J. Ind. Soc. Agril. Statist.
62(2), 2008 : 183-194

Computer-Generated Efficient Two-Level Supersaturated Designs

V.K. Gupta, Rajender Parsad, Basudev Kole and Lalmohan Bhar
Indian Agricultural Statistics Research Institute, New Delhi

SUMMARY

The co-ordinate column-wise exchange algorithms of Nguyen (1996) and Lejeune (2003) have been modified to generate efficient supersaturated designs (SSDs) for two-level factorial experiments. Designs have been generated for two different settings viz. (a) the design is balanced in the sense that for an n run design, the +1 and –1 levels appear n/2 times each for all the factors; (b) the design is balanced and there is a subset of factors that are mutually orthogonal. The upper bound on the number of active factors that are estimable through the design has also been obtained by computing the rank of design matrix **X** using the necessary condition of Srivastava (1975). Catalogues of efficient SSDs have been prepared for number of runs $10 \le n \le 20$, n = 2t in (a) and n = 4t in (b), t a positive integer, and number of factors $n + 2 \le m \le 2n$. A comparison of the designs generated has been made with the designs available in the literature.

Key words: Co-ordinate column-wise exchange algorithm, E(s^2)-efficiency criteria.

1. INTRODUCTION

In industrial, biological and agricultural experiments, there occur situations where a large number of factors are to be tested but only few of the factors are active. It is not known as to how many and which of the many factors tried in the experiment are active. The experimenter's endeavor is to minimize the number of runs to identify the active factors for efficient utilization of resources and minimization of cost and time. A Supersaturated Design (SSD) is essentially a fractional factorial design in which the degrees of freedom for all its main effects and the intercept term exceed the number of design runs. The huge advantage of these designs is that they reduce the experimental cost drastically. But their critical disadvantage is the confounding involved in the statistical analysis. Because of their run size economy, these designs can be broadly exploited to screen active factor main effects when experimentation is expensive and the number of potentially active factors is large.

SSDs were introduced by Satterthwaite (1959) for two-level factorials. Satterthwaite suggested the construction of balanced SSDs by a randomization procedure. A design is said to be balanced if the number of times each level appears in a column is same, i.e., for a balanced two-level design the number of +1's and –1's is equal in each column of the design. Let $\mathbf{X} = (x_{ij})$ be an $n \times m$ design matrix for a factorial experiment in m factors and n runs. For a two-level design, $x_{ij} = +1$ or –1. The design matrix **X** is called orthogonal if $\mathbf{X}'\mathbf{X}$ is a diagonal matrix. Let s_{ij} be the $(i, j)^{th}$ element of $\mathbf{X}'\mathbf{X}$. For a two-level factors SSD, $n - 1 < m$, and the inner product of two columns does not always vanish to zero, *i.e.*, the matrix $\mathbf{X}'\mathbf{X}$ is never a diagonal matrix. Therefore, a distinctive aspect of SSDs is the measure of non-orthogonality between two columns. The popular $E(s^2)$ criterion, proposed by Booth and Cox (1962), is to minimize

$$E(s^2) = \sum_{1 \le i < j \le m} s_{ij}^2 \Big/ \binom{m}{2}. \tag{1.1}$$

Nguyen (1996) obtained a lower bound to $E(s^2)$ as

$$E(s^2) \ge \frac{n^2(m-n+1)}{(m-1)(n-1)}. \tag{1.2}$$

Bulutoglu and Cheng (2004) gave an improved bound to $E(s^2)$ which is given in the following theorem:

Theorem 1.1. (Bulutoglu and Cheng, 2004). Suppose m is a positive integer such that $m > (n - 1)$. Then there is a unique q such that $-2n + 2 < m - q(n - 1) < 2n - 2$ and $(m + q) \equiv 2(\text{mod } 4)$. Let $g(q) = (m + q)^2 n - q^2 n^2 - mn^2$.

1. If $n \equiv 0$ (mod 4), then

$$E(s^2) \geq \frac{g(q) + 2n^2 - 4n}{m(m-1)}, \quad \text{when} \left|m - q(n-1)\right| < n - 1$$

$$= L[E(s^2)]$$

$$E(s^2) \geq \frac{g(q) - 2n^2 + 4n + 4n\left|m - q(n-1)\right|}{m(m-1)}, \quad \text{when } n - 1 < \left|m - q(n-1)\right| \leq \frac{3}{2}n - 2$$

$$= L[E(s^2)]$$

$$E(s^2) \geq \frac{g(q) + 4n^2 - 4n}{m(m-1)}, \quad \text{when} \left|m - q(n-1)\right| \geq \frac{3}{2}n - 2$$

$$= L[E(s^2)].$$

2. If $n \equiv 2$(mod 4) and q is even, then

$E(s^2) \geq \max[(h(q), 4)] = L[E(s^2)]$, where

$$h(q) = \frac{g(q) + 2n^2 - 4n + 8}{m(m-1)}, \quad \text{when} \left|m - q(n-1)\right| < n - 1$$

$$h(q) = \frac{g(q) - 2n^2 + 20n + (4n - 8)\,|\,m - q(n-1)\,| - 24}{m(m-1)}, \quad \text{when } n - 1 < \left|m - q(n-1)\right| \leq \frac{3}{2}n - 3$$

$$h(q) = \frac{g(q) + 4n^2 - 4n}{m(m-1)}, \quad \text{when} \left|m - q(n-1)\right| \geq \frac{3}{2}n - 3.$$

