Title
Construction of Flexible Maximin Latin Hypercube Designs Based on Good Lattice Point Sets

Author
Zhu, Yongkai

Publication Date
2019

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Construction of Flexible Maximin Latin Hypercube Designs Based on Good Lattice Point Sets

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Yongkai Zhu

2019

ABSTRACT OF THE THESIS

Construction of Flexible Maximin Latin Hypercube Designs
Based on Good Lattice Point Sets

by

Yongkai Zhu
Master of Science in Statistics
University of California, Los Angeles, 2019
Professor Hongquan Xu, Chair

Maximin distance Latin hypercube designs are becoming increasingly prevalent in computer experiments. As addressed by Wang,Xiao and Xu (2018), $p \times (p-1)$ optimal designs, or asymptotically optimal designs based on good lattice point sets have been successfully constructed using Williams transformation; in this paper, we would like to further this idea for more general, or more flexible, designs, such as $N \times \frac{(N-1)}{2}$ for N equal to primes, prime multiples and prime powers, and implement a similar construction algorithm to build optimal or asymptotically optimal designs of such dimension.

The thesis of Yongkai Zhu is approved.

Qing Zhou

Frederic R. Paik Schoenberg

Hongquan Xu, Committee Chair

University of California, Los Angeles

2019

To my parents . . .

for the countless sponsor and support

they have endowed me with in the past

25 years, and also to the professors here

for imparting valuable statistical knowledge to me

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

Computer experimenters are currently seeking designs with points that fill a design region as uniformly as possible, or in short, space-filling designs [Lin and Tang (2015); Wang, Xiao and Xu (2018)]. It was proposed that the most suitable type for computer experiments is space-filling Latin hypercube designs [Xiao and Xu (2017)]. Moreover, since maximin distance criterion, among multiple types of criteria to measure space-filling property, is the ideal one for generating robust space-filling designs [Xiao and Xu (2017)] and is asymptotically optimal under a Bayesian setting [Johnson et al. (1990); Xiao and Xu (2017)], the construction of such maximin distance Latin hypercube designs becomes one of the main concerns for those experiments. According to Wang, Xiao and Xu (2018), the previously proposed algorithms for constructing orthogonal or nearly orthogonal Latin hypercubes were not space-filling in higher dimensions and they instead proposed using Williams transformation and its modified version in the construction, and found success in the optimality of the resulting $p \times (p-1)$, or $N \times \phi(N)$ in general for some natural number N (generally primes, prime multiples or prime powers), and $\phi(N)$ stands for Euler's Phi function that measures the number of natural numbers between 1 and $N-1$ that are coprime to N. Specifically, it was claimed that the resulting designs have weakly correlated columns and, in addition, the average pairwise column correlation tends to zero as the design sizes increase, and thus resolving the issue of lack of space filling in high dimensions in the previously constructed designs [Wang, Xiao and Xu (2018)].

However, the problem becomes much trickier when only a few of the columns are needed. The difficulty might not be realized if the design only has to be reduced by removing one or two columns since one might just try all possibilities or use brutal force to solve the

issue without wasting too much time, but it gradually arises as more columns have to be removed, since brutal force is no longer applicable due to the computational expensiveness. This concern remains untouched in Wang, Xiao and Xu's work in 2018 and most of the other earlier works on this field; also, even though Zhou and Xu (2015) mentioned a simple algorithm to construct LHDs with $\phi(N)/2$ columns using the first half of the set of positive integers coprime to N, the results were not good according to their Theorem 4: they have categorized four different kinds of positive integers N of concern (primes or prime powers, $2p$ for some odd prime p, p_1p_2 for odd primes p_1 and p_2 and powers of 2), and only in the first case, where N is a prime or prime power, did they produce a minimal row-wise distance barely satisfactorily large (with a relative efficiency of 75%; we are going to introduce the concept of efficiency in Chapter 3). Furthermore, we cannot adopt a random procedure to pick the columns, since for large $N's$ there are a myriad of possible column combinations and thus we could end up running 1000 times without getting a single good result; also, a good result on one N does not guarantee a good result on another N value.

In this paper, a selection trick using primitive roots is introduced. For primes p, whereas the $p \times \phi(p)$ design was constructed using Williams tranformation in the same way as stated by Wang, Xiao and Xu (2018), we choose the columns with indices given by the values of a randomly chosen primitive root's even powers in a Galois field $\mathbb{F}_p$; the situation becomes more complex when selection within $n \times \phi(n)$ (n is some prime multiples: $2p$, $3p$, p_1p_2 for some odd primes p, p_1 and p_2, or some prime powers p^k for some integer k) designs is concerned, however, as there is no guarantee that such n has primitive roots, and even if it does, the period of the powers of such primitive roots does not correspond with the number of columns, $\phi(n)$ in the original design. Nonetheless, a similar trick is applied using primitive root in the selection of columns under this circumstance. It is found that the consequent maximin LHDs perform well regarding L_1 distance, and their performance improves with larger p and n; moreover, since the Cauchy-Schwarz inequality claims that L_1 distance provides a lower bound for L_2 distance, these LHDs also have a good performance regarding L_2 distance [Wang, Xiao and Xu (2018)].

This paper is organized as follows; in Chapter 2 the selection algorithm using primitive

roots is elaborated; Chapter 3 presents some results from applying the method onto the $p \times (p-1)$ and $n \times \phi(n)$ maximin LHDs, with some supporting plots (also, some complementary tables can be found in the Appendix); next, patterns from the results, shown in formulas, tables and plots, and justifications for some part of the algorithm are provided in Chapter 4. Last but not least, Chapter 5 gives some further insights and improvements on the aforementioned algorithms and a general conclusion.

CHAPTER 2

Construction Algorithm

A Latin hypercube design (LHD) is defined to have each of its columns as a permutation of a set with equally spaced elements. We start with building a $p \times (p-1)$, or $N \times \phi(N)$ in general (where N includes all primes, prime multiples and prime powers), LHD, denoted as D, formed according to the formula $Mh(mod\, N)$, where $M = \begin{bmatrix} 1 & 2 & \dots & N-1 & 0 \end{bmatrix}^T$, and h represents a row vector of dimension $1 \times \phi(N)$ containing all the positive integers less than and coprime to N [Zhou and Xu (2015)]. As a result, each column in D is a permutation of all the elements in $\mathbb{F}_N : \{0, 1, \dots, N-1\}$, with 0 as the last entry. Then, following from Wang, Xiao and Xu (2018)'s work, we also introduce a linear shift b, where $b \in \mathbb{F}_N$ is added to every entry of the current design to get $D_b = D + b(mod\, N)$, and Williams transformation to improve the design's performance regarding the maximin L_1 distance criterion. The Williams transformation is defined in Wang, Xiao and Xu (2018) as follows:

$$W(x) = \begin{cases} 2x & 0 \leqslant x \leqslant \frac{N}{2} \\ 2(N-x)-1 & otherwise \end{cases}$$

Thus, Williams transformation serves as a re-permutation of the columns in D_b, while increasing the maximum of the minimum pairwise distances between the rows of D_b.

Now we introduce the notion of primitive roots to perform column selection on this existing $N \times \phi(N)$ LHD. We say g is a primitive root of N if every positive integer less than and coprime to N is congruent to a power of g modulo N. In plain words, if we raise g to the powers $\{1, 2, \dots, N-1\}$ and examine the remainder of this number divided by N, we will get a permutation of all the positive integers less than and coprime to N, with or without some repeated values, depending on whether N is a prime. Such primitive roots exist for 2,4,p^a and $2p^a$, where p is a prime greater than or equal to 3 and $a \geqslant 1$. Thus, when $p \times (p-1)/2$

designs are considered, we propose a way of column selection using the properties of the primitive roots of such p; that is, we randomly pick a primitive root of p and choose the columns indicated by $g^k(mod\, p)$, where $k \in \{1, 3, \ldots, p-2\}$ or $\{2, 4, \ldots, p-1\}$. In fact, for an arbitrarily chosen p, every primitive root, when raised by the same set of powers, gives the same set of columns permuted in a different order; this result will be proved in Chapter 4; moreover, the two designs, resulted by an odd set and an even set of k, also perform equally well for given p and b. For simplicity, this paper will only concern the even exponents when selecting the columns. To illustrate our procedure, consider $p = 11$. We start from the following 11×10 design matrix. Notice that, except for the last row, each row contains a permutation of elements in $\mathbb{F}_{11}$ excluding 0, and each column contains a permutation of all elements in $\mathbb{F}_{11}$.

$$\left[\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
2 & 4 & 6 & 8 & 10 & 1 & 3 & 5 & 7 & 9 \\
3 & 6 & 9 & 1 & 4 & 7 & 10 & 2 & 5 & 8 \\
4 & 8 & 1 & 5 & 9 & 2 & 6 & 10 & 3 & 7 \\
5 & 10 & 4 & 9 & 3 & 8 & 2 & 7 & 1 & 6 \\
6 & 1 & 7 & 2 & 8 & 3 & 9 & 4 & 10 & 5 \\
7 & 3 & 10 & 6 & 2 & 9 & 5 & 1 & 8 & 4 \\
8 & 5 & 2 & 10 & 7 & 4 & 1 & 9 & 6 & 3 \\
9 & 7 & 5 & 3 & 1 & 10 & 8 & 6 & 4 & 2 \\
10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right]$$

$\mathbb{F}_{11}$ has primitive roots 2,6,7 and 8, and we will have the set {1,3,4,5,9} if we raise any of these primitive roots to the even powers $\{2, 4, 6, 8, 10\}$. The matrix given on the left shows the selected columns; the middle one is obtained by adding $b = 8$ (the most optimal b for $p = 11$) to the matrix on the left and getting values in $\mathbb{F}_{11}$, and the one on the right is Williams transformed after we add in the linear translation. As a result, the minimal row-wise distance occurs between the last row and each of the other rows, and it is equal to 15 (Take the first and the last row, for example: $d = |3-5|+|0-5|+|2-5|+|4-5|+|9-5| = 2+5+3+1+4 = 15$).

Table 6.1 in the Appendix is going to show that this distance translates to a relative efficiency of 75% using the upper bound for the minimal distances given by Lemma 1 in Wang, Xiao and Xu (2018). Even though a 75% may not be look convincing, we are going to show in the table that as p increases, the corresponding efficiency will also get boosted.

$$\begin{bmatrix} 1 & 3 & 4 & 5 & 9 \\ 2 & 6 & 8 & 10 & 7 \\ 3 & 9 & 1 & 4 & 5 \\ 4 & 1 & 5 & 9 & 3 \\ 5 & 4 & 9 & 3 & 1 \\ 6 & 7 & 2 & 8 & 10 \\ 7 & 10 & 6 & 2 & 8 \\ 8 & 2 & 10 & 7 & 6 \\ 9 & 5 & 3 & 1 & 4 \\ 10 & 8 & 7 & 6 & 2 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow{\text{b=8}} \begin{bmatrix} 9 & 0 & 1 & 2 & 6 \\ 10 & 3 & 5 & 7 & 4 \\ 0 & 6 & 9 & 1 & 2 \\ 1 & 9 & 2 & 6 & 0 \\ 2 & 1 & 6 & 0 & 9 \\ 3 & 4 & 10 & 5 & 7 \\ 4 & 7 & 3 & 10 & 5 \\ 5 & 10 & 7 & 4 & 3 \\ 6 & 2 & 0 & 9 & 1 \\ 7 & 5 & 4 & 3 & 10 \\ 8 & 8 & 8 & 8 & 8 \end{bmatrix} \xrightarrow{\text{Williams}} \begin{bmatrix} 3 & 0 & 2 & 4 & 9 \\ 1 & 6 & 10 & 7 & 8 \\ 0 & 9 & 3 & 2 & 4 \\ 2 & 3 & 4 & 9 & 0 \\ 4 & 2 & 9 & 0 & 3 \\ 6 & 8 & 1 & 10 & 7 \\ 8 & 7 & 6 & 1 & 10 \\ 10 & 1 & 8 & 7 & 6 \\ 9 & 4 & 0 & 3 & 2 \\ 7 & 10 & 8 & 6 & 1 \\ 5 & 5 & 5 & 5 & 5 \end{bmatrix}$$

In addition, regarding the construction of the $p \times (p-1)$ LHDs and the column selection, other techniques including the SLHD package proposed by Ba, Myers and Brenneman (2015) and the inverse operation to our "exponent-of-primitive-roots" – taking log over the Galois field (Gilbert method), were also suggested [Xiao and Xu (2017)]. Specifically, the results from SLHD were obtained from running "maximinSLHD" function in R for 10 times for each prime value with the default settings in the function, and the median and maximum efficiencies for each prime were shown in Figure 3.2; on the other hand, we applied the Gilbert method as described in Xiao and Xu (2017). First, we constructed the $p \times (p-1)$ LHD using the aforementioned $D = Mh \,(mod\, p)$ formula; then, for each c in $\{1, ..., p-1\}$, we iterated through the positive even i values in $\mathbb{F}_p$ ($i \in \{2, 4, ...p-1\}$) so that we had $\frac{p-1}{2}$ a_i values serving as the column indices, where $a_i = log_\beta(i) + 1 - c \,(mod\, p-1)$ and β is a random primitive root mod p (We picked four β values for each prime, since this time β values have some effects on the results, unlike in our algorithm). Note that our $a_i's$ are exactly the same as the $b_i's$ defined in their Definition 4 (we are renaming it to differentiate from the linear

shift b). This logarithmic function is the inverse operation of raising primitive roots to the even exponents we mentioned before; in other words, $log_\beta(i)$ returns the power which we have to raise for the primitive root β in order to get the output i in $\mathbb{F}_p$. We repeated this process for each c in the given set; also, we iterated through all the possible linear shift $b's$ afterwards and applied the Williams transformation to get the final design matrix. Because, based on the descriptions of these two algorithms, they are more computationally expensive than our algorithm (for example, the Gilbert method has a computation time on the order of $O(4p^2-4)$ while ours is only on the order of $O(p)$), Figure 3.2 only shows the comparison of the three algorithms for primes up to 97. Yet, even based on the limited results, we can conclude that SLHD and Gilbert methods have inferior performances regarding the maximin L_1 distance criterion.

