AN ALGORITHMIC APPROACH TO CONSTRUCTING ORTHOGONAL AND NEAR-ORTHOGONAL ARRAYS*

NAM-KY NGUYEN†

June 10, 2006

Abstract

Due to run size constraints, near-orthogonal arrays (near-OAs) and supersaturated designs, a special case of near-OA, are considered good alternatives to OAs. This paper shows (i) a combinatorial relationship between a mixed-level array and a non-resolvable incomplete block design (IBD) with varying replications (and its dual, a resolvable IBD with varying block size); (ii) the relationship between the criterion $E(d^2)$ proposed by Lu and Sun (2001) or $E(f_{\mathrm{NOD}})$ proposed by Fang et al. (2003) used in the (near-) OA construction and the (M,S)-optimality criterion used in the IBD construction (the tighter bound for $E(d^2)$ is accordingly established); (iii) how to modify the IBD algorithm of Nguyen (1994) to obtain efficient (near-) OA algorithms and the relationship between these algorithms and the one of Xu (2002). Some new OAs will be presented and some near-OAs are compared with designs constructed by other authors. Examples showing the use of the constructed arrays will be given.

Keywords: computer-generated designs; Cramer's V; screening designs; supersaturated designs; incomplete block designs.

---

*Paper presented at the *2006 International Conference on Design of Experiments and Its Applications.* Nankai University, Tianjin, China.

†School of Mathematics, Statistics and Computer Science, University of New England, Australia; e-mail: nkn@turing.une.edu.au.

1 Introduction

We will begin by providing two examples to illustrate the use of near-OAs:

Example 1. A wood scientist was asked to develop plywood of certain strength which was needed for the floor of cargo containers. As the strength could not be determined from first principles and because test data would be necessary to convince the regulatory authorities once a product was developed, she had to investigate a number of combinations of four timber species, four adhesive types, four different initial moisture contents, three hot press pressures, two cold press times, two levels of filler added to the adhesive resin, two levels of insecticide added to the adhesive resin and two types of fungicides. An OA for three 4-level factors, one 3-level factor and four 2-level factors requires a run size that is divisible by 4×4, 4×3, 4×2, 3×2, and 2×2, so the $L_{48}(4^3 3^1 2^4)$ in 48 runs (cf. `http://support.sas.com/techsup/technote/ts723.html`) is the smallest possible OA. However, because of the time and cost constraints, at most half of the number of suggested runs are allowed. What should be the suitable array for this experiment?

Example 2. Nguyen & Cheng (2005) described a passenger-impact crash test experiment on a planned new four-wheel-drive range whose objective is to find a subset of 54 safety features. They proposed a supersaturated design with (n, m)=(27,54) which only used 27 car prototypes. Now assume that the R & D Department wants to incorporate an additional 3-factor level into this experiment, i.e. car speed and is keen to know how this can be done.

Before discussing the near-OA solutions to the above problems, we discuss OA. A strength 2 OA of size n with k s_j-level columns ($j = 1, \ldots, k$), denoted by $L_n(s_1, \ldots, s_k)$ is an $n \times k$ matrix in which all possible combinations of levels in any two columns appear the same number of times (Rao 1947). There is an OA library of over 200 OAs maintained by Prof. N. J. A. Sloane (`http://www.research.att.com/~njas/oadir/`). This library has been recently updated by Dr. W. F. Kuhfeld of SAS at his OA site (`http://support.sas.com/techsup/technote/ts723.html`). This site contains all OAs listed in the Appendix of Kuhfeld & Tobias (2005) as well as new ones contributed by other authors. A simple introduction to OA can be found in most textbook on design of experiments (e.g. Chapter

7 of Wu & Hamada, 2000). More comprehensive references of OA are Hedayat et al. (1999) and Dey & Mukerjee (1999).

In a near-OA $L'_n(s_1...s_k)$, to reduce the run size, the orthogonality of some pairs of columns is necessarily sacrificed. The concept of near-OA (Taguchi 1959, Wang & Wu 1992, Nguyen 1996b, Ma et al. 2000, Xu 2002, Lu et al. 2006) provide a genuine answer to situations when OAs are not available. An array is called a saturated design when $\sum(s_i - 1) = n - 1$ (e.g. a Hadamard matrix) and is called a supersaturated design when $\sum(s_i - 1) > n - 1$. The 2-level supersaturated designs were discussed in Booth & Cox (1962), Lin (1993), Nguyen (1996a), Tang & Wu (1997), Cheng (1997) and Section 8.6 of Wu & Hamada (2000). The multi- and mixed-level supersaturated designs were discussed in Lu et al. (2003), Fang et al. (2002, 2003, 2004) and Liu & Fang (2005).

