Biometrika (2017), **104**, 2, *pp.* 455–464
Printed in Great Britain

doi: 10.1093/biomet/asx006
Advance Access publication 28 February 2017

Construction of maximin distance Latin squares and related Latin hypercube designs

By QIAN XIAO and HONGQUAN XU

Department of Statistics, University of California, Box 951554, Los Angeles, California 90095, U.S.A.

xiaoqian1990v@gmail.com hqxu@stat.ucla.edu

Summary

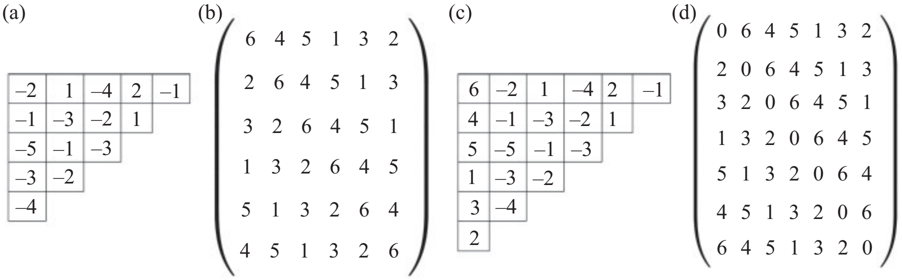
Maximin distance Latin hypercube designs are widely used in computer experiments, yet their construction is challenging. Based on number theory and finite fields, we propose three algebraic methods to construct maximin distance Latin squares as special Latin hypercube designs. We develop lower bounds on their minimum distances. The resulting Latin squares and related Latin hypercube designs have larger minimum distances than existing ones, and are especially appealing for high-dimensional applications.

Some key words: Computer experiment; Costas array; Cyclic design; Maximin distance design; Space-filling design.

1. Introduction

Computer experiments are increasingly being used to investigate complex systems (Santner et al., 2013; Fang et al., 2006; Morris & Moore, 2015). The most suitable designs for such experiments are space-filling Latin hypercube designs (Lin & Tang, 2015). Several criteria have been proposed to measure space-filling, including discrepancy criteria via reproducing kernel Hilbert spaces (Hickernell, 1998) and maximin and minimax distance criteria (Johnson et al., 1990). In this paper, we adopt the maximin distance criterion, which maximizes the minimum distance between design points. This criterion optimizes the worst case, thus generating robust space-filling designs. Johnson et al. (1990) showed that maximin distance designs are asymptotically optimal under a Bayesian setting. Morris & Mitchell (1995), Joseph & Hung (2008), Ba et al. (2015) and many others proposed algorithms to construct maximin Latin hypercube designs; see Lin & Tang (2015) for a summary. To the best of our knowledge, the R package SLHD by Ba et al. (2015) implements the most efficient current algorithm. Zhou & Xu (2015) proposed to construct maximin Latin hypercube designs via good lattice point sets.

Morris (1991) and Kleijnen (1997) gave many computer models involving several hundred factors, which may require run-economic designs. Under such a situation, it is not unreasonable to assume effect sparsity, that is, relatively few active factors. Loeppky et al. (2009) provided an informal rule of thumb that the number of runs for a computer experiment should be around ten times the input dimension, but also suggested that, under effect sparsity, the run size should be around ten times the effective dimension, given good a priori knowledge on the number of active factors. In order to identify active factors from a large number of factors with limited budgets or runs, saturated or even supersaturated Latin hypercube designs are useful; see, for



(a)

$$\begin{array}{|c|c|c|c|c|}\hline -2 & 1 & -4 & 2 & -1 \\ \hline -1 & -3 & -2 & 1 \\ \cline{1-4} -5 & -1 & -3 \\ \cline{1-3} -3 & -2 \\ \cline{1-2} -4 \\ \cline{1-1}\end{array}$$

(b)

$$\begin{pmatrix} 6 & 4 & 5 & 1 & 3 & 2 \\ 2 & 6 & 4 & 5 & 1 & 3 \\ 3 & 2 & 6 & 4 & 5 & 1 \\ 1 & 3 & 2 & 6 & 4 & 5 \\ 5 & 1 & 3 & 2 & 6 & 4 \\ 4 & 5 & 1 & 3 & 2 & 6 \end{pmatrix}$$

(c)

$$\begin{array}{|c|c|c|c|c|c|}\hline 6 & -2 & 1 & -4 & 2 & -1 \\ \hline 4 & -1 & -3 & -2 & 1 \\ \cline{1-5} 5 & -5 & -1 & -3 \\ \cline{1-4} 1 & -3 & -2 \\ \cline{1-3} 3 & -4 \\ \cline{1-2} 2 \\ \cline{1-1}\end{array}$$

(d)

$$\begin{pmatrix} 0 & 6 & 4 & 5 & 1 & 3 & 2 \\ 2 & 0 & 6 & 4 & 5 & 1 & 3 \\ 3 & 2 & 0 & 6 & 4 & 5 & 1 \\ 1 & 3 & 2 & 0 & 6 & 4 & 5 \\ 5 & 1 & 3 & 2 & 0 & 6 & 4 \\ 4 & 5 & 1 & 3 & 2 & 0 & 6 \\ 6 & 4 & 5 & 1 & 3 & 2 & 0 \end{pmatrix}$$

Fig. 1. (a) Difference triangle and (b) cyclic Latin square from Costas array $(6,4,5,1,3,2)$; (c) difference triangle and (d) cyclic Latin square from vector $(0,6,4,5,1,3,2)$.

example, Butler (2001, 2007). Yet, the construction of such maximin Latin hypercube designs is challenging.

