RASHI 1 (2) :41-50 (2016)

Construction of optimal Two- Level Supersaturated Designs

Basudev Kole and Gourav Kumar Rai
Department of Statistics, Mathematics & Computer Application
Bihar Agricultural University, Bihar, India

ABSTRACT

Gupta et. al. (2008, 2010) constructed two level SSDs through computer algorithm. They used lower bound to $E(s^2)$ to measure the efficiency of the generated design. In this article we modify the algorithm of Gupta et. al. (2008, 2010) to construct two level SSDs for both balance as well as nearly balanced cases. Instead $E(s^2)$ we have used r_{max} and f_{max} criterion to compare the design among the class of $E(s^2)$ optimal designs. This algorithm generates design in such a way that all the columns of the design matrix are distinct and no two columns are fully aliased. A catalogue of 30 optimal supersaturated designs which are best with respect to r_{max} and f_{max} criterion among the class of $E(s^2)$ optimal SSDs have also been prepared.

AMS Subject Classification: *62K15, 62K05, 62K99.*

Keywords : Co-ordinate exchange algorithm, efficiency criteria, supersaturated design.

1. Introduction

A design having n runs and m factors is a supersaturated design (SSD) when the degrees of freedom for all the main effects of the m factors and the intercept term exceed the total number of distinct factor level combinations, n, of the design. The huge advantage of these designs is the reduction in the experimental cost drastically, but the critical disadvantage is the confounding involved while analyzing. As the run size is very less and the experimentation cost is very less SSDs have been studied extensively in the literature.

Supersaturated designs (SSDs) were introduced by Satterthwaite (1959) for two-level factorials. He used randomization procedure to construct balanced SSDs. Booth and Cox (1962) also introduced SSDs for two-level factors. Since then, number of research works have been done in generating two-level SSDs (see e.g., Bulutoglu and Cheng, 2004; Bulutoglu, 2007; Ryan and Bulutoglu, 2007; Gupta et al., 2008; Das *et al*., 2008; Nguyen and Cheng, 2008; Suen and Das, 2010).

Let $X = (X_{ij})$ be an $n \times m$ design matrix with $X_{ij} = \pm 1$. Let s_{ij} be the $(i, j)^{th}$ element of $X'X$. The design matrix X is called column-orthogonal if $X'X$ is a diagonal matrix. As soon as $m > n$, in case of SSD, the column orthogonality is disturbed, and the column non-orthogonality is measured with the popular $E(s^2) = \sum_{1 \le i < j \le m} s_{ij}^2 \Big/ \binom{m}{2}$ criterion, proposed by Booth and Cox (1962). On the other hand lots of research works have been done in developing the lower bound to both for balanced as well as unbalanced cases.

In this research, we propose a computer algorithm for generating two-level balanced as well as nearly balanced SSDs. The algorithm generates efficient SSDs for both the cases. The algorithms proposed by Nguyen (1996), Lejeune (2003), Ryan and Bulutoglu (2007), Gupta *et al*. (2008) and Gupta *et al*. (2010) have been modified to construct efficient SSDs for both balanced as well as nearly balanced cases. The proposed algorithm have been constructed in such a way that it generates an SSD where all the columns are distinct and no column can be generated from any other column. The algorithm generates balanced SSDs for even n and nearly balanced SSDs for odd n. The algorithm checks the efficiency of the constructed balanced design by using lower bound to $E(s^2)$, given by Das *et al.* (2008) and nearly balanced design by using the lower bound to $E(s^2)$ given by Suen and Das (2010). For balanced SSDs the algorithm has been so constructed that the generated design has less number of orthogonal pair of columns. And for nearly balanced design the constructed design is efficient for all $E(s^2)$, r_{max} and f_{max} criterion.

Some preliminaries are given in Section 2, in Section 3 the proposed algorithm is given. Illustration of the algorithm with an example have been given in Section 4. Conclusion and a small catalogue of 30 optimal designs obtained through the algorithm are also tabulated in section 5.

