An algorithm for constructing mixed-level $k$-circulant supersaturated designs

Min-Qian Liu, Li Zhang

Department of Statistics, School of Mathematical Sciences and LPMC, Nankai University, Tianjin 300071, China

**ABSTRACT**

Supersaturated design (SSD) has received much interest because of its potential in factor screening experiments. Recently, Chen and Liu [Chen, J., Liu, M.Q., 2008a. Optimal mixed-level $k$-circulant supersaturated designs. J. Statist. Plann. Inference 138, 4151–4157] explored the construction of $E(f_{\text{NOD}})$-optimal mixed-level SSDs using $k$-cyclic generators. As an extension, this paper further studies how to construct optimal mixed-level $k$-circulant SSDs under the $\chi^2(D)$ criterion. A sufficient condition under which the generator vector produces a $\chi^2(D)$-optimal SSD is obtained, and some other properties of the resulting SSDs are investigated. Note that, the construction of optimal mixed-level $k$-circulant SSDs under the $\chi^2(D)$ criterion is much more complicated than that under $E(f_{\text{NOD}})$. Hence, an algorithm for finding generators of (nearly) $\chi^2(D)$-optimal designs is provided, and many (nearly) $\chi^2(D)$-optimal mixed-level SSDs are then constructed.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

A supersaturated design (SSD) is a fractional factorial design whose run size is not enough for estimating all the main effects. It is commonly used in the initial stage of an industrial or scientific experiment, where the goal is to identify sparse and dominant active effects with low cost. The data collected by SSDs are typically analyzed under the assumption of effect sparsity, which says that only a few dominant factors actually affect the response. Satterthwaite (1959) stands as a pioneer and proposed the idea of SSD in random balance designs. Booth and Cox (1962) first examined SSDs systematically. After that, such designs were not studied further until the appearance of the work by Lin (1993) and Wu (1993). SSDs have become increasingly popular in recent years because of their potential in factor screening experiments. Most studies have focused on the construction and analysis of symmetrical SSDs, see Liu et al. (2007), Nguyen and Cheng (2008), and the references therein. However, much practical experience indicates that mixed-level SSDs also have wide uses.

Researches on mixed-level SSDs include the early work by Fang et al. (2000, 2003) who proposed the $E(f_{\text{NOD}})$ criterion and the FSOA method for constructing mixed-level SSDs, and by Yamada and Matsui (2002) and Yamada and Lin (2002) who used $\chi^2(D)$ to evaluate mixed-level SSDs. And then Fang et al. (2004) and Koukouvinos and Mantas (2005) constructed many $E(f_{\text{NOD}})$-optimal mixed-level SSDs. Li et al. (2004) derived a lower bound of $\chi^2(D)$ along with the sufficient and necessary condition for achieving it. Recent work on mixed-level SSDs includes Xu (2003), Xu and Wu (2005), Liu et al. (2006), Yamada et al. (2006), Ai et al. (2007), Zhang et al. (2007), Tang et al. (2007), Nguyen and Liu (2008), Chen and Liu (2008a,b), Liu and Lin (2009), and Liu and Cai (2009). In particular, Chen and Liu (2008a) explored the construction of $E(f_{\text{NOD}})$-optimal mixed-level SSDs using $k$-cyclic generators. Their method generalized the cyclic method proposed by Nguyen (1996), Liu and Zhang (2000), and Liu and Dean (2004) for two-level SSDs and the one due to Georgiou and Koukouvinos (2006) for the multi-level case.

* Corresponding author.
E-mail address: mqliu@nankai.edu.cn (M.-Q. Liu).