An $n \times n$ Latin square is a supersaturated Latin hypercube design where each row and each column is a permutation of n levels. We propose three algebraic methods for constructing $n \times n$ maximin Latin squares, where $n = q, q-1$ or $q-2$ and q is a prime or a prime power. We study their properties and derive lower bounds on their minimum distances. The generated Latin squares and related saturated $n \times (n-1)$ Latin hypercube designs have larger minimum distances than existing ones. Our methods are associated with Costas arrays, which are introduced next.

2. Costas arrays and the Welch method

Costas arrays are widely used in radar and sonar applications due to their ideal autocorrelation properties (Costas, 1984; Drakakis, 2006). A Costas array of order n can be represented geometrically by allocating n points on an $n \times n$ checker-board, such that each row and column has only one point and all of the $n(n-1)/2$ displacement vectors between each pair of points are distinct. Costas arrays can be represented algebraically as permutation vectors, which are used in this paper.

Definition 1 (Difference triangle). *For any vector $a = (a_1, \ldots, a_n)$, the difference triangle $\mathcal{T}(a)$ is $(t_{i,j})$, where $t_{i,j} = a_{i+j} - a_j$ for $i = 1, \ldots, n-1$ and $j = 1, \ldots, n-i$.*

Definition 2 (Costas array). *Let $a = (a_1, \ldots, a_n)$ be a permutation of $1, \ldots, n$. Then a is a Costas array of order n if and only if no row in the difference triangle $\mathcal{T}(a)$ contains a repeated value.*

Figure 1(a) shows the difference triangle, $\mathcal{T}(a)$, for a permutation vector $a = (6,4,5,1,3,2)$. All elements in each row of $\mathcal{T}(a)$ are distinct, so a is a Costas array.

An $n \times k$ Latin hypercube design is an $n \times k$ matrix where each column is a permutation of n equally-spaced levels, which are denoted by n consecutive numbers, say, 1 to n or 0 to $n-1$. The minimum distance of a design D, denoted by $d_{\min}(D)$, is the minimum distance between any two distinct rows. In this paper we consider L_1-distance, also known as the rectangular or Manhattan distance. For any $n \times k$ Latin hypercube design, the average row pairwise L_1-distance is $(n+1)k/3$ (Zhou & Xu, 2015). The minimum distance cannot exceed the integer part of the average; thus we have the following upper bound.

Lemma 1. *For any $n \times k$ Latin hypercube design D, $d_{\min}(D) \leqslant d_{\text{upper}} = \lfloor (n+1)k/3 \rfloor$, where $\lfloor x \rfloor$ is the integer part of x.*

Let p be a prime throughout the paper. In Galois field $\mathbb{F}_p$, a number α is a primitive root modulo p if and only if for every nonzero element i in $\mathbb{F}_p$ there exists an integer k such that $\alpha^k = i \mod p$. In other words, if α is a primitive root modulo p, the vector $(\alpha, \alpha^2, \ldots, \alpha^{p-1}) \mod p$ is a permutation of $1, \ldots, p-1$. The Welch–Costas array is defined as follows; see Golomb (1984) and Drakakis (2006).

Definition 3 (Welch–Costas array). *Let α be a primitive root modulo p. For $i = 1, \ldots, p-1$, let $a_i = \alpha^{i-1+c} \mod p$ where c is an integer and $1 \leqslant c \leqslant p-1$. The permutation vector $a = (a_1, \ldots, a_{p-1})$ is a Costas array of order $p-1$.*

From a Welch–Costas array of order $p-1$, we can generate a $(p-1) \times (p-1)$ cyclic Latin square by right shifting the vector $p-2$ times. We can also generate a $p \times p$ cyclic Latin square by augmenting the vector with an additional element 0.

Example 1. For $p = 7$, the Welch–Costas array with primitive root $\alpha = 3$ and parameter $c = 3$ is $a = (6, 4, 5, 1, 3, 2)$. Figure 1(b) shows the 6×6 cyclic Latin square generated by a. Its minimum distance is 12. To construct a 7×7 Latin square, we use $a_* = (0, a)$ as the generator which is the first row of the cyclic design. Figures 1(c) and (d) show the difference triangle $\mathcal{T}(a_*)$ and the Latin square, respectively. Its minimum distance is 18.

Lemma 2. *For any $n \times n$ cyclic Latin square D with generator a, there are at most $\lfloor n/2 \rfloor$ distinct pairwise L_1-distances, and its ith $(i = 1, \ldots, \lfloor n/2 \rfloor)$ possible distance is the sum of the absolute values of all elements in the ith and $(n-i)$th row of the difference triangle $\mathcal{T}(a)$.*

As an illustration, the 6×6 cyclic design in Fig. 1(b) has three possible distances: 14, 12 and 18 which are calculated via the (1st, 5th), (2nd, 4th) and (3rd, 3rd) rows of $\mathcal{T}(a)$ in Fig. 1(a), respectively. With Lemma 2, for an $n \times n$ cyclic design, it requires only $O(n^2)$ operations to determine the minimum distance, while for a general $n \times n$ design, it requires $O(n^3)$ operations.

Proposition 1. *All possible $(p-1) \times (p-1)$ cyclic Latin squares via generators of Welch–Costas arrays with order $p-1$ are equivalent under row and column permutations.*

From the proof of Proposition 1, we can see that all such cyclic Latin squares are equivalent to the leave-one-out good lattice point designs in Zhou & Xu (2015). Thus, the minimum distance of all such designs is $(p^2-1)/4$ by Theorem 4 and Proposition 2 in Zhou & Xu (2015).

The $(p-1) \times (p-1)$ Welch designs have bad two-dimensional projections. For example, the points of the Welch design in Fig. 1(b) lie on the diagonal when projected onto the first and fourth columns. Here we propose a simple modification: replace $p-1$ with 0 when constructing $(p-1) \times (p-1)$ Welch designs. The modified Welch designs not only have improved projections and column correlations, but also have larger minimum distances when $p > 7$, though for $p = 5$ and $p = 7$, they have smaller minimum distances. See § 5 for details.