E-mail : basudevkole@gmail.com

2. PRILIMINARIES

Let the design matrix for a two-level supersaturated design is denoted by X having m factors (columns) and n runs (rows). A design is said to be balanced if the number of +1's and –1's are equal in each column of the design. For a design with *n* runs the number of +1's and –1's in any column is equal to n/2. Obviously, n is even when the design is balanced.

When *n* is odd, the design cannot be balance. For odd *n*, Bulutoglu and Ryan (2008) proposed lower bounds to E(s^2) and also gave a method of generation through computer algorithm. But their algorithm construct the class of designs where the levels +1 appears [n/2] times and the level –1 appears n – $\left[\frac{n}{2}\right]$ times in any column of X, where, [.] denotes the greatest integer function. Bulutoglu and Ryan (2008) obtained a number of optimal SSDs using their algorithm and their proposed lower bounds. Nguyen and Cheng (2008) also provide lower bound to E(s^2) for both even and odd n. Recently Suen and Das (2010)improved the lower bound to E(s^2) for odd *n*.

Gupta et al. (2010) introduce nearly balanced designs for odd *n*. They define nearly balance design where the design is unbalanced but a balance is achieved in the appearance of the two levels +1 and –1 over the columns of X. A design is said to be nearly balanced if the frequencies of occurrence of levels +1 and –1 differ at most by one in such a way that in each of the first $\left[\frac{m}{2}\right]$ columns of X, the frequencies of the occurrence of levels + l and –1 is $\left[\frac{n}{2}\right]$ and n – $\left[\frac{n}{2}\right]$, respectively and in each of the remaining $m-\left[\frac{m}{2}\right]$ columns of X, the frequency of occurrence of levels + l and –1 is n – $\left[\frac{n}{2}\right]$ and $\left[\frac{n}{2}\right]$, respectively, [.] denotes the greatest integer function. For nearly balance design, Gupta et al. (2010) proposed a method of construction through computer algorithm.

For any balanced SSD with m factors and n runs, Nguyen (1996) and Tang and Wu (1997) independently gave the following lower bound to E(s^2):

$$E(s^2) \geq \frac{n^2(m-n+1)}{(m-1)(n-1)} \qquad (2.1)$$

When $n \equiv 0$ (*mod* 4), this bound can be achieved only if m is a multiple of $n - 1$; when $n \equiv 2$ (*mod* 4), *m* needs to be an even multiple of $n - 1$.

Further improvements on the lower bound given by Nguyen (1996) was proposed by Butler *et al*. (2001), Bulutoglu and Cheng (2004), Ryan and Bulutoglu (2007) and Das *et al*. (2008).

Das et al. (2008) proposed the sharper lower bound to E(s^2) for balanced SSDs which is given in the following Theorem.

Theorem 2.1 (Das *et al.*; 2008). For a supersaturated design with n runs and $m = p(n-1) \pm r$ factors (*p* positive, $0 \leq r < \frac{n}{2}$), $E(s^2)$ is greater than or equal to the lower bound LB, where LB is as defined below:

1. Let $n \equiv 0$ (*mod* 4). Then,

$$L.B. = \frac{n^2(m-n+1)}{(n-1)(m-1)} + \frac{n}{m(m-1)}\left\{D(n,r) - \frac{r^2}{n-1}\right\} \qquad (2.2)$$

where

$$D(n,r) = \begin{cases} n+2r-3 & for\ r \equiv 1(mod\ 4), \\ 2n-4 & for\ r \equiv 2(mod\ 4), \\ n+2r+1 & for\ r \equiv 3(mod\ 4), \\ 4r & for\ r \equiv 0(mod\ 4). \end{cases}$$

2. Let $n \equiv 2$ (*mod* 4). Then,

$$L.B. = \max\left\{\frac{n^2(m-n+1)}{(n-1)(m-1)} + \frac{n}{m(m-1)}\left\{D(n,r) - \frac{r^2}{n-1}\right\}, 4\right\} \qquad (2.3)$$

where

(i) When *p* is even,

$$D(n,r) = \begin{cases} n+2r-3+x/n & for\ r \equiv 1(mod\ 4), \\ 2n-4+8/n & for\ r \equiv 2(mod\ 4), \\ n+2r+1 & for\ r \equiv 3(mod\ 4), \\ 4r & for\ r \equiv 0(mod\ 4). \end{cases}$$