0167-9473/ - see front matter © 2008 Elsevier B.V. All rights reserved.
Note that, the $E(f_{\text{nod}})$ criterion assigns the same weight to all the factors, while the $\chi^2(D)$ considers different weights for factors with different levels. As remarked in Chen and Liu (2008a), a sufficient condition for the $\chi^2(D)$-optimality can be derived. However, the construction of optimal mixed-level $k$-circulant SSDs under the $\chi^2(D)$ criterion is much more complicated. In this paper, we further investigate the construction of $\chi^2(D)$-optimal mixed-level SSDs using $k$-cyclic generators. Section 2 introduces the mixed-level $k$-circulant method and the necessary and sufficient conditions due to Chen and Liu (2008a) for the resulting designs to be balanced. In Section 3, the sufficient condition under which the generator vector produces a $\chi^2(D)$-optimal SSD is obtained, a nice property for $\chi^2(d^i, d^j)$ is derived. Section 4 provides an efficient construction algorithm for finding $k$-cyclic generators of (nearly) $\chi^2(D)$-optimal designs, and many (nearly) $\chi^2(D)$-optimal SSDs are tabulated.

The rest of this section is devoted to the $\chi^2(D)$ criterion that will be used. A mixed-level design that has $n$ runs and $m$ factors with $q_1, \ldots, q_m$ levels respectively is denoted by $D(n; q_1, \ldots, q_m)$. A $D(n; q_1, \ldots, q_m)$ can be expressed as an $n \times m$ matrix $D = (d_{ij})$. Let $d_i$ be the $i$th row of $D$ and $d^j$ be the $j$th column which takes values from a set of $q_j$ symbols $\{1, \ldots, q_j\}$. If each column $d^j$ has the equal occurrence property of the $q_j$ symbols, we say that $D$ is a balanced design. In a factorial design, two columns are called fully aliased if one column can be obtained from the other by permuting levels. When $\sum_{j=1}^m (q_j - 1) > n - 1$, the design is called a supersaturated design (SSD). When some $q_i$’s are equal, we use the notation $D(n; q_1^n, \ldots, q_m^n)$. And if all the $q_i$’s are equal, the design is said to be symmetrical.

For a $D(n; q_1, \ldots, q_m)$, the $\chi^2(D)$ criterion defined by Yamada and Matsui (2002) is to minimize

$$
\chi^2(D) = \sum_{1 \leq i < j \leq m} \chi^2(d^i, d^j),
$$

where

$$
\chi^2(d^i, d^j) = \sum_{a=1}^q \sum_{b=1}^q \left( \frac{n_{ab}^{(ij)} - n}{q_i q_j} \right)^2 \sqrt{n/q_i q_j},
$$

and $n_{ab}^{(ij)}$ is the number of $(a, b)$-pairs in $(d^i, d^j)$. Another class of design criteria is the maximum $\chi^2(d^i, d^j)$ values

$$
\chi^2_{max}(q_u, q_v) = \max\{\chi^2(d^i, d^j) | 1 \leq i < j \leq m, \text{ both } d^i \text{ and } d^j \text{ have } q_u \text{ levels}, \}
$$

$$
\chi^2_{max}(q_u, q_v) = \max\{\chi^2(d^i, d^j) | 1 \leq i, j \leq m, d^i \text{ has } q_u \text{ levels, } d^j \text{ has } q_v \text{ levels, } q_u \neq q_v \}. \tag{3}
$$

It is known that when two $q_u$-level columns $d^i$ and $d^j$ are fully aliased, all possible values of $n_{ab}^{(ij)}$ take $n/q_u$ and 0 with numbers $q_u$ and $q_u^2 - q_u$, respectively. So from (1), we can easily obtain that

$$
\chi^2_{max}(q_u, q_v) = \frac{n^2}{n(q_u(n/q_u - n/q_u^2)^2 + (q_u^2 - q_u)(0 - n/q_u^2)^2)} = n(q_u - 1),
$$

which can be regarded as an upper bound of $\chi^2_{max}(q_u, q_v)$.

The $E(f_{\text{nod}})$ criterion proposed by Fang et al. (2000, 2003) for comparing mixed-level SSDs is to minimize

$$
E(f_{\text{nod}}) = \frac{2}{m(m-1)} \sum_{1 \leq i < j \leq m} \chi^2(d^i, d^j)n/q_i q_j.
$$

Note that the $\chi^2(D)$ considers different weights for factors with different levels, while $E(f_{\text{nod}})$ does not.

There are also several other criteria for evaluating mixed-level SSDs. As argued in Liu et al. (2006) and Liu and Lin (2009), in this paper, we will mainly adopt $\chi^2(D)$ as the optimality criterion for assessing mixed-level SSDs.