Speaking of the more general $n's$ (prime powers and prime multiples), the construction of an $n \times \frac{\phi(n)}{2}$ will be more complicated as they generally do not have primitive roots. However, depending on the prime factorizations of the numbers and the values of $\phi(n)$, we will examine the results of a selection of five methods applied onto the numbers:

- Method (i): Proposed by Zhou and Xu (2015); take the set of positive integers less than and coprime to n, and just use the first half of this set to generate the resulting design. Look for the best linear translation b and then apply Williams transformation to form the final design matrix.

  –For example, take $n = 21$; this method suggests using its first 6 coprimes out of the 12: $\{\mathbf{1}, \mathbf{2}, \mathbf{4}, \mathbf{5}, \mathbf{8}, \mathbf{10}, 11, 13, 16, 17, 19, 20\}$. Define the set consisting of only the bolded values to be h, and a column vector $\begin{bmatrix} 0 & 1 & \dots & 20 \end{bmatrix}^T$ to be M, then we can use $Mh \,(mod\, 21)$ to form a design with the proper dimensions, and then apply the linear shift and Williams transformation to it.

- Method (ii): In case $\phi(n) + 1$ is a prime, we take a random primitive root of this number, and raise it to the even exponents in $\mathbb{F}_{\phi(n)+1}$ to get the column indices to form an $n \times \frac{\phi(n)}{2}$ matrix. Look for the best linear transformation b and apply Williams transformation to form the final design matrix.

– For example, take $n = 21$; $\phi(21) = 12$, and so we are going to take the even exponents of a primitive root mod 13 as our column indices ($\{1, 3, 4, 9, 10, 12\}$). We next apply linear shift and William transformation to the matrix.

- Method (iii): In case $\frac{\phi(n)}{2}+1$ is a prime, we take a random primitive root of this number and raise it to the even exponents in $\mathbb{F}_{\frac{\phi(n)}{2}+1}$. Now, we have half of the column indices; then, regarding the other half, we subtract these column indices from $\phi(n)+1$, and take the resulting differences as the other half of the column indices to construct the design of dimension $n \times \frac{\phi(n)}{2}$. Look for the best linear transformation b and apply Williams transformation to form the final design matrix (This is motivated by the pattern of the even exponents given by a random primitive root for primes equal to $4k + 1$ for some integer k).

  – For example, $\frac{\phi(21)}{2} = 6$; 7 is a prime, so we can obtain one half of the column indices using a primitive root mod 7 ($\{1, 2, 4\}$); as a result, the other half is going to be given by the set $\{9, 11, 12\}$. We next apply linear shift and Williams transformation to the matrix.

- Method (iv): There can be cases where methods (ii) and (iii) fail to apply due to their constraints and method (i) cannot give good results. In that case, we propose this method (iv), or the generalized version of method (ii); that is, we instead look for the next prime after $\phi(n)$, take a primitive root of this number, and raise it to the even exponents in $\mathbb{F}_{\phi(n)+k}$. We remove the elements greater than $\phi(n)$ and take the remaining set as the column indices. Look for the best linear transformation b and apply Williams transformation to form the final design matrix (There is no guarantee that under some cases this method will produce the exact $n \times \frac{\phi(n)}{2}$ design; sometimes we will end up having 1-3 more columns than required using this method).

- Method (v): This is a generalized version of method (iii); we take the next prime after $\frac{\phi(n)}{2}$, do the same thing and get a portion of the column indices. Likewise, we obtain the other column indices via subtraction from $\phi(n) + 1$; notice that now we might end up getting repeated values in our column indices due to the fact that the first half

of the indices are not coming from $\mathbb{F}_{\frac{\phi(n)}{2}+1}$ anymore. So, we are going to remove the duplicated indices in the set and use the unique values to construct our design matrix, and then look for the best linear transformation b and apply Williams transformation to form the final design matrix (There is no guarantee that under some cases this method will produce the exact $n \times \frac{\phi(n)}{2}$ design; sometimes we will end up having around 5 more columns than required using this method; because of the relatively large discrepancy in the dimensions produced and the dimensions required, this method will not be touched until Chapter 5 when we talk about possible improvements).

CHAPTER 3

Implementation and Results

3.1 Primes $p \times (p-1)/2$

According to Zhou and Xu (2015), in a general $N \times \phi(N)$ LHD for some positive integer N, the maximum of the minimum pairwise L_1 distance between rows is bounded by $\left\lfloor \frac{(N+1)\phi(N)}{3} \right\rfloor$. Also, they proposed to search for the best b that maximizes such distance. It is believed that a good way of constructing these maximin LHDs should not have the efficiency of the resulting design D, or the ratio between its maximin L_1 distance and the upper bound, fall below 75% under the optimal b values. Thus, we have implemented the aforementioned construction and column selection algorithms and examined the optimal $b's$ and the corresponding efficiency. Table 6.1 shows the maximin L_1 distances and the optimal $b's$ for a selection of $p \times (p-1)/2$ LHDs, up to $p = 499$.

As shown by the table, the design has always attained an efficiency of at least 75%, and the value improves from 75% when $p = 11$ to around 95% when p is close to 500. Also, except for few $p's$, the efficiencies are constantly increasing when p gets larger, and Figure 3.1, illustrating the trend between prime values and efficiencies, confirms this. The performances of the aforementioned and previously described construction methods (SLHD [Ba, Myers and Brenneman (2015)] and Gilbert [Xiao and Xu (2017)]) for small p up to 100, due to computational expensiveness, are shown in Figure 3.2, together with that of our algorithm for the same $p's$. As shown by the plot, our algorithm has outperformed the median values from the 10 runs of SLHD and the Gilbert methods for all p from 11 to 97, and the maximum values from SLHD runs except for the first few $p's$ under 30.

Moreover, the optimal $b's$ producing the largest distances shown in the table are worth

noticing. For all p, there are exactly 2 or 4 optimal $b's$ except for the case where $p = 11$. In fact, a close examination of those optimal $b's$ reveal that they come in pairs. When we compare the $b's$ to $(p-1)/2$, we can see that for the primes having 2 optimal $b's$, the $b's$ are either both less than $(p-1)/2$ or both greater than that value. As we move toward larger $p's$, the magnitude of the pairs less than $(p-1)/2$ also moves up; same thing happens for the pairs greater than $(p-1)/2$. Additionally, when $b's$ are both less than $(p-1)/2$, their sum always equals $(p-1)/2$; on the other hand, when they are both greater than $(p-1)/2$, the difference from $(p-1)/2$ to the smaller b always equals the difference from the larger b to p. These patterns can be more clearly seen if we look at the values of $W(b)$, obtained via applying Williams transformation on the optimal $b's$: for p with 2 optimal $b's$, the Williams transformed $b's$ add up to $p-1$ (for $p = 11$, $W(b) + W(b)$ also gives $p-1$; for the $p's$ with 4 optimal $b's$, we can obtain two pairs of $W(b)$ that add up to $p-1$). Thus, we can say that in these maximin LHDs, the $b's$ are "symmetric" in some sense. Last but not least, through examination of the next few optimal $b's$ could we find out a well-established linear relationship between b and p, and we will leave that discussion to Chapter 4 for now.

Figure 3.1: Efficiency Trend for Primes

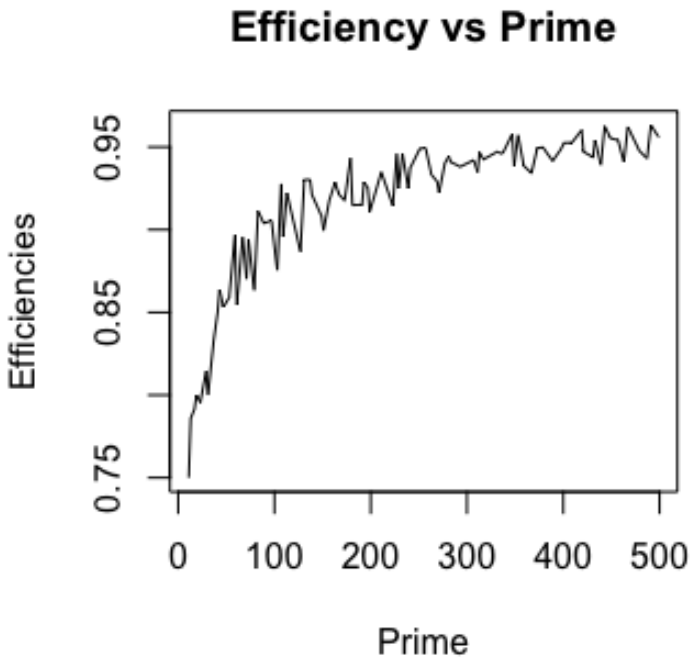
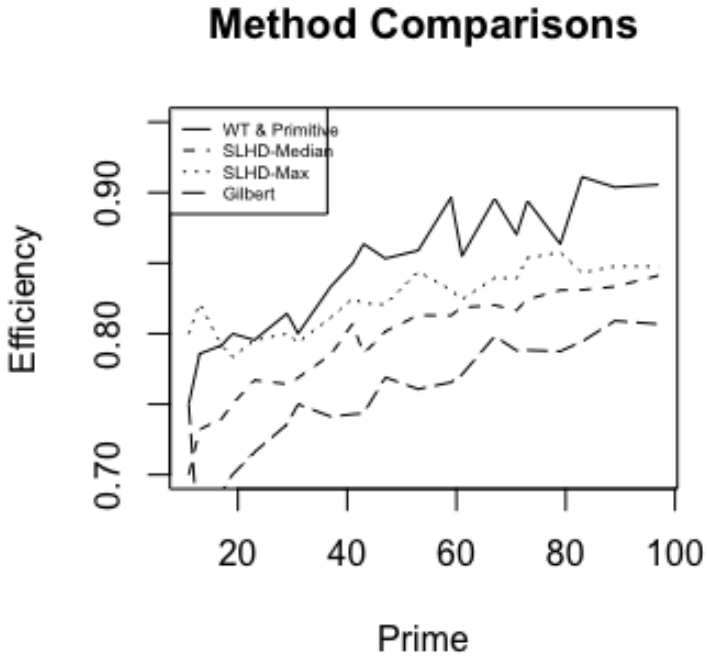


Figure 3.2: Our Construction and Selection Algorithm vs SLHD Method and Gilbert Method



3.2 Prime Multiples $n \times \phi(n)/2$

Moving away from primes and going towards prime multiples, the former algorithm for column selection might not work, as these numbers are not guaranteed to have primitive roots; moreover, even if they do, the exponents of the roots will not be able to span the set of positive integers from 1 to the number of columns in such LHDs. Zhou and Xu (2015) mentioned generating the design only using the first half of the h vector defined in Chapter 2 (method (i)). However, our experiment suggests that this method of construction does not perform well for large n and especially even n.

To fix this problem, a similar logic may be applied here. If $\phi(n) + 1$ is a prime number, then we can simply pick one of its primitive roots and do the same thing – raise to the even powers and accordingly select the columns (method (ii)). Also, motivated by a pattern in the even exponents of the primitive roots for a prime p in the form $4k + 1$ in $\mathbb{F}_p$, where k is an integer (it consists of $\frac{p-1}{4}$ pairs, each adding up to p; for example, the even orders of each primitive root of 17 give 9,13,15,16,8,4,2,1), we also separate the case where $\phi(n)/2 + 1$ is a prime; in these cases, we look for a primitive root of such a prime p, take its even exponents in $\mathbb{F}_p$, and get half of the columns. Regarding the other half, we subtract the chosen column indices from $\phi(n) + 1$; the columns with indices equal to the differences are also going to be chosen (method (iii)). Further experiments show that this way of column selection not only works well for $n = 4k + 1$, but rather some other $n's$ as well. Admittedly, these two algorithms both have disadvantages as they are unable to deal with the cases where neither of $\phi(n) + 1$ and $\phi(n)/2 + 1$ are primes. In that case, we propose to simply look for the next prime after $\phi(n)$, take the even powers of its primitive roots and discard the ones that exceed $\phi(n)$, which refer to some nonexistent columns (method (iv)). For the cases that the former two algorithms do not apply, this method also works better than just taking the first half of the generator vector h to construct the design. Table 6.2 summarizes the best L_1 distance from the four methods and the corresponding $b's$ for prime multiples up to 500; first of all, the efficiencies reveal that these methods work better for odd numbers than for even ones; this is also corroborated by Figure 3.3, as the black line lies below the other three for most

of the times. A possible explanation might be due to the smaller value of $\phi(n)$ when n is even; a factor of 2 excludes half of the preceding positive integers from the list of coprimes, thus resulting in a $\phi(n)/2$ that is around $n/4$. In contrast, the efficiencies for kp when $k \geqslant 7$ seem to be higher than the other cases on average.

Speaking of the comparison of the methods, we can see that while method (i) shows some optimality for a few prime multiples under 100, the other methods start to dominate once n exceeds 100. To illustrate this, Figures 3.4 and 3.5 are provided; we have combined methods (ii) and (iv) in the plots since (ii) is simply a special case of (iv). The plots on the top row give the overall comparisons between (i) and (ii)/(iv) and between (i) and (iii). It is demonstrated that, overall speaking, method (ii) starts to produce higher efficiencies when $n \geqslant 50$ (the numerous troughs in the graph imply inferior efficiencies given by both methods for even numbers), whereas method (iii) improves slower and only outcompetes (i) when n is larger than 100 (the scarcity of data for method (iii) is due to the fact that it is rather inapplicable to most $n's$). On the second row, some more specific comparisons are shown. Note that we only give a selection of the comparisons because method (ii) does not apply to the cases where $n = 3p$. For these n, $\phi(n) = 2(p-1)$, and we don't usually have that $2p-1$ is a prime; on the other hand, when considering $n = 2p$ for example, $\phi(n) = p-1$, and thus under this circumstance we are always able to create designs with exactly equal dimensions as the ones coming from method (i) to perform efficiency comparisons. Similarly, method (iii) is good for the cases where $n = 3p$ but not the other choices of n, which justifies the lack of data for most values of n shown on the top right grid. According to these comparisons, it can perhaps be concluded that, speaking of maximin L_1 distances, method (i) is only good for $n = 3p$ under 100, as the solid lines are well below the dashed ones in all other situations.

