# UCLA UCLA Electronic Theses and Dissertations

Title

Construction of Maximin Distance Designs via Level Expansion

Permalink https://escholarship.org/uc/item/4dp4n5nm

**Author** XIAO, QIAN

Publication Date 2015

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

# Construction of Maximin Distance Designs via Level Expansion

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

by

Qian Xiao

2015

© Copyright by Qian Xiao 2015

#### Abstract of the Thesis

## Construction of Maximin Distance Designs via Level Expansion

by

#### Qian Xiao

Master of Science in Statistics University of California, Los Angeles, 2015 Professor Hongquan Xu, Chair

Maximin distance designs as an important class of space-filling designs are widely used today, yet their constructions are challenging. In this article, we develop a 3-step procedure which can efficiently generate maximin distance Latin hypercube designs and maximin distance fractional factorial designs. This new method selects existing efficient low-level designs to generate high-level maximin distance designs via level expansion. The generated maximin distance designs are of flexible run and factor sizes and also have robust pairwise correlations. To justify this method, we derive the relationships of the distance distributions between the initial low-level designs and the generated high-level designs. We also prove the relationships between the generalized word length patterns of the initial low-level designs and the distance distributions of the generated high-level designs. Examples are presented to show that this new method outperforms many current prevailing methods in generating maximin distance designs. The thesis of Qian Xiao is approved.

Jingyi Li

Qing Zhou

Hongquan Xu, Committee Chair

University of California, Los Angeles 2015

I dedicate my dissertation work to my parents. A special feeling of gratitude to my loving parents who support me both emotionally and economically to finish my master degree.

## TABLE OF CONTENTS

1	Introduction	1
2	Some Theoretical Results	6
3	The Maximin Level Expansion (MmLE) Method	11
	3.1 Procedures of MmLE	11
	3.2 The Threshold Accepting Algorithm in MmLE	12
	3.3 Justifications for the procedures of MmLE	13
4	Results and Comparison	16
	4.1 Construction of Maximin LHDs	16
	4.2 Construction of Maximin Factorial Designs	19
5	Multi-phase Maximin Level Expansion Method	23
6	Conclusions	26
7	Appendix	<b>28</b>
	7.1 Proof of Lemma 1	28