(ii) When *p* is odd,

$$D(n,r) = \begin{cases} 2r-8r/n+n-16/n+9 & for\ r \equiv 1(mod\ 4), \\ 4r-8r/n-8/n+8 & for\ r \equiv 2(mod\ 4), \\ 2r+n+8/n-3 & for\ r \equiv 3(mod\ 4), \\ 2n-4+x/n & for\ r \equiv 0(mod\ 4). \end{cases}$$

and $x = 32\ if\ \left\{\frac{m-1-2i}{4} + \left[\frac{m+(1+2i)(n-1)}{4(n-1)}\right]\right\} \equiv (1-i)$ (*mod* 2) *for* $i = 0$ *or* 1 ; *else* $x = 0$.

When the number of runs n is odd, Nguyen and Cheng (2008) proposed a lower bound to $E(s^2)$. Bulutoglu and Ryan (2008) and Suen and Das (2010) sharpened the lower bound to $E(s^2)$ for odd n. We have used the lower bound given by Suen and Das (2010) to calculate the lower bounds to $E(s^2)$ for nearly balanced SSDs and is given below.

Theorem 2.2 (Suen and Das; 2010). For an odd integer n and an integer $m \geq n$, let q be the integer such that $m + q \equiv 2 \ (mod\ 4)$ and $-2n \leq qn - m \leq 2n$, and let

$g(q) = n\,(m + q)^2 - 2mq - (m + q)^2\, n^2$. Also let p^*, d and d^* be defined as

$$p^* = \left[\left\{ n - \sqrt{(|qn - m| - n)(n-1) + n} \right\} / 2 \right],$$

$d = 4p^*(n - p^*) - (2n - |qn - m|)(n-1)$ and

$d^* = 4(n + 1 - 2p^*)$. Then,

$$LB_{SD} = \begin{cases} \dfrac{2(n-1)^2 + g(q)}{m(m-1)} & if\ \ |qn - m| \leq n-1, \\ \dfrac{4(n-1)(|qn-m|-n) + 8p^*(n-p^*) + g(q)}{m(m-1)} & \\ if\ |qn - m| > n-1,\ d \leq d^*/2, & \\ \dfrac{4n(n-1) - 8(p^*-1)(n-p^*+1) + g(q)}{m(m-1)} & \\ if\ |qn-m| > n-1,\ d > d^*/2. & \end{cases} \quad (2,4)$$

Gupta *et al*. (2010) proposed the following theorem which provides the maximum number of distinct columns for balanced and nearly balanced SSDs. The columns are distinct in the sense that no two columns are same and no columns can be obtained from any other columns by interchanging the symbols.

Theorem 2.3(Gupta et al. 2010). For a two-level SSD with n runs and m factors, the upper bound on the number of factors such that no two factors are identical or are linear function of any other column(s) is given by

(i) When n is even, i.e., n = 2t, t being a positive integer

$$m \leq \frac{1}{2}\left(\frac{n!}{t!t!}\right) \quad (2.5)$$

(i) When n is odd, *i.e.*, $n = 2t + 1$, t being a positive integer

$$m \leq \frac{n!}{t!(t+1)!} \quad (2.6)$$

To construct efficient SSDs the lower bound to $E(s^2)$ given by Das et al. (2008) for balanced SSDs and the lower bound to $E(s^2)$ given by Suen and Das (2010) for nearly balanced SSDs have been used where the efficiency of the constructed designs has been computed by using the efficiency defined as

$$Efficiency = \frac{LB}{E(s^2)} \quad (2.7)$$

To compute the degree of association between the columns of an SSD r_{max} and f_{max} are defined as $r_{\max} = \max(|r_{ij}|)$ and $f_{\max}$ = (frequency of occurence of r_{max} in the upper diagonal of R) $= \sum_{j=1}^{m} \sum_{i(<j)=1}^{m} I_{r_{\max}}(|r_{ij}|)$

where correlation matrix $R = (r_{ij}), i \neq j = 1, 2, ..., m$, and the indicator function $I_a(b) = 1$ if $a = b$ and is 0 otherwise.