Comparing Figs. 1(a) and (c), $\mathcal{T}(a_*)$ is equivalent to $\mathcal{T}(a)$ adding the Costas array $(6, 4, 5, 1, 3, 2)$ as the first column. Even if a is a Costas array, $a_* = (0, a)$ may or may not be one.

Lemma 3. *Let $a_* = (0, a)$. The difference triangle $\mathcal{T}(a_*)$ is equivalent to $\mathcal{T}(a)$ adding the vector a as the first column.*

Theorem 1. *Let $p \geqslant 5$ be any prime and a be any Welch–Costas array of order $p-1$. The $p \times p$ cyclic Latin square D with generator $a_* = (0, a)$ has $d_{\min}(D) \geqslant (p^2+7)/8 + 2$.*

This bound is very conservative and in practice the results are much better. If α is a primitive root modulo p, $\beta = \alpha^{-1} \mod p$ is another primitive root modulo p. The number of different primitive roots modulo p can be calculated by the Euler's totient function $\phi(p-1)$, which counts the number of integers up to $p-1$ that are coprime to $p-1$; $\phi(n) = n\prod_{t|n}(1-1/t)$, where the product is over all distinct prime numbers t dividing n.

Example 2. For $p = 7$, the primitive roots are 3 and 5. From either primitive root, we can construct six Welch–Costas arrays with order 6, and then construct six 7×7 Latin squares. Five designs have $d_{\min} = 16$ and one design has $d_{\min} = 18$. The lower bound in Theorem 1 is 9 and the upper bound in Lemma 1 is 18. The best design from our construction achieves the upper bound, and the worst designs have much larger minimum distance than the lower bound.

Proposition 2. *Let α be a primitive root modulo p and $\beta = \alpha^{-1} \mod p$. Let a and b be two Welch–Costas arrays with primitive roots α and β, and parameters c_1 and c_2, respectively. When $c_1 + c_2 = 1 \mod (p-1)$, the $p \times p$ cyclic Latin squares with generators $a_* = (0, a)$ and $b_* = (0, b)$ have the same distance distribution.*

As an illustration, when $p = 13$, using two Welch–Costas arrays with $\alpha = 2$, $c_1 = 8$ and $\beta = 7$, $c_2 = 5$, we can generate two 13×13 designs with the same distance distribution and minimum distance of 56. Proposition 2 shows that it is equivalent to use primitive root α and $\beta = \alpha^{-1} \mod p$ in the construction. Thus, in all we only need to compare $\phi(p-1)(p-1)/2$ possible designs.

3. Gilbert method

The Gilbert construction was proposed by Gilbert (1965) and called the logarithmic Welch construction by Costas (1984). Gilbert (1965) used these arrays to construct Latin squares without repeated diagrams. Our purpose and use of these arrays are different from his.

Definition 4 (Gilbert–Costas array). *Let β be a primitive root modulo p. For $i = 1, \ldots, p-1$, let $b_i = \log_\beta(i) + 1 - c \mod (p-1)$, where $c = 1, \ldots, p-1$; if $b_i = 0$ set $b_i = p-1$. The permutation vector $b = (b_1, \ldots, b_{p-1})$ is a Costas array of order $p-1$.*

Gilbert–Costas arrays are inverse permutations of Welch–Costas arrays. As any permutation is a bijection, if $\{f(1), \ldots, f(n)\}$ is a permutation of $\{1, \ldots, n\}$, its inverse permutation is $\{f^{-1}(1), \ldots, f^{-1}(n)\}$.

Example 3. For $p = 7$, with primitive root 3 and parameter $c = 1$, the corresponding Welch–Costas array is $a = (3, 2, 6, 4, 5, 1)$ and the Gilbert–Costas array is $b = (6, 2, 1, 4, 5, 3)$. It is clear that b is the inverse permutation of a. The 6×6 cyclic Latin square with generator b is an equal distance design with all pairwise distances equal to 14. The 7×7 cyclic Latin square with generator $b_* = (0, b)$ has $d_{\min} = 14$.

Theorem 2. *Let $p \geqslant 5$ be a prime and b be a Gilbert–Costas array of order $p-1$. The $(p-1) \times (p-1)$ cyclic Latin square D with generator b has $d_{\min}(D) \geqslant (p-1)(p+3)/8$ when $p = 1 \mod 4$, and $d_{\min}(D) \geqslant (p+1)^2/8$ when $p = 3 \mod 4$.*

This lower bound in Theorem 2 is tight for $p = 5$ or 7. For example, the 6×6 design with generator $b = (5, 1, 6, 3, 4, 2)$, a Gilbert–Costas array with primitive root 3 and parameter $c = 2$, has $d_{\min} = 8$ which equals the lower bound.

Proposition 3. *Let a and b be two Gilbert–Costas arrays with primitive roots α and β, and parameters c_1 and c_2, respectively. The $(p-1) \times (p-1)$ cyclic Latin squares with generators a and b have the same distance distribution under either of the following conditions:* (i) $\alpha = \beta$ *and* $c_1 - c_2 = (p-1)/2 \mod (p-1)$*;* (ii) $\beta = \alpha^{-1} \mod p$ *and* $c_1 + c_2 = 1 \mod (p-1)$.

By Proposition 3, there are at most $\phi(p-1)(p-1)/4$ designs with different minimum distances via the Gilbert construction. For example, when $p = 7$, the generated cyclic designs via primitive root 3 and parameters 1, 2, 3, 4, 5 and 6 have the same distance distribution as the designs via primitive root 5 and parameters 6, 5, 4, 3, 2 and 1, and their minimum distances are 14, 8, 12, 14, 8 and 12, respectively.