2. Existence of balanced mixed-level $k$-circulant SSDs

The method for constructing two-level $k$-circulant SSDs was proposed by Liu and Dean (2004), and then it was generalized to the multi-level and mixed-level cases by Georgiou and Koukouvinos (2006) and Chen and Liu (2008a), respectively. A $k$-circulant SSD can be produced by using a generator vector and its $k$-cyclic permutations as rows. As an illustration, suppose we have a generator vector

$$
G = (2, 5, 1, 3, 2, 5, 1, 2, 1, 4, 1, 1, 2, 2, 2, 3, 2, 4, 1, 6, 2, 6). \tag{5}
$$

Repeat cycling $k = 2$ elements of $G$ to the right and moving the last two elements to the first two positions for 10 steps, we obtain the first 11 rows of a balanced $2$-circulant SSD $D(12; 2^{11}6^{11})$. A 12th row of 1’s is added to give the design as is shown in Table 1.

The next lemma obtained by Chen and Liu (2008a) shows us the necessary and sufficient conditions for the existence of balanced mixed-level $k$-circulant SSDs.

**Lemma 1.** Suppose design $D(n; q_1, \ldots, q_m)$ is obtained from the generator vector $(g_1, \ldots, g_m)$ by cycling $k$ elements to the right at each step and adding a row of 1’s, where $m = sk$ for some positive integer $s$, $g_{i+k} \ldots, g_{i+(s-1)k}$ take values from a set of $q_i$ symbols $\{1, \ldots, q_i\}$ for $i = 1, \ldots, k$. Then the necessary and sufficient conditions for $D(n; q_1, \ldots, q_m)$ to be balanced are

(i) $s = n - 1, n = q_it$, for some positive integers $t_i$, $i = 1, \ldots, k$;
Let $D$ be a balanced 2-circulant SSD. For any balanced design $D$ (Liu 2004), for a balanced design $D$ (ii), each of the levels \( l_i = 1, \ldots, q_i \),
\[
\sum_{u=0}^{n-2} \delta_{u,l_i+uk} = t_i - \delta_{l_i,1},
\]
where $\delta_{i,j} = 1$ if $i = j$ and 0 otherwise.

It can be checked that for the design generated by (5), the conditions in Lemma 1 are satisfied, thus it is a balanced SSD.

3. \( \chi^2(D) \)-optimality of \( k \)-circulant SSDs

This section mainly investigates how to ensure the \( \chi^2(D) \)-optimality of the resulting \( k \)-circulant SSDs. For given \( m, q_1, \ldots, q_m \), let
\[
\Delta = \left\{ \sum_{u=1}^{m} q_u \delta_u : \delta_u = 0, 1, \text{ for } u = 1, \ldots, m \right\}.
\]
In $\Delta$, let $\delta_l$ and $\delta_U$ be the two nearest values to
\[
\delta = \sum_{u=1}^{m} (n - q_u)/(n - 1),
\]
satisfying $\delta_l \leq \delta < \delta_U$. For any two rows $d_i$ and $d_j$, let
\[
\rho_{ij} = \sum_{u=1}^{m} q_u \delta_{d_i,\,d_j},
\]
i.e., $\rho_{ij}$ is the natural weighted coincidence number between the two rows. According to Theorem 1 of Li et al. (2004) and Corollary 2 of Liu et al. (2006), the lower bound of \( \chi^2(D) \) is provided below along with the necessary and sufficient condition to achieve it.

Lemma 2. For any balanced design $D(n; q_1, \ldots, q_m)$,
\[
\chi^2(D) \geq \frac{\sum_{1 \leq i < j \leq n} \rho_{ij}^2}{2n} + C(D),
\]
and the lower bound of \( \chi^2(D) \) on the right-hand side of (9) can be achieved, if and only if, for any row $d_i$, among the $n - 1$ values of $\rho_{ij}$ ($j \neq i$), there are \( (n - 1)[(\delta_U - \delta)/(\delta_U - \delta_l)] \) with the value $\delta_l$ and \( (n - 1)((\delta - \delta_l)/(\delta_U - \delta_l)) \) with the value $\delta_U$, where $C(D) = \left[ s(s-1)(n-1) - s(n-1)(m-s)\right]/2$, $s = \sum_{j=1}^{m} (q_j - 1)/(n - 1)$.