Figure 3.3: Efficiency Trends for Prime Multiples

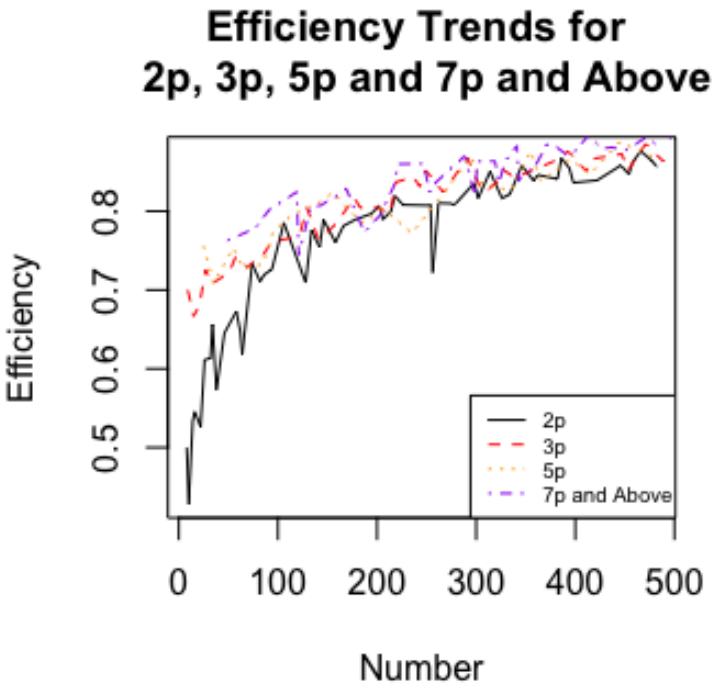


Figure 3.4: Efficiency Comparisons Between Methods (i) and (ii)/(iv)

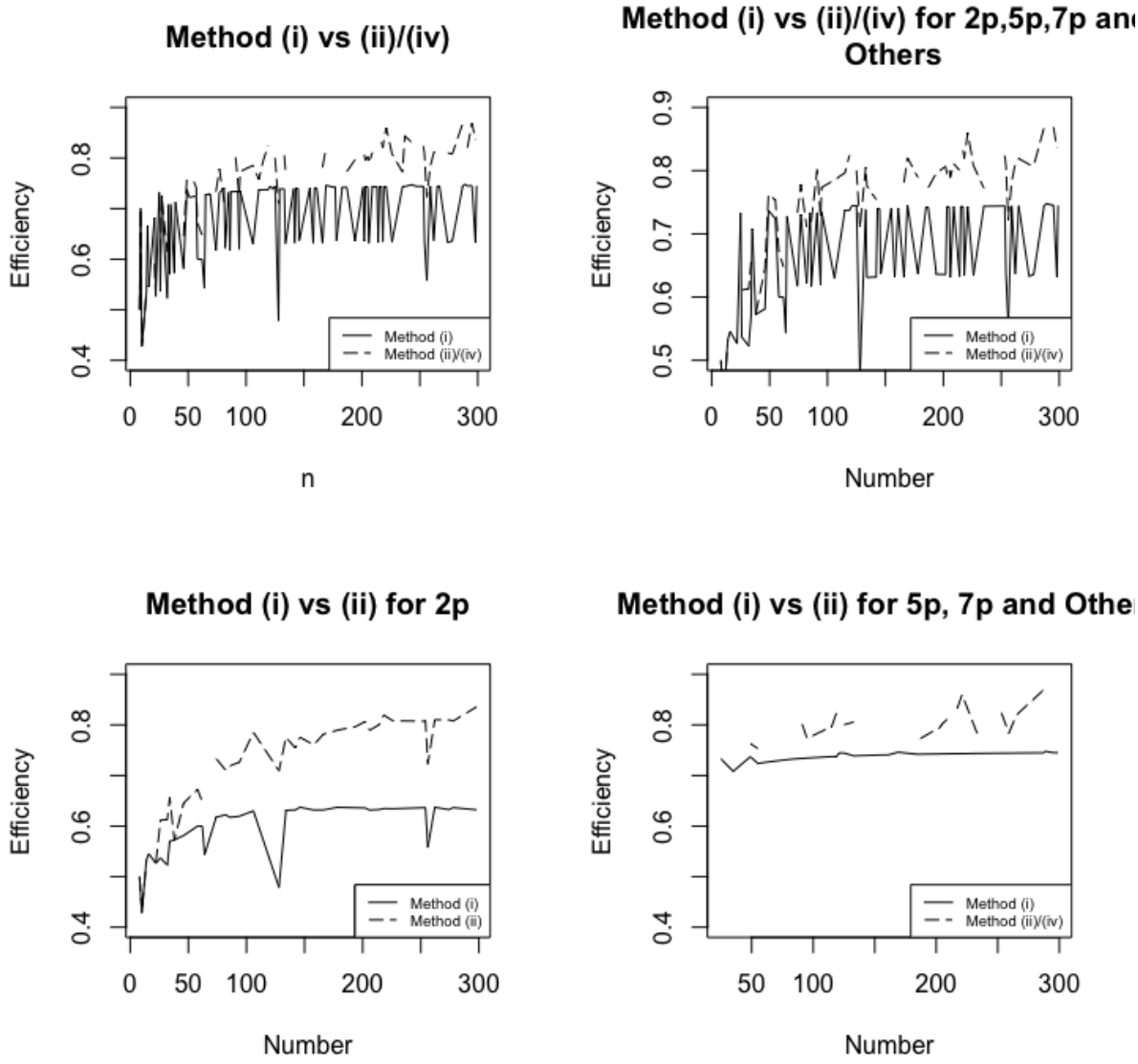
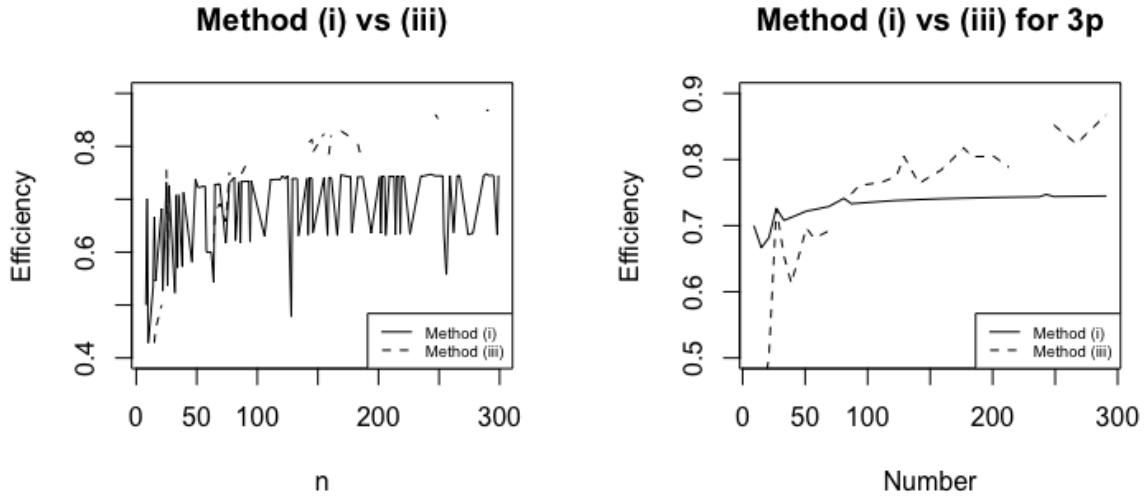


Figure 3.5: Efficiency Comparisons Between Methods (i) and (iii)

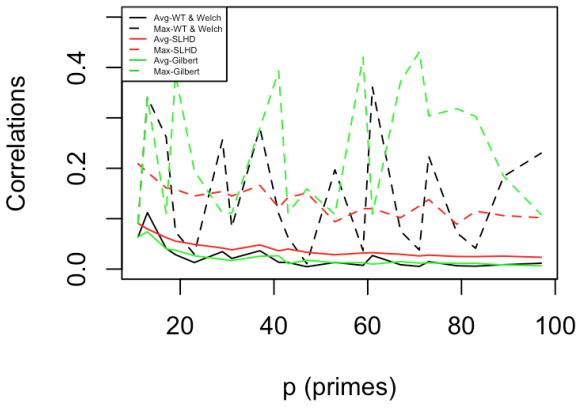


3.3 Correlations

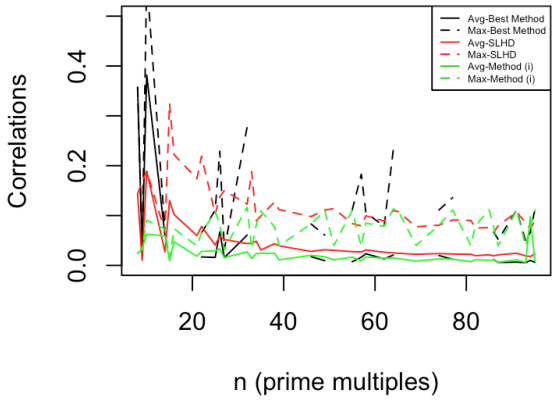
Next, we calculated the average and maximum pairwise correlations among the columns of the designs mentioned in the last two sections; moreover, we compared the correlations to the ones using Gilbert method and also the corresponding average results from running Ba, Myers and Brenneman's algorithm "maximinSLHD" ten times (only for $n < 100$ due to computational expensiveness; the averages and maxima were reported for the run yielding the largest L_1 distance among the ten only). We only considered the magnitudes of the correlations and thus took the absolute values of the correlations before computing the averages and maxima; the corresponding results are shown in Table 6.3 and 6.4, respectively. For the parameters (b and/or c), we used the optimal ones giving the largest L_1 distances; for most of the times, the simultaneously optimal parameters yield identical correlation matrices for a given p or n, except for $n = 10$, where the optimal $b = 1, 3, 4, 6, 8, 9$ under both methods (i) and (ii) give different correlations. Especially for method (ii), $b = 1, 6$ produced a design where the two columns are almost uncorrelated at 0.00606, whereas $b = 3, 4, 8, 9$ resulted in correlations as high as close to 0.6.

Overall speaking, the average correlations tend to 0 for all the algorithms shown in Figure 3.6, whether we are concerning primes or prime multiples. Specifically, when dealing with $p's$, our method (linear permutation + Williams transformation + Welch-Costas Array [Xiao and Xu (2017)]) outperforms the other ones considering the average correlations; however, the maximum correlations for both our and Gilbert methods fluctuate a lot from 0.1 to 0.4, whereas the maximum correlations for SLHD designs constantly decrease, though very slowly, from 0.2. In fact, a close examination of the maximum correlations reveals that our method did not control the values for $p = 4k + 1$ very well, whereas for $p = 4k + 3$, most of the maximum correlations under our algorithm can go below 0.1. On the other hand, when working with prime multiples $n's$, even though our methods (ii) and (iii) described in Section 3.2 could produce comparable average correlations as method (i) for a few $n's$ below 100, where either of them is preferred over method (i), they were not very optimal concerning the maximum correlations, as they were the only ones giving max correlations above 0.2 among the methods considered under most circumstances. Thus, our methods for column selections were not as optimal for prime multiples as for primes, speaking of the maximum correlations (in fact, for most of the prime multiples under 100, our distance efficiencies were outperformed by designs of equivalent size generated using SLHD).

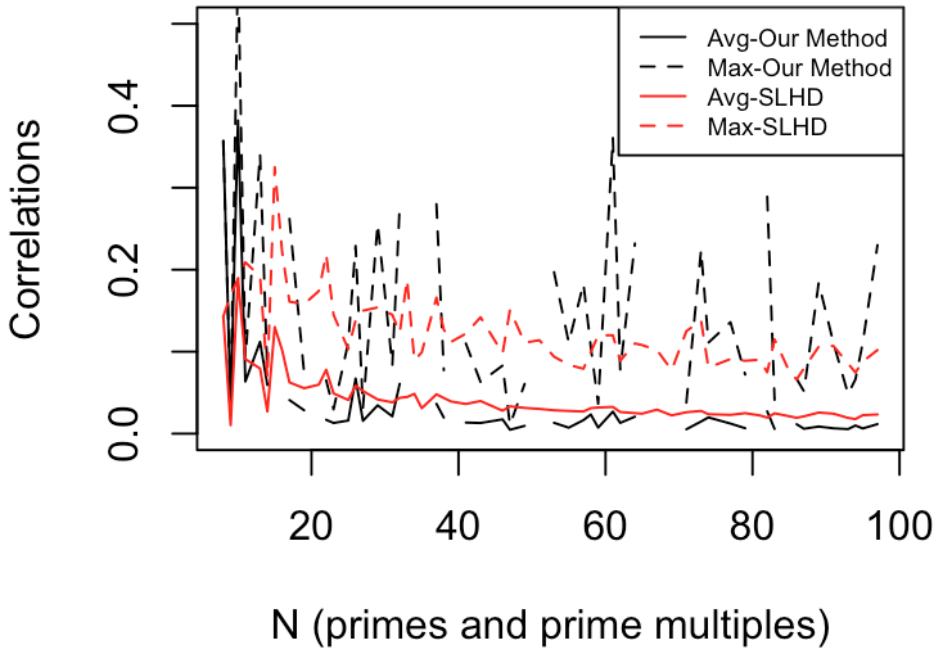
Figure 3.6: Correlation Comparisons



(a) p=primes



(b) n=prime multiples



(c) N=combined primes and prime multiples

CHAPTER 4

Patterns and Proofs

4.1 Patterns

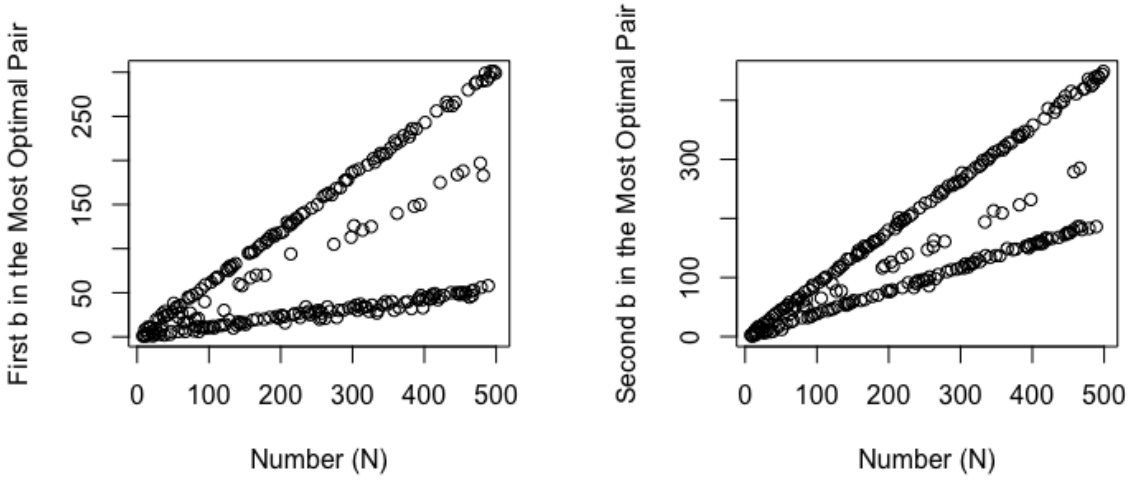
We concluded in Section 3.1 a relationship between $W(b)$ and $(p-1)/2$ and p; however, this relationship is built upon using one b to find another. To find a direct relationship connecting the optimal $b's$ with p or n, we had to consider not only the first optimal, but also the first up to three optimal pairs of $b's$ for each n; the most optimal pair for a random n can be both less than $n/2$ or both greater than the value. However, among the three optimal pairs, we could usually see both scenarios. Once we separate the two cases from one another, we found that the patterns between n and b could be nicely modeled using linear regressions.