Theorem 3. *Let $p \geqslant 5$ be a prime and b be a Gilbert–Costas array of order $p-1$. The $p \times p$ cyclic Latin square D with generator $b_* = (0, b)$ has $d_{\min}(D) \geqslant (p^2 + 7)/4$.*

This lower bound in Theorem 3 is roughly 75% of d_{upper} in Lemma 1 for large p, which nearly doubles the lower bound in Theorem 1.

Proposition 4. *Let b be any Gilbert–Costas array of order $p-1$ via primitive root β modulo p. All possible $p \times p$ cyclic Latin squares with generators $b_* = (0, b)$ are equivalent under row and column permutations.*

By Proposition 4, the $p \times p$ cyclic Latin squares generated via Gilbert–Costas arrays do not depend on parameter c. Thus, in all we have $\phi(p-1)$ possible designs.

4. Golomb method

Let $q = p^m$ be a prime power and consider the Galois field $\mathbb{F}_q$. If $m = 1$, elements and primitive roots are integers in $\mathbb{F}_p$. If $m \geqslant 2$, the elements and primitive roots in $\mathbb{F}_q$ are polynomials. If α is a primitive root, $\alpha^{q-1} = 1$ and $(\alpha, \alpha^2, \ldots, \alpha^{q-1})$ is a permutation vector of nonzero elements in $\mathbb{F}_q$. There are $\phi(q-1)$ primitive roots in $\mathbb{F}_q$. Golomb (1984) constructed the following Costas arrays.

Definition 5 (Golomb–Costas array). *Let α and β be two primitive roots in $\mathbb{F}_q$ where $q = p^m$. For $i, j = 1, \ldots, q-2$, let $g_i = j$ if $\alpha^i + \beta^j = 1$ in $\mathbb{F}_q$. The permutation vector $g = (g_1, \ldots, g_{q-2})$ is a Costas array of order $q-2$.*

The two primitive roots α and β are not necessarily different. By switching α and β, we obtain another Golomb–Costas array, which is the inverse permutation.

Theorem 4. *Let $q = p^m \geqslant 7$ be a prime power and g be a Golomb–Costas array of order $q-2$. The $(q-2) \times (q-2)$ cyclic Latin square D with generator g has $d_{\min}(D) \geqslant q^2/8$ for even q and $d_{\min}(D) \geqslant (q^2 - 1)/8$ for odd q.*

Theorem 5. *Let $q = p^m \geqslant 7$ be a prime power and g be a Golomb–Costas array of order $q-2$. The $(q-1) \times (q-1)$ cyclic Latin square D with generator $g_* = (0, g)$ has $d_{\min}(D) \geqslant (q-1)(q-3)/4 + 2$ for odd q and $d_{\min}(D) \geqslant (q-2)^2/4 + 3$ for even q.*

PROPOSITION 5. *Given the same primitive root β and possible different α in $\mathbb{F}_q$, all $(q-1) \times (q-1)$ cyclic Latin squares with generators $g_* = (0, g)$ are equivalent under row and column permutations.*

The lower bounds of $d_{\min}$ in Theorems 4 and 5 are roughly 37·5% and 75% of the upper bound d_{upper} in Lemma 1 for large p, respectively. These bounds are conservative and in practice the minimum distances of Golomb designs are much larger.

Example 4. Let $q = 2^4 = 16$ and set the irreducible polynomial as $x^4 + x + 1$ over $\mathbb{F}_{16}$. Set primitive roots $\alpha = \beta = x$. For $i = 1, \ldots, 14$, solving equations $x^i + x^j = 1$ in $\mathbb{F}_{16}$, we find solution pairs (i, j) which are $(1, 4)$, $(2, 8)$, $(3, 14)$, $(6, 13)$, $(11, 12)$, $(7, 9)$ and $(5, 10)$ where i and j are interchangeable in the solution pairs since $\alpha = \beta$. By Definition 5, this Golomb–Costas array is $g = (4, 8, 14, 1, 10, 13, 9, 2, 7, 5, 12, 11, 6, 3)$. The 14×14 Latin square with generator g has minimum distance of 62, and a ratio $(d_{\min}/d_{\text{upper}})$ of 89%. This is much better than the lower bound in Theorem 4 which is 32. The 15×15 Latin square with generator $g_* = (0, g)$ has minimum distance of 70 and a ratio $(d_{\min}/d_{\text{upper}})$ of 88%, where the lower bound in Theorem 5 is 58.

Setting $m = 1$, the Golomb method can efficiently generate $(p-2) \times (p-2)$ and $(p-1) \times (p-1)$ maximin designs. In generating $(p-1) \times (p-1)$ designs, this lower bound in Theorem 5 nearly doubles the lower bound in Theorem 2 where the Gilbert method is used.

Example 5. For $p = 13$, there are four primitive roots 2, 6, 7 and 11, and thus in all 16 possible Golomb–Costas arrays g. With generators g, we can construct four 11×11 designs with $d_{\min} = 38$ and twelve designs with $d_{\min} = 40$. By Proposition 5, with generators $g_* = (0, g)$, we can fix $\alpha = 2$ and there are four possible 12×12 designs whose $d_{\min}$ are 38, 40, 42 and 48, respectively. As a comparison, the best 12×12 Gilbert design has $d_{\min} = 46$.

5. RESULTS AND COMPARISONS

In this section, we compare our three methods with the R package SLHD by Ba et al. (2015) and the good lattice point method by Zhou & Xu (2015). The following lemma is straightforward.