This paper has four sections. Section 2 shows a combinatorial relationship between a mixed-level array (OA and near-OA) and an IBD. Section 3 shows the relationship between the popular criterion $E(d^2)$ (or $E(f_{\mathrm{NOD}})$) used in the construction of the mentioned type of designs and the (M, S)-optimality criterion used in IBD construction. In this Section, we will show the derivation of a tighter bound for $E(d^2)$. Section 4 describes two new (near-) OA algorithms which are modifications of the IBD algorithm of Nguyen (1994). Section 5 compares some near-OAs constructed by the new algorithm and by other authors in terms of the D-efficiency of the designs and other goodness criteria.

2 The relationship between an array and an IBD

There is a relationship between certain combinatorial structures with IBDs. Box & Behnken (1960) and Nguyen (2005) used regular graph designs (RGDs) to construct 3-level response surface designs. Nguyen (1996) and Liu & Zhang (2000) used cyclic balanced IBDs to construct 2-level supersaturated designs. Nguyen & Cheng (2005) used RGDs to construct saturated and supersaturated designs. Lu et al. (2003), Fang et al. (2002, 2003, 2004) and Liu & Fang (2005) used resolvable balanced IBDs and resolvable group divisible designs to construct multi- and mixed-level supersaturated designs. Additional work of Professor Kai-Tai Fang and his coworkers on this area of research can be found at `http://www.math.hkbu.edu.hk/UniformDesign/`.

Consider the following mixed-level near-OA $L'_6(3^1 2^3)$:

Table 1: $L'_6(3^1 2^3)$

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

If we use the *dummy* coding to code the near-OA in Table 1, we will get the following X matrix:

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

It can be seen that (1) is the N' matrix (transpose of the incidence matrix) of the non-resolvable IBD of with varying replications of size (v, b, k)=(9,6,4) in Table 2:

Table 2: IBD of size (v, b, k)=(9, 6, 4)†

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

†Blocks are rows.

A non-resolvable IBD of size (v, b, k) has v varieties, each replicated r_i times ($i = 1, \ldots, v$), set out in b blocks of size $k(< v)$, i.e. $\sum r_i = bk$. We assume that no variety occurs more than once in a block. Note that the 1st position of each block of the IBD in Table 2 has varieties 0-2 which corresponds to the three levels of column 1 of the array in Table 1. Similarly, the 2nd position of each block of this IBD has varieties 2-3 which correspond to the two levels of column 2 of this array, etc. Associated with each IBD is the *incidence* matrix $N_{v\times b}$ whose (ij)th element equals 1 if variety i occurs in block j and 0 otherwise.

It can be seen that (1) is also the incidence matrix of the resolvable IBD (RIBD) with varying block sizes of size (v, b, r)=(6,9,4) in Table 3:

Table 3: RIBD of size (v, b, r)=(6,9,4)†

0_0	0_3	0_5	1_7
1_0	2_3	3_5	3_7
2_1	4_3	5_5	4_7
3_1	1_4	1_6	0_8
4_2	3_4	2_6	2_8
5_2	5_4	4_6	5_8

†Subscripts denote block number.

An RIBD of size (v, b, r) has v varieties, each replicated r times, set out in b blocks, each of size k_i $(i = 1, \dots, b)$, i.e. $\sum k_i = vr$. These blocks can be divided into subsets, each of which represents a complete replication of the varieties. Each column of the RIBD in Table 3 represents a replicate. The first replicate, for example, has three blocks (0, 1), (2, 3) and (4, 5). This IBD is, in fact the *dual* of the *primal* IBD in Table 2. The dual of an IBD is a new IBD obtained by swapping the varieties and block symbols in the original design(cf. p. 39-41 of John & Williams 1995).

As an additional example, the (coincidence) matrix on p. 357 of Xu (2002) which is associated with the $L_{12}(3^1 2^4)$ on the same page is also the concurrence matrix of an RIBD of size (v, b, r)=(12, 21, 5). The first replicate of this IBD, for example, has three blocks (0, 1, 2, 3), (4, 5, 6, 7), and (8, 9, 10, 11).