A design with efficiency 1 is an optimal design. And an optimal design having less values of r_{max} and f_{max} is desired. We now describe a procedure for generation of optimal/efficient two level SSDs.

3. PROPOSED ALGORITHM FOR CONSTRUCTING TWO-LEVEL SSDS

The algorithms proposed by Nguyen (1996), Lejeune (2003), Ryan and Bulutoglu (2007) and Gupta *et al*. (2008, 2010) have been modified in the proposed algorithm. The algorithm generates balanced SSDs for even n and nearly balanced SSDs for odd n lower bound to $E(s^2)$, given by Das et al. (2008) and Suen and Das (2010) have been used to check the efficiency of the constructed design for balanced and nearly balanced SSDs respectively. In case of balanced SSDs the algorithm generates design with less number of orthogonal pair of columns. And in case of nearly balanced design the algorithm generates design which is efficient for all $E(s^2)$, r_{max} and f_{max} criterion.

Step1: initialize the generation procedure

(i) Input the parameter viz. n (the number of runs) and m (the number of factors).

(ii) Check the parameter and decide the layout of the design to be generated. The criterion of $n - 1 < m$ and the values of n and m have been checked by Theorem 2.3. If the generation is possible the algorithm goes to next step otherwise ask for new parameter.

(iii) Based on the values of n and m the algorithm calculates the lower bound to $E(s^2)$. The bound is calculated using Theorem 2.1 or Theorem 2.2 for the case of balanced or nearly balanced SSDs, respectively.

Step 2: generation of random design

(i) Based on the input parameter the algorithm generates a random matrix **X** of order $n \times m$ with entries as –1, +1 in each column.

(ii) If n is even the random matrix **X** is generated in such a way that the frequencies of the occurrence of levels + l and –1 in each column is $\left[\frac{n}{2}\right]$.

(iii) If n is odd the random matrix **X** is generated in such a way that the in such a way that in each of the first $\left[\frac{m}{2}\right]$ columns of **X**, the frequencies of the occurrence of levels + l and –1 is $\left[\frac{n}{2}\right]$ and $n - \left[\frac{n}{2}\right]$, respectively and in each of the remaining $m - \left[\frac{m}{2}\right]$ columns of **X**, the frequency of occurrence of levels + l and –1 is $n - \left[\frac{n}{2}\right]$ and $\left[\frac{n}{2}\right]$, respectively.

(iv) After generating initial design Algorithm compute, $E\left(s^2\right) = \sum_{1 \le i < j \le m} s_{ij}^2 \Big/ \binom{m}{2}$ where s_{ij} is the $(i, j)^{th}$ entry of $X'X$.

(v) Check the efficiency of the design and decide whether improvement is necessary or not. For improvement of the initial design algorithm move to step3.

Step 3: Improvement of design

(i) Compute $S_j^2 = \sum_{\substack{i=1 \\ i \ne j}}^{n} s_{ij}^2$ for j = 1,2...., m, where s_{ij} is the $(i, j)^{th}$ entry of X'X and Select the k^{th} column for modification if $S_k^2 = \max\left(S_j^2\right)$, j = 1, 2,...,m.

(ii) For the selected k^{th} column, swap the first entry of this column with all the other entries in that column that have opposite signs and calculate the value of $E(s^2)$ at each swapping.

(iii) The swapping of entries $[\pm 1 \to \mp 1]$ continues for remaining entries of the k^{th} column and keep calculating the value of $E(s^2)$ at each swapping.

(iv) Accept the swapping of entries in the column which leads to maximum reduction in the value of $E(s^2)$.

(v) Repeat (iv) and (v) of Step 2 and decide for next step.

Step 4: Termination of swapping in step 3.

This improvement of design stops when either of the following two conditions arrived:

(i) $E(s^2)$ attains the lower bound for the given parameter.

(ii) No further reduction in the value of $E(s^2)$ is possible through swapping.

