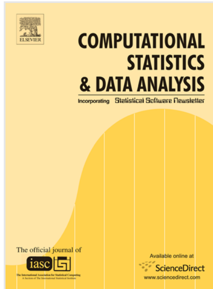
Accepted Manuscript



An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays

Nam-Ky Nguyen, Min-Qian Liu

PII: S0167-9473(08)00256-9
DOI: 10.1016/j.csda.2008.05.004
Reference: COMSTA 4034

To appear in: *Computational Statistics and Data Analysis*

Received date: 31 July 2007
Revised date: 18 April 2008
Accepted date: 7 May 2008

An algorithmic approach to constructing mixed-level orthogonal and near-orthogonal arrays

Nam-Ky Nguyen [a], Min-Qian Liu [b,*]

[a] *Centre for High-Performance Computing, Hanoi University of Science, 334 Nguyen Trai, Thanh Xuan, Hanoi, Vietnam*

[b] *Department of Statistics, School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, China*

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 sizes); (ii) the relationship between the criterion $E(d^2)$ proposed by Lu and Sun (2001) or $E(f_{NOD})$ proposed by Fang et al. (2003b) used in the (near-) OA construction and the (M, S)-optimality criterion used in the IBD construction; (iii) the derivation of a tighter bound for $E(d^2)$; (iv) how to modify the IBD algorithm of Nguyen (1994) to obtain efficient (near-) OA algorithms. Some new (near-) OAs are presented and some near-OAs are compared with arrays constructed by other authors. Examples showing the use of the constructed arrays are given.

Key words: Computer-generated design; Cramer's V; Screening design; Supersaturated design

---

* Corresponding author. Tel.: +86-22-23504709; fax: +86-22-23506423.
Email addresses: nguyen.namky@gmail.com (Nam-Ky Nguyen), mqliu@nankai.edu.cn (Min-Qian Liu).

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 design for this experiment?

Example 2 Nguyen and Cheng (2008) 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 2-level 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-level factor 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 and Tobias (2005) as well as new ones contributed by other authors.

A near-OA, denoted by $L'_n(s_1 \cdots s_k)$, is an array in which the orthogonality requirement is nearly satisfied. The concept of near-OA (Taguchi (1959), Wang and Wu (1992), Nguyen (1996b), Ma et al. (2000), Xu (2002), Lu et al. (2006)) provides 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 Plackett-Burman design) and is called a supersaturated design when $\sum(s_i - 1) > n - 1$. Supersaturated designs were first examined by Booth and Cox (1962) systematically, and were not studied further until the appearance of the work by Lin (1993) and Wu (1993). Since then, there has been a large number of papers on this subject, see some most recent papers, e.g. Nguyen and Cheng (2008), Chen and Liu (2008a, b), Liu and Lin (2008), and the references therein.

Note that, although almost all the near-OAs and supersaturated designs studied in existing papers, except perhaps those in Nguyen and Cheng (2008) and Chen and Liu (2008b), are U-type designs, i.e. arrays in which all levels appear equally often for any column (Fang et al. (2006)), the results we developed here, i.e. the relationships, tighter bound and algorithms, are not restricted to U-type designs, see for example the solution for Example 2 and the detailed discussions in the following sections.

This paper is organized as follows. 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 $E(f_{NOD})$) 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 Relationship between an array and an IBD

There is a relationship between certain combinatorial structures with IBDs. Box and Behnken (1960) used balanced IBDs and partially balanced IBD to construct 3-level response surface designs. Nguyen and Borkowski (2008) used regular graph designs (RGDs) to construct this type of designs. Nguyen (1996a) and Liu and Zhang (2000) used cyclic balanced IBDs to construct 2-level supersaturated designs. Lu et al. (2003), Fang et al. (2002, 2003a, 2004a,b,c) and Liu and Fang (2005) used resolvable balanced IBDs, resolvable group divisible designs, packing designs and large sets to construct multi- and mixed-level supersaturated designs. Nguyen and Cheng (2008) used RGDs to construct saturated and supersaturated designs.

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

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.

Consider the mixed-level near-OA $L'_6(3^1 2^3)$ given in Table 1. If we use the *dummy* coding to code this near-OA, 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 with varying replications of size (v, b, k)=(9,6,4) in Table 2.