Remarks:

1. Associated of each IBD is the *concurrence* matrix NN' whose (ii)th element is r_i and (ij)th element is the number of blocks in which both varieties i and j appear. The concurrence matrix of the IBD in Table 2 is:

$$\begin{pmatrix} 2 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 3 & 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 & 3 & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 3 & 0 & 1 & 2 \\ 1 & 1 & 1 & 2 & 1 & 0 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 2 & 3 & 0 \\ 1 & 1 & 1 & 2 & 1 & 2 & 1 & 0 & 3 \end{pmatrix} \tag{2}$$

and the concurrent matrix of the RIBD in Table 3 (or $N'N$) is:

$$\begin{pmatrix} 4 & 1 & 2 & 1 & 1 & 2 \\ 1 & 4 & 1 & 2 & 2 & 1 \\ 2 & 1 & 4 & 1 & 2 & 1 \\ 1 & 2 & 1 & 4 & 1 & 2 \\ 1 & 2 & 2 & 1 & 4 & 1 \\ 2 & 1 & 1 & 2 & 1 & 4 \end{pmatrix} \tag{3}$$

2. We have shown that there is a combinational relationship between a mixed-level array of size n with k s_j-level columns ($j = 1, \ldots, k$) and a non-resolvable IBD of size $(v, b, k) = (\sum s_j, n, k)$ (and its dual, an RIBD of size $(v, b, r) = (n, \sum s_j, k)$). What this means is that we can construct this array indirectly by constructing either of the IBD which involves the minimization of the sum of squares of the elements of either (2) or (3). This is called the (M, S)-optimality criterion in the IBD literature (cf. p. 34-35 of John & Williams 1995).

3 Relationship between the criterion $E(d^2)$ and the (M, S)-optimality criterion

Given a near-OA $L'_n(s_1, \ldots, s_k)$, following Lu & Sun (2001) and Fang et al. (2003), we define "a measure of departure from orthogonality" for two columns i and j of this array as:

$$d_{ij}^2 = \sum_{u=0}^{s_i-1} \sum_{w=0}^{s_j-1} (n_{uw}^{ij} - \frac{n}{s_i s_j})^2 \tag{4}$$

and the global measure of departure from orthogonality of an array is defined as:

$$E(d^2) = \sum_{i=1}^{k} \sum_{j=i+1}^{k} d_{ij}^2 / \binom{k}{2}. \tag{5}$$

Here n_{uw}^{ij} is the observed frequencies of rows whose column i takes symbol u and column j takes symbol w. $n/(s_i s_j)$ is the expected frequency for each level combination. For the near-OA in Table 1, readers can verify that $d_{12} = d_{13} = d_{14} = 0$ and $d_{23} = d_{24} = d_{34} = 1$. $E(d^2)$ of this array is thus 0.5.

Note that the sum of the elements of NN' and $N'N$ are nk^2 and $\sum r_i^2$ respectively which are constants. As such, the (M, S)-optimality criterion only involves the minimization of the sum of squares of the elements in either NN' or $N'N$, i.e. minimizing trace$(NN')^2$ or trace$(N'N)^2$ (note that trace$(NN')^2$= trace$(N'N)^2$). It can then be shown that:

$$\binom{k}{2}E(d^2) = \sum_i \sum_{j>i} \sum_u \sum_w (n_{uw}^{ij})^2 - C = \frac{1}{2}(\text{trace}(NN')^2 - \sum r_i^2) - C \tag{6}$$

where $C = \sum_i \sum_{j>i} n^2/(s_i s_j)$, a constant.

(6) establishes the relationship between between $E(d^2)$ and the (M, S)-optimality criterion. It is also the generalization of the results of Fang et al. (2003, 2004) which requires the run size n be divisible by s_i. We can use this relationship to find a better lower bound for $E(d^2)$.

First, let's use the primal IBD. NN' can be partitioned into $\binom{k}{2}$ sub-matrices Λ_{ij}, $(i, j = 1, \ldots, k)$. The sum of the elements in Λ_{ij} is n, and the sum of squares of the elements in this matrix is minimal if it equals $S_{ij} = l_1\lambda^2 + l_2(\lambda + 1)^2$ (i.e. each Λ_{ij} has l_1 values λ and l_2 values $\lambda + 1$) where $\lambda = [n/(s_i s_j)]$, $l_2 = n - \lambda s_i s_j$ and $l_1 = s_i s_j - n_2$. Thus, the first lower bound for $E(d^2)$ is:

$$B_p = (\sum_i \sum_{j<i} S_{ij} - C)/\binom{k}{2} \tag{7}$$

This derivation of B_p is parallel to the one in Ma et al. (2000) and Lu et al. (2006) (see also p. 81 of John & Williams 1995). Obviously, when the array is an OA, (7) becomes 0.