Step 5: Finalize design

If (i) of Step 4 achieved, algorithm checks the following property of the design (say **D**) otherwise algorithm goes to (v) of this step.

(i) If any pair of columns of D are fully aliased algorithm randomly select any one of the column and replace with a freshly generated column having same number of +1 and –1. While replacing with new column it also ensures that the new column is not aliased with any other column of **D.**

(ii) After replacing all aliased columns algorithm take the new design as initial design and goes to (iv) of Step 2.

(iii) If no two columns are aliased in **D** the algorithm computes the correlation matrix $R = \left(r_{ij}\right), i \ne j = 1, 2, ..., m$ and then the values of r_{max} and f_{max} for the design **D**,

(iv) If no two columns are aliased in **D** algorithm identifies the number of pair wise orthogonal columns in **D** in the case of $n \equiv 0 \pmod 4$.

(v) Select the k^{th} column by using (i) of Step 3 and generate a new design $\mathbf{D_1}$ by deleting the selected column. Taking $\mathbf{D_1}$ as initial design algorithm repeat the same process to construct an optimal SSD with parameter SSD (n, m-1) satisfying the desired property.

(vi) If algorithm becomes successful to get $\mathbf{D_1}$ algorithm generate a fresh column having same number of +1 and –1 as of the deleted column and add the new column to $\mathbf{D_1}$. While adding new column it also ensures that the new column is not aliased with any other column of $\mathbf{D_1}$. In this way the algorithm again get a new initial design **D** and goes to (iv) of Step 2.

(vii) Algorithm repeat the whole procedure to generate 5 design with same efficiency named $\mathbf{ID_1}$, $\mathbf{ID_2}$, $\mathbf{ID_3}$, $\mathbf{ID_4}$ and $\mathbf{ID_5}$ (say). Out of these five designs the algorithm finalize and report the design having less values of r_{max} and f_{max} .

Using the algorithm, we generate a large number of optimal designs having less values of r_{max} and f_{max}. A small catalogue of 20 optimal designs for number of runs is given in section 5. The layout of these designs is available with the author.

4. IMPLEMENTATION OF THE PROPOSED ALGORITHM

We now describe the step by step implementation of the proposed algorithm through an example of nearly balanced SSDs.

Example4.1 Two level Nearly Balanced SSD (7, 16)

Suppose we want to construct a nearly balanced two-level SSD for 7 runs having 16 factors. As the parameter satisfied the necessary condition of the parameter of SSDs Step 1 of the proposed algorithm calculates Lower Bound to $E(s^2)$ as 5.400 and Step 2 of the algorithm generate the following random two level balanced SSD (7, 16):

-1	-1	-1	1	-1	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	1	-1	-1	1	1	-1	1	1	1	1	-1
1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	1	-1	1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	-1	1	-1
1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	1	1	-1	1	-1	1	1	1	1	1	1	1
-1	1	-1	1	1	-1	1	-1	1	-1	1	1	-1	-1	1	1

X^TX **Calculation of $E(s^2)$**

7	-1	3	-1	-1	-1	3	-1	1	-3	-3	-3	1	1	-3	1
-1	7	3	-1	-1	-1	-1	-5	5	1	-3	1	-3	1	1	1
3	3	7	-5	-1	-1	-1	-1	1	1	-3	1	1	5	-3	1
-1	-1	-5	7	-1	-1	3	-1	1	-3	1	-3	-3	-7	1	-3
-1	-1	-1	-1	7	-1	3	3	-3	-3	5	5	1	1	1	5
-1	-1	-1	-1	-1	7	-1	-1	1	1	-3	1	5	1	5	-3
3	-1	-1	3	3	-1	7	-1	1	-7	1	1	1	-3	1	1
-1	-5	-1	-1	3	-1	-1	7	-7	1	5	1	1	1	-3	1
1	5	1	1	-3	1	1	-7	7	-1	-5	-1	-1	-1	3	-1
-3	1	1	-3	-3	1	-7	1	-1	7	-1	-1	-1	3	-1	-1
-3	-3	-3	1	5	-3	1	5	-5	-1	7	3	-1	-1	-1	3
-3	1	1	-3	5	1	1	1	-1	-1	3	7	3	3	3	3
1	-3	1	-3	1	5	1	1	-1	-1	-1	3	7	3	3	-1
1	1	5	-7	1	1	-3	1	-1	3	-1	3	3	7	-1	3
-3	1	-3	1	1	5	1	-3	3	-1	-1	3	3	-1	7	-1
1	1	1	-3	5	-3	1	1	-1	-1	3	3	-1	3	-1	7
112	**136**	**144**	**176**	**168**	**128**	**144**	**176**	**176**	**144**	**184**	**144**	**128**	**176**	**136**	**128**