Based on Lemma 2, we can obtain the following theorem which provides us a sufficient condition under which the generator vector produces a \( \chi^2(D) \)-optimal mixed-level \( k \)-circulant SSD.

Theorem 1. Let $D(n; q_1, \ldots, q_m)$ be a balanced $k$-circulant SSD obtained from the generator vector $(g_1, \ldots, g_m)$. The sufficient condition for this design to be \( \chi^2(D) \)-optimal is that the values of $c_v$ for $v = 1, \ldots, [(n + 1)/2]$ take either $\delta_l$ or $\delta_U$, where
\[
c_v = \sum_{l=1}^{n} q_l \delta_{d_l,\,d_l + v}, \quad v = 1, \ldots, [(n - 1)/2], \quad c_{[(n+1)/2]} = \sum_{l=1}^{n} q_l (t_l - 1).
\]
Proof. For any two rows \( d_i \) and \( d_j \) with \( |i - j| = v \) or \( |i - j| = n - 1 - v \), it can be easily shown from the construction method that

\[
\rho_{ij} = \sum_{u=1}^{m} q_u \delta_{d_iu} \delta_{d_ju} = \sum_{i=1}^{m} q_i \delta_{c_i, d_i + uk} = c_v
\]

with any \( l + vk > m \) replaced by \( l + vk - m \), where \( 1 \leq i, j \leq n - 1, v = 1, \ldots, [(n - 1)/2] \). Meanwhile, based on (6), for \( j = 1, \ldots, n - 1 \),

\[
\rho_{ij} = \sum_{i=1}^{k} q_i \sum_{u=0}^{n-2} \delta_{c_i, d_i + uk} = \sum_{i=1}^{k} q_i (t_i - 1) = c_{(n+1)/2}.
\]

Thus from Lemma 2, when the values of \( c_v \) for \( v = 1, \ldots, [(n + 1)/2] \) take either \( \delta_l \) or \( \delta_d \), i.e. the natural weighted coincidence numbers between any two distinct rows take either \( \delta_l \) or \( \delta_d \), the design is \( \chi^2(D) \)-optimal. This completes the proof. \( \square \)

Remark 1. From Theorem 1, we can judge whether a mixed-level \( k \)-circulant SSD achieves the lower bound of \( \chi^2(D) \) in (9). Note that, this theorem is parallel to Theorem 2 in Chen and Liu (2008a) for the \( E(f_{opt}) \)-optimality. However, it is much more complicated to judge whether a mixed-level \( k \)-circulant SSD satisfies the sufficient condition given here. To do this, we first need to find out the two nearest values \( \delta_l \) and \( \delta_d \) in \( \Delta \) for given \( m \) and \( q_1, \ldots, q_m \) by enumeration, then calculate \( c_1, \ldots, c_{[(n+1)/2]} \) and compare them with \( \delta_l \) and \( \delta_d \).

For the balanced mixed-level 2-circulant SSD \( D(12; 2^{11}6^{11}) \) produced from the generated vector in (5), it can be checked that \( \delta_l = 16, \delta_d = 18 \), and \( c_1 = \ldots = c_6 = 16 \), thus the design is \( \chi^2(D) \)-optimal. Further, the following property of a \( k \)-circulant SSD can be easily obtained

**Theorem 2.** For a \( k \)-circulant SSD \( D(n; q_1, \ldots, q_m) \) obtained from the generator vector \( (g_1, \ldots, g_m) \),

\[
\chi^2(d^i, d^j) = \chi^2(d^{i+vk}, d^{j+vk}),
\]

where \( i, j = 1, \ldots, m, v = 1, \ldots, n - 2, i + vk \) and \( j + vk \) are replaced by \( i + vk - m \) and \( j + vk - m \) when \( i + vk > m \) and \( j + vk > m \), respectively.