Now, let's use the dual IBD. The sum of the upper diagonal elements of $N'N$ can be $S = \frac{1}{2}(\sum r_i^2 - nk)$. The sum of squares of the elements above the diagonal of $N'N$ is minimal if it equals $S_d = m_1\kappa^2 + m_2(\kappa + 1)^2$ (i.e. these elements consist of m_1 values κ and m_2 values $\kappa+1$) where $\kappa = [S/\binom{n}{2}]$, $m_2 = S - \kappa\binom{n}{2}$ and $m_1 = \binom{n}{2} - m_2$. In this case, the sum of squares of the elements above the diagonal elements of NN' is $S_p = \frac{1}{2}(2S_d + nk^2 - \sum r_i^2)$ where $2S_d + nk^2$ is the sum of squares of the elements of $N'N$ (or NN') (readers can verify that the C, S, S_d and S_p values of the associated with the array in Table are 45, 21, 33 and 48 respectively). Thus the second lower bound for $E(d^2)$ is:

$$B_d = (S_p - C)/\binom{k}{2}. \tag{8}$$

The derivation of $B_d(E(d^2)$ simplifies and generalizes the one in Fang et al. (2003, 2004), which restricts the run size n be divisible by s_i. The lower bound for $E(d^2)$ is thus:

$$B = \max(B_p, B_d) \tag{9}$$

Remarks:

1. The J_2 of Xu (2002) is the sum of squares of the elements above the diagonal of the $N'N$ matrix associated with the array. This J_2 reaches Xu's lower bound for J_2 when $J_2 = \frac{1}{2}(2C - nk^2 + \sum r_i^2)$. In Xu's $L_{12}(3^1 2^4)$ example, J_2=330, C=312, nk^2=300 and $\sum r_i^2$=336. Xu's lower bound for J_2 is useful to check whether the constructed array is an OA but cannot be used to check whether the constructed array is an $E(d^2)$-optimal near-OA.

2. Fang et al. (2003) showed that $E(d^2) = \frac{1}{4}E(s^2)$ where $E(s^2)$ is a criterion used for supersaturated designs with factors at two levels ± 1.

3. $E(d^2)$'s of the array in Table 1 and several near-OAs in Section 5 reach both lower bounds in (7) and (8). However, there are situations in which the $E(d^2)$ value of a particular near-OA reaches B_p but not B_d and vice versa. The $E(d^2)$ of the two $L'_{18}(2^1 3^8)$ in Table 7 of Xu (2002) reaches B_d (=0.5) but not B_p (=0). The $E(d^2)$ of the near-OA $L'_{24}(3^{10})$ in Table A7 of Lu et al. (2006) reaches B_p (=2) but not B_d (=0). Similarly, the $E(d^2)$ of the near-OA $L'_{10}(5^1 2^5)$ in Table 4 reaches B_p (=0.6666) but not B_d (=0).

Table 4: $L'_{10}(5^1 2^5)$

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

4. The $E(d^2)$ criterion used in the (near-) OA construction, like the (M, S)-optimality criterion used in the IBD construction, are approximate criteria in design construction. Table 3 of Lu (2006) lists six $L'(3^1 2^9)$. It can be seen that the arrays with the largest value of D (D-efficiency) in this Table (i.e. the ones reported in Nguyen 1996 and Xu 2002) are not necessarily the ones with the smallest $E(d^2)$.

4 Algorithms for constructing (near-) OAs

We have two algorithms for array construction. The primal algorithm makes use of the relationship between an array and a non-resolvable IBD. The dual algorithm makes use of the relationship between an array and an RIBD. Both algorithms use the $E(d^2)$ criterion. This criterion is akin to the (M, S)-optimality criterion which involved the minimization of the sum of squares of the elements above the diagonal elements of NN' (or $N'N$). Before discussing our algorithms, we give details of the update of our objective function f $(= \binom{k}{2}E(d^2))$ and NN' matrix that are crucial in speeding up our algorithm.

Let i be a variety in position j of block I and t be a variety in another position of this block. Let u be a variety in position j of block U and t' be a variety in another position of this block. The pairwise swapping of i and u will increase all λ_{tm}'s and $\lambda_{t'i}$'s by 1 and decrease all λ_{ti}'s and $\lambda_{t'm}$'s by 1. This means f will be increased by an amount:

$$\Delta f = 2\{\sum(\lambda_{tm} - \lambda_{ti} + 1) + \sum(\lambda_{t'i} - \lambda_{t'm} + 1)\}. \tag{10}$$

The steps of the primal algorithm making use of the update formula in (10) are:

1. Construct a starting array $L_n^{(')}(s_1, \ldots, s_k)$ by allocating s_j symbols 0,...,$s_j - 1$ to column j such that the numbers of each symbol differs by at most 1. Randomize the positions of each symbol. Convert this array to an IBD of size $(v, b, k) = (\sum s_j, n, k)$. Construct the NN' matrix of this IBD. Deduct each element of the sub-matrix Λ_{ij} $(i, j = 1, \ldots, k,\ j > i)$ by an amount $n/(s_i s_j)$ and calculate f, the sum of squares of the elements of these sub-matrices.