3. If $n \equiv 2$(mod 4) and q is odd, then

$E(s^2) \geq \max\ [h(q), 4] = L[E(s^2)]$, where

$$h(q) = \frac{g(q) + 2n^2 - 4n}{m(m-1)}, \quad \text{when} \left|m - q(n-1)\right| < n - 1$$

$$h(q) = \frac{g(q) - 2n^2 + 4n + 4n\left|m - q(n-1)\right|}{m(m-1)}, \quad \text{when } n - 1 < \left|m - q(n-1)\right| \leq \frac{3}{2}n - 1$$

$$h(q) = \frac{g(q) + 4n^2 - 12n + 8\left|m - q(n-1)\right| + 8}{m(m-1)}, \quad \text{when } n\left|m - q(n-1)\right| \geq \frac{3}{2}n - 1$$

This lower bound may be used for judging the suitability of a SSD by computing lower bound to its efficiency defined as

$$\text{Efficiency} = \frac{L\left[E(s^2)\right]}{E(s^2)}. \tag{1.3}$$

A design is optimal when the lower bound to efficiency is equal to 1.00. A design with high efficiency is an acceptable design. A large number of methods of construction of SSDs are available in literature. For excellent reviews on SSDs for two level factorials, one may refer to Lin (1993, 2000), Cheng (1997), Gupta and Chatterjee (1998) and Xu and Wu (2003) and references cited therein. For a more recent update on SSDs, one may refer to Nguyen and Cheng (2008). These methods of construction are algebraically quite involved. Therefore, it was thought to generate designs through computer-aided search for any given parametric combination (n, m). The attempt is also to obtain designs that have high efficiency in terms of $E(s^2)$. However, the generation ease compensates for the loss in efficiency that one may have to bear. There might be still a scope to improve the algorithm to improve the efficiency of the design generated.

For computer aided construction of SSDs a column-wise exchange algorithm has been first proposed by Heavlin and Finnegan (1993, 1998). In this procedure level of one or more factors are changed. Nguyen (1996) proposed algorithms for computer-aided search of efficient SSDs by repeatedly searching a pair of coordinates having different signs in j^{th} column of **X** such that the swap of these coordinates results in biggest reduction in $E(s^2)$. $E(s^2)$, **X** and **X′X** are then updated. This is repeated for every column j of **X**, j = 1, 2 ..., m.

Li and Wu (1997) and Lejeune (2003) proposed another sort of column-wise exchange algorithm which consists of selecting one column and trying to increase the efficiency of the design by switching one pair of elements of this column. Kohli (2006) also developed an algorithm for computer-aided search of efficient SSDs.

In the present investigation, we have modified algorithm proposed by Nguyen (1996) and coordinate column-wise exchange algorithm of Lejeune (2003) for construction of SSDs. The algorithm has been developed for two different situations, viz. (a) the design is balanced in the sense that for an n run design, the +1 and –1 levels appear n/2 times each for all the factors; (b) the design is balanced and there is a subset of factors that are mutually orthogonal in the sense that the inner product of any two columns in this subset vanishes to zero. The algorithms developed are given in Sections 2.1 and 2.3. The implementation of algorithm is illustrated with the help of an example in Section 2.2. The designs obtained through the algorithm are described in Section 3. A comparison of these designs with the designs available in the literature is also given in Section 3. A discussion on some future directions of research in this area is given in Section 4.

2. COLUMN-WISE COORDINATE EXCHANGE ALGORITHM FOR GENERATION OF EFFICIENT SUPERSATURATED DESIGNS

In this section, we describe the column-wise coordinate exchange algorithm for generation of two-level SSDs. The algorithm is an amalgamation and modification of the algorithms of Nguyen (1996) and Lejune (2003). The algorithm enables to generate designs for the following situations:

(a) The design is balanced.

(b) The design is balanced and there is a subset of factors that are mutually orthogonal.

The algorithm is described in the sequel.

2.1 Generation of SSD for Situation (a)

Step1: Input n (the number of runs) and m (the number of factors) $n - 1 < m$, $n = 2t$, t a positive integer. Generate a matrix **X** of order $n \times m$ with entries ±1 in the m columns such that number of + 1's is equal to the number of –1's in each of the m columns. Randomize the order of +1 and –1 in each column of **X**.

Step 2: Compute $\mathbf{X'X} = (s_{ij});\ i, j = 1, 2, ..., m$ and $E(s^2)$ as defined in (1.1).