Similar to the \( E(f_{opt}) \) criterion, the \( \chi^2(D) \) criterion is also not enough to prevent the existence of fully aliased columns in designs. To evaluate our designs, we also use the values of \( \chi^2_{max}(q_u, q_v) \) and \( \chi^2_{max}(q_u, q_v) \) defined in (2) and (3), respectively, not only to avoid fully aliased columns, but also to keep small dependency between their columns.

Sometimes, it is very difficult to find a mixed-level \( k \)-circulant SSD that attains the lower bound of \( \chi^2(D) \) in (9), especially when \( n, k \) and \( q_1, \ldots, q_m \) become larger. Let \( LB(\chi^2) \) denote this lower bound, we define the \( \chi^2 \)-efficiency as

\[
\chi^2 \text{-efficiency} = \frac{LB(\chi^2)}{\chi^2(D)}.
\]

A design is \( \chi^2(D) \)-optimal when its \( \chi^2 \)-efficiency equals one. Mixed-level \( k \)-circulant SSDs with high \( \chi^2 \)-efficiencies can be found in the subsequent section.

4. Construction algorithm

Now we present an algorithm for constructing \( \chi^2(D) \)-optimal mixed-level \( k \)-circulant SSDs.

Step 1. Input \( n, m, s, k \) and \( q_1, \ldots, q_k \), where \( m = sk \). Set \( N = 0 \).

Step 2. Search a (new) generator satisfying the conditions in Lemma 1 to ensure the balanced property of the resulting design. If such a generator cannot be found, then go to Step 5 when \( N \geq 1 \), or stop the algorithm without obtaining any balanced design when \( N = 0 \); otherwise go to Step 3.

Step 3. Calculate \( \delta_l, \delta_d, \), and \( c_1, \ldots, c_{[(n+1)/2]} \) for the generator found in Step 2. If each \( c_v \) satisfies the condition in Theorem 1, set \( N = N + 1 \) and goto Step 4, otherwise goto Step 2.

Step 4. Calculate \( \chi^2_{max}(q_u, q_v) \)'s and \( \chi^2_{max}(q_u, q_v) \)'s for the generator that passed Step 3. Then goto Step 2.

Step 5. From the \( N \) generators we obtained, find out the one (or ones) that has the smallest \( \chi^2_{max}(q_u, q_v) \)'s and (or) \( \chi^2_{max}(q_u, q_v) \)'s.

Based on the SSD generated from the vector in (5), we now give some explanations for the algorithm. In Step 2, the odd positions of the generator must have five 1's and six 2's, and the even ones must have one 1's and two 2's, 3's, 4's, 5's and 6's, respectively. The generator \( G \) in (5) meets this condition.

In Step 2, it can be calculated from (7) and (8) and Lemma 2 that \( \delta_l = \delta = 16, \delta_d = 18 \), and the necessary and sufficient condition to achieve the lower bound is that for any row \( d_i \), among the 11 values of \( \rho_{ij} \) \( (j \neq i) \), there are 11 with the value \( \delta_l = 16 \) and 0 with the value \( \delta_d = 18 \), hence the sufficient condition in Theorem 1 is that all the values of \( c_v \) for \( v = 1, \ldots, 6 \) take 16. If this condition is satisfied, the algorithm goes to Step 4. The \( G \) in (5) satisfies this condition.

In Step 4, for \( G \), we obtain \( \chi^2_{max}(2, 2) = 0, \chi^2_{max}(2, 6) = 8, \chi^2_{max}(6, 6) = 24 \) and \( \chi^2 \)-efficiency\( = 1 \).
In practice, if we could not find a generator for a generator $G_i$ having one of the smaller numbers of candidates of the generator in the tables. It can be seen that these newly generated designs also have smaller values of maximum $\chi^2$($d',d'$) compared to their upper bounds. Some designs even have zero values of $\chi^2_\text{max}(q_u, q_v)$, which means that in these designs, all the $q_v$-level columns are orthogonal to each other. For any two designs with the same parameters of $n, q_u, r_i$ for $i = 1, 2$ in the tables, one has a smaller $\chi^2_\text{max}(q_u, q_v)$ value (or values) and the other has a smaller $\chi^2_\text{max}(q_u, q_v)$ value (or values).