2. Repeat searching a pair of varieties i and m in position j $(j = 1, \ldots, k)$ in two difference blocks such that the swap of these two varieties results in the biggest reduction of f. If the search is successful, update f, NN' and the IBD. If f cannot be reduced further, go to the next position. This process is repeated until f reaches its lower bound (i.e. $\binom{k}{2}B$ or cannot be reduced further.

3. Convert the IBD in step 2 to the array $L_n^{(')}(s_1, \ldots, s_k)$ and calculate some goodness statistics for this array such as the D, $V_{\max} = \max(V_{ij})$, where V_{ij} is the Cramer's V coefficient of association between two columns i and j (`http://www2.chass.ncsu.edu/garson/pa765/assocnominal.htm`) and the $f_{\max}$, the frequency of $V_{ij} = V_{\max}$.

The basic algorithm (i.e. steps 1-3) is repeated a number of times to avoid the local optima. Each time is called a *try*. Among a large number of tries, the best one with respect to a chosen goodness criterion is selected. Our algorithm uses D in conjunction with $V_{\max}$ and $f(V_{\max}$ as the goodness criterion. f is used instead of D the design is supersaturated.

Remarks:

1. With the dual algorithm, the dual of the IBD used in the primal algorithm and $N'N$ will be used instead. Varieties in different blocks of the same replicate will be swapped. There is a resemblance of this algorithm and the one of Xu (2002) as both work with matrix $N'N$. Both primal and dual algorithms work better than algorithms which maximizes the D-efficiency such as the Fedorov exchange algorithm (cf. Nguyen &Piepel 2005) in terms

of speed and the number of pairs of orthogonal columns.

2. New arrays can be constructed by adding new columns to an existing array. The primal algorithm requires less calculations than the dual one in this type of array construction as it only works with a sub-matrix of NN' which involves new columns.

3. There are situations in which experimenters are interested in arrays with minimal $\max(d^2)$ (and the minimum number of $d_{ij} = \max(d^2)$). This type of array can be indirectly constructed by the primal algorithm by minimizing $\max(\delta_{uw}^{ij})$ where $\delta_{uw}^{ij} = |n_{uw}^{ij} - n/(s_i s_j)|$ and the frequency of $\delta_{uw}^{ij} = \max(\delta_{uw}^{ij})$. The stopping rule for this *minimax* algorithm is that each $\delta_{uw}^{ij} < 1$.

4. There are also situations in which experimenters consider certain factors (columns) as more important than the remaining ones. In other words, they prefer the former to be orthogonal (or close to orthogonal) among themselves and to the latter. Again, this type of array can be easily obtained via the primal algorithm by defining a second objective function calculated from elements of a sub-matrix of NN' which involves the mentioned factors.

Table 5: Comparison of near-OAs in terms of D and N_p.

#	Array	Wang & Wu†	Ma et al.†	Xu†	Nguyen†	$V_{\max}$‡
1	$L'_6(3^1 2^3)$	901 (3)	901 (3)	901 (3)	901 (3)§	333 (3)
2	$L'_{10}(5^1 2^5)$	883 (10)	967 (10)	967 (10)	967 (10)§	200 (10)
3	$L'_{12}(4^1 3^4)$	946 (6)	946 (6)	946 (6)	946 (6)§	250 (6)
4	$L'_{12}(2^3 3^4)$	946 (6)	946 (6)	946 (6)	946 (6)§	250 (6)
5	$L'_{12}(6^1 2^5)$	911 (6)	911 (3)	959 (4)	959 (4)	333 (4)
6	$L'_{12}(6^1 2^6)$		909 (4)	947 (6)	947 (6)	333 (6)
7	$L'_{12}(3^1 2^9)$	867 (9)	905 (5)	933 (6)	933 (8)	333 (8)
8	$L'_{12}(2^1 3^5)$	877 (10)		877 (10)	877 (10)	250 (10)
9	$L'_{12}(3^2 2^7)$		899 (6)	909 (6)	888 (8)	333 (7)
10	$L'_{12}(3^3 2^5)$		877 (6)	877 (6)	925 (9)	333 (6)
11	$L'_{15}(5^1 3^5)$	882 (10)	882 (10)	882 (10)	882 (10)§	200 (10)
12	$L'_{18}(2^1 3^8)$			967 (3)	967 (3)§	289 (3)
13	$L'_{18}(2^3 3^7)$	970 (3)	970 (7)	970 (3)	970 (3)§	333 (3)
14	$L'_{18}(9^1 2^8)$	985 (28)	985 (28)	985 (28)	985 (28)§	111 (28)
15	$L'_{20}(5^1 2^{15})$	838 (30)	623 (14)	925 (19)	956 (18)	200 (18)
16	$L'_{24}(8^1 3^8)$	897 (28)	845 (31)	897 (28)	897 (28)§	125 (28)
17	$L'_{24}(3^1 2^{21})$	853 (21)	953 (14)	968 (23)	968 (8)	333 (8)
18	$L'_{24}(6^1 2^{15})$	870 (18)	934 (12)	994 (1)	994 (1)	333 (1)
19	$L'_{24}(6^1 2^{18})$		761 (18)	974 (6)	974 (6)	333 (6)
20	$L'_{24}(2^1 3^{11})$	871 (55)		895 (56)	895 (55)	177 (11)
21	$L'_{24}(3^1 4^7)$	594 (21)		858 (21)	874 (21)	236 (2)
22	$L'_{36}(3^{13} 2^9)$				978 (8)	333 (8)
23	$L'_{50}(5^{11} 2^5)$				994 (10)	200 (10)
24	$L'_{54}(3^{25} 2^3)$				990 (3)§	333 (3)