Step 3: Select column k of **X** where $S_k^2 = \max_{(j=1,2,\ ...,\ m)} S_j^2$, and $S_j^2 = \sum_{i(\neq j)=1}^{m} s_{ij}^{2}$.

Step 4: For column k in Step 3, repeat searching a pair of coordinates having different signs in this column such that the swap of these coordinates will result in biggest reduction in $E(s^2)$. Update $E(s^2)$, **X** and **X′X**.

Step 5: Repeat Steps 3 and 4 until $E(s^2)$ becomes stable (cannot be reduced further) or $E(s^2)$ reaches its lower bound as defined in Theorem 1.1.

Step 6: Compute the lower bound to efficiency of the design as defined in (1.3). Steps 1-5 make up one attempt or try. 100 tries were given for each (n, m) combination. The result of the best try is reported.

The algorithm just described is an improvement over the algorithm of Nguyen (1996) in the sense that we select a column k of **X** where $S_k^2 = \max_{(j\ =\ 1,\ 2,\ ...,\ m)} S_j^2$, and then exchange the elements with opposite signs in column k to reduce $E(s^2)$ rather than exchange elements of opposite signs in every column of **X**, as in case of Nguyen (1996). Further, we have used improved lower bound given by Bulutoglu and Cheng (2004) for computation of efficiency of the generated SSD, which was not the case with Nguyen (1996) and Lejeune (2003). Generation of SSDs with a subset of mutually orthogonal factors (described in Section 2.3) is another modification of the algorithms of Nguyen (1996) and Lejeune (2003).

2.2 Implementation of Algorithm

We now describe the implementation of the algorithm through an example given in Kohli (2006). Suppose that a tractor manufacturer wants to study the effect of safety features on tractor's safety system using the following 11 equipments / tools: Roll Over Protection Structures (ROPS), Reflectors, Seat Belts, Enclosed Cabins, Power Take Off (PTO) Cover, Integrated Brake System, Counter Weights, Rear View, Spark (exhaust pipe), Hydraulic (fluid) Power, Hand Support System (for mounting/dismounting). Each safety system could either be present or absent. This is a typical factorial experiment. Single replication of full factorial would require 2048 tractors to be tested. No tractor manufacturing company would allow so many tractors to be tested. The main interest of the experimenter is to identify important safety systems to be imbedded in the tractor. Such experimentation would require a design that can estimate orthogonally the mean and the main effects under the assumption that two factor and higher order interactions are absent. A saturated orthogonal main effect plan in 12 runs obtained through an Hadamard matrix of order 12 may be useful for this situation. However, the tractor manufacturing company allows only 8 tractors to be tested. The problem now is how to design an experiment with 8 runs for identification of active factors out of these 11 factors. The above algorithm was used for generation of a SSD for 11 factors in 8 runs.

The randomly selected design at Step 1 through the algorithm is

–1	1	–1	–1	–1	–1	1	1	–1	1	1
–1	1	1	1	1	1	1	1	1	1	1
1	–1	–1	1	1	1	–1	–1	–1	1	–1
1	–1	1	–1	1	1	1	1	1	–1	–1
1	1	1	–1	–1	–1	–1	–1	1	1	–1
–1	1	–1	–1	–1	–1	–1	–1	1	–1	1
1	–1	1	1	1	1	–1	–1	–1	–1	1
–1	–1	–1	1	–1	–1	1	1	–1	–1	–1

The design is balanced because in each column, there are four +1's and four –1's.

Computation of **X′X** and $E(s^2)$

X′X

8	–4	4	0	4	4	–4	–4	0	0	–4
–4	8	0	–4	–4	–4	0	0	4	4	4
4	0	8	0	4	4	0	0	4	0	0
0	–4	0	8	4	4	0	0	–4	0	0
4	–4	4	4	8	8	0	0	0	0	0
4	–4	4	4	8	8	0	0	0	0	0
–4	0	0	0	0	0	8	8	0	0	0
–4	0	0	0	0	0	8	8	0	0	0
0	4	4	–4	0	0	0	0	8	0	0
0	4	0	0	0	0	0	0	0	8	0
–4	4	0	0	0	0	0	0	0	0	8

$$S_j^2 = \sum_{i(\neq j)=1}^{m} s_{ij}^2 =$$

112	112	64	64	128	128	80	80	48	16	32

$$\sum_{1 \le i < j \le m} s_{ij}^2 =$$

112	96	48	48	64	0	64	0	0	0	0

The column with maximum value of S_j^2 is to be selected for modification. Here S_5^2 and S_6^2 both are same as 128. The algorithm selects the last column with maximum S_j^2 ; column 6 is, therefore, selected for modification. All possible coordinate exchange steps are now implemented in this column and for each exchange the value of $E(s^2)$ is computed. The best exchange is when coordinates of rows 6 and 7 are exchanged in this column. After the exchange of the mentioned coordinates, $E(s^2)$ is reduced to 6.1091 and the design becomes