LEMMA 4. *Let D be a Latin square with levels 1 to n and D' be the $(n+1) \times n$ design obtained by adding a row of zeros to D. Then $d_{\min}(D') = d_{\min}(D)$.*

With Lemma 4, we generate $p \times (p-1)$ Latin hypercube designs by adding a row of zeros to our $(p-1) \times (p-1)$ Latin squares from the Welch, Gilbert or Golomb method. Table 1 compares $p \times (p-1)$ Latin hypercube designs constructed via different methods. The $p \times (p-1)$ Welch designs are equivalent to good lattice point designs whereas the modified Welch designs have larger minimum distances than good lattice point designs when $p > 7$. For the modified Welch designs we add a row of $(p-1)$s to the $(p-1) \times (p-1)$ Latin squares whose levels are from 0 to $p-2$. The Gilbert and Golomb designs outperform good lattice point designs for all cases and outperform linearly permuted good lattice point designs for most cases. For the R package SLHD, we run the command maximinSLHD with option $t = 1$ and default settings for 100 times, and choose the best results. The best of the Gilbert and Golomb methods are comparable to the R package SLHD, especially for large p. All of our three methods are much faster than the R package SLHD. For example, it takes about an hour for the 97×96 case using the R package

Table 1. *Comparison of minimum L_1-distances for $p \times (p-1)$ Latin hypercube designs*

p	mWel	Gil	Gol	GLP	LGLP	SLHD	p	mWel	Gil	Gol	GLP	LGLP	SLHD
7	10	14	14	12	13	15	47	596	672	668	552	676	672
11	32	34	34	30	34	37	53	752	848	856	702	846	857
13	52	46	48	42	54	50	59	926	1056	1050	870	1050	1067
17	82	86	80	72	84	87	61	988	1134	1130	930	1132	1135
19	104	102	106	90	106	108	67	1186	1372	1378	1122	1362	1370
23	152	154	158	132	154	159	71	1328	1518	1538	1260	1516	1541
29	236	250	244	210	250	253	73	1402	1632	1634	1332	1596	1628
31	268	276	292	240	280	289	79	1636	1888	1898	1560	1872	1919
37	376	408	404	342	408	411	83	1802	2122	2112	1722	2090	2120
41	458	512	498	420	508	510	89	2066	2442	2456	1980	2382	2435
43	502	558	542	462	562	562	97	2446	2902	2872	2352	2886	2898

mWel, modified Welch method; Gil, Gilbert method; Gol, Golomb method; GLP, good lattice point method; LGLP, linearly permuted good lattice point method; SLHD, R package SLHD.

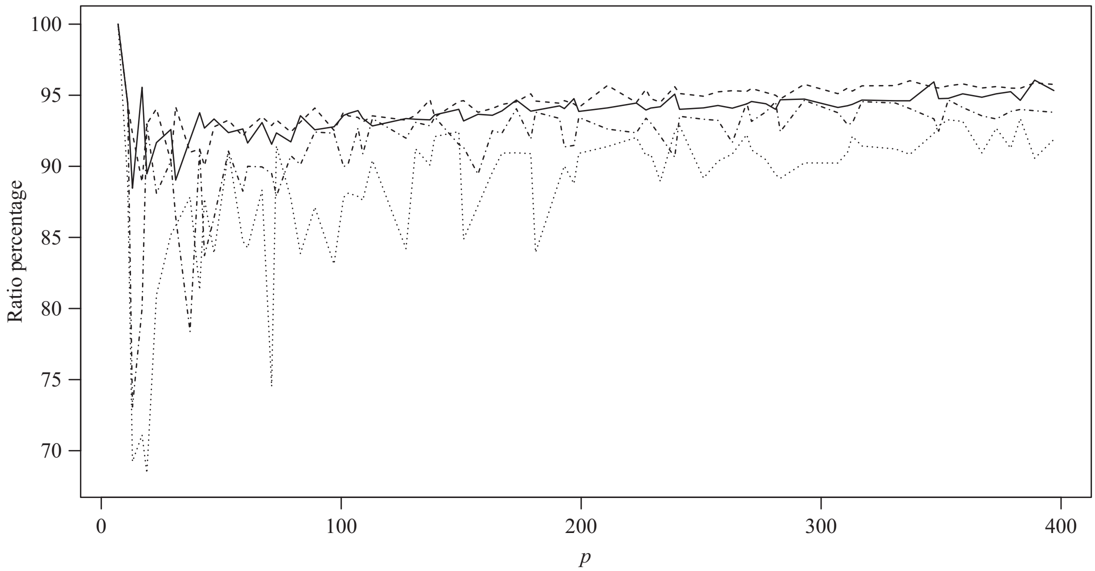


Fig. 2. Ratio percentages for $(p-1) \times (p-1)$ Latin squares generated by Gilbert method (solid), Golomb method (dashed), simplified Gilbert method (dotted), and simplified Golomb method (dot-dash).

SLHD on a laptop with an Intel 2·50GHz I7 CPU, while our algebraic methods take only a few seconds. The minimum distances of our designs can be further improved in some cases by permuting levels as Zhou & Xu (2015) did. We do not pursue this here.

The Gilbert method outperforms the Welch method and the R package SLHD for constructing $p \times p$ Latin hypercube designs when $p \geqslant 29$, and the Golomb method outperforms the R package SLHD in most cases for constructing $(p-2) \times (p-2)$ Latin hypercube designs; see the Supplementary Material.