†$10^3 D$ (the larger the better) and N_p (the smaller the better).

‡$10^3 V_{\max}$ (the smaller the better) and $f_{\max}$ of Nguyen's array.

§$E(d^2)$-optimal arrays.

5 Discussion

Table 5 gives a listing of 24 near-OAs constructed by Wang & Wu (1992), Ma et al. (2000), Xu (2002) and the author in terms of the D and N_p (the number of non-orthogonal pairs). Our arrays also give details of the $V_{\max}$ and $f_{\max}$. Our arrays are restricted to those with $V_{\max} \leq 0.333$. As a result, two of our arrays are not as good as arrays of other authors with respect to other measures of goodness. For $L'_{12}(3^1 2^9)$ (# 7), both Xu's array and ours have D=0.933 (Table 6). The N_p of the Xu's array is 6 and of ours is 8. However, the $V_{\max}$ of the former is 0.408 and of the latter is 0.333. For this $L'_{12}(3^1 2^9)$, the first 3-level column of the array of Lu et al (2006) and ours is orthogonal to the remaining columns. The N_p of Lu's array is 7 and of ours is 8. However, the $V_{\max}$ of the former is 0.667 and of the latter is 0.333.

Table 6: Two $L'12(3^1 2^9)$'s†

1	2	3	4	5	6	7	8	9	10	6'	7'	8'	9'	10'
0	0	1	0	1	1	1	0	0	0	0	0	0	1	1
0	1	0	0	1	1	0	0	1	0	0	0	1	0	0
0	0	1	1	0	1	0	1	1	1	1	1	1	0	1
0	1	0	1	0	0	1	1	0	0	1	1	0	1	0
1	0	0	0	0	0	0	0	0	1	0	1	0	0	1
1	1	1	0	0	0	1	0	1	1	0	1	1	1	0
1	0	1	1	1	0	0	1	1	0	1	0	0	0	0
1	1	0	1	1	1	1	1	0	1	1	0	1	1	1
2	0	0	1	0	1	1	0	1	0	0	0	1	1	0
2	1	1	0	0	1	0	1	0	0	1	0	0	0	1
2	0	0	0	1	0	1	1	1	1	1	1	1	1	0
2	1	1	1	1	0	0	0	0	1	0	1	0	0	1

†Columns 1-5 form an $L_{12}(3^1 2^4)$. This OA and columns 6-10 form Xu's array. This OA and columns 6'-10' form ours.

Similarly, for $L'_{12}(3^2 2^7)$ (# 9), the D and N_p of Xu's array are 0.909 and 6 and of ours are 0.888 and 8. However, the $V_{\max}$ of the former is 0.408 and of the latter is 0.333. In terms of D, we were able to improve three arrays of Xu in Table 5 (i.e. # 10, # 15, and # 21). In terms of N_p, we were able to improve three arrays of Xu in this Table (i.e. # 15, # 17, and # 20). 10 out of 24 arrays in this table are $E(d^2)$-optimal. Arrays in this Table are of the form $L'(s_1^{k_1} s_2^{k_2})$. The first k_1 columns of our arrays are always orthogonal to the remaining k_2 columns. It is clear that arrays # 7 and # 9 of Xu do not have this feature and it is not clear that the other arrays of Xu have this feature.