−1	1	−1	−1	−1	−1	1	1	−1	1	1
−1	1	1	1	1	1	1	1	1	1	1
1	−1	−1	1	1	1	−1	−1	−1	1	−1
1	−1	1	−1	1	1	1	1	1	−1	−1
1	1	1	−1	−1	−1	−1	−1	1	1	−1
−1	1	−1	−1	−1	1	−1	−1	1	−1	1
1	−1	1	1	1	−1	−1	−1	−1	−1	1
−1	−1	−1	1	−1	−1	1	1	−1	−1	−1

Now again repeat the same steps and identify the column with maximum S_j^2. The algorithm selects column 2 for modification. All the possible co-ordinate exchange steps are now implemented in column 2 and for each exchange the value of $E(s^2)$ is computed. The best exchange is when co-ordinates of rows 1 and 8 are exchanged in column 2. After exchange of the co-ordinates, the new design becomes

−1	−1	−1	−1	−1	−1	1	1	−1	1	1
−1	1	1	1	1	1	1	1	1	1	1
1	−1	−1	1	1	1	−1	−1	−1	1	−1
1	−1	1	−1	1	1	1	1	1	−1	−1
1	1	1	−1	−1	−1	−1	−1	1	1	−1
−1	1	−1	−1	−1	1	−1	−1	1	−1	1
1	−1	1	1	1	−1	−1	−1	−1	−1	1
−1	1	−1	1	−1	−1	1	1	−1	−1	−1

For this design $E(s^2) = 5.2364$. Now again repeat the same steps and identify the column with maximum S_j^2. The algorithm selects column 1 for modification. All the possible co-ordinate exchange steps are now implemented in column 1 and for each exchange the value of $E(s^2)$ is computed. The best exchange is when coordinates of rows 1 and 7 are exchanged in column 1. After exchange of the co-ordinates, the new initial design is

1	−1	−1	−1	−1	−1	1	1	−1	1	1
−1	1	1	1	1	1	1	1	1	1	1
−1	−1	−1	1	1	1	−1	−1	−1	1	−1
1	−1	1	−1	1	1	1	1	1	−1	−1
1	1	1	−1	−1	−1	−1	−1	1	1	−1
−1	1	−1	−1	−1	1	−1	−1	1	−1	1
1	−1	1	1	1	−1	−1	−1	−1	−1	1
−1	1	−1	1	−1	−1	1	1	−1	−1	−1

For this design $E(s^2) = 4.6545$. At this step, we see that the value of $E(s^2)$ attains the lower bound of Theorem 1.1. The efficiency of the design selected is then computed and it comes out 1.000 and the search is terminated here.

2.3 Generation of SSD for Situation (b)

In this situation, the SSD has q orthogonal columns. The remaining m – q columns need not be orthogonal. The design has n runs. Let a semi-normalized Hadamard matrix $\mathbf{H}_n$, of order n, be expressed as $\mathbf{H}_n = [\mathbf{1}\ \mathbf{A}^*\ \mathbf{A}]$. Here $\mathbf{1}_u$ is a u-component vector of all ones, $\mathbf{A}^*$ is an $n \times q$ matrix and $\mathbf{A}$ is an $n \times (n - q - 1)$ matrix of remaining $n - q - 1$ columns of $\mathbf{H}_n$. The q orthogonal columns in SSD are obtained by selecting $\mathbf{A}^*$. Obviously, in this situation, $n \equiv 0 \bmod(4)$.

The steps in the generation of SSDs for this situation are similar to those in Section 2.1 except that in Step 1 we augment a matrix $\mathbf{X}^*$ with entries ±1 of size $n \times (m - q)$ to $\mathbf{A}^*$. This makes $\mathbf{X} = [\mathbf{A}^*\ \ \mathbf{X}^*]$ the required initial design. In Step 4, exchanges only take place in column k belonging to $\mathbf{X}^*$. However, for column k belonging to $\mathbf{A}^*$, swap the entire row of $\mathbf{A}^*$ instead of swapping the coordinates in the selected column and compute the value of $E(s^2)$ for this pair of exchange. Repeat searching for a pair of rows such that the swap of these rows will result in biggest reduction in $E(s^2)$. Update $E(s^2)$, $\mathbf{X}$ and $\mathbf{X'X}$.

3. CATALOGUE OF DESIGNS OBTAINED

The above algorithm was converted into a C program and this program was used for computer aided search of efficient designs. Using this program, one can generate a large number of designs for given parameters (n, m, q). However, catalogues of efficient SSDs have been prepared for number of runs, $10 \leq n \leq 20$, n = 2t in (a) and n = 4t in (b), t a positive integer, and number of factors $n + 2 \leq m \leq 2n$. Some selected designs are given in Tables 1.1 and 1.2 in the Appendix. These designs are also available at design resources server under the link (http://www.iasri.res.in/design/Supersaturated_Design/Supersaturated.html).