Our algebraic construction methods are suitable for constructing high-dimensional designs. As p gets larger, the Gilbert and Golomb methods tend to produce better designs in the sense that the ratios of $d_{\text{min}}/d_{\text{upper}}$ become higher as shown in Fig. 2, where d_{upper} is the upper bound given in Lemma 1. Here we further introduce two simplified methods which avoid searching primitive roots and parameters. The simplified Gilbert method uses the smallest primitive root and parameter $c = 1$. The simplified Golomb method uses the smallest primitive root as α and the second smallest primitive root as β. Figure 2 shows that all of our methods perform well when p is large. The simplified Golomb method is better than the simplified Gilbert method, and the ratios of $d_{\text{min}}/d_{\text{upper}}$ are near or above 90% when $p > 100$ for the former method. It would be interesting to find the explicit forms of d_{min} for the Gilbert and Golomb methods or to study their asymptotical properties.

From a Latin square, we can generate many Latin hypercube designs by deleting one or more columns. Deleting one column from an $n \times n$ Latin square reduces the minimum distance by at most $n - 1$. If we start with an $n \times n$ design with large $d_{\text{min}}/d_{\text{upper}}$ ratio, we can drop a small number of columns which will lead to good designs with large minimum distances. To drop a comparatively large number of columns, one can adopt a searching scheme such as threshold accepting, which has been thoroughly discussed in Fang et al. (2006).

Based on the Welch, Gilbert and Golomb constructions, there are some secondary constructions of Costas arrays with orders of p, $p - 2$, $p - 3$, p^m, $p^m - 1$, $p^m - 3$, $p^m - 4$ and $p^m - 5$; see Beard (2006) and Drakakis et al. (2011). As a generalization of our methods, we can also use these Costas arrays to construct cyclic Latin squares. It is straightforward to extend all theoretical results in this paper using the L_2-distance.

ACKNOWLEDGEMENT

This research was supported by the U.S. National Science Foundation. The authors thank the editor and two reviewers for their helpful comments.

SUPPLEMENTARY MATERIAL

Supplementary material available at *Biometrika* online includes tables comparing the Welch, Gilbert and Golomb methods and the R package SLHD in generating $p \times p$ and $(p-2) \times (p-2)$ Latin hypercube designs, and proofs of Theorems 2 and 4 and Propositions 3, 4 and 5.

APPENDIX

Proof of Lemma 2. Denote $x_1 = (a_1, \ldots, a_n)$ and $x_i = (a_{n-i+2}, \ldots, a_n, a_1, \ldots a_{n-i+1})$ for $2 \leqslant i \leqslant n$. Let $a_0 = a_n$ for convenience. For any $i < j$, x_j is obtained from x_i by applying $k = j - i$ steps of right-cyclic shift, and their L_1-distance is $\sum_{i=1}^{n} \mid a_{(i+k) \text{ mod } n} - a_i \mid$ which is denoted as d_k here. Further, $d_{n-k} = \sum_{i=1}^{n} \mid a_{(i+n-k) \text{ mod } n} - a_i \mid = \sum_{i=1}^{n} \mid a_{(i-k) \text{ mod } n} - a_i \mid = \sum_{i=1}^{n} \mid a_{(i+k) \text{ mod } n} - a_i \mid = d_k$. Thus, all pairwise L_1-distances can be categorized into $\lfloor n/2 \rfloor$ groups which are represented by the L_1-distances between its 1st row and its 2nd, …, $(\lfloor n/2 \rfloor + 1)$th row. Furthermore, $d_k = \sum_{i=1}^{n} \mid a_{(i+k) \text{ mod } n} - a_i \mid = \sum_{i=1}^{n-k} \mid a_{i+k} - a_i \mid + \sum_{i=n-k+1}^{n} \mid a_{i+k-n} - a_i \mid = \sum_{i=1}^{n-k} \mid a_{i+k} - a_i \mid + \sum_{j=1}^{k} \mid a_j - a_{n-k+j} \mid = \sum_{i=1}^{n-k} \mid t_{k,i} \mid + \sum_{j=1}^{k} \mid t_{n-k,j} \mid$, where $t_{k,j}$ is the jth element in the kth row of the difference triangle $\mathcal{T}(a)$. This completes the proof. □

Proof of Proposition 1. Let $D_{\alpha,c}$ be the $(p - 1) \times (p - 1)$ generated design using the Welch–Costas array with primitive root α and parameter c. Denote the (ith, jth) element in design $D_{\alpha,c}$ where $c \neq 0$ as $x_{i,j}$, and in design $D_{\alpha,0}$ as $y_{i,j}$. Let $j' = j + c \mod (p - 1)$ and if $j' = 0$ set $j' = p - 1$. We have $x_{i,j} = x_{1,j-i+1 \text{ mod } (p-1)} = \alpha^{j-i+c \text{ mod } (p-1)} \mod p = \alpha^{j'-i \text{ mod } (p-1)} \mod p = y_{i,j'}$. Thus, any $D_{\alpha,c}$ where $c \neq 0$ is equivalent to $D_{\alpha,0}$ under column permutations. Without loss of generality, let $c = 0$ in the following proof. For any two different primitive roots α and β, there exists a unique integer t which is coprime to $p - 1$, such that $\beta = \alpha^t \mod p$. Denote the (ith, jth) element in design $D_{\beta,0}$ as $z_{i,j}$. Let $i' = ti$

mod $(p-1)$ and $j' = tj \mod (p-1)$. Then $j' - i' = t(j-i) \mod (p-1)$ and $\beta^{j'-i'} = \beta^{t(j-i)} = \alpha^{j-i}$ mod p. This leads to $z_{i',j'} = y_{i,j}$. Thus, $D_{\alpha,0}$ and $D_{\beta,0}$ are equivalent under row and column permutations. This completes the proof. □

Proof of Theorem 1. With Lemmas 2 and 3 and the Costas property of $\mathcal{T}(a)$ that there are no repeated values in any row of the difference triangle, for the $p \times p$ generated design, the lower bound of the pairwise distances can only occur between the 1st and $\{(p+1)/2\}$th row under the following situations.