We have two solutions for array $L'_{24}(6^1 2^{15})$ (# 18). The 2nd solution obtained by the minimax criterion has $D=0.988$ instead of 0.994 and N_p=8 instead of 1 (Table 7). However, its $V_{\max}$ is 0.167 instead of 0.333. To many experimenters, this solution is a preferred one despite its low D.

Table 7: Two $L'_{24}(6^1 2^{15})$'s†

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

†Columns 1-15 form an $L_{24}(6^1 2^{14})$. This OA and column 16 form the 1st near-OA. This OA and column 16' form the 2nd near-OA.

The solution for Example 1 is the following $E(d^2)$-optimal $L'_{24}(4^3 3^1 2^4)$ (Table 8). It has D=0.978 and $V_{\max}$=0.193 with $f_{\max}$=3. The last five columns of this array form an OA and the first three columns of this array are orthogonal to the remaining columns.

Table 8: $L'_{24}(4^3 3^1 2^4)$

0	0	0	1	1	0	0	0
0	0	1	0	0	0	1	0
0	1	3	2	0	0	1	1
0	2	0	0	1	1	1	1
0	2	2	2	1	1	0	0
0	3	1	1	0	1	0	1
1	0	1	2	1	0	0	1
1	0	3	0	0	1	1	0
1	1	2	0	1	0	0	0
1	1	3	1	1	1	0	1
1	2	0	2	0	0	1	1
1	3	2	1	0	1	1	0
2	0	2	2	0	1	0	1
2	1	0	0	0	1	0	0
2	1	2	1	0	0	1	1
2	2	1	1	1	0	1	0
2	3	3	0	1	0	0	1
2	3	3	2	1	1	1	0
3	0	0	1	1	1	1	1
3	1	1	2	1	1	1	0
3	2	1	0	0	1	0	1
3	2	3	1	0	0	0	0
3	3	0	2	0	0	0	0
3	3	2	0	1	0	1	1

The solution for Example 2 is an $E(d^2)$-optimal $L'_{27}(3^1 2^5 4)$ with $V_{\max}$=0.421 and $f_{\max}$=3. All near-OAs in Table 5 and the solutions for the two examples can be found at `http://designcomputing.net/gendex/noa`.

The work of Lu et al. (2006) becomes relevant in light of this research. Table 1 of Lu provides details of 13 near-OAs consisting of 2- and 3-level factors. Out of these 13 arrays, we were able to improve the D's of nine of them. These arrays are # 1, # 2, # 3, # 4, # 5 , # 8, # 10, # 11, and # 13 in this table. The D's of Lu et al. (2006) for these arrays are 0.905, 0.948, 0.962, 0.881, 0.833, 0.837, 0.772 and 0.854 compared with 0.933, 0.954, 0.909, 0.888, 0.950, 0.877, 0.861, 0.967 and 0.909 for the algorithm in Section 4. There is evidence that this Table was made with insufficient tries (e.g. their algorithm stops as soon as $E(d^2)$ is reached). Despite this, we were not able to obtain the $E(d^2)$-optimal $L'_{21}(3^1 0)$ reported in this Table after a very large number of tries. Basically, this suggests that no algorithm is good for all situations.

As mentioned, one of the main features of our algorithm is its ability to add additional columns to existing arrays. Several new OAs and near-OAs can be constructed this way. Our new $L_{36}(2^{13}3^2 6^1)$, $L_{60}(2^{15}6^1 10^1)$, $L_{84}(2^{14}6^1 14^1)$ and $L_{100}(10^4 2^4)$ are listed at `http://support.sas.com/techsup/technote/ts723.html`. The $L_{100}(10^4 2^4)$, for example, was constructed by adding four additional columns to the well known $L_{100}(10^4)$. Our new $E(d^2)$-optimal $L'_{84}(2^8 6^1 14^1 3^2)$ and $L'_{100}(10^4 2^4 3^2)$ and other smaller near-OAs are listed at `http://designcomputing.net/gendex/noa/`.

Both algorithms reported in Section 4 are very fast. For small arrays such as the $L'_{12}(3^3 2^5)$, the primal algorithm takes less than four seconds on our Pentium 1.83GHz laptop to obtain 1,000 tries. Out of these 1,000 tries, 143 have D=0.925. For larger arrays such as the $L'_{24}(6^1 4^6)$, this algorithm takes 210 seconds on this laptop to obtain 10,000 tries. Out of these 10,000 tries, 32 are $E(d^2)$-optimal and only two out of these 32 tries have D=0.928. These algorithms are implemented in two Java programs. Please contact the author regarding their availability.

Acknowledgement

The author would like to thank Dr. Warren Kuhfeld of SAS for valuable comments and the Univerisity of New England for the PVC Research Grant. This paper is dedicated to Professor Aloke Dey, the author's former Ph.D. supervisor.