To illustrate this, we have fitted regression models of the most optimal b pair on the number N (including all the primes and prime multiples). For the smaller b (b_1) in the pair, the model is $\mathbf{b_1 = 2.156 + 0.330N}$, predicting b_1 is approximately $\frac{1}{3}$ of N, with an R^2 of 0.35 only; on the other hand, the larger b's model, $\mathbf{b_2 = -0.673 + 0.660N}$, predicting b_2 is close to $\frac{2N}{3}$, yields an R^2 close to 0.67. The graphs of the most optimal b pair vs N also show that even though apparently there are trends in the plot, they cannot be properly captured by linear models.

Table 4.1 shows the three most optimal b pairs for the last few primes and prime multiples we have tried, as for these larger numbers the $b's$ are much better formatted such that each pair contains exactly two b values than for smaller numbers and thus it is easier to convey our idea of linear model constructions to the readers. We constructed four different linear regression models, one for each b at "distinct locations" in $\mathbb{F}_N$ (N stands for the collection of all the primes and prime multiples under 500); that is, for example, for $p = 499$, we took

the first four $b's$, 299,449,50 and 199 as four response values, and modeled each against 499; similarly, for $n = 497$, $b = 301, 444, 52, 196$ are the ones we used. What's more, we collected the $b's$ in similar regions in each $\mathbb{F}_N$: for example, 50 for 499, 52 for 497, 55 for 493, 50 for 491 and so on are closer to 0 than to $N/2$ and N, so they are in the same regression model; likewise, we grouped 199 for 499, 196 for 497, 191 for 493 and 195 for 491 in the second regression model as they are closer to and smaller than $n/2$ in $\mathbb{F}_N$. Similarly, we built two more regression models on the $b's$ on the other end of $\mathbb{F}_N$. Consequently, these models account for the trends in b pairs much better than the previous models built on raw data.

Figure 4.1: Most Optimal b Pair vs N (Collection of Primes and Prime Multiples)



(a) $b = b_1$; smaller b (b) $b = b_2$; larger b

Table 4.1: Three Most Optimal b Pairs

Number	b
499	299,449; 50,199; 300,448
497	301,444; 52,196; 53,195
493	301,438; 55,191; 54,192
491	295,441; 50,195; 296,440
489	58,186; 302,431; 57,187
487	291,439; 48,195; 292,438
485	299,428; 56,186; 57,185
482	183,424; 57,298; 58,299

481	290,431; 49,191; 50,190
479	56,183; 295,423; 55,184
⋮	⋮

The resulting models each has R^2 above 95%, implying a good fit. The formulas for the four regressions are shown below and the corresponding plots for each fit are also provided:

$$\mathbf{b_1 = 0.446 + 0.104N}$$

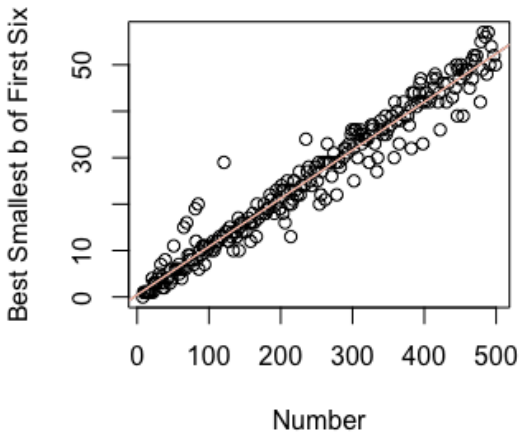
$$\mathbf{b_2 = -1.867 + 0.397N}$$

$$\mathbf{b_3 = 0.415 + 0.604N}$$

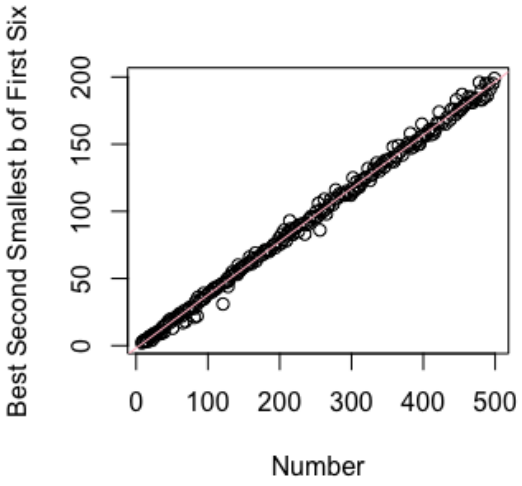
$$\mathbf{b_4 = -1.297 + 0.897N}$$

The plots in Figure 4.2 corroborated the good fit of the four linear models; even the plot for the smallest b, having the most number of deviated points, showed a strong linear trend well captured by our regression line shown in pink. From the dots with the significant deviations in the topleft panel, we could see that unexpectedly large $b_1's$ were more likely to happen for smaller N values, approximately below 120, whereas the significant residuals, if any, were mostly negative once $N > 200$. A close examination of the data reveals that the most significant deviance from the regression line occurs at $n = 121$, where none of the first six optimal b values are close to 13.03 suggested by the b_1 regression model. Also worth noticing, 121 is the only prime power/multiple we have dealt with that none of the proposed methods (ii), (iii) and (iv) could apply to yield a design of $121 \times \frac{\phi(121)}{2} = 121 \times 55$ (methods (ii), with $p = 109$ selected, and (iv) can only give 121×54 designs, and method (iii) can only produce a 121×52 design using $p = 53$; this lack of information about one or more additional columns made method (i) the only applicable one in this case, even though the relative efficiencies of methods (ii) and (iv) beat that of (i)). Table 4.2 specifies the results of all methods applied to $n = 121$.

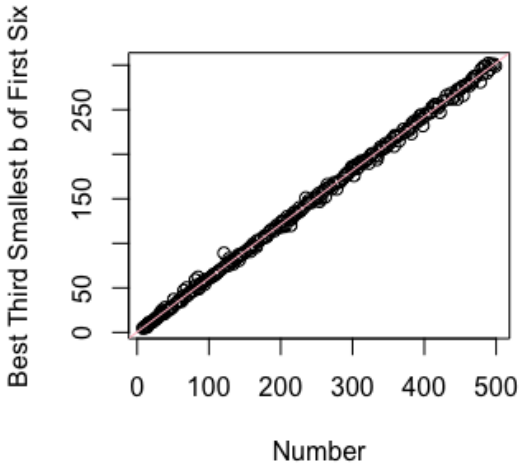
Figure 4.2: Four Most Optimal $b's$ vs N



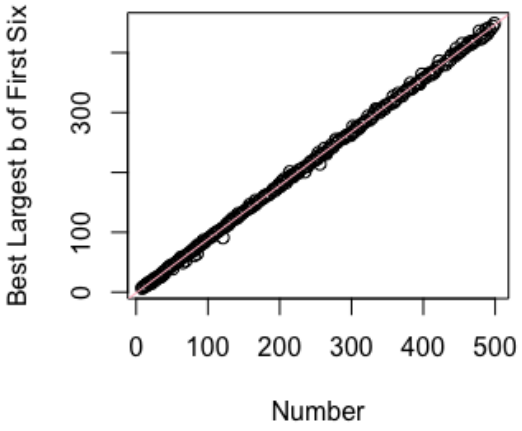
(a) $b_1 = 0.446 + 0.104N$; smallest b



(b) $b_2 = -1.867 + 0.397N$; second smallest b



(c) $b_3 = 0.415 + 0.604N$; third smallest b



(d) $b_4 = -1.297 + 0.897N$; largest b

Table 4.2: Examination of All Methods for $n = 121$

Number	Column	b	Distance	Efficiency	Method
121	55	30	1665	74.46%	(i)
121	55	90,91	1612	72.09%	
121	55	29,31	1562	69.86%	
121	55	89,92	1515	67.75%	
121	54	76,105	1683	76.64%	(ii)
121	54	15,45,75,106	1682	76.59%	
121	52	10,50	1535	72.61%	(iii)
121	52	70,111	1534	72.56%	
121	52	71,110	1508	71.33%	
121	54	18,42	1653	75.27%	(iv)
121	54	78,103	1649	75.09%	
121	54	17,43	1645	74.91%	

4.2 Proofs

Apart from the relationship between N and b, we have also found a relationship between the best L_1 distance and N and b, but before we dive into the model connecting these terms together, we would like to show and prove some important claims and propositions for the column selection algorithm involving primitive roots.

Proposition 1: Suppose g and h are two of the primitive roots mod p, where p is an arbitrary odd prime. Then, $g^i \,(mod\, p) \neq h$ for any i such that $gcd(i, p-1) \neq 1$ ("gcd" refers to their greatest common divisor).

Proof: Without loss of generality, let's suppose $g^2 = h \,(mod\, p)$ as a contradiction. Then, it follows that $g^4 = h^2 \,(mod\, p)$, $g^6 = h^3 \,(mod\, p)$ and so on. Now, if we examine $g^{p-1} \,(mod\, p)$, we will have the equality $g^{p-1} \,(mod\, p) = h^{\frac{p-1}{2}} \,(mod\, p)$, following from our supposition. By the properties of a primitive root, $g^{p-1} = 1 \,(mod\, p)$ for any primitive roots g mod p; then, this says that $h^{\frac{p-1}{2}} \,(mod\, p) = 1 \,(mod\, p)$ as well. However, this does not hold because primitive roots are always of order $p-1$, implying that $h^i \,(mod\, p)$ is going to span the entire set $\{1, 2, ...p-1\}$ for $i = 1, 2, ..., p-1$ if h is a primitive root mod p. Once

$h^{\frac{p-1}{2}} \, (mod \, p) = 1 \, (mod \, p)$ and $h^{p-1} \, (mod \, p) = 1 \, (mod \, p)$, this spanning property no longer holds and a contradiction occurs.

Then, following from Proposition 1, we can justify our statement in Chapter 2 saying that:

Proposition 2: For an arbitrary odd prime p, every primitive root mod p raised to the even powers $2, 4, ..., p-1$ gives the same set of numbers in $\mathbb{F}_p$ up to different permutations.

Proof: Based on Proposition 1, given two primitive roots g and h mod p, h can only be obtained via g by $h = g^k \, (mod \, p)$, where $gcd(k, p-1) = 1$. Thus, for each i in $\{2, 4, ..., p-1\}$, we can find a q and an even number $r : 0 \leqslant r < p$ such that $ki = q(p-1) + r$; also, when $r = 0$, we set q to be $q - 1$ such that r gets transformed to $p - 1$. It can be seen that r for each i in the set is going to be different from each other. Thus, we have that $g^{ki} \, (mod \, p) = g^{q(p-1)+r} \, (mod \, p)$, and, again since $g^{q(p-1)} = 1 \, (mod \, p)$ for a primitive root g mod p, $g^{ki} \, (mod \, p) = h^i \, (mod \, p) = g^r \, (modp)$, and this finishes the proof.

We are going to show that the even exponents of a primitive root mod p can actually be represented using the following terms in Proposition 3, but before we talk about Proposition 3, we would like to introduce a concept called **quadratic residue**.

Definition: Suppose m is a positive integer. If there exists an x such that $x^2 = a \, (mod \, m)$ for some a such that $gcd(a, m) = 1$, then a is called a **quadratic residue** modulo m; otherwise, a is called a **quadratic nonresidue** of mod m.

Proposition 3: Suppose p is an odd prime. Then, there are exactly $\frac{p-1}{2}$ quadratic residues and $\frac{p-1}{2}$ quadratic nonresidues. Moreover, each element in the quadratic residues is congruent to one and only one number in the following sequence mod p: $1^2, 2^2, ..., (\frac{p-1}{2})^2$.

Sketch of the Proof: We first note that the elements in $\mathbb{F}_p$ can be rewritten as $\{-\frac{p-1}{2}, -\frac{p-1}{2} + 1, ..., -1, 1, \frac{p-1}{2} - 1, \frac{p-1}{2}\}$. Thus, it is straightforward to see that, if a is a quadratic residue modulo p, then a has to be congruent to some of the numbers in the following sequence: $\{(-\frac{p-1}{2})^2, (-\frac{p-1}{2} + 1)^2, ..., (-1)^2, 1^2, ..., (\frac{p-1}{2} - 1)^2, (\frac{p-1}{2})^2\}$ mod p. Notice that this sequence

has $\{1^2, 2^2, ..., (\frac{p-1}{2})^2\}$ repeated twice, and since each number in $\{1, 2^2, ..., (\frac{p-1}{2})^2\}$ corresponds to an unique solution or "root" in the set $\{1, 2, ..., \frac{p-1}{2}\}$, $\{1^2, 2^2, ..., (\frac{p-1}{2})^2\}$ comprises all the $\frac{p-1}{2}$ quadratic residues in $\mathbb{F}_p$.

