, chap 7-8).

An alternative of the previous plan, as mentioned in the preface, is to use an $OA'(18, 2^1 3^8)$. Table 8 lists two $OA'(18, 2^1 3^8)$'s. The first array, given in Nguyen (1996b), is obtained by adding one column to an $OA(18, 2^1 3^7)$. The second array is constructed by the algorithm. The comparison of these two NOAs is given in Table 9. For reference, the original plan of combined factors is also included in Table 9.

Table 9 clearly shows that both NOAs are superior to the original plan. Either NOA has a D efficiency of 0.967, which implies that all main effects can be estimated efficiently. Both NOAs have the same overall nonorthogonality, i.e., $A_2 = 0.5$. Nguyen array has one pair of nonorthogonal columns and our array has three pairs of nonorthogonal columns. On the other hand, the nonorthogonal pair of Nguyen array has only six (among nine) different level combinations and each nonorthogonal pair of our array has all nine level combinations. As a consequence, the aliasing between any nonorthogonal pair of our array is one third of the aliasing between the nonorthogonal pair of Nguyen array. Because, when planning the experiment, the experimenter does not know which factors will turn out to be significant, it is important to minimize the maximum aliasing of any two factors. Our array has the desired property, that is, the nonorthogonality is uniformly spread among three pairs so that the nonorthogonality of each pair is minimized. Therefore, our array is preferred to Nguyen's.

5 Concluding Remarks

This paper proposes an efficient algorithm for constructing mixed-level OAs and NOAs. The basic idea is to sequentially adding columns such that the resultant array is optimal with respect to some optimality criteria. In particular, the J_2-optimality is used for simplicity in programming and efficiency in computation. The algorithm has the following advantages: (a) it is easy of use for practitioners, (b) it is flexible for constructing various mixed-level arrays, and (c) it outperforms existing algorithms in both speed and efficiency.

The sequential procedure avoids an exhaustive search of columns for improvement and is computationally efficient. As all other algorithms do, this program may end up with a local minimum. To overcome this problem, it is necessary to rerun the program M times, where M ranges from a few to the thousands.

Traditionally, OAs are used for estimating main effects only. As main effect plans, all OAs are equally good. For example, Cheng (1980) showed that OAs are universally optimal. Nevertheless, recent advance in analysis strategies shows that OAs may entertain some interactions besides the main effects under the effect sparsity principle (Hamada and Wu 1992). For this very reason, not all OAs are equivalent any more. 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 construct many new OAs with different projective property efficiently. For example, Cheng and Wu (2001) constructed an $OA(27, 3^8)$ via an ad hoc combinatorial method. They showed that the new array is much superior to commonly used ones for their dual purposes of factor screening and response surface exploration. It is not surprising that our algorithm can construct many $OA(27, 3^8)$'s that are superior to their array. The algorithm has produced many other new OAs with good statistical properties, which will be reported elsewhere.

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

Acknowledgments

This paper forms part of the author's Ph.D. thesis at the University of Michigan under the supervision of Professor C. F. Jeff Wu. The author is grateful to the Editor and an Associate Editor for their valuable comments. This research was supported by NSF Grant DMS-0072489.

Appendix: Proofs

Proof of (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

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

$$\geq \sum_{k=1}^{n} w_k^2 N^2/s_k + \sum_{1\leq k\neq l\leq n} w_k w_l [N^2/(s_k s_l)] - N\left(\sum_{k=1}^{n} w_k\right)^2$$
$$= \left(\sum_{k=1}^{n} N w_k/s_k\right)^2 + \left(\sum_{k=1}^{n}(s_k-1)(Nw_k/s_k)^2\right) - N\left(\sum_{k=1}^{n} w_k\right)^2,$$

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, i.e., **d** is an OA. □

Proof of (6). Because A_2 is invariant with respect to the choice of treatment contrasts, it is convenient, as done in Xu and Wu (2001), to use the *complex* contrasts. For $k = 1, \ldots, n$, let $[z_{ip}^{(k)}]$ be the standardized contrast coefficients of the kth factor such that

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

for any $p, q = 1, \ldots, s_k - 1$, where $\overline{z}$ is the complex conjugate of z. In particular, we use the following complex contrasts

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

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

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

$$- \sum_{k=1}^{n} [s_k^2 + 2(n-1)s_k] \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(x_{ik}, x_{jk}) + N^2 n(n-1).$$

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,

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

□

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," *Annals of Statistics*, 8, 447–453.

Cheng, C. S. (1995), "Some Projection Properties of Orthogonal Arrays," *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*, to appear.

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.

DeCock, 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," *Statistics 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," *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.

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., 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.

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.

Nguyen, N.-K. (1996b), "A Note on the Construction of Near-Orthogonal Arrays with Mixed Levels and Economic Run Size," *Technometrics*, 38, 279–283.