References

Booth, K.H.V. & Cox, D.R. (1962) Some systematic supersaturated designs. *Technometrics* 4, 489-495.

Box, G.E.P. & Behnken, D.W. (1960) Some new three level designs for the study of quantitative variables. *Technometrics* 2, 455-475.

Cheng, C.S. (1997) $E(s^2)$-optimal supersaturated designs. *Statistica Sinica* 7, 929-939.

Dey, A & Mukerjee, M. (1999) *Fractional Factorial Plans* New York: Wiley.

Fang, K. T., Ge, G. N. and Liu, M. Q. (2002) Uniform supersaturated design and its construction. *Sci. China Ser. A* 45 (8), 1080-1088.

Fang, K-T. , Lin, D. K. J. & Liu, M-Q. (2003) Optimal mixed-level supersaturated design. *Metrika* 58, 279-291.

Fang, K. T., Ge, G. N., Liu, M. Q. and Qin, H. (2004) Combinatorial constructions for optimal supersaturated designs. *Discrete Math* 279, 191–202.

Hedayat, A. S., Sloane, N. J. A. & Stufken J. (1999) *Orthogonal arrays: Theory and Applications*, New York:Springer-Verlag.

John, J.A. & Williams E.R. (1987) *Cyclic designs and computer-generated designs.* NY: Chapman & Hall.

Kuhfeld, W. F. & Tobias, R. D. (2005) Large factorial designs for product engineering and market research applications. *Technometrics* 47, 122-132.

Lin, D.K.J. (1993) A new class of supersaturated designs. *Technometrics* 35, 28-31.

Liu, M.Q. & Zhang, R.C. (2000) Construction of $E(s^2)$-optimal Supersaturated Designs Using Cyclic BIBDs. *J. of Statistical Planning & Inference* 91, 139-150.

Liu, M. Q. & Fang, K. T. (2005) Some results on resolvable incomplete block designs. *Science in China Ser. A Mathematics* 2005 48, 503512

Lu, X. & Sun, Y. (2001) Supersaturated designs with more than two levels. *Chinese Ann. Math.* B 22, 183-194.

Lu, X., Hu, W. & Zheng, Y. (2003) A systematical procedure in the construction of multi-level supersaturated designs. *J. of Statistical Planning & Inference* 115, 287-310.

Lu, X. Li, W. & Xie, M. (2006) A class of nearly orthogonal arrays. *J. of Quality Technology* 38, 148-161.

Ma, C-X., Fang, K-T & Liski, E. (2000) A new approach in constructing orthogonal and nearly orthogonal arrays. *Metrika* 50, 255-268.

Nguyen, N-K. (1994) Construction of optimal incomplete block designs by computer. *Technometrics* 36, 300-307.

Nguyen, N-K. (1996) An algorithmic approach to constructing supersaturated designs. *Technometrics* 38, 205-209.

Nguyen, N-K. (1996) A note on the construction of near-orthogonal arrays with mixed levels and economic run size. *Technometrics* 38, 279-283.

Nguyen, N-K (2005) New 3-level response surface designs constructed from incomplete block designs (Invited paper). Proc. of the International Conference on Design of Experiments: Theory & Applications, The University of Memphis, Memphis (USA).

Nguyen, N-K & Piepel, G.F. (2005) Computer-generated experimental designs for irregular-shaped regions, *Quality Technology & Quantitative Management* 2, 147-160

Nguyen, N-K & Cheng, C-S (2005) New $E(s^2)$-optimal supersaturated designs constructed from from incomplete block designs. Submitted.

Rao, C. R. (1947) Fractional experiments derivable from combinatorial arrangements of arrays. *J. of the Statistical Society (Supp.)* 9, 128-139.

Taguchi, G. (1959) Linear graphs for orthogonal arrays and their applications to experimental designs, with the aid of various techniques. Report of Statistical Applications Research, Japanese Union of Scientists and Engineers 6, 1-43.

Tang, B. & Wu, C.F.J. (1997) A method for constructing supersaturated designs and its $E(s^2)$-property. *Canadian J. of Statistics* 25, 191-201

Wang, J.C. & Wu, C.F.J. (1992) Nearly orthogonal arrays with mixed levels and small runs. Technometrics 34, 409-422.

Wu, C.F.J & M. Hamada (2000) *Experiments: Planning, Analysis and Parameter Design Optimization.* New York: Wiley.

Xu, H. (2002) An algorithm for constructing orthogonal and nearly-orthogonal arrays with mixed levels and small runs. *Technometrics* 44, 356-368.