**Acknowledgements**

The authors thank the Co-Editor, an Associate Editor and two anonymous referees for their valuable comments. This work was supported by the Program for New Century Excellent Talents in University (NCET-07-0454) of China and the NNSF of China grant 10671099.

### Table 2

$\chi^2(D)$-optimal 2-circulant mixed-level SSDs with generator vectors $G$.

<table>
<thead>
<tr>
<th>$D(n; q_1^m q_2^r)$</th>
<th>$\chi^2_\text{max}(q_1, q_1)(\text{UB})$</th>
<th>$\chi^2_\text{max}(q_1, q_1)(\text{UB})$</th>
<th>$\chi^2_\text{max}(q_2, q_2)(\text{UB})$</th>
<th>$\chi^2$-efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D(8; 2^4 7^4)$</td>
<td>0(8)</td>
<td>4(8)</td>
<td>8(24)</td>
<td>1</td>
</tr>
<tr>
<td>$G = (24211312221423)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(10; 2^6 5^9)$</td>
<td>3.6(10)</td>
<td>4(8)</td>
<td>20(40)</td>
<td>0.9901</td>
</tr>
<tr>
<td>$G = (142215211223231524)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(12; 2^3 3^1 5^1)$</td>
<td>0(12)</td>
<td>6(8)</td>
<td>6(24)</td>
<td>1</td>
</tr>
<tr>
<td>$G = (12211112132321321322)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(12; 3^1 4^1 5^1)$</td>
<td>6(24)</td>
<td>10(18)</td>
<td>9.3(36)</td>
<td>0.9986</td>
</tr>
<tr>
<td>$G = (1332111424323213232324)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(12; 2^6 11^3 13^1)$</td>
<td>0(12)</td>
<td>8(12)</td>
<td>24(60)</td>
<td>1</td>
</tr>
<tr>
<td>$G = (2513251214112223241626)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(15; 3^{18} 5^{14})$</td>
<td>15(30)</td>
<td>10(22)</td>
<td>26.7(54)</td>
<td>0.9953</td>
</tr>
<tr>
<td>$G = (22253331153225123332132343414)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(16; 2^{15} 4^{15})$</td>
<td>1(16)</td>
<td>8(16)</td>
<td>10(48)</td>
<td>1</td>
</tr>
<tr>
<td>$G = (12232421211414221232112323)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(16; 2^{15} 4^{15})$</td>
<td>1(16)</td>
<td>6(16)</td>
<td>12(48)</td>
<td>1</td>
</tr>
<tr>
<td>$G = (1422321214222224122132112213)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(18; 3^{17} 6^{17})$</td>
<td>8(36)</td>
<td>18(36)</td>
<td>38(90)</td>
<td>0.9947</td>
</tr>
<tr>
<td>$G = (2335211616243232133633242234152511)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$D(18; 3^{17} 6^{17})$</td>
<td>12(36)</td>
<td>14(36)</td>
<td>38(90)</td>
<td>0.9947</td>
</tr>
<tr>
<td>$G = (23313513261114132636322424251522)$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UB: upper bound.

**Remark 2.** Usually, the number of candidates of the generator in Table 2 is very large for large $n, m, s, k$ and $q_1, \ldots, q_k$. So it is really quite difficult to enumerate all the generators because of the limit of the computer’s computing capacity. Therefore we provide a simplified method to control the searching times and gain the chance to find $\chi^2(D)$-optimal designs. First, we obtain one initial sub-generator $G_i = (g_{i1}, g_{i+k}, \ldots, g_{i+(c-1)k})$ of a $q_i$-level 1-circulant SSD satisfying the conditions in Lemma 1 for $i = 1, \ldots, k$, respectively. Thus we have $k$ sub-generators such that each has $s$ elements. A generator $G = (g_1, g_2, \ldots, g_m)$ can then be created for $D(n; q_1, \ldots, q_m)$ by merging the elements of these $k$ sub-generators. For instance, the generator vector in (5) is formed by merging the following elements of $G_1 = (2, 1, 2, 1, 1, 2, 2, 1, 2)$ and $G_2 = (5, 3, 5, 2, 4, 1, 2, 3, 4, 6, 6)$.