Nguyen, N.-K., 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*, Supplementary, 9, 128–139.

Tang, B., and Deng, L. Y. (1999), "Minimum G_2-Aberration for Non-regular Fractional Factorial Designs," *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.

Wang, J. C., and Wu, C. F. J. (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," *Annals of Statistics*, to appear.

Table 1: Performance in the Construction of OAs

	$T_1 = 1$		$T_1 = 10$		$T_1 = 100$		$T_1 = 1000$	
Array	No.	Time	No.	Time	No.	Time	No.	Time
$OA(9, 3^4)$	1000	0.001	1000	0.001	1000	0.001	1000	0.001
$OA(12, 2^{11})$	936	0.002	935	0.002	959	0.002	946	0.010
$OA(16, 8^1 2^8)$	22	0.002	968	0.006	1000	0.007	1000	0.007
$OA(16, 2^{15})$	72	0.002	999	0.009	1000	0.008	1000	0.008
$OA(16, 4^5)$	56	0.003	154	0.013	157	0.107	175	0.889
$OA(18, 3^7 2^1)$	3	0.003	429	0.022	827	0.051	818	0.260
$OA(18, 6^1 3^6)$	9	0.004	164	0.017	186	0.115	162	1.110
$OA(20, 2^{19})$	0	0.005	549	0.029	634	0.090	638	0.444
$OA(20, 5^1 2^8)$	0	0.003	44	0.020	322	0.107	335	0.782
$OA(24, 2^{23})$	0	0.009	91	0.055	304	0.370	284	1.903
$OA(24, 4^1 2^{20})$	0	0.008	165	0.056	455	0.309	434	1.609
$OA(24, 3^1 2^{16})$	0	0.008	4	0.049	35	0.398	33	2.568
$OA(24, 12^1 2^{12})$	0	0.006	281	0.059	988	0.104	985	0.129
$OA(24, 4^1 3^1 2^{13})$	0	0.007	4	0.045	56	0.383	53	2.507
$OA(24, 6^1 4^1 2^{11})$	0	0.006	12	0.045	101	0.327	89	2.339
$OA(25, 5^6)$	5	0.009	93	0.077	120	0.608	107	5.989
$OA(27, 9^1 3^9)$	0	0.012	104	0.106	970	0.433	1000	0.450
$OA(27, 3^{13})$	0	0.013	0	0.091	2	0.878	3	7.782
$OA(28, 2^{27})$	0	0.015	0	0.078	14	0.764	8	5.544
$OA(32, 16^1 2^{16})$	0	0.017	0	0.144	881	1.079	1000	1.158
$OA(32, 8^1 4^2 2^{18})$	0	0.018	13	0.134	381	1.364	400	5.644
$OA(40, 20^1 2^{20})$	0	0.037	0	0.258	81	3.146	689	13.972

NOTE: For each array and each T_1, the algorithm is repeated for 1000 times. The entries in the columns show the total number of OAs constructed and the average time in seconds per repetition.

Table 2: $OA(36, 6^2 3^2 2^6)$ and $OA(36, 6^2 3^3 2^4)$

Run	1	2	3	4	5	6	7	8	9	10
1	0	0	0	2	0	0	1	0	0	1
2	0	1	2	1	1	1	1	0	0	0
3	0	2	1	1	1	0	1	0	1	0
4	0	3	0	0	0	0	0	1	1	0
5	0	4	1	0	0	1	0	1	0	1
6	0	5	2	2	1	1	0	1	1	1
7	1	0	1	0	1	1	0	0	0	1
8	1	1	0	1	1	0	1	1	0	0
9	1	2	0	1	0	1	0	1	1	1
10	1	3	2	0	1	0	0	0	1	0
11	1	4	2	2	0	0	1	0	1	1
12	1	5	1	2	0	1	1	1	0	0
13	2	0	2	2	1	1	0	0	1	0
14	2	1	1	0	0	0	0	1	1	1
15	2	2	0	2	0	0	0	0	0	0
16	2	3	1	1	1	1	1	1	0	1
17	2	4	2	1	0	1	1	1	1	0
18	2	5	0	0	1	0	1	0	0	1
19	3	0	2	1	0	0	1	1	1	1
20	3	1	0	0	0	1	1	0	1	1
21	3	2	2	0	1	1	0	1	0	0
22	3	3	1	2	0	1	1	0	0	0
23	3	4	0	2	1	0	0	1	0	0
24	3	5	1	1	1	0	0	0	1	1
25	4	0	0	0	1	1	1	1	1	0
26	4	1	1	2	0	1	0	0	1	0
27	4	2	1	2	1	0	1	1	1	1
28	4	3	2	1	0	0	0	0	0	1
29	4	4	0	1	1	1	0	0	0	1
30	4	5	2	0	0	0	1	1	0	0
31	5	0	1	1	0	0	0	1	0	0
32	5	1	2	2	1	0	0	1	0	1
33	5	2	2	0	0	1	1	0	0	1
34	5	3	0	2	1	1	1	1	1	1
35	5	4	1	0	1	0	1	0	1	0
36	5	5	0	1	0	1	0	0	1	0