Proposition 3 can be generalized to our case to say that, for a primitive root g mod p, its even exponents are congruent to $\{1, 2^2, ..., (\frac{p-1}{2})^2\}$ in $\mathbb{F}_p$ because $g^{ki}\,(mod\,p)$ for an even i such that $0 < i \leqslant p-1$ can simply be rewritten as $(g^{\frac{ki}{2}})^2\,(mod\,p)$.

Following from Proposition 3, we can conclude that our eventual design matrix is:

$$\begin{bmatrix} W(1 \bigoplus b) & W(2^2 \bigoplus b) & \dots & W((\frac{p-1}{2})^2 \bigoplus b) \\ W(2 \bigoplus b) & W(2 \cdot 2^2 \bigoplus b) & \dots & W(2(\frac{p-1}{2})^2 \bigoplus b) \\ \vdots & \vdots & \vdots & \vdots \\ W((p-1) \bigoplus b) & W((p-1)2^2 \bigoplus b) & \dots & W((p-1)(\frac{p-1}{2})^2 \bigoplus b) \\ W(b) & W(b) & \dots & W(b) \end{bmatrix},$$

where $1 \bigoplus b$ symbolizes $(1+b)\,mod\,p$.

Proposition 4: An attainable lower bound of the maximin L_1 distance for such a design matrix with fixed p and varying b from $\{0, 1, ...p-1\}$ is $\frac{p-1}{2}$, and it is only attained when $b = 0$ or $b = \frac{p-1}{2}$.

Proof: If we examine the i^{th} column of this design matrix, $1 \leqslant i \leqslant \frac{p-1}{2}$, we are going to find that it consists of $p-1$ different numbers in $\mathbb{F}_p$. To prove for the distinctness of the elements inside a column, it suffices to prove that $\{i^2 \bigoplus b, 2i^2 \bigoplus b, ...(p-1)i^2 \bigoplus b, b\}$ does not contain repeated elements, as the Williams transformation is a one-to-one transformation from $\mathbb{F}_p$ to $\mathbb{F}_p$.

Suppose $x \cdot i^2 \bigoplus b = y \cdot i^2 \bigoplus b$ for some x and y satisfying $0 \leqslant x < y \leqslant p-1$. Then, it follows that $(x \cdot i^2)\,mod\,p = (y \cdot i^2)\,mod\,p$, or $(x-y)i^2 = 0\,mod\,p$. Note that i^2 does not contain a factor of p for all i assumed, implying that $gcd(i^2, p) = 1$. Thus, $(x-y)i^2 = 0\,mod\,p$ indicates that $x - y = 0\,mod\,p$. Then, we have reached a contradiction because $0 \leqslant x < y \leqslant p-1$ by assumption. Hence, in each column, each value is distinct from one another, and the L_1 distance for a random b is lower bounded by $\sum_{i=1}^{\frac{p-1}{2}} 1 = \frac{p-1}{2}$.

When $b = 0$, we can take the k^{th} row $[W(k) \quad W(2^2 \cdot k) \quad \dots \quad W((\frac{p-1}{2})^2 \cdot k)]$ and the

$(p-k)^{th}$ row $[W(p-k) \quad W((p-k)2^2) \quad \dots \quad W((p-k)(\frac{p-1}{2})^2)]$ of the design matrix, and, as a result, these two rows give the smallest L_1 pairwise distance: $\frac{p-1}{2}$ for all k between 1 and $p-1$. Denote a to be such that $a = i^2 \, (mod \, p)$, where i is the column index of a random column in the design matrix ($1 \leqslant i \leqslant \frac{p-1}{2}$).

Then, regarding the corresponding entries on the k^{th} and $(p-k)^{th}$ rows, we have that:

$$W(ki^2) = \begin{cases} 2(ka - qp) & ka \, (mod \, p) \leqslant \frac{p-1}{2} \\ 2(p - (ka - qp)) - 1 = (q+2)p - 2ka - 1 & ka \, (mod \, p) > \frac{p-1}{2} \end{cases}$$

, and

$$W((p-k)i^2) = W(p - ki^2)$$

, where

$$W(p - ki^2) = \begin{cases} 2(p - (p - (ka - qp)) - 1 = 2ka - 2qp - 1 & p - ka \, (mod \, p) > \frac{p-1}{2} \\ 2(p - (ka - qp)) = (q+2)p - 2ka & p - ka \, (mod \, p) \leqslant \frac{p-1}{2} \end{cases}$$

Here, q refers to a random scalar such that $ka - qp \in \mathbb{F}_p$. Notice that the first cases for $W(ki^2)$ and $W((p-k)i^2)$ correspond with each other perfectly; namely, $ka \, (mod \, p) \leqslant \frac{p-1}{2}$ if and only if $p - ka \, (mod \, p) > \frac{p-1}{2}$, and it follows for the second cases also. Thus, $W(ki^2)$ and $W((p-k)i^2)$ only differ by 1 regardless of the choice of i. In other words, $|W((p-k)i^2) - W(ki^2)| = 1$ for all i. Thus, when $b = 0$, $\sum_{i=1}^{\frac{p-1}{2}} |W(ki^2) - W((p-k)i^2)| = \frac{p-1}{2}$ is the minimum pairwise L_1 distance among the rows of the design matrix.

For $b = \frac{p-1}{2}$, the logic is the same. It is just that the values of the $W's$ will be switched between the two cases, because a centering factor of $\frac{p-1}{2}$ will map the elements originally on the first half of $\mathbb{F}_p$ to the second half, and vice versa.

$$W(ki^2 \bigoplus \frac{p-1}{2}) = \begin{cases} 2(p - (ka - qp)) - 1 = (q+2)p - 2ka - 1 & ka \, (mod \, p) \leqslant \frac{p-1}{2} \\ 2(ka - qp) & ka \, (mod \, p) > \frac{p-1}{2} \end{cases}$$

,and

$$W((p-k)i^2) = W(p - ki^2)$$

, where

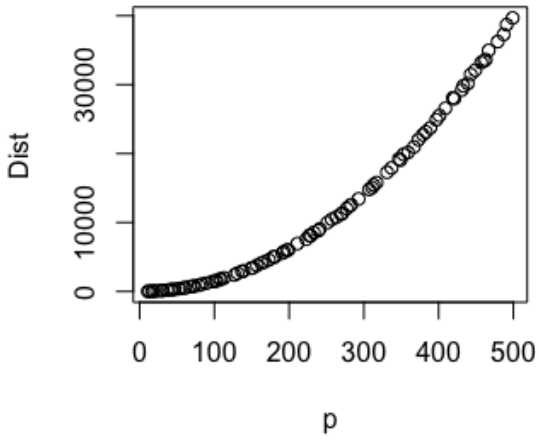
$$W(p-ki^2) = \begin{cases} 2(p-(ka-qp)) = (q+2)p - 2ka & p - ka\,(mod\,p) > \frac{p-1}{2} \\ 2(p-(p-(ka-qp)) - 1 = 2ka - 2qp - 1 & p - ka\,(mod\,p) \leqslant \frac{p-1}{2} \end{cases}$$

Thus, we have reached the same conclusion as in the case of $b = 0$.

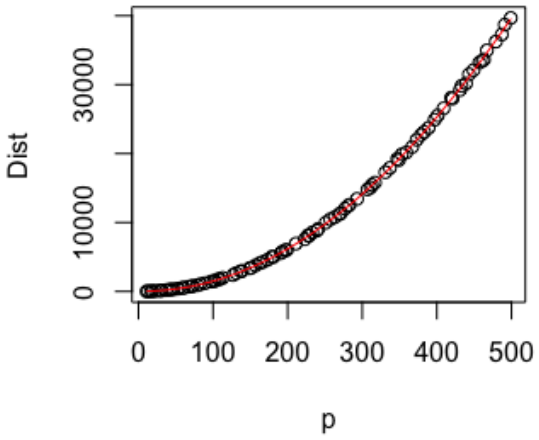
4.3 Continued Patterns

Further analysis on the optimal $b's$, $N's$ and also the distances also reveals strong connections among them, especially for primes p. The left panel of Figure 4.3 shows a strong parabolic relationship between the maximin L_1 distances and their corresponding p values. Consequently, we have fitted a regression of these maximin distances on $p's$ to the second power and also the Williams transformed optimal $b's$, and got a model with an R^2 as high as 0.9999. The resulting model is $\mathbf{distance = 36.051 - 1.619p + 0.162p^2 - 0.112W(b_1)}$ (We are only including $W(b_1)$ as $W(b_2)$ is highly correlated with $W(b_1)$ and thus including both will possibly cause a multicollinearity issue. The summary of this fit is shown on the right panel of Figure 4.3; even though the points seem to be perfectly following the line, the limits of the y-axis are on a scale of 0 to 40000. In fact, looking at the residuals of this model, we have found out that they range from -409 to 426. Though they seem decent given that the $p's$ can go as large as 500 and produce a maximin distance on the order of 10^4, problems will arise if some of the large residuals are actually coming from predicting the distances for small p values. Thus, to further evaluate the performance of this model, we have provided on the bottom panel of Figure 4.3 the ratio between the absolute values of the residuals and the actual distanes versus p (the three red reference lines are each representing a ratio of 0.1, 0.05 and 0.01, respectively). Indeed, the model has a poor prediction of the maximin distances for small $p's$, especially for the first five observations. Yet, we have also found that the formula $\frac{1}{2}(\frac{p^2-1}{3} - \left\lfloor \frac{(\frac{p-1}{2})^2-1}{3} \right\rfloor)$ is good for predicting the distances for at least the first five $p's$, as illustrated by Table 4.3.

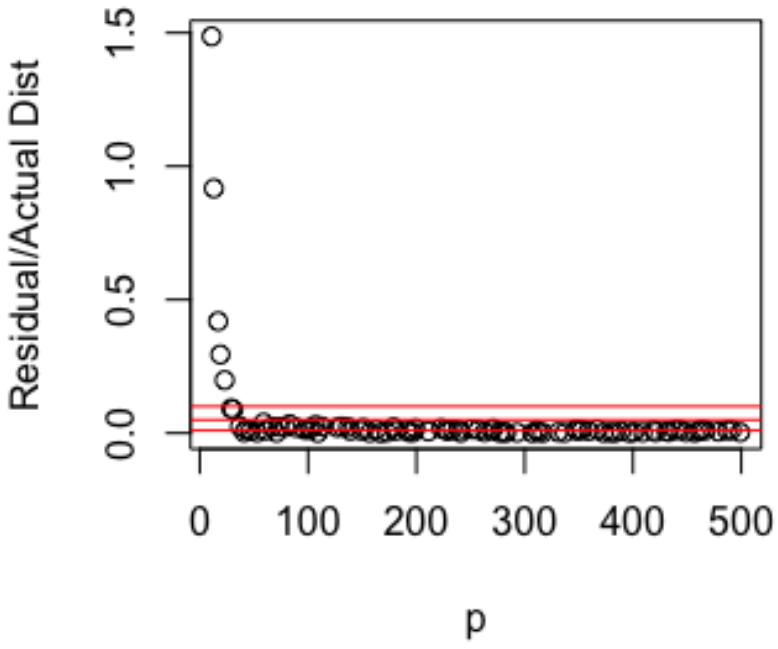
Figure 4.3: Pattern Between Distances and $p's$ (primes)



(a) Pattern



(b) Pattern with Fitted Curve



(c) Residual and Actual Value Ratio vs Prime

Table 4.3: Formulaic Distance Prediction for Small p Values

Prime	Predicted Distance	Actual Distance
11	16	15
13	22.5	22
17	37.5	38
19	47	48
23	68	70
29	107.5	114
31	123	128
37	174.5	190
41	213.5	238
43	235	266
47	280	314

As for prime multiples, unfortunately there is no model that captures the overall pattern between the maximin distances and $n's$ and $W(b)'s$. This is corroborated by the top panel in Figure 4.4, which summarizes the cases for all prime multiples under 500. It shows a clear trichotomous pattern toward the end of the plot and thus suggests that we perhaps should break these multiples into different cases before finding patterns. Indeed, the plots on the second and third row of Figure 4.4, presenting the cases of $n = 2p$, $3p$, $5p$ and $7p$ and above separately, suggest much stronger quadratic patterns without any branching ($n's$ were categorized by their smallest prime factors). Therefore, after we fit four different linear models to the cases, we find out that the R^2 are also very decent at 0.996 or even higher. The corresponding models are shown below (The term $W(b_2)$ is again omitted due to its high correlation with $W(b_1)$):

$$\text{distance} = \begin{cases} \mathbf{70.321 - 2.710n + 0.0769n^2 + 0.00255W(b_1)} & n = 2p \\ \mathbf{70.366 - 2.942n + 0.102n^2 + 0.262W(b_1)} & n = 3p \\ \mathbf{209.863 - 4.795n + 0.125n^2 - 0.596W(b_1)} & n = 5p \\ \mathbf{209.546 - 6.077n + 0.142n^2 + 0.464W(b_1)} & n = 7p, 11p, ... \end{cases}$$

The models for $n = 2p$, $3p$ and $5p$ produced residuals on the same level as the ones for primes, but some residuals for $7p$ and others were on the order of 1000+. The relative

inaccuracies of the model predictions in the four cases generally follow a trend that the models predicted poorly for very small n and thus small numbers of columns (for example, for $n = 8$ and 10, the number of columns is only 2, resulting in maximin distances equal to 3, while the model predicted the distances to be 20+), but the performances improved very fastly with increasing $n's$ so that the relative inaccuracies quickly dropped to around or below 0.05, which is indicated by the red lines in Figure 4.5. However, the plot for $7p$ and others again stood out, as the relative inaccuracies did not follow any pattern, but rather were randomly scattered around 0.04.