$$E\left(s^2\right) = \sum_{1 \le i < j \le m} s_{ij}^2 \bigg/ \binom{m}{2} = 808/120 = 6.733$$

Using (v) of Step 2 we get the efficiency of the design is 0.8012 therefore the algorithm goes to Step 3 and compute S_j^2 for each column by using (i) of Step 3 and we select column number 11 for modification. Using (ii), (ii) and (iii) of Step 3 on the design generated, we get the following Table:

Table 4.1: Exchange of coordinate *i* with coordinate *j* (*j* > *i*) for column 11

Sl. No.	**(*i*, *j*)**	**$E(s^2)$**		**Sl. No.**	**(*i*, *j*)**	**$E(s^2)$**
1	(1, 2)	6.733		7	(3, 4)	6.733
2	(1, 4)	6.733		8	(3, 5)	6.733
3	(1, 5)	6.600		9	(4, 6)	6.733
4	(2, 3)	6.467		10	(4, 7)	6.667
5	(2, 6)	5.933		11	(5, 6)	6.600
6	(2, 7)	6.667		12	(5, 7)	6.800

Table 4.1 shows that 7th exchange *i.e.* the exchange of pair of coordinates 2 and 6 in column 11 leads to maximum reduction in $E(s^2)$. Retaining this best exchange the design obtained is the following with $E(s^2) = 5.933$:

-1	-1	-1	1	-1	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	1	-1	-1	1	1	1	1	1	1	1	-1
1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	1	-1	1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	-1	1	-1
1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	1	1	-1	1	-1	1	-1	1	1	1	1	1
-1	1	-1	1	1	-1	1	-1	1	-1	1	1	-1	-1	1	1

Using (iv) and (v) of Step 2 we get that further modification is necessary and Step 3 select column 14 for modification. Using (iii) of Step 3 on the design generated, we get the following Table:

Table 4.2: Exchange of coordinate *i* with coordinate *j* (*j*>*i*) for column 14

Sl. No.	**(*i*, *j*)**	**$E(s^2)$**		**Sl. No.**	**(*i*, *j*)**	**$E(s^2)$**
1	(1, 2)	6.000		7	(4, 5)	5.867
2	(1, 3)	6.000		8	(4, 6)	5.733
3	(1, 5)	5.933		9	(6, 7)	5.867
4	(1, 6)	5.800		10	(5, 7)	6.000
5	(2, 4)	5.667		11	(2, 3)	5.800
6	(3, 4)	5.933		12	(2, 7)	5.800

From Table 4.2 it is clear that 5th exchange *i.e.* the exchange of pair of coordinates 2 and 4 in column 14 leads to maximum reduction in $E(s^2)$. After exchange the design obtained is the following with $E(s^2) = 5.667$.

-1	-1	-1	1	-1	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	1	-1	-1	1	1	1	1	1	-1	1	-1
1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	1	-1	1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	1	1	-1
1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	1	1	-1	1	-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 exchange of coordinates continues again after selecting column number 8. And the values of $E(s^2)$ after each exchange is given below:

Table 4.3: Exchange of coordinate *i* with coordinate *j* (*j*>*i*) for column 8

Sl. No.	(*i*, *j*)	$E(s^2)$		Sl. No.	(*i*, *j*)	$E(s^2)$
1	(1, 2)	5.800		7	(3, 4)	5.733
2	(1, 4)	6.000		8	(3, 5)	5.667
3	(1, 5)	5.933		9	(3, 7)	5.667
4	(1, 7)	5.933		10	(4, 6)	5.667
5	(2, 3)	5.800		11	(5, 6)	5.867
6	(2, 6)	5.467		12	(6, 7)	5.867

Table 4.3 shows that the maximum reduction in $E(s^2)$ is with the exchange of pair of coordinates 2 and 6 in column 8. After exchange the design obtained is the following with $E(s^2) = 5.467$.