(i) $OA(36, 6^2 3^2 2^6)$

Run	1	2	3	4	5	6	7	8	9
1	0	0	1	2	0	0	0	0	1
2	0	1	1	0	2	1	1	0	1
3	0	2	2	2	1	0	1	0	0
4	0	3	0	0	1	1	0	1	0
5	0	4	2	1	0	1	0	1	1
6	0	5	0	1	2	0	1	1	0
7	1	0	0	0	1	0	0	0	1
8	1	1	2	1	1	1	0	0	1
9	1	2	1	2	0	1	1	1	0
10	1	3	0	2	2	1	1	1	1
11	1	4	1	0	0	0	1	0	0
12	1	5	2	1	2	0	0	1	0
13	2	0	1	1	2	1	1	1	1
14	2	1	0	1	1	0	1	0	0
15	2	2	2	0	1	1	1	1	1
16	2	3	1	0	0	0	0	1	0
17	2	4	0	2	2	0	0	0	0
18	2	5	2	2	0	1	0	0	1
19	3	0	2	0	2	1	0	1	0
20	3	1	0	2	0	1	1	1	0
21	3	2	0	0	2	0	0	0	1
22	3	3	2	1	0	0	1	0	1
23	3	4	1	1	1	0	1	1	1
24	3	5	1	2	1	1	0	0	0
25	4	0	0	1	0	1	1	0	0
26	4	1	2	0	0	0	0	1	0
27	4	2	1	1	2	1	0	0	0
28	4	3	2	2	2	0	1	0	1
29	4	4	0	2	1	1	0	1	1
30	4	5	1	0	1	0	1	1	1
31	5	0	2	2	1	0	1	1	0
32	5	1	1	2	2	0	0	1	1
33	5	2	0	1	0	0	0	1	1
34	5	3	1	1	1	1	0	0	0
35	5	4	2	0	2	1	1	0	0
36	5	5	0	0	0	1	1	0	1

(ii) $OA(36, 6^2 3^3 2^4)$

Table 3: $OA(36, 6^1 3^3 2^8)$ and $OA(36, 6^1 3^4 2^6)$

Run	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	1	0	1	1	1	0	1
2	0	1	0	2	1	0	1	0	1	0	0	0
3	0	2	1	1	0	0	1	1	0	1	1	1
4	0	0	2	1	1	0	0	0	1	0	1	1
5	0	1	1	0	0	1	0	0	0	1	1	0
6	0	2	2	2	1	1	1	1	0	0	0	0
7	1	0	1	2	0	0	1	1	1	1	1	0
8	1	1	0	0	0	0	0	0	0	0	0	0
9	1	2	2	1	1	1	0	1	0	1	0	1
10	1	0	1	2	1	1	0	1	1	0	1	0
11	1	1	0	1	1	0	1	0	0	1	1	1
12	1	2	2	0	0	1	1	0	1	0	0	1
13	2	0	2	0	1	1	1	0	0	1	1	0
14	2	1	1	1	1	1	0	1	1	1	0	1
15	2	2	0	2	0	0	0	1	1	0	0	1
16	2	0	2	0	0	0	1	1	0	0	1	1
17	2	1	1	1	1	1	1	0	1	0	0	0
18	2	2	0	2	0	0	0	0	0	1	1	0
19	3	0	1	2	0	1	0	0	0	0	0	1
20	3	1	2	1	0	0	0	1	1	0	1	0
21	3	2	1	2	1	0	1	0	0	1	0	1
22	3	0	2	1	0	0	1	0	1	1	0	0
23	3	1	0	0	1	1	1	1	0	0	1	1
24	3	2	0	0	1	1	0	1	1	1	1	0
25	4	0	0	2	1	1	1	0	1	1	1	1
26	4	1	1	0	0	0	1	1	1	1	0	1
27	4	2	1	1	0	1	0	0	0	0	1	0
28	4	0	0	1	0	1	1	1	0	0	0	0
29	4	1	2	2	1	0	0	1	0	0	1	1
30	4	2	2	0	1	0	0	0	1	1	0	0
31	5	0	0	1	1	0	0	1	0	1	0	0
32	5	1	2	2	0	1	1	1	0	1	0	0
33	5	2	0	1	0	1	1	0	1	0	1	1
34	5	0	1	0	1	0	0	0	0	0	0	1
35	5	1	2	2	0	1	0	0	1	1	1	1
36	5	2	1	0	1	0	1	1	1	0	1	0