For $i = 1, \ldots, k$, to get new sub-generators, we compare the $j$th element with the $l$th one of $G_i$, where $1 \leq j < l \leq s$, if they are not equal, exchanging these two elements can result in a new sub-generator. Many new sub-generators can be obtained in this way. Furthermore, at Step 2, we can add control sentences in the program to control the repeating times according to different computing capacities.

**Remark 3.** In practice, if we could not find a generator for a $\chi^2(D)$-optimal SSD by the above algorithm, we would broaden the judging condition in Step 3. In this paper, let $\epsilon = \delta_u - \delta_4$, and the judging condition is set to be $\delta - t \epsilon \leq c_i \leq \delta + t \epsilon$ for some positive $t$, e.g., $t = 1, 2$. Although some designs found under this condition may not be $\chi^2(D)$-optimal, they still have high $\chi^2$-efficiencies as will be evidenced in the following tables.
Table 3
\(\chi^2(D)-\)optimal 3-circulant mixed-level SSDs with generator vectors \(G\).

<table>
<thead>
<tr>
<th>(D(n; q_1^1, q_2^2))</th>
<th>(\chi_{\max}^2(q_1, q_1)(UB))</th>
<th>(\chi_{\max}^2(q_1, q_2)(UB))</th>
<th>(\chi_{\max}^2(q_2, q_2)(UB))</th>
<th>(\chi^2)-efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D(10; 2^{10}5^8))</td>
<td>3.6(10)</td>
<td>4(8)</td>
<td>30(40)</td>
<td>0.996</td>
</tr>
<tr>
<td>(G = (11422112221215123223115224))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{22}3^{11}))</td>
<td>1.3(12)</td>
<td>6(8)</td>
<td>6(24)</td>
<td>0.9977</td>
</tr>
<tr>
<td>(G = (22223233232123112321232111211))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{13}3^{23}))</td>
<td>1.3(12)</td>
<td>6(8)</td>
<td>7.5(24)</td>
<td>0.9986</td>
</tr>
<tr>
<td>(G = (22223233232123112321232111211))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{13}3^{23}))</td>
<td>5.3(12)</td>
<td>2(8)</td>
<td>15(24)</td>
<td>0.9972</td>
</tr>
<tr>
<td>(G = (121133233321231123223112222212))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{22}3^{11}))</td>
<td>1.3(12)</td>
<td>6.7(12)</td>
<td>12(36)</td>
<td>1</td>
</tr>
<tr>
<td>(G = (1142241132112242211212122112221))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{13}3^{23}))</td>
<td>1.3(12)</td>
<td>6.7(12)</td>
<td>14.7(36)</td>
<td>1</td>
</tr>
<tr>
<td>(G = (112121412213421122223211222312233))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 3^{22}4^{11}))</td>
<td>10.5(24)</td>
<td>16(18)</td>
<td>12(36)</td>
<td>0.9987</td>
</tr>
<tr>
<td>(G = (3211241321233323431222233133214))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 3^{22}4^{11}))</td>
<td>12(24)</td>
<td>10(18)</td>
<td>12(36)</td>
<td>0.9987</td>
</tr>
<tr>
<td>(G = (23123421321123341121231223224))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 3^{13}4^{22}))</td>
<td>6(24)</td>
<td>12(18)</td>
<td>14.7(36)</td>
<td>0.9995</td>
</tr>
<tr>
<td>(G = (233321242433411222242131334311))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{22}6^{11}))</td>
<td>1.3(12)</td>
<td>8(12)</td>
<td>24(60)</td>
<td>1</td>
</tr>
<tr>
<td>(G = (1122222411511621321212422121222212))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(12; 2^{11}6^{22}))</td>
<td>0(12)</td>
<td>8(12)</td>
<td>30(60)</td>
<td>1</td>
</tr>
<tr>
<td>(G = (16523625315122413311614224265222))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(D(16; 2^{8}3^{45}))</td>
<td>4(16)</td>
<td>8(16)</td>
<td>14(48)</td>
<td>1</td>
</tr>
<tr>
<td>(G = (1132221242211113121242222121121124223123124))</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UB: upper bound.

References