-1	-1	-1	1	-1	-1	-1	1	-1	1	1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	1	-1	1	1	1	1	1	1	-1	1	-1
1	-1	1	-1	1	-1	1	1	-1	-1	1	1	1	1	-1	1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	-1	1	1	1	-1
1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	1	1	-1	-1	-1	1	-1	1	1	1	1	1
-1	1	-1	1	1	-1	1	-1	1	-1	1	1	-1	-1	1	1

This coordinate exchange of various columns selected in (i) of Step 3 terminates with the following design with $E(s^2) = 5.6$, since there is no further reduction in the values of $E(s^2)$. The efficiency of this design D is 0.964.

-1	1	1	-1	-1	1	1	1	1	1	-1	1	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	1	1
1	-1	1	-1	1	1	1	-1	-1	1	1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	-1	1	1	1	1	1	-1
-1	-1	-1	1	-1	1	-1	-1	-1	1	1	1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	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 efficiency of this design is 0.964. Therefore, the algorithm goes to (v) of Step 5.column 8 is selected for deletion . Deleting column 8 we get the new initial design D_1 as follows.

-1	1	1	-1	-1	1	1	1	1	-1	1	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
1	-1	1	-1	1	1	1	-1	1	1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1	1	-1	1	1	1	1	1	-1
-1	-1	-1	1	-1	1	-1	-1	1	1	1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	1	1	-1	1	-1	-1	-1
-1	-1	-1	1	1	-1	1	1	-1	-1	1	1	1	1	1

Using (v) and (vi) of step 5 gives the following optimal initial design $\mathbf{ID_1}$ having r_{max}=0.75 and f_{max}= 6. The design is given below.

-1	1	1	-1	-1	1	1	-1	1	1	-1	1	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
1	-1	1	-1	1	1	1	-1	-1	1	1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	-1	1	1	1	1	1	-1
-1	-1	-1	1	-1	1	-1	1	-1	1	1	1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	1	1	1	-1	1	-1	-1	-1

Using (vii) of Step 5 the algorithm generates 5 initial design names $\mathbf{ID_1}$, $\mathbf{ID_2}$, $\mathbf{ID_3}$, $\mathbf{ID_4}$ and $\mathbf{ID_5}$ which is $E(s^2)$ optimal. For each of these five design algorithm calculate r_{max} and f_{max} and report the design having less values of r_{max} and f_{max}. Step 5 report the final design as given bellow which is $E(s^2)$ optimal having r_{max}=0.75 and f_{max}= 6.

-1	1	1	-1	-1	1	1	-1	1	1	-1	1	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
1	-1	1	-1	1	1	1	-1	-1	1	1	-1	1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	-1	1	1	1	1	1	-1
-1	-1	-1	1	-1	1	-1	1	-1	1	1	1	-1	1	-1	-1
-1	1	-1	-1	1	-1	-1	1	1	1	1	-1	1	-1	-1	-1
-1	-1	-1	1	1	-1	1	1	1	-1	-1	1	1	1	1	1

5. CONCLUSION

In this article we propose a computer algorithm for generating two-level balanced as well as nearly balanced SSDs. An important feature of this algorithm is that it generates an SSD where all the columns are distinct and no column can be generated from any other column. The main emphasis of this work has been given to generate the best design based on r_{max} and f_{max} criterion among the class of $E(s^2)$ optimal SSDs.

Using the algorithm, we can generate a number of optimal designs. We have given a catalogue of 30 optimal designs for number of runs $5 \leq n \leq 12$ in the catalogue. The values of $E(s^2)$ and their lower bounds and the values of r_{max} and f_{max} of the designs generated have also been given in the catalogue.