(i) When $p = 4k+1$ and $k \geqslant 2$, the lower bound occurs when the $(2k)$th row of $\mathcal{T}(a)$ consists of numbers: $-1, 1, \ldots, -k, k$, the $(2k+1)$th row of $\mathcal{T}(a)$ consists of numbers: $-1, 1, \ldots, -(k-1), (k-1), -k$ or k, and the two elements added at the first position are 1 and 2. Under such a situation, by Lemma 2, $d_{\min}(D) \geqslant 4 \times \{1 + \cdots + (k-1)\} + 3k + 1 + 2 = 2k^2 + k + 3 = (p^2+7)/8 + 2$. The bound also holds for $p = 5$.

(ii) When $p = 4k+3$ and $k \geqslant 1$, the lower bound occurs when the $(2k+1)$th row of $\mathcal{T}(a)$ consists of numbers: $-1, 1, \ldots, -(k+1)$ or $(k+1)$, the $(2k+2)$th row of $\mathcal{T}(a)$ consists of numbers: $-1, 1, \ldots, -k, k$, and the two elements added at the first position are 1 and 2. Under such a situation, by Lemma 2, $d_{\min}(D) \geqslant 4 \times (1 + \cdots + k) + (k+1) + 1 + 2 = 2k^2 + 3k + 4 = (p^2+7)/8 + 2$. □

Proof of Proposition 2. Denote two Welch–Costas arrays as a and b where $a_j = \alpha^{j+c_1-1} \mod p$ and $b_j = \beta^{j+c_2-1} \mod p$. Given $\beta = \alpha^{-1} \mod p$ and $c_1 + c_2 = 1 \mod (p-1)$, $b_j = \alpha^{-j-c_2+1} = \alpha^{p-j-c_2} = \alpha^{p-j+c_1-1} = a_{p-j} \mod p$. Thus, b is the inverse reflection of a. When only considering absolute values and ignoring the order, elements are the same for every uth $(u = 1, \ldots, p-2)$ row of difference triangles $\mathcal{T}(a)$ and $\mathcal{T}(b)$. Define $a_* = (0, a)$ and $b_* = (0, b)$. With Lemma 3, for $\mathcal{T}(a_*)$ and $\mathcal{T}(b_*)$, the sum of the first element of the uth $(u = 1, \ldots, (p-1)/2)$ and $(p-u)$th rows are the same. Further, by Lemma 2 the $p \times p$ designs with generators of a_* and b_* have the same distance distribution. □

Proof of Theorem 3. We first prove a claim that in the difference triangle $\mathcal{T}(b)$, if number v exists in the uth row, $2 \leqslant u \leqslant (p-1)/2$, then number $-v$ cannot exist in the $(p-u)$th row. Suppose otherwise, by Definitions 1 and 4, there exist integers i and j where $1 \leqslant i, j \leqslant p-1$, $1 \leqslant i+u \leqslant p-1$, $1 \leqslant j+p-u \leqslant p-1$ and $1 \leqslant\mid v \mid\leqslant p-1$, such that $\log_\beta(i) - \log_\beta(i+u) = v \mod (p-1)$ and $\log_\beta(j) - \log_\beta(j+p-u) = -v \mod (p-1)$. Then, we have $i = (i+u)\beta^v \mod p$ and $j+p-u = j\beta^v$ mod p· This leads to $ij\beta^v = (i+u)(j-u)\beta^v \mod p$. Since $\beta^v \not\equiv 0 \mod p$, we have $ij = (i+u)(j-u)$ mod p or $u(j-i-u) = 0 \mod p$. Since $u \not\equiv 0 \mod p$, we have $j = u+i \mod p$. Since $1 \leqslant j \leqslant p-1$ and $1 \leqslant i+u \leqslant p-1$, we have $j = u+i$. But for $1 \leqslant j+p-u \leqslant p-1$, we have $1 \leqslant i+p \leqslant p-1$ which is a contradiction to $1 \leqslant i \leqslant p-1$. Thus, our claim is proved.

With Definition 2, Lemma 3 and the proved claim above, ignoring the first column of $\mathcal{T}(b_*)$, for any $u = 2, \ldots, (p-1)/2$, considering the absolute values of elements in the uth and $(p-u)$th row of $\mathcal{T}(b_*)$ together, no value can appear more than twice. Since β is a primitive root modulo p, $\beta^{(p-1)/2} = p-1 \mod p$ and $\log_\beta(p-1) = (p-1)/2$. Then $p-u = (p-1)u \mod p$ and $\log_\beta(p-u) = \log_\beta(p-1) + \log_\beta(u) = (p-1)/2 + \log_\beta(u) \mod (p-1)$. This implies $b_{p-u} = b_u + (p-1)/2 \mod (p-1)$. Therefore, with Lemma 2, the lower bound is $d_{\min}(D) \geqslant 2 \times \{1 + \cdots + (p-3)/2\} + (p-1)/2 + 1 + 1 + (p-1)/2 = (p^2+7)/4$. When considering $u = 1$, by Definition 2 and Lemma 2, it is straightforward that the above lower bound stands. □