A non-resolvable IBD of size (v, b, k) has v varieties, each replicated r_i times $(i = 0, \ldots, v-1)$, 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 3–4 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 N 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 $(i = 0, \ldots, b-1)$, 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. pp. 39–41 of John and Williams (1995)). So we conclude that

Theorem 1 *There exists a one-to-one correspondence between a mixed-level array of size n with k s_j-level columns $(j = 1, \ldots, k)$, $L_n^{(\prime)}(s_1 \cdots s_k)$, and a non-resolvable IBD of size $(v, b, k) = (\sum s_j, n, k)$(and its dual, an RIBD of size $(v, b, r) = (n, \sum s_j, k)$).*

Remarks:

1. Associated with 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 and the one of the RIBD in Table 3 (or $N'N$) are:

$$\begin{pmatrix} 2 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 2 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 3 & 0 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 0 & 3 & 2 & 1 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 3 & 0 & 1 & 2 \\ 1 & 1 & 1 & 2 & 1 & 0 & 3 & 2 & 1 \\ 1 & 1 & 1 & 1 & 2 & 1 & 2 & 3 & 0 \\ 1 & 1 & 1 & 2 & 1 & 2 & 1 & 0 & 3 \end{pmatrix} \text{ and } \begin{pmatrix} 4 & 1 & 2 & 1 & 1 & 2 \\ 1 & 4 & 1 & 2 & 2 & 1 \\ 2 & 1 & 4 & 1 & 2 & 1 \\ 1 & 2 & 1 & 4 & 1 & 2 \\ 1 & 2 & 2 & 1 & 4 & 1 \\ 2 & 1 & 1 & 2 & 1 & 4 \end{pmatrix} \qquad (2)$$

respectively.

2. As a result of the combinational relationship shown in Theorem 1, we can construct the array indirectly by constructing either IBD which involves the minimization of the sum of squares of the elements of either matrix in (2). This is called the (M, S)-optimality criterion in the IBD literature (cf. pp. 34–35 of John and Williams (1995)).

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

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

$$d_{ij}^2 = \sum_{u=0}^{s_i-1} \sum_{w=0}^{s_j-1} (n_{uw}^{ij} - \frac{n}{s_i s_j})^2, \text{ and}$$
$$E(d^2) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} d_{ij}^2 \Big/ \binom{k}{2},$$

respectively. 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.

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:

Theorem 2

$$\binom{k}{2} E(d^2) = \sum_i \sum_{j>i} \sum_u \sum_w (n_{uw}^{ij})^2 - C$$
$$= (\text{trace}(NN')^2 - \sum r_i^2)/2 - C, \tag{3}$$

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

Equation (3) establishes the relationship between $E(d^2)$ and the (M,S)-optimality criterion. It is also the generalization of the results of Fang et al. (2003b, 2004b) 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 us use the primal IBD. The upper-diagonal of NN' contains $\binom{k}{2}$ sub-matrices Λ_{ij} $(i = 1, \ldots, k-1, j = i+1, \ldots, k)$. 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 $\lambda = \lfloor n/(s_i s_j) \rfloor$, $l_2 = n - \lambda s_i s_j$ and $l_1 = s_i s_j - l_2$, $\lfloor l \rfloor$ is the integer part of l. Thus, the first lower bound for $E(d^2)$ is:

$$B_p = (\sum_i \sum_{j>i} S_{ij} - C) \Big/ \binom{k}{2}. \tag{4}$$

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 and Williams (1995)). Obviously, when the array is an OA, (4) becomes 0.

Now, let us use the dual IBD. The sum of the upper diagonal elements of $N'N$ can be $S = (\sum r_i^2 - nk)/2$. 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 $\kappa = \lfloor S/\binom{n}{2} \rfloor$, $m_2 = S - \kappa\binom{n}{2}$ 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 $S_p = (2S_d + nk^2 - \sum r_i^2)/2$ where $2S_d + nk^2$ is the sum of squares of the elements of $N'N$ (or NN'). Thus the second lower bound for $E(d^2)$ is:

$$B_d = (S_p - C)\Big/\binom{k}{2}. \tag{5}$$

The derivation of B_d simplifies and generalizes the ones in Fang et al. (2003b, 2004b) which restrict the run size n be divisible by s_i. Thus we get the lower bound for $E(d^2)$:

Theorem 3

$$E(d^2) \geq \max(B_p, B_d).$$

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 $J_2 = (2C - nk^2 + \sum r_i^2)/2$. 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 near-OA is $E(d^2)$-optimal.