The algorithm is also efficient in consuming CPU time. Any designs given in the catalogue take less than 20 CPU seconds using Intel® Core i5 @ 2.30 GHz CPU with 4 GB memory. A reason for taking less time is that the algorithm works only one column at a time which contributes maximum to the value of $E(s^2)$ instead of working with all the columns.

Table 5.1: Catalogue of optimal SSDs

Sl. No.	*n* (No. of runs)	*m* (No. of factors)	$E(s^2)$	Lower Bounds		$r_{\max}$	f_{max}
				Suen and Das (2010)	Das et al. (2008)		
1	5	8	3.571	3.571	-	0.667	9
2	5	9	3.667	3.667	-	0.667	12
3	5	10	3.667	3.667	-	0.667	15
4	6	9	4.000	-	4.000	0.333	36
5	6	10	4.000	-	4.000	0.333	45
6	7	12	4.636	4.636	-	0.750	4
7	7	13	4.692	4.692	-	0.750	6
8	7	14	4.956	4.956		0.750	4
9	7	15	5.114	5.114	-	0.750	3
10	8	14	4.923	-	4.923	0.500	17
11	8	15	5.486	-	5.486	0.500	36
12	8	16	5.867	-	5.867	0.500	44
13	8	17	6.118	-	6.118	0.500	52
14	8	18	6.275	-	6.275	0.500	60
15	8	20	6.400	-	6.400	0.500	76
16	9	16	5.667	5.667	-	0.800	2
17	9	17	5.705	5.705	-	0.550	1
18	9	18	5.706	5.706	-	0.800	3
19	10	16	5.867	-	5.867	0.200	64
20	10	17	5.882	-	5.882	0.600	8
21	10	18	5.882	-	5.882	0.600	9
22	10	19	6.433	-	6.433	0.600	13
23	10	20	6.863	-	6.863	0.600	17
24	11	16	5.333	5.333	-	0.633	1
25	11	17	5.706	5.706	-	0.633	1
26	11	20	6.684	6.684	-	0.633	2
27	12	18	5.961	-	5.961	0.666	1
28	12	19	6.456	-	6.456	0.666	1
29	12	20	6.821	-	6.821	0.666	2
30	12	24	7.826	-	7.826	0.666	5

REFERENCES

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

Bulutoglu, D.A. (2007): Cyclically constructed $E(s^2)$-optimal supersaturated designs. *Journal of Statistical Planning and Inference*, **137**, 2413–2428.

Bulutoglu, D.A. and Cheng, C.S. (2004): Construction of $E(s^2)$-optimal supersaturated designs. *Annals of Statistics*, **32**, 1662–1678.

Bulutoglu, D.A. and Ryan, K.J. (2008): $E(s^2)$-Optimal supersaturated designs with good minimax properties when N is odd. *Journal of Statistical Planning and Inference*, **138**, 1754–1762.

Gupta, V.K., Parsad, R., Bhar, L.M. and Kole, B. (2008): Supersaturated designs for asymmetrical factorial experiments. *Journal of Statistical Theory and Practice*, **2**, 95–108.

Gupta, V.K., Singh P., Kole, B. and Parsad, R. (2010): Construction of Efficient Balanced and Nearly Balanced Two-level Supersaturated Designs. *Journal of Statistics and Applications*, **5**, 179–194.

Lejeune, M.A. (2003): A coordinate-columnwise exchange algorithm for construction of supersaturated, saturated and non-saturated experimental designs. *American Journal of Mathematical and Management Sciences*, **23**, 109–142.

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.

Ryan, K.J. and Bulutoglu, D.A. (2007): $E(s^2)$-Optimal supersaturated designs with good minimax properties. *Journal of Statistical Planning and Inference*, **137**, 2250–2262.

Satterthwaite, F. (1959): Random balance experimentation. *Technometrics*, **1**, 111–137.

Suen, C.S. and Das, Ashish (2010): $E(s^2)$-optimal supersaturated designs with odd number of runs. *Journal of Statistical Planning and Inference*, **140**,1398–1409.