For the designs generated through the algorithm, Rank(**X**) was also obtained. If Rank(**X**) is a, then maximum number of active factors that are estimable through the design is [a/2], where [.] denotes the greatest integer function. This is based on the necessary condition given by Srivastava (1975). The tables also give the upper bound to the number of active factors that can be captured from the design. The value of $r_{max} = \max\left(\left|r_{ij}\right|\right)$, where r_{ij} is the correlation between columns i and j of **X** is also given in the Tables 1.1 and 1.2. This is another measure of goodness of the design. Smaller values of r_{max} indicates that the design is good.

A comparison of the designs generated by the proposed algorithm has been made with those catalogued by Nguyen (1996) and also those obtained by other authors and listed in the catalogue of Nguyen (Table 2 of Nguyen). This comparison is given in Table 1.3 in the Appendix. In this catalogue, there were some designs which were outside the parametric range considered in the present investigation. It can easily be seen that for the parametric combinations catalogued by Nguyen (1996), the designs obtained in the present investigation are more efficient than those obtained by Booth and Cox (1962), Lin (1993), Wu (1993) and are only inferior to the balanced incomplete block design (BIBD)-based SSDs of Nguyen (1996).

4. DISCUSSION

In this article a computer aided search is made for efficient two-level SSDs. The designs are generated for two situations viz. (a) the design is balanced in the sense that for an n run design, the +1 and –1 levels appear n/2 times each for all the factors; (b) the design is balanced and there is a subset of factors that are mutually orthogonal in the sense that the inner product of any two columns in this subset vanishes to zero.

As earlier, let a semi-normalized Hadamard matrix $\mathbf{H}_n$, of order n, be expressed as $\mathbf{H}_n = [\mathbf{1}\ \mathbf{A}^*\ \mathbf{A}]$. For the present investigation, semi-normalized Hadamard matrix has been taken from http://www.iasri.res.in/WebHadamard/WebHadamard.htm. Other choices of a Hadamard matrix, however, are possible. For making a random start of the initial design for the case (b) in Section 2.3, the algorithm at step 1 selects q columns of $\mathbf{A}^*$. The remaining m – q columns are selected randomly. However, one can select any q columns out of $\binom{n-1}{q}$ possibilities from $[\mathbf{A}^*\ \ \mathbf{A}]$. We illustrate this for q = 2. For n = 8 and q = 2, we choose first two columns of $\mathbf{A}^*$ of a semi-normalized Hadamard matrix of order 8. However, this is not the only choice. It is a matter of convenience only. One can choose any two columns from $[\mathbf{A}^*\ \ \mathbf{A}]$ in 21 ways. This would lead to 21 randomly selected designs at Step 1 of the algorithm. It is felt that each random choice may lead to a design with different efficiency. The efficiencies of all the 21 different combinations were calculated and it was found that the efficiency of the design changes with the choice of the random design generated at step 1 of the algorithm in Section 2.3. This fact has also been implemented in the algorithm to obtain the most efficient designs. The designs generated by this approach are given in Table 1.2 and are marked with an asterisk (*). However, a similar idea needs to be implemented for the case when one needs more than 2 orthogonal factors.

We have seen above that the random choice of a design is made by selecting any of the pairs of $\binom{n-1}{2}$ columns of **A** and then making a fresh search for m – 2 columns as well. Yet another approach could be to start with a search for the most efficient design for given n and m – 2. Having obtained the most efficient design, add two columns out of $\binom{n-1}{2}$ pair of columns and obtain the efficiency of this design. Repeat this for all the $\binom{n-1}{2}$ pair of columns and obtain the efficiency of the resulting design. Pick the design with the maximum efficiency. For example, to search for an efficient SSD for n = 8, m = 16, we begin the search for an efficient design for n = 8, m = 14 and q = 0. When the first two columns of $\mathbf{A}^*$ obtained from $\mathbf{H}_8$ are added to this design, the resulting design has efficiency 0.894. However, if we replace the two columns by the remaining 20 pairs of columns, we get designs with different efficiencies. The design with maximum efficiency as 0.9143 is obtained for the pair of columns (2, 5). The design obtained is

n = 8, m = 16, q = 2, $E(s^2) = 6.000$; Lower bound = 5.867, Efficiency = 0.9778; Rank ($\mathbf{X}$) = 7, Upper bound number of active factors that can be detected is 3 (columns are runs) :

$$\begin{matrix}
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
-1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 \\
-1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 \\
-1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 \\
-1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\
-1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\
-1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 \\
-1 & -1 & 1 & -1 & -1 & 1 & 1 & 1 \\
-1 & 1 & 1 & -1 & -1 & 1 & 1 & -1
\end{matrix}$$

Finally, the algorithm needs to include some steps for the analysis of the design. To begin with, the stepwise regression analysis can be included. This would make the search complete.

ACKNOWLEDGEMENTS

The authors express their gratefulness to the referee for making some very useful suggestions that has led to a considerable improvement in the manuscript.