2. Fang et al. (2003b) showed that $E(d^2) = E(s^2)/4$ where $E(s^2)$ is a criterion proposed by Booth and Cox (1962) and used for supersaturated designs with factors coded at two levels ± 1.

3. The $E(d^2)$'s of the array in Table 1 and several near-OAs in Section 5 reach both lower bounds in (4) and (5). 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 $L'_{18}(2^1 3^8)$'s 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).

4. The $E(d^2)$ criterion used in the (near-) OA construction, like the (M, S)-optimality criterion used in the IBD construction, is an approximate criterion in design construction. Table 3 of Lu et al. (2006) lists six $L'_{12}(3^1 2^9)$'s. It can be seen that the arrays with the largest value of D (D-efficiency) in this table

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

(i.e. the ones reported in Nguyen (1996b) 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 m be a variety in position j of block M and t' be a variety in another position of this block. The pairwise swapping of i and m 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:

$$\Delta f = 2\{\sum(\lambda_{tm} - \lambda_{ti} + 1) + \sum(\lambda_{t'i} - \lambda_{t'm} + 1)\}. \tag{6}$$

The steps of the primal algorithm making use of the update formula in (6) are:

Step 1. Construct a starting array $L_n^{(')}(s_1, \ldots, s_k)$ by allocating s_j symbols $0, \ldots, s_j - 1$ to column j such that the numbers of these symbols differ 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.

Step 2. Repeat searching a pair of varieties i and m in position j $(j = 1, \ldots, k)$ in two different 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. $\max(B_p, B_d)\binom{k}{2}$) or cannot be reduced further.

Step 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 $V_{ij} = V_{\max}$.

Step 4. 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 $f_{\max}$ as the goodness criterion. f is used instead of D when 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 and 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 $\max(d^2) = \max\{d_{ij}^2\}$ (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.

5 Discussion

Table 5 gives a listing of 24 near-OAs constructed by Wang and Wu (1992), Ma et al. (2000), Xu (2002) and the authors 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 the $L'_{12}(3^1 2^9)$ (#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, Li and Xie'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.

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 $L'_n(s_1^{k_1} s_2^{k_2})$. 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.

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. The solution for Example 2 is an $E(d^2)$-optimal

Table 5

Comparison of near-OAs in terms of D and N_p

#	Array	Wang and Wu†	Ma, Fang and Liski†	Xu†	Authors†	$V_{\max}$‡
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 authors' array.

§$E(d^2)$-optimal arrays.

$L'_{27}(3^1 2^{54})$ 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 et al. (2006) 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 eight of them. These arrays are #1, #2, #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, 0.882, 0.881, 0.833, 0.837, 0.772 and 0.854 compared with 0.933, 0.954, 0.888, 0.950, 0.877, 0.891, 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

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.

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.

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 $E(d^2)$-optimal $L'_{21}(3^{10})$ 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 2-level 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 about two seconds on our Core Duo 2 GHz 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 2 minutes 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 first author regarding their availability.

Acknowledgements

The first author would like to dedicate this paper to Professor Aloke Dey, his former Ph.D. supervisor on his retirement from the Indian Statistical Institute, Delhi Centre, Delhi. This work was supported by the PVC Research Grant of the University of New England, the Program for NCET in University (NCET-07-0454) of China, the NNSF of China Grant 10671099 and the SRFDP of China Grant 20050055038. The authors would like to thank Dr. Warren Kuhfeld of SAS, the Co-Editor and two referees for their valuable comments.

References

[1] Booth, K. H. V., Cox, D. R., 1962. Some systematic supersaturated designs. Technometrics 4, 489–495.

[2] Box, G. E. P., Behnken, D. W., 1960. Some new three-level designs for the study of quantitative variables. Technometrics 2, 455–475.

[3] Chen, J., Liu, M. Q., 2008a. Optimal mixed-level k-circulant supersaturated designs. J. Statist. Plann. Inference, doi: 10.1016/j.jspi.2008.03.025. Available online 18 March 2008.

[4] Chen, J., Liu, M. Q., 2008b. Optimal mixed-level supersaturated design with general number of runs. Statist. Probab. Lett., doi:10.1016/j.spl.2008.02.025. Available online 18 March 2008.