Figure 4.4: Pattern Between Distances and Prime Multiples n

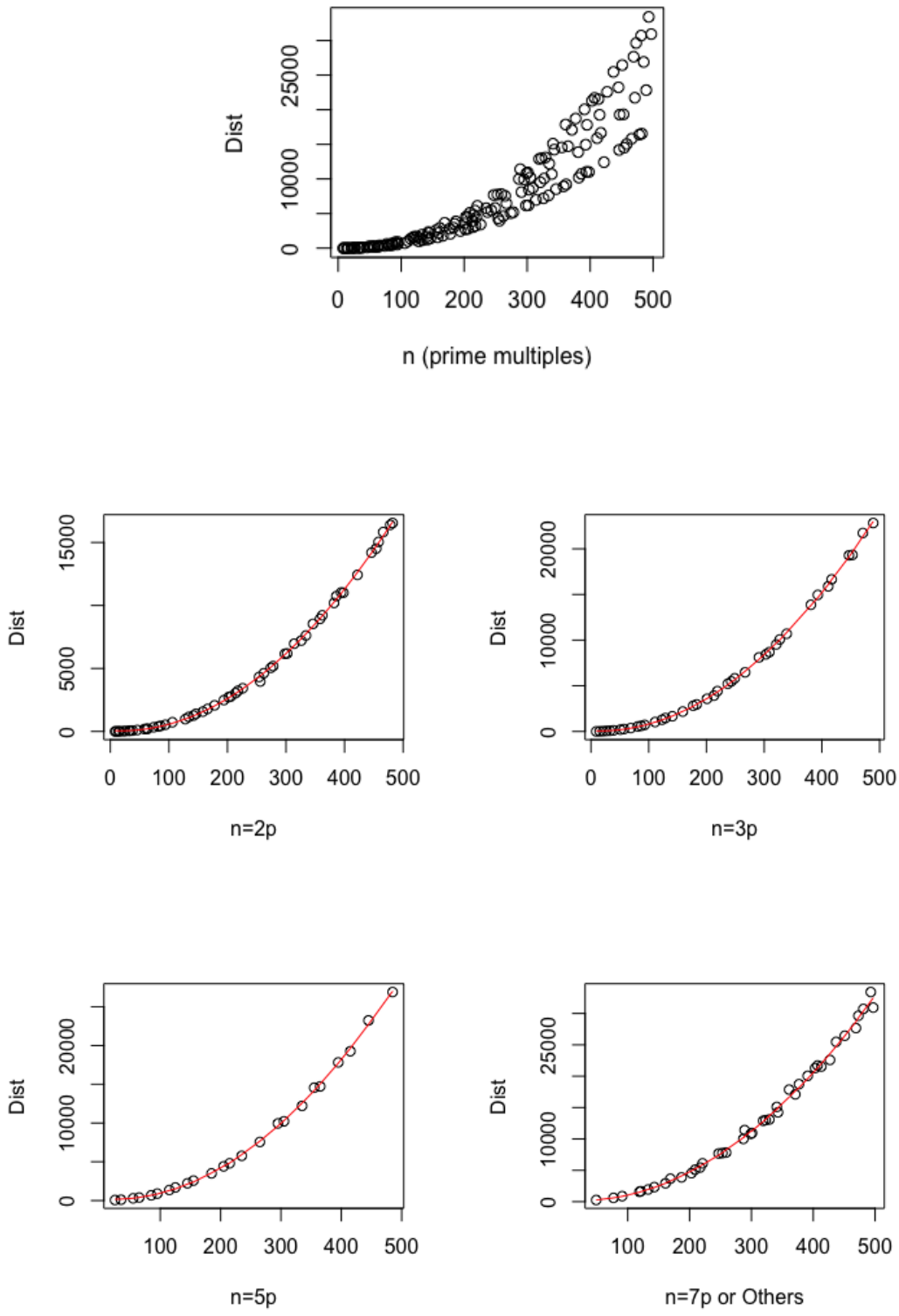
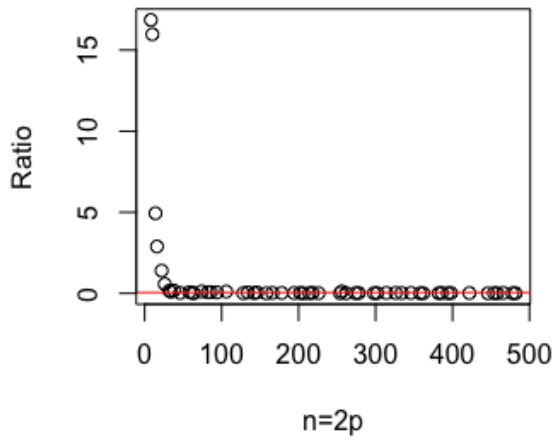
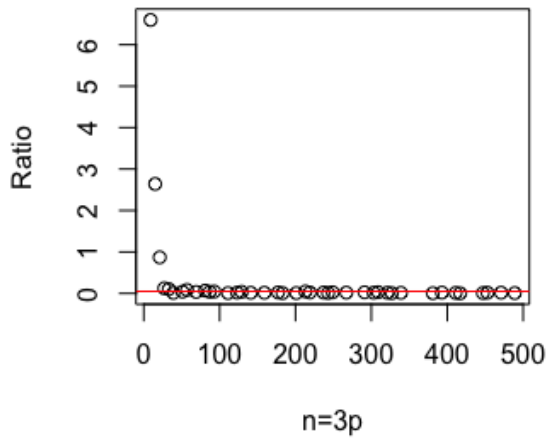


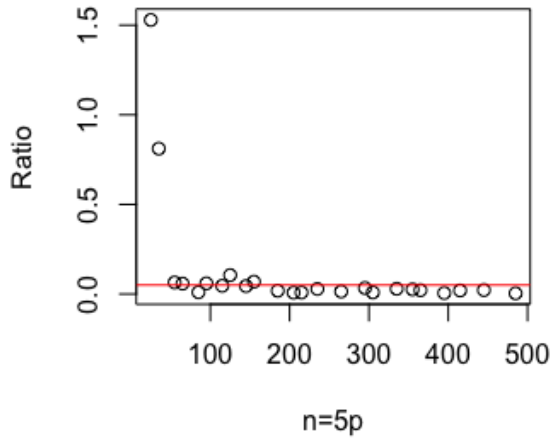
Figure 4.5: Relative Inaccuracy of Distance Predictions for Prime Multiples n (Residual and Actual Value Ratio vs Prime Multiple)



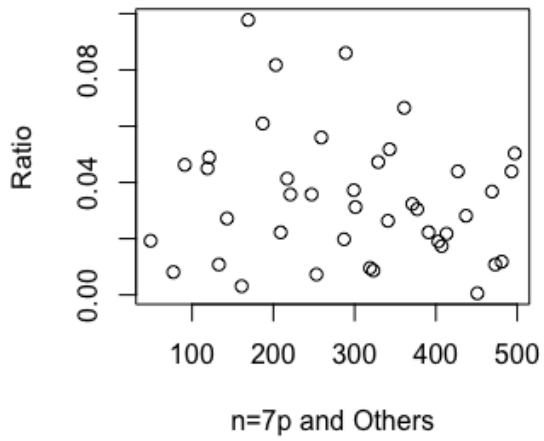
(a) 2p



(b) 3p



(c) 5p

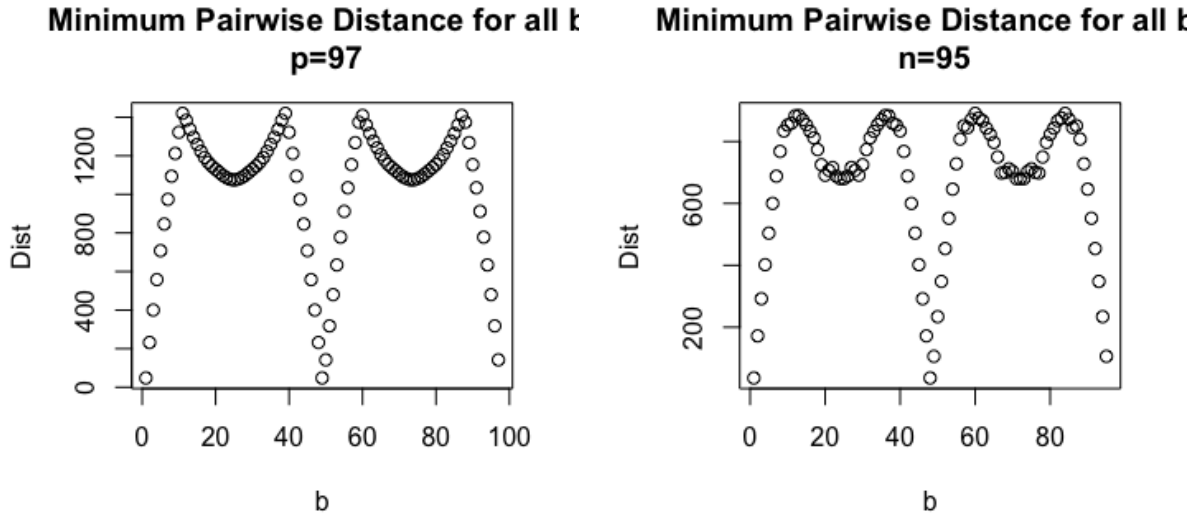


(d) 7p and Above

Unfortunately, we were unable to deduce any useful patterns for more general $b's$; even though Proposition 3 from Section 4.2 specifies which columns are selected in such LHDs, it

is rather unclear what values from $\{0, 1, ..., p-1\}$ are missing on a certain row (except for the last one), unlike in a $p \times \frac{p-1}{2}$ design, where it is always $W(b)$ missing from the entries on all the rows but the last one. However, we have found that, generally for a fixed p or n, the distances for different $b's$ look like two M shaped curves, and each M is perfectly symmetric around its center (but the two M's are not necessarily symmetric reflections of each other). Yet, the stepsizes for moving along these M shaped curves, or the extents of increment or decrement from one b value to the next, remain unknown.

Figure 4.6: Distance Trends for random p and n



(a) p=97

(b) n=95

CHAPTER 5

Extensions, Further Improvements and Conclusions

5.1 Extensions and Further Improvements

In fact, for prime multiples n, just like how we could generalize method (ii) to form method (iv) and accomodate for most of the cases where neither of $\phi(n)+1$ and $\frac{\phi(n)}{2}+1$ are primes, we could also apply the same generalization onto method (iii): we instead look for the next prime p_{next} after $\frac{\phi(n)}{2}$, randomly choose one of its primitive roots, raise it to the even exponents in $\mathbb{F}_{p_{next}}$ to give us approximately half of the columns in the final design matrix; then, we again subtract these column indices from $\phi(n)+1$. Because it is highly possible that the first half of the column indices may contain numbers larger than $\frac{\phi n}{2}$ and thus will perhaps cause some duplicates in the second half of the indices after we do subtraction, we remove the duplicated values from the final index vector to construct the final design matrix. Admittedly, a potential problem with this method (denoted as (v)), similar to method (iv), is that finally we might end up getting more columns than desired, and, in fact, after running this algorithm, we have found that, unlike method (iv) which occasionally produces 1-3 more columns, this method (iv) under some cases might give 5-8 more columns than required (for example, it gives 124 columns for $n = 295$ while only 116 are required), and it is also a heavy pain to pick 5-8 columns to remove from some large designs, say with dimension 295×124. Yet, this method has produced better efficiencies on several of the other prime multiples, especially $2p$, but except for $3p$, where the outputs from this method (v) is exactly the same as those from its special case, method (iii). Figure 5.1 depicts the comparisons among method (i) and the generalized methods (iv) and (v); for most of the times, method (v) has given the designs with the highest efficiency for a given n, and method (iv) is a close

second: its efficiency paths have a few more significant drops while those for method (v) are more stable around a higher point. Method (i) on the other hand is outperformed due to its ineffectiveness in dealing with even $n's$ and also $n > 100$.

Table 5.1: Maximin L_1 Distances and $b's$ for $n \times \phi(n)/2$ LHD Using Method (v)

Number	Columns	Maximin Distance	Upper Bound	Efficiency	b	Improved?
8	2	3	6	50%	1,3,5,7	✗
9	**4** (3)	8	13	61.54%	1,3,5,6,7,8	✗
10	2	3	7	42.86%	1,3,4,6,8,9	✗
14	**4** (3)	11	20	55%	1,2,5,8,9,12	✓(1.67%)
15	4	NA	NA	NA	NA	NA
16	4	10	22	45.45%	1,2,3,5,6,7,9,10,11,13,14,15	✗
21	6	NA	NA	NA	NA	NA
22	**6** (5)	30	46	65.21%	8,19	✓(12.59%)
25	10	65	86	75.58%	3,9	✗
26	6	31	54	57.41%	3,9,16,22	✗
27	**10** (9)	63	93	67.5%	3,10,17,23	✗
32	**10** (8)	72	110	65.45%	13,29	✓(4.09%)
33	10	NA	NA	NA	NA	NA
34	**10** (8)	82	116	70.69%	4,12,21,29	✓(5.10%)
35	12	97	144	67.36%	3,5,12,14,23,29	✗
38	**10** (9)	94	130	72.31%	14,33	✓(15.04%)
39	12	NA	NA	NA	NA	NA
46	**12** (11)	130	188	69.15%	6,29	✓(4.61%)
49	**22** (21)	277	366	75.68%	5,19	✗
51	16	NA	NA	NA	NA	NA
55	**22** (20)	308	410	75.12%	6,21	✗
57	18	NA	NA	NA	NA	NA
58	14	188	275	68.36%	7,36	✓(1.09%)
62	**14** (15)	NA	NA	NA	NA	NA
64	16	214	346	61.85%	9,25,41,57	✗
65	**26** (24)	427	572	74.65%	8,24	✓(1.92%)
69	22	NA	NA	NA	NA	NA
74	18	296	450	65.78%	9,46	✗
77	30	586	780	75.13%	48,67	✗
81	**28** (27)	588	765	76.86%	50,71	✓(2.74%)
82	**22** (20)	454	608	74.67%	9,50	✓(3.60%)
85	**36** (32)	758	1032	73.45%	10,32	✓(0.17%)
86	**22** (21)	461	638	72.26%	11,54	✓(0.34%)
87	28	NA	NA	NA	NA	NA
91	36	847	1104	76.72%	56,80	✗

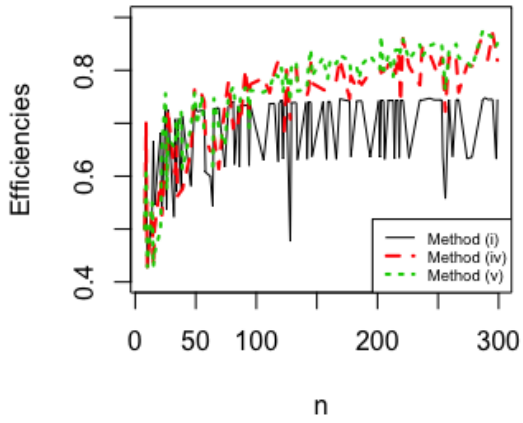
93	30	NA	NA	NA	NA	NA
94	**24** (23)	526	760	69.21%	12,59	✗
95	36	878	1152	76.22%	58,84	✗