REFERENCES

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

Bulutoglu, Dursun A. and Cheng, C.S. (2004). Construction of $E(s^2)$-optimal supersaturated designs. *Ann. Statist.,* **32(4)**, 1662-1678.

Cheng, C.S. (1997). $E(s^2)$-optimal supersaturated designs. *Statist. Sinica,* **7(4)**, 929-939.

Gupta, S. and Chatterjee, K. (1998). Supersaturated designs: A review. *J. Combi.: Info. Sys. Sci.,* **23(1-4)**, 475-488.

Heavlin, W. D. and Finnegan, G. P. (1993). Adapting experiments with sequentially processed factors. *Proceedings of the Physical and Engineering Sciences Section, American Statistical Association,* 44- 49.

Heavlin, W.D. and Finnegan, G.P. (1998). Columnwise construction of response surface designs. *Statist. Sinica,* **8 (1)**, 185-206.

Kohli, P. (2006). A Study on Supersaturated Designs. Unpublished M.Sc. Thesis, IARI, New Delhi.

Lejeune, Miguel A. (2003). A coordinate-columnwise exchange algorithm for the construction of supersaturated, saturated, and non-saturated experimental designs. Modern mathematical, management, and statistical sciences. II: Advances in theory and application. *Amer. J. Math. Manage. Sci.,* **23 (1-2)**, 109–142.

Li, W.W. and Wu, C.F.J. (1997). Columnwise-pairwise algorithms with applications to the construction of supersaturated designs. *Technometrics,* **39**, 171-179.

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

Lin, D. K.J. (2000). Supersaturated design: Theory and application. *Statistical Process Monitoring and Optimization*, S.H. Park and G.G. Vining (eds.), Marcel Dekker.

Nguyen, N.K. (1996). An algorithmic approach to constructing supersaturated designs. *Technometrics,* **38**, 69-73.

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

Satterthwaite, F.E. (1959). Random balance experimentation. *Technometrics,* **1**, 111-137.

Srivastava, J.N. (1975). Designs for searching non-negligible effects. In : *A Survey of Statistical Designs and Linear Models*. J.N. Srivastava (ed.), North Holland, Amsterdam, 507-519.

Wu, C.F.J. (1993). Construction of supersaturated designs through partially aliased interactions. *Biometrika,* **80(3)**, 661-669.

Xu, H. and Wu, C.F.J. (2003). *Construction of Optimal Multi-level Supersaturated Designs*. UCLA, Department of Statistics, Electronic Publication.(http://repositories.cdlib.org/uclastat).

APPENDIX

Table 1.1. Two-level SSDs

Number of runs	Number of factors	$E(s^2)$	$L[E(s^2)]$	Efficiency	Upper bound to number of active factors	r_{max}
10	14	5.407	5.0549	0.935	4	0.600
10	15	7.048	5.5238	0.784	4	0.600
10	16	6.667	5.8667	0.880	4	0.600
10	17	6.588	5.8824	0.893	4	0.600
10	18	6.510	5.8824	0.904	4	0.600
10	19	6.620	6.4323	0.972	4	0.600
12	16	5.460	5.2000	0.952	5	0.333
12	18	6.379	5.9608	0.934	5	0.333
12	19	6.643	6.4561	0.972	5	0.333
12	20	6.989	6.8211	0.976	5	0.667
12	21	7.314	6.8571	0.938	5	0.333
12	22	7.550	6.8571	0.908	5	0.667
12	23	8.095	7.3990	0.914	5	0.667
12	24	8.116	7.8260	0.964	5	0.667
14	16	5.600	4.0000	0.714	6	0.429
14	17	7.059	4.9412	0.700	6	0.429
14	18	6.510	5.6732	0.871	6	0.429
14	19	7.368	6.0585	0.822	6	0.429
14	20	7.705	6.3579	0.825	6	0.429
14	21	8.114	6.6667	0.822	6	0.429
14	22	7.879	6.9091	0.877	6	0.429
14	23	8.300	7.4150	0.893	6	0.429
14	24	8.406	7.8261	0.931	6	0.714
14	25	8.587	7.8400	0.913	6	0.714
14	26	9.218	7.8400	0.851	6	0.429
14	27	9.652	8.3875	0.869	6	0.429
14	28	9.503	8.8042	0.926	6	0.714
16	18	5.961	4.1830	0.702	7	0.250
16	19	6.737	4.4912	0.667	7	0.250
16	20	7.747	5.3895	0.696	7	0.500
16	21	7.543	6.0952	0.808	7	0.500
16	22	7.550	6.6494	0.881	7	0.500
16	23	8.221	7.0830	0.862	7	0.500
16	24	8.232	7.4203	0.901	7	0.500
16	25	9.067	7.6800	0.847	7	0.500
16	26	9.305	7.8769	0.847	7	0.500