(i) $OA(36, 6^1 3^3 2^8)$

Run	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	1	0	1	0	0	0	0	0
2	0	1	1	0	1	0	0	0	1	0	1
3	0	2	1	2	0	1	1	0	1	1	1
4	0	0	0	2	1	0	1	1	1	0	0
5	0	1	2	0	2	0	0	1	0	1	0
6	0	2	2	1	2	1	1	1	0	1	1
7	1	0	1	0	1	0	1	1	0	1	0
8	1	1	2	0	0	1	1	0	1	0	1
9	1	2	2	2	1	1	0	0	1	1	0
10	1	0	1	2	2	0	1	1	0	0	1
11	1	1	0	1	0	1	0	1	0	1	1
12	1	2	0	1	2	0	0	0	1	0	0
13	2	0	0	0	2	1	0	0	1	1	0
14	2	1	1	2	2	1	0	1	0	1	1
15	2	2	2	1	1	0	1	0	0	0	1
16	2	0	2	0	0	0	1	0	1	1	1
17	2	1	1	1	0	0	0	1	1	0	0
18	2	2	0	2	1	1	1	1	0	0	0
19	3	0	2	1	1	1	0	1	1	0	1
20	3	1	2	2	1	0	0	0	0	1	0
21	3	2	1	0	0	1	0	1	0	0	0
22	3	0	1	1	2	1	1	0	1	1	0
23	3	1	0	2	0	0	1	0	0	0	1
24	3	2	0	0	2	0	1	1	1	1	1
25	4	0	0	2	0	0	0	1	1	1	1
26	4	1	2	1	2	0	1	1	1	0	0
27	4	2	1	0	1	1	0	1	1	0	1
28	4	0	2	2	2	1	0	0	0	0	1
29	4	1	0	0	1	1	1	0	0	1	0
30	4	2	1	1	0	0	1	0	0	1	0
31	5	0	2	0	0	1	1	1	0	0	0
32	5	1	1	2	2	1	1	0	1	0	0
33	5	2	2	2	0	0	0	1	1	1	0
34	5	0	1	1	1	0	0	0	0	1	1
35	5	1	0	1	1	1	1	1	1	1	1
36	5	2	0	0	2	0	0	0	0	0	1

(ii) $OA(36, 6^1 3^4 2^6)$

Table 4: Comparison of NOAs with Run Size ≤ 24

	Wang & Wu			Nguyen			Ma et al.			The Author		
Array	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
$OA'(12, 2^7 3^2)$	-	-	-	-	-	-	.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
$OA'(18, 3^7 2^3)$	.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	?[a]	.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
$OA'(24, 6^1 2^{18})$	-	-	-	.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. Wang and Wu arrays are based on the construction method in their Section 6.

[a] The value can not be determined because no design is given in Ma et al.

Table 5: $OA'(12, 3^1 2^9)$

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 first five columns form an $OA(12, 3^1 2^4)$. The pairs of non-orthogonal columns (1, 6) and (1, 10) have an A_2 value of 0.167; the pairs of non-orthogonal columns (2, 9), (3, 7), (4, 8), and (6, 10) have an A_2 value of 0.111; and all other pairs of columns are orthogonal.

Table 6: $OA'(20, 5^1 2^{15})$

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	0	1	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	1	0	0	0	1	1	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	1	0	1	1	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 non-orthogonal columns are (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), and each pair has an A_2 value of 0.04.

Table 7: $OA'(24, 3^1 2^{21})$

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	0	1	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	0	1	0	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	0	1	0	0	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	1	0	0	0	0	1	1	1	1	1	1
18	2	1	1	0	0	0	1	0	1	1	0	0	0	1	1	1	0	0	1	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	0	1	0	1	0	0	0	0	1	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 non-orthogonal columns $(5, 19)$ has an A_2 value of 0.111; the pairs of non-orthogonal 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 $(21, 22)$ have an A_2 value of 0.028; and all other pairs of columns are orthogonal.

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

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

(i) Nguyen

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

(ii) The Author

NOTE: The first eight columns of each design form an $OA(18, 2^1 3^7)$. (i) The pair of non-orthogonal columns (2, 9) has an A_2 value of 0.5. (ii) The pairs of non-orthogonal columns (3, 9), (5, 9), and (8, 9) have an A_2 value of 0.167.

Table 9: Comparison of $OA'(18, 2^1 3^8)$

Array	D	A_2	Np	a_2
Original Plan	0.000	2.0	1	2.000
Nguyen	0.967	0.5	1	0.500
The Author	0.967	0.5	3	0.167

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