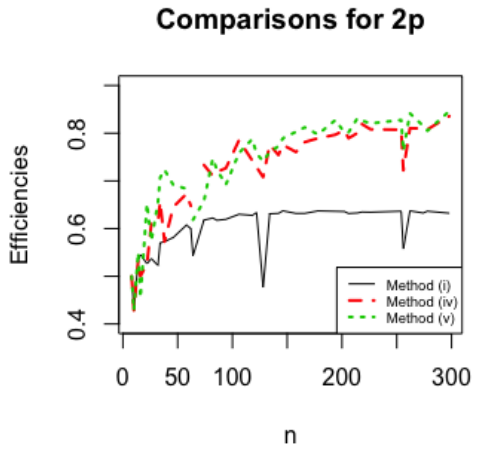
* NA is used for $n = 3p$ to indicate that the corresponding entries are no different from the ones given by method (iii), or for a lack of columns in other cases (e.g. $n = 62$);

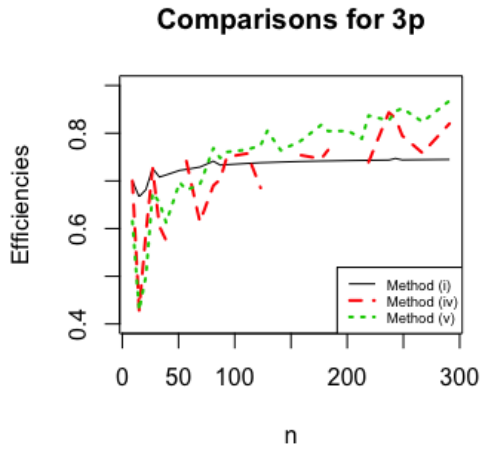
* The bolded values in the table and the values in the parentheses next to them are respectively specifying the number of columns in the generated design matrix and the actual value of $\frac{\phi(n)}{2}$ (if the bolded value is smaller than the value in the parentheses, then we have insufficient information about some columns that should be in the final matrix and thus the results from that run do not count);

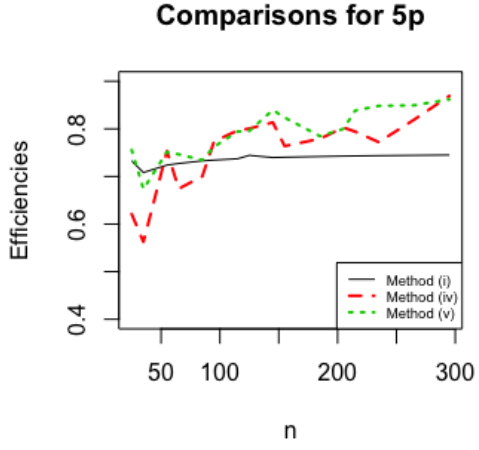
* Percentages next to ✓are indicating how much the efficiency has improved from the best of methods (i), (ii), (iii) and (iv) to here.

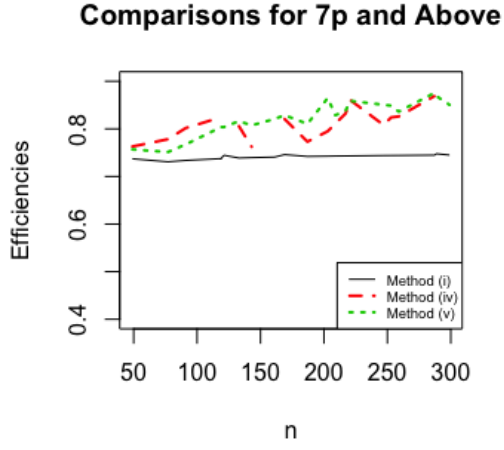
Figure 5.1: Efficiency Comparisons for Prime Multiples - Methods (i), (iv) and (v)











Furthermore, our ideas of constructing such flexible LHDs can be analogously generalized to build designs with only $\phi(N)/3$ and $\phi(N)/4$ columns. The results of applying these ideas to a selection of primes under 500 are presented in Table 5.2 and 5.3. Note that such algorithms can only be applied to $p = 3c + 1$ and $4c + 1$, respectively, since the designs will be varied upon different choices of primitive roots otherwise (Suppose $p = 3c + 2$ or $4c + 3$ for some positive integer c; then, following from the proof for Proposition 2 in Section 4.2, we have g and h as two primitive roots for p and $h = g^k \, (mod \, p)$, where $gcd(k, p - 1) = 1$; for $i \in \{3, 6, ...p - 2\}$ or $\{4, 8, ..., p - 3\}$, $ki = q(p - 2) + r$ for some r equal to a multiple of 3 from 0 to $p - 2$, or $ki = q(p - 3) + r$ for some r equal to a multiple of 4 from 0 to $p - 3$. Nonetheless, since none of the primitive roots g mod p has its $(p - 2)^{th}$ or $(p - 3)^{th}$ power equal to 1 in $\mathbb{F}_p$, we won't have the same set of column indices under different primitive roots).

As shown by Figure 5.2, the three lines all exhibit a pattern of efficiency increment with larger p, but it seems that the efficiencies for smaller fraction designs also tend to be smaller.

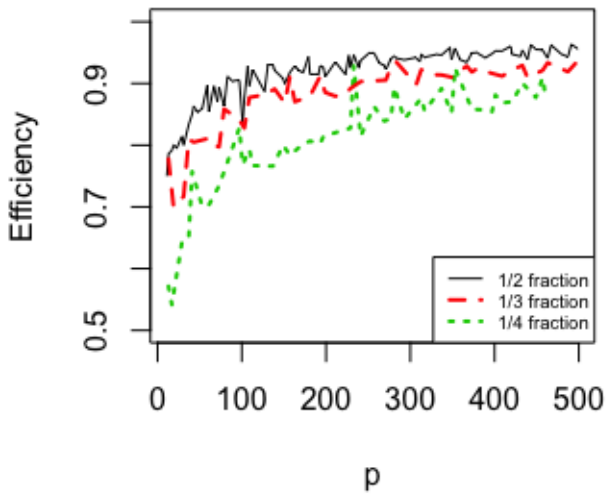
Table 5.2: Maximin L_1 Distance and $b's$ for $p \times \frac{p-1}{3}$ LHD

Number	Column	Dist	Upper Bound	Efficiency	Best b
13	4	14	18	77.78%	8,11
19	6	28	40	70%	2,7,12,16
31	10	76	106	71.70%	19,27
37	12	123	152	80.92%	4,14
43	14	165	205	80.49%	26,38
61	20	335	413	81.11%	36,55
67	22	398	498	79.91%	43,57
73	24	472	592	79.73%	46,63
79	26	595	693	85.86%	9,30
97	32	879	1045	84.11%	60,85
⋮	⋮	⋮	⋮	⋮	⋮
439	146	19482	21413	90.98%	46,173
457	152	21374	23205	92.11%	49,179
463	154	22234	23818	93.35%	50,181
487	162	24218	26352	91.90%	297,433
499	166	25920	27666	93.69%	302,446

Table 5.3: Maximin L_1 Distance and $b's$ for $p \times \frac{p-1}{4}$ LHD

Number	Column	Dist	Upper Bound	Efficiency	Best b
13	3	8	14	57.14%	8,11
17	4	13	24	54.17%	1,2,6,7,10,11,14,15
29	7	45	70	64.29%	7,19,24
37	9	74	114	64.91%	7,11,25,30
41	10	106	140	75.71%	5,15
53	13	164	234	70.09%	6,20,33,46
61	15	218	310	70.32%	42,49
73	18	326	444	73.42%	44,65
89	22	525	660	79.55%	10,34
97	24	649	784	82.78%	12,36
⋮	⋮	⋮	⋮	⋮	⋮
421	105	12856	14770	87.04%	260,371
433	108	14068	15624	90.04%	50,166
449	112	14739	16800	87.73%	49,175
457	114	15826	17404	90.93%	282,403
461	115	15470	17710	87.35%	292,399

Figure 5.2: Efficiency Comparisons for 1/2, 1/3 and 1/4 LHDs



5.2 Conclusion

Our method, using the primitive roots to construct flexible LHDs, has been successful in terms of maximizing the minimum L_1 distances between rows, that are about 80% of the upper bound, $\left\lfloor \frac{(n+1)\phi(n)}{6} \right\rfloor$ in the general $n \times \frac{\phi(n)}{2}$ design case, or $\left\lfloor \frac{(p+1)p}{6} \right\rfloor$ in the $p \times \frac{p-1}{2}$ design case, according to Wang, Xiao and Xu (2018). Moreover, the average correlations and maximum correlations for most $p's$ and $n's$ also seem to be adequate at around 0.01 and 0.1, whereas some maximum correlations reach 0.2 or 0.3.

The trends for the maximin distances and correlations seem to be well-established: when n or p gets larger, the distance and its corresponding efficiency go up, and the correlations decrease. Our method seems to have the best performances on primes, and a little worse but still decent performances on odd prime multiples, while the performances on even prime multiples are still yet to be improved. Furthermore, results on designs with $(p-1)/3$ and $(p-1)/4$ columns when $p = 3c+1$ and $4c+1$ are also poorer than the ones on regular designs with $(p-1)/2$ columns for general odd primes. A plausible explanation of these phenomena is that the fewer columns a design has, the smaller the distance efficiency is and the larger the correlations are: smaller n and p have smaller $\phi(n)$ and $p-1$; even prime multiples or 2^k for some integer k have one or more factors of 2 and thus are coprime with fewer positive integers than a comparable odd prime multiple or an odd prime; smaller fraction also diminishes the number of columns in the final design.

For most of p and n, there is usually a pair of two $b's$ ($0 \leqslant b \leqslant p-1$) that produce the best design in terms of the maximin distances, and these optimal $b's$ have a strong relationship with their corresponding $p's$ and $n's$; they are generally close to one of 0.1, 0.4, 0.6 and 0.9 times the value of p or n. Also, the overall trend for maximin distances of a random LHD constructed using our algorithm resembles two M-shaped curves, implying that the worst choices of $b's$ occur at the two ends and also the exact middle of the corresponding Galois fields, and each M-shaped curve is symmetric around its center. Even though the degree of increment or decrement from one b to the next is still yet to be determined, the best and worst distances for each p and n can be approximately or even exactly predicted using the

squared p or n coupled with the corresponding best b value after Williams transformation, or just using p or n itself. Nonetheless, further experiments are required to seek for patterns between a general b and its corresponding maximin distance.

CHAPTER 6

Appendix

Table 6.1: Maximin L_1 Distance and $b's$ for $p \times (p-1)/2$ LHD

Prime	Maximin Distance	Upper Bound	Efficiency	b	W(b)
11	15	20	75%	8	5
13	22	28	78.57%	1,5	2,10
17	38	48	79.17%	10,15	3,13
19	48	60	80.00%	11,17	3,15
23	70	88	79.55%	3,8,15,19	6,16,15,7
29	114	140	81.43%	4,10,19,24	8,20,19,9
31	128	160	80%	20,26	9,21
37	190	228	83.33%	22,33	7,29
41	238	280	85%	25,36	9,31
43	266	308	86.36%	25,39	7,35
47	314	368	85.33%	6,17	12,34
53	402	468	85.90%	32,47	11,41
59	520	580	89.66%	35,53	11,47
61	530	620	85.48%	36,37,54,55	49,47,13,11
67	670	748	89.57%	6,27	12,54
71	731	840	87.02%	45,61	19,51
73	794	888	89.41%	44,65	17,55
79	898	1040	86.35%	7,32	14,64
83	1046	1148	91.11%	49,75	15,67
89	1193	1320	90.38%	54,79	19,69
97	1420	1568	90.56%	10,38	20,76
101	1504	1700	83.56%	11,39	22,78
103	1548	1768	87.56%	62,92	21,81
107	1770	1908	92.77%	10,43	20,86
109	1774	1980	89.60%	66,97	23,85
113	1962	2128	92.20%	68,101	23,89
127	2384	2688	88.69%	75,77,113,115	103,99,27,23
131	2660	2860	93%	78,118	25,105
137	2910	3128	93.03%	83,122	29,107
139	2966	3220	92.11%	13,56	26,112

149	3360	3700	90.81%	16,58	32,116
151	3418	3800	89.95%	14,61	28,122
157	3770	4108	91.77%	95,140	33,123
163	4112	4428	92.86%	97,147	31,131
167	4282	4648	92.13%	20,63	40,126
173	4578	4988	91.78%	105,154	37,135
179	5036	5340	94.31%	107,161	35,143
181	4995	5460	91.48%	110,161	39,141
191	5564	6080	91.51%	114,116,170,172	153,149,41,37
193	5766	6208	92.87%	117,172	41,151
197	5984	6468	92.52%	21,77	42,154
199	6012	6600	91.09%	118,180	37,161
211	6936	7420	93.48%	126,190	41,169
223	7580	8288	91.46%	134,200	45,177
227	8121	8588	94.56%	22,91	44,182
229	8086	8740	92.52%	139,204	49,179
233	8558	9048	94.58%	141,208	49,183
239	8806	9520	92.50%	29,90	58,180
241	9072	9680	93.72%	25,95	50,190
251	9964	10500	94.90%	150,226	49,201
257	10452	11008	94.95%	27,101	54,202
263	10760	11528	93.34%	161,163,231,233	203,199,63,59
269	11204	12060	92.90%	29,105	58,210
271	11292	12240	92.25%	161,245	51,219
277	12022	12788	94.01%	168,247	59,217
281	12428	13160	94.44%	30,110	60,220
283	12560	13348	94.10%	169,255	55,227
293	13422	14308	93.81%	178,261	63,229
⋮	⋮	⋮	⋮	⋮	⋮
487	37276	39528	94.30%	291,439	95,391
491	38704	40180	96.33%	295,441	99,391
499	39672	41500	95.60%	299,449	99,399