16	27	8.889	8.3874	0.944	7	0.500
16	28	9.354	8.8042	0.941	7	0.500
16	29	9.576	8.8275	0.922	7	0.500
16	30	10.262	8.8275	0.860	7	0.500
16	31	10.529	9.3591	0.889	7	0.500
16	32	10.129	9.8064	0.968	7	0.750
18	20	7.705	4.0000	0.519	8	0.333
18	21	8.267	5.1428	0.622	8	0.555
18	22	8.433	5.8000	0.688	8	0.333
18	23	8.174	6.1509	0.752	8	0.333
18	24	9.217	6.6667	0.723	8	0.333
18	25	8.907	7.2000	0.808	8	0.555
18	26	9.120	7.6430	0.838	8	0.555
18	27	9.288	8.1020	0.872	8	0.333
18	28	10.688	8.4867	0.794	8	0.555
18	29	9.990	8.6897	0.870	8	0.555
18	30	10.179	8.8552	0.870	8	0.555
18	31	10.125	9.3677	0.925	8	0.555
18	32	10.903	9.8065	0.899	8	0.555
18	33	11.152	9.8182	0.880	8	0.555
18	34	11.301	9.8181	0.869	8	0.555
18	35	11.368	10.3460	0.910	8	0.555
18	36	12.025	10.8063	0.899	8	0.555
20	22	7.965	4.1558	0.522	9	0.400
20	23	7.652	4.4268	0.579	9	0.200
20	24	7.884	5.5072	0.699	9	0.400
20	25	8.427	6.4000	0.759	9	0.400
20	26	8.714	6.8923	0.791	9	0.400
20	27	8.980	7.2934	0.812	9	0.400
20	28	9.905	7.8307	0.791	9	0.400
20	29	10.404	8.2758	0.795	9	0.400
20	30	10.152	8.6436	0.851	9	0.400
20	31	10.529	9.1183	0.866	9	0.400
20	32	10.839	9.5161	0.878	9	0.600
20	33	10.879	9.6969	0.891	9	0.400
20	34	11.180	9.8395	0.880	9	0.400
20	35	11.213	10.3520	0.923	9	0.400
20	36	11.911	10.7936	0.906	9	0.400
20	37	11.772	10.8108	0.918	9	0.400
20	38	12.222	10.8108	0.885	9	0.400
20	39	12.588	11.3360	0.901	9	0.400
20	40	12.964	11.7940	0.910	9	0.600

Table 1.2. Two level SSDs with a subset of mutually orthogonal factors. Maximum has been picked up from all possible two level combinations.

Number of runs	Number of factors	Number of orthogonal factors	$E(s^2)$	$L[E(s^2)]$	Efficiency	columns chosen	Upper bound to number of active factors	r_{max}
12*	14	2	8.440	4.2198	0.500	3, 7	5	0.667
12	15	2	6.857	4.5714	0.667	7, 8	5	0.667
12	16	2	7.067	5.2000	0.736	5, 7	5	0.333
12	17	2	7.176	5.6471	0.787	4, 6	5	0.333
12	17	10	6.353	5.6471	0.889	1 - 10	5	0.667
12*	18	2	7.320	5.9608	0.814	1, 3	5	0.333
12	18	9	6.797	5.9608	0.877	1 - 9	5	0.667
12	18	10	6.797	5.9608	0.877	1 - 10	5	0.667
12*	19	2	7.860	6.4561	0.821	3, 10	5	0.667
12	19	8	7.579	6.4561	0.852	1 - 8	5	0.667
12	19	9	7.111	6.4561	0.908	1 - 9	5	0.667
12	19	10	6.737	6.4561	0.958	1 - 10	5	0.667
12*	20	2	7.747	6.8211	0.880	3, 11	5	0.333
12	20	4	7.663	6.8211	0.890	1 - 4	5	0.667
12	20	6	8.253	6.8211	0.826	1 - 6	5	0.667
12	20	9	7.663	6.8211	0.890	1 - 9	5	0.667
12	20	10	7.074	6.8211	0.964	1 - 10	5	0.667
12*	21	2	7.924	6.8571	0.865	2, 5	5	0.667
12	21	6	7.619	6.8571	0.900	1 - 6	5	0.667
12	21	7	8.457	6.8571	0.810	1 - 7	5	0.667
12	21	8	7.390	6.8571	0.928	1 - 8	5	0.667
12	21	9	7.543	6.8571	0.909	1 - 9	5	0.667
12	21	10	8.000	6.8571	0.857	1 - 10	5	0.667
12*	22	2	8.312	6.8571	0.825	9, 11	5	0.667
12	22	3	9.004	6.8571	0.762	1, 2, 3	5	0.667
12	22	4	9.004	6.8571	0.762	1 - 4	5	0.667
12	22	5	8.380	6.8571	0.818	1 - 5	5	0.667
12	22	6	8.589	6.8571	0.798	1 - 6	5	0.667
12	22	7	7.688	6.8571	0.892	1 - 7	5	0.667
12	22	8	8.035	6.8571	0.853	1 - 8	5	0.667
12	22	9	8.589	6.8571	0.798	1 - 9	5	0.667
12	22	10	7.550	6.8571	0.908	1 - 10	5	0.667
12*	23	2	8.032	7.3990	0.921	1, 4	5	0.667
12	23	7	7.905	7.3990	0.936	1 - 7	5	0.667
12	23	8	7.842	7.3990	0.944	1 - 8	5	0.667