Proof of Theorem 5. First we prove a claim that for Golomb–Costas array $g = (g_1, \ldots g_{q-2})$ where $q = p^m$ with primitive roots α and β, in the difference triangle $\mathcal{T}(g)$, if number v exists in the uth row where $2 \leqslant u \leqslant (q-1)/2$, then number $-v$ cannot exist in the $(q-1-u)$th row. Suppose otherwise, by Definition 5 there exist integers i and j where $1 \leqslant i, j, g_i, g_j \leqslant q-2$, $1 \leqslant i+u \leqslant q-2$, $1 \leqslant j+q-1-u \leqslant q-2$

and $1 \leqslant| v |\leqslant q-2$, such that in Galois field $\mathbb{F}_q$,

$$\begin{cases}\alpha^i+\beta^{g_i}=1,\\ \alpha^{i+u}+\beta^{g_i+v}=1,\\ \alpha^j+\beta^{g_j}=1,\\ \alpha^{j+q-1-u}+\beta^{g_j-v}=1,\end{cases} \Rightarrow \begin{cases}\alpha^i\beta^v+\beta^{g_i+v}=\beta^v,\\ \alpha^{i+u}+\beta^{g_i+v}=1,\\ \alpha^j+\beta^{g_j}=1,\\ \alpha^{j-u}\beta^v+\beta^{g_j}=\beta^v,\end{cases} \Rightarrow \begin{cases}\alpha^i(\beta^v-\alpha^u)=\beta^v-1,\\ \alpha^{j-u}(\beta^v-\alpha^u)=\beta^v-1.\end{cases}$$

In $\mathbb{F}_q$, if $\beta^v-\alpha^u \neq 0$, we have $\alpha^i=\alpha^{j-u}$. Since α is primitive root and given the range of i and j, we have $j=i+u$. Then $j+q-1-u=i+q-1>q-2$ contradicts with the condition $1 \leqslant j+q-1-u \leqslant q-2$. In $\mathbb{F}_q$, if $\beta^v-\alpha^u=0$, we have $\alpha^i+\beta^{c_i}=1$ and $\alpha^{i+u}+\beta^{c_i+v}=1$. Thus, we have $\alpha^i+\beta^{c_i}=1$ and $\alpha^u(\alpha^i+\beta^{c_i})=1$. Then, $\alpha^u=1$ and $u=q-1$. This contradicts with the range $2 \leqslant u \leqslant (q-1)/2$. Thus, our claim is proved. With this claim, similar to the proof of Theorem 3, we can compute the lower bound as follows. When q is odd, $d_{\min}(D) \geqslant 2 \times \{1+\cdots+(q-3)/2\}+1 \times 2=(q-1)(q-3)/4+2$; when q is even, $d_{\min}(D) \geqslant 2 \times \{1+\cdots+(q-4)/2\}+(q-2)/2+1+2=(q-2)^2/4+3$. □

REFERENCES

BA, S., MYERS, W. R. & BRENNEMAN, W. A. (2015). Optimal sliced Latin hypercube designs. *Technometrics* **57**, 479–87.

BEARD, J. K. (2006). Generating Costas arrays to order 200. In *The 40th Annual Conference on Information Sciences and Systems*. IEEE, pp. 1130–3.

BUTLER, N. A. (2001). Optimal and orthogonal Latin hypercube designs for computer experiments. *Biometrika* **88**, 847–57.

BUTLER, N. A. (2007). Supersaturated Latin hypercube designs. *Commun. Statist.* A **34**, 417–28.

COSTAS, J. P. (1984). A study of a class of detection waveforms having nearly ideal range-doppler ambiguity properties. *Proc. IEEE* **72**, 996–1009.

DRAKAKIS, K. (2006). A review of Costas arrays. *J. Appl. Math.* **2006**, 26385.

DRAKAKIS, K., IORIO, F. & RICKARD, S. (2011). The enumeration of Costas arrays of order 28 and its consequences. *Adv. Math. Commun.* **5**, 69–86.

FANG, K.-T., LI, R. & SUDJIANTO, A. (2006). *Design and Modeling for Computer Experiments*. New York: Chapman and Hall/CRC.

GILBERT, E. N. (1965). Latin squares which contain no repeated digrams. *SIAM Rev.* **7**, 189–98.

GOLOMB, S. W. (1984). Algebraic constructions for Costas arrays. *J. Combin. Theory* **37**, 13–21.

HICKERNELL, F. (1998). A generalized discrepancy and quadrature error bound. *Math. Comp. Am. Math. Soc.* **67**, 299–322.

JOHNSON, M. E., MOORE, L. M. & YLVISAKER, D. (1990). Minimax and maximin distance designs. *J. Statist. Plan. Infer.* **26**, 131–48.

JOSEPH, V. R. & HUNG, Y. (2008). Orthogonal-maximin Latin hypercube designs. *Statist. Sinica* **18**, 171–86.

KLEIJNEN, J. P. (1997). Sensitivity analysis and related analyses: a review of some statistical techniques. *J. Statist. Comp. Simul.* **57**, 111–42.

LIN, C. D. & TANG, B. (2015). Latin hypercubes and space-filling designs. In *Handbook of Design and Analysis of Experiments*, A. Dean, M. Morris, J. Stufken & D. Bingham, eds. New York: Chapman and Hall/CRC, pp. 593–625.

LOEPPKY, J. L., SACKS, J. & WELCH, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics* **51**, 366–76.

MORRIS, M. D. (1991). Factorial sampling plans for preliminary computational experiments. *Technometrics* **33**, 161–74.

MORRIS, M. D. & MITCHELL, T. J. (1995). Exploratory designs for computational experiments. *J. Statist. Plan. Infer.* **43**, 381–402.

MORRIS, M. D. & MOORE, L. M. (2015). Design of computer experiments: Introduction and background. In *Handbook of Design and Analysis of Experiments*, A. Dean, M. Morris, J. Stufken & D. Bingham, eds. New York: Chapman and Hall/CRC, pp. 577–91.

SANTNER, T. J., WILLIAMS, B. J. & NOTZ, W. I. (2013). *The Design and Analysis of Computer Experiments*. New York: Springer.

ZHOU, Y. & XU, H. (2015). Space-filling properties of good lattice point sets. *Biometrika* **102**, 959–66.

[*Received on* 1 *September* 2016. *Editorial decision on* 2 *January* 2017]