[5] Fang, K. T., Ge, G. N., Liu, M. Q., 2002. Uniform supersaturated design and its construction. Sci. China Ser. A 45, 1080–1088.

[6] Fang, K. T., Ge, G. N., Liu, M. Q., Qin, H., 2003a. Construction of minimum generalized aberration designs. Metrika 57, 37–50.

[7] Fang, K. T., Lin, D. K. J., Liu, M. Q., 2003b. Optimal mixed-level supersaturated design. Metrika 58, 279–291.

[8] Fang, K. T., Ge, G. N., Liu, M. Q., 2004a. Construction of optimal supersaturated designs by the packing method. Sci. China Ser. A 47, 128–143.

[9] Fang, K. T., Ge, G. N., Liu, M. Q., Qin, H., 2004b. Combinatorial constructions for optimal supersaturated designs. Discrete Math. 279, 191–202.

[10] Fang, K. T., Ge, G. N., Liu, M. Q., Qin, H., 2004c. Construction of uniform designs via super-simple resolvable t-designs. Util. Math. 66, 15–32.

[11] Fang, K. T., Li, R., Sudjianto, A., 2006. Design and Modeling for Computer Experiments. Chapman & Hall, Boca Raton.

[12] John, J. A., Williams, E. R., 1995. Cyclic Designs and Computer Generated Designs. Chapman and Hall, New York.

[13] Kuhfeld, W. F., Tobias, R. D., 2005. Large factorial designs for product engineering and market research applications. Technometrics 47, 122–132.

[14] Lin, D. K. J., 1993. A new class of supersaturated designs. Technometrics 35, 28–31.

[15] Liu, M. Q., Fang, K. T., 2005. Some results on resolvable incomplete block designs. Sci. China Ser. A 48, 503–512.

[16] Liu, M. Q., Lin, D. K. J., 2008. Construction of optimal mixed-level supersaturated designs. Statist. Sinica, to appear.

[17] Liu, M. Q., Zhang, R. C., 2000. Construction of $E(s^2)$-optimal supersaturated designs using cyclic BIBDs. J. Statist. Plann. Inference 91, 139–150.

[18] Lu, X., Hu, W., Zheng, Y., 2003. A systematical procedure in the construction of multi-level supersaturated designs. J. Statist. Plann. Inference 115, 287–310.

[19] Lu, X., Li, W., Xie, M., 2006. A class of nearly orthogonal arrays. J. Quality Technol. 38, 148–161.

[20] Lu, X., Sun, Y., 2001. Supersaturated designs with more than two levels. Chinese Ann. Math. B 22, 183–194.

[21] Ma, C. X., Fang, K. T., Liski, E., 2000. A new approach in constructing orthogonal and nearly orthogonal arrays. Metrika 50, 255–268.

[22] Nguyen, N-K., 1994. Construction of optimal incomplete block designs by computer. Technometrics 36, 300–307.

[23] Nguyen, N-K., 1996a. An algorithmic approach to constructing supersaturated designs. Technometrics 38, 205–209.

[24] Nguyen, N-K., 1996b. A note on the construction of near-orthogonal arrays with mixed levels and economic run size. Technometrics 38, 279–283.

[25] Nguyen, N-K., Borkowski, J. J., 2008. New 3-level response surface designs constructed from incomplete block designs. J. Statist. Plann. Inference 138, 294–305.

[26] Nguyen, N-K., Cheng, C. S., 2008. New $E(s^2)$-optimal supersaturated designs constructed from incomplete block designs. Technometrics 50, 26–31.

[27] Nguyen, N-K., Piepel, G. F., 2005. Computer-generated experimental designs for irregular-shaped regions. Quality Technol. Quant. Manag. 2, 147–160.

[28] Rao, C. R., 1947. Fractional experiments derivable from combinatorial arrangements of arrays. J. Roy. Statist. Soc. Suppl. 9, 128–139.

[29] 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.

[30] Wang, J. C., Wu, C. F. J., 1992. Nearly orthogonal arrays with mixed levels and small runs. Technometrics 34, 409–422.

[31] Wu, C. F. J., 1993. Construction of supersaturated designs through partially aliased interactions. Biometrika 80(3), 661–669.

[32] Xu, H., 2002. An algorithm for constructing orthogonal and nearly-orthogonal arrays with mixed levels and small runs. Technometrics 44, 356–368.