12	23	9	8.095	7.3990	0.914	1 - 9	5	0.667
12	23	10	7.399	7.3990	1.000	1 - 10	5	0.667
12*	24	2	8.174	7.8260	0.957	6, 9	5	0.333
12	24	3	8.812	7.8260	0.888	1, 2, 3	5	0.667
12	24	6	9.043	7.8260	0.865	1 - 6	5	0.667
12	24	7	8.174	7.8260	0.957	1 - 7	5	0.667
12	24	8	8.058	7.8260	0.971	1 - 8	5	0.667
12	24	9	8.174	7.8260	0.957	1 - 9	5	0.667
12	24	10	7.942	7.8260	0.985	1 - 10	5	0.667
16*	24	2	10.551	7.4203	0.703	1, 7	7	0.500
16*	25	2	10.613	7.6800	0.724	8, 15	7	0.500
16*	26	2	10.437	7.8769	0.755	6, 13	7	0.500
16*	27	2	10.849	8.3874	0.773	13, 14	7	0.500
16*	28	2	11.132	8.8042	0.791	4, 9	7	0.500
16*	29	2	10.877	8.8275	0.812	4, 15	7	0.500
16*	30	2	11.145	8.8275	0.792	12, 13	7	0.500
16*	31	2	11.286	9.3591	0.829	3, 11	7	0.500
16*	32	2	11.323	9.8064	0.866	12, 13	7	0.750
20*	22	2	13.576	4.1558	0.306	2, 10	9	0.400
20*	23	2	13.091	4.4268	0.338	6, 18	9	0.400
20*	24	2	13.217	5.5072	0.417	7, 19	9	0.400
20*	25	2	12.747	6.4000	0.502	9, 12	9	0.400
20*	26	2	13.785	6.8923	0.500	11, 17	9	0.400
20*	27	2	12.627	7.2934	0.578	1, 16	9	0.400
20*	28	2	13.630	7.8307	0.575	10, 15	9	0.400
20*	29	2	13.438	8.2758	0.616	5, 8	9	0.600
20*	30	2	13.720	8.6436	0.630	2, 14	9	0.400
20*	31	2	13.935	9.1183	0.654	1, 17	9	0.400
20*	32	2	14.226	9.5161	0.669	9, 17	9	0.600
20*	33	2	13.121	9.6969	0.739	11, 17	9	0.400
20*	34	2	13.861	9.8395	0.710	1, 7	9	0.600
20*	35	2	14.682	10.3520	0.705	4, 10	9	0.600
20*	36	2	14.222	10.7936	0.759	7, 19	9	0.400
20*	37	2	14.751	10.8108	0.733	1, 6	9	0.400
20*	38	2	14.566	10.8108	0.742	4, 17	9	0.600
20*	39	2	13.970	11.3360	0.811	1, 6	9	0.600
20*	40	2	14.441	11.7940	0.817	2, 16	9	0.600

* indicates that the design reported is with maximum efficiency among all designs generated by taking all possible pairs of **A** in $\mathbf{H}_n$.

Table 1.3. Comparison of designs obtained (q = 0) with designs given in Table 2 of Nguyen (1996)

n	m	$E(s^2)$				
		Booth & Cox (1962)	Lin (1993)	Wu (1993)	Nguyen (1996)	Obtained through proposed algorithm
6	10	–	4.00	–	4.00	4.00
8	14	–	–	–	4.92	4.92$
10	18	–	5.88	–	5.88	6.51
12	16	7.06	6.27	6.00	5.20	5.47
12	18	9.68	6.59	6.59	5.96	6.38
12	22	–	6.86	7.40	6.86	7.55
12	24	10.26	–	8.17	7.83	7.94$
14	26	–	7.84	–	7.84	9.22
16	30	–	–	–	8.83	10.26
18	24	13.04	9.22	–	7.13	9.22
18	30	15.36	9.74	–	9.37	10.18
18	34	–	9.82	–	9.82	11.30
18	36	16.44	–	–	10.96	12.03
20	38	–	–	11.36	10.81	12.22
22	42	–	11.80	–	11.80	14.52
24	30	12.06	11.59	9.27	7.91	9.60
24	46	–	12.80	13.29	12.80	15.20
26	49	–	13.80	–	13.80	15.62
26	50	–	–	–	13.80	15.91
28	54	–	–	15.33	14.79	16.94
30	58	–	15.79	–	15.79	17.69

$ indicates that the design has some orthogonal factors.