Table 6.2: Maximin L_1 Distance and $b's$ for $n \times \phi(n)/2$ LHDs

Number	Columns	Maximin Distance	Upper Bound	Efficiency	b	Method
8	2	3	6	50%	1,3,5,7	(i), (ii)
9	3	7	10	70%	2	(i),(ii)
10	2	3	7	42.86%	1,3,4,6,8,9	(i),(ii)

14	3	8	15	53.33%	4,11	(i),(ii)
15	4	14	21	66.67%	11	(i)
16	4	12	22	54.55%	3,7,11,15	(i)
21	6	30	44	68.18%	5	(i)
22	5	20	38	52.63%	1,12	(i),(ii)
25	10	65	86	75.58%	3,9	(iii)
26	6	33	54	61.11%	9,22	(ii)
27	9	61	84	72.62%	20	(i),(ii),(iii)
32	8	54	88	61.36%	3,11,19,27	(ii)
33	10	80	113	70.80%	8	(i)
34	8	61	93	65.59%	12,29	(ii)
35	12	102	144	70.83%	26	(i)
38	9	67	117	57.26%	2,21	(i),(ii)
39	12	114	160	71.25%	29	(i)
46	11	111	172	64.53%	3,20,26,43	(ii)
49	21	267	350	76.29%	28,45	(ii)
51	16	200	277	72.20%	38	(i)
55	20	281	373	75.34%	34,48	(ii)
57	18	258	348	74.14%	6,22	(ii)
58	14	185	275	67.27%	22,51	(ii)
62	15	204	315	64.76%	27,58	(ii)
64	16	214	346	61.85%	9,25,41,57	(iii)
65	24	384	528	72.73%	16	(i)
69	22	374	513	72.90%	17	(i)
74	18	330	450	73.33%	27,64	(ii)
77	30	607	780	77.82%	47,68	(ii)
81	27	547	738	74.12%	20	(i)
82	20	393	553	71.07%	12,53	(ii)
85	32	672	917	73.28%	21	(i)
86	21	438	609	71.92%	6,49	(ii)
87	28	614	821	74.79%	11,32	(iii)
91	36	885	1104	80.16%	11,34	(ii)
93	30	715	940	76.06%	11,35	(iii)
94	23	529	728	72.66%	40,87	(ii)
95	36	891	1152	77.34%	59,83	(ii)
106	26	728	927	78.53%	12,65	(ii)
111	36	1029	1344	76.56%	67,99	(iii)
115	44	1355	1701	79.66%	13,44	(ii)
118	29	866	1150	75.30%	50,109	(ii)
119	48	1583	1920	82.45%	13,46	(ii)
121	55	1665	2236	74.46%	30	(i)
122	30	962	1230	78.21%	46,107	(ii)

123	40	1280	1653	77.43%	14,47	(iii)
125	50	1681	2100	80.05%	77,110	(ii)
128	32	976	1376	70.93%	19,51,83,115	(iv)
129	42	1465	1820	80.49%	80,113	(iii)
133	54	1943	2412	80.56%	81,118	(ii)
134	33	1152	1485	77.58%	10,77	(ii)
141	46	1663	2177	76.39%	17,53,87,124	(iii)
142	35	1258	1668	75.42%	60,131	(ii)
143	60	2325	2880	80.73%	15,56	(iii)
145	56	2217	2725	81.36%	16,56	(ii)
146	36	1392	1764	78.91%	58,131	(iii)
155	60	2570	3120	82.37%	95,137	(iii)
158	39	1572	2067	76.05%	67,146	(ii)
159	52	2175	2773	78.43%	96,142	(iii)
161	66	2925	3564	82.07%	19,61	(iii)
166	41	1783	2282	78.13%	70,153	(ii)
169	78	3663	4420	82.87%	102,151	(iii)
177	58	2813	3441	81.75%	20,68	(iii)
178	44	2072	2625	78.93%	70,159	(ii)
183	60	2960	3680	80.43%	111,163	(iii)
185	72	3503	4464	78.47%	22,70	(iii)
187	80	3875	5013	77.30%	115,165	(iv)
194	48	2486	3120	79.68%	23,120	(ii)
201	66	3576	4444	80.47%	24,76	(iii)
202	50	2728	3383	80.64%	24,77,125,178	(ii)
203	84	4541	5712	79.50%	119,126,178,185	(iv)
205	80	4409	5493	80.27%	23,79	(iv)
206	51	2778	3519	78.94%	16,119	(ii)
209	90	5109	6300	81.10%	130,183	(ii)
213	70	3936	4993	78.83%	25,81	(iii)
214	53	3039	3798	80.02%	94,201	(ii)
215	84	4803	6048	79.41%	128,194	(iv)
217	90	5441	6540	83.20%	130,195	(ii)
218	54	3230	3942	81.94%	25,134	(ii)
219	72	4419	5280	83.69%	26,83	(iv)
221	96	6109	7104	85.99%	134,197	(ii)
226	56	3424	4237	80.81%	27,140	(ii)
235	**95** (92)	5770	7473	77.21%	34,83	(iv)
237	78	5219	6188	84.34%	27,91	(ii)
243	81	5476	6588	83.12%	146,218	(ii)
247	108	7682	8928	86.04%	27,96	(iii)
249	82	5822	6833	85.20%	29,95	(iii)

253	**111** (110)	7744	9398	82.40%	30,96	(iv)
254	63	4323	5355	80.73%	20,147	(ii)
256	64	3959	5482	72.22%	22,86,150,214	(iv)
259	108	7827	9360	83.62%	159,229	(iii)
262	65	4617	5698	81.03%	21,152	(ii)
265	104	7554	9221	81.92%	34,98	(iv)
267	88	6476	7861	82.38%	163,237	(iii)
274	68	5048	6233	80.99%	105,242	(ii)
278	69	5184	6417	80.79%	22,161	(ii)
287	120	10006	11520	86.86%	30,113	(ii)
289	136	11413	13146	86.82%	177,256	(iii)
291	96	8111	9344	86.80%	178,258	(iii)
295	116	9948	11445	86.92%	33,114	(ii)
298	74	6163	7375	83.57%	113,262	(ii)
299	132	10805	13200	81.86%	186,262	(iv)
⋮	⋮	⋮	⋮	⋮	⋮	⋮
489	162	22832	26460	86.29%	58,186	(iii)
493	224	33414	36885	90.59%	301,438	(ii)
497	210	30932	34860	88.73%	301,444	(ii)

* (i) refers to the selection method using the first $\phi(n)/2$ coprimes;

* (ii) refers to the selection method using $\phi(n) + 1$ when $\phi(n) + 1$ is a prime (a special case of (iv))

* (iii) refers to the selection method using $\phi(n)/2 + 1$ when $\phi(n)/2 + 1$ is a prime;

* (iv) refers to the selection method using the next prime after $\phi(n)$ (a more general case of (ii))

* Bolded values (235 and **95**, 253 and **111**) are used to indicate that the number of columns produced using method (iv) is more than needed ($\phi(235)/2 = 92$ and $\phi(253)/2 = 110$).

Table 6.3: Columnwise Correlations for $p \times (p-1)/2$ Designs

Number	Columns	Avg. Cor	Max Cor	Avg. Cor SLHD	Max. Cor SLHD	Avg. Cor Gilbert	Max. Cor Gilbert
11	5	0.0636	0.0909	0.0909	0.209	0.0636	0.0909
13	6	0.112	0.341	0.0795	0.192	0.0740	0.341
17	8	0.0410	0.262	0.0625	0.161	0.0402	0.110
19	9	0.0285	0.0737	0.0554	0.158	0.0373	0.386
23	11	0.0130	0.0296	0.0491	0.145	0.0265	0.195
29	14	0.0346	0.256	0.0417	0.154	0.0192	0.111
31	15	0.0211	0.0831	0.0382	0.145	0.0168	0.111
37	18	0.0364	0.280	0.0480	0.166	0.0255	0.280
41	20	0.0135	0.110	0.0362	0.122	0.0264	0.393
43	21	0.0130	0.0630	0.0398	0.142	0.0107	0.111
47	23	0.00473	0.0106	0.0333	0.151	0.0176	0.159
53	26	0.0132	0.197	0.0285	0.0939	0.0125	0.108
59	29	0.00733	0.0365	0.0319	0.120	0.0128	0.420
61	30	0.0269	0.361	0.0326	0.106	0.00977	0.109
67	33	0.00862	0.0755	0.0293	0.102	0.0148	0.372
71	35	0.00534	0.0379	0.0261	0.125	0.0119	0.431
73	36	0.0148	0.224	0.0277	0.138	0.0125	0.304
79	39	0.00662	0.0721	0.0250	0.0889	0.0113	0.318
83	41	0.00576	0.0413	0.0248	0.115	0.0114	0.303
89	44	0.00870	0.184	0.0257	0.106	0.00822	0.184
97	48	0.0116	0.230	0.0234	0.102	0.00657	0.108
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
281	140	0.00320	0.207	⋮	⋮	0.00344	0.391
283	141	0.00245	0.0846	⋮	⋮	0.00288	0.195
293	146	0.00209	0.198	⋮	⋮	0.00264	0.219
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
487	243	0.00128	0.0846	⋮	⋮	0.00181	0.271
491	245	0.000941	0.0325	⋮	⋮	0.00167	0.257
499	249	0.00131	0.0875	⋮	⋮	0.00175	0.299

Table 6.4: Columnwise Correlations for $n \times \phi(n)/2$ Designs

Number	Columns	Avg. Cor (Best)	Max Cor (Best)	Avg. Cor SLHD	Max. Cor SLHD	Avg. Cor (i)	Max. Cor (i)

8	2	0.357	0.357	0.143	0.143	0.0238	0.0238
9	3	0.0333	0.0333	0.104	0.169	0.0333	0.0333
10	2	**0.382**	**0.382**	0.189	0.189	**0.0626**	**0.0626**
14	3	0.0593	0.0725	0.0271	0.0681	0.0593	0.0725
15	4	**NA**	**NA**	0.130	0.325	0.00952	0.0143
16	4	NA	NA	0.102	0.222	0.0471	0.0735
21	6	NA	NA	0.0598	0.174	0.0182	0.0390
22	5	0.0169	0.0649	0.0780	0.219	0.0282	0.0649
25	10	0.0160	0.111	0.0411	0.103	0.0291	0.108
26	6	0.0672	0.229	0.0584	0.138	0.0333	0.0824
27	9	0.0156	0.0317	0.0520	0.150	0.0156	0.0317
32	8	0.0608	0.278	0.0437	0.123	0.0271	0.117
33	10	NA	NA	0.0447	0.188	0.0138	0.0341
34	8	0.0488	0.176	0.0486	0.0894	0.0238	0.0814
35	12	NA	NA	0.0310	0.0992	0.0244	0.109
38	9	0.0196	0.0774	0.0435	0.127	0.0243	0.0774
39	12	NA	NA	0.0392	0.112	0.0111	0.0397
46	11	0.0174	0.0831	0.0280	0.0980	0.0202	0.0831
49	21	0.00949	0.0607	0.0314	0.111	0.0172	0.110
51	16	NA	NA	0.0302	0.114	0.0112	0.0399
55	20	0.00728	0.111	0.0277	0.0839	0.0166	0.110
57	18	0.0165	0.183	0.0270	0.0792	0.00958	0.0381
58	14	0.0234	0.106	0.0311	0.0995	0.0168	0.0848
62	15	0.0131	0.0783	0.0264	0.0890	0.0145	0.0783
64	16	0.0207	0.232	0.0250	0.110	0.0149	0.113
65	24	NA	NA	0.0244	0.108	0.0137	0.111
69	22	NA	NA	0.0223	0.0775	0.00861	0.0399
74	18	0.0198	0.111	0.0237	0.0806	0.0130	0.0804
77	30	0.0123	0.136	0.0229	0.0910	0.0126	0.111
81	27	NA	NA	0.0222	0.0902	0.00728	0.0399
82	20	0.0272	0.289	0.0196	0.0748	0.0113	0.0800
85	32	NA	NA	0.0212	0.0760	0.0103	0.111
86	21	0.0115	0.0668	0.0194	0.0661	0.0106	0.0774
87	28	0.00597	0.0517	0.0211	0.0796	0.00659	0.0392
91	36	0.00654	0.111	0.0244	0.107	0.0111	0.111
93	30	0.00552	0.0484	0.0192	0.0855	0.00601	0.0393
94	23	0.00994	0.0662	0.0178	0.0746	0.00976	0.0787
95	36	0.00633	0.111	0.0225	0.0879	0.00948	0.111
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
295	116	0.00284	0.147	⋮	⋮	0.00319	0.111
298	74	0.00423	0.110	⋮	⋮	0.00325	0.0777

299	132	0.00197	0.111	⋮	⋮	0.00347	0.111
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
489	162	0.00120	0.0705	⋮	⋮	0.00130	0.0400
493	224	0.00195	0.167	⋮	⋮	0.00210	0.111
497	210	0.00252	0.225	⋮	⋮	0.00205	0.111

* For $n = 10$ different optimal b values give different and very contrasting correlations; thus, the resulting average correlations are calculated by taking the average of the absolute values of all correlations under all optimal $b's$, and the maxima are calculated likewise.

* NA's in the first two columns indicate that method (i) is already the best method of the four for that n.

REFERENCES

[1] Ba, S., Myers, W.R. and Brenneman, W.A. (2015). Optimal sliced Latin hypercube designs. *Technometrics* **57** 479-487.

[2] Johnson, M.E., Moore, L. M. and Ylvisaker, D. (1990). Minimax and maximin distance designs. *J. Statist. Plann. Inference* **26** 131-148.

[3] Lin, C.D. and Tang, B. (2015). Latin hypercubes and space filling designs. In *Handbook of Design and Analysis of Experiments* (A. Dean, M. Morris, J. Stufken and D. Bingham, eds.) 593-625. Chapman & Hall/CRC, London.

[4] Wang, L., Xiao, Q. and Xu, H. (2018). Optimal Maximin L_1-Distance Latin Hypercube Designs Based on Good Lattice Point Designs. *The Annals of Statistics* **46** 3741-3766.

[5] Xiao, Q. and Xu, H. (2017). Construction of maximin distance Latin squares and related Latin hypercube designs. *Biometrika* **104** 455-464.

[6] Zhou, Y. and Xu, H. (2015). Space-filling properties of good lattice point sets. *Biometrika* **102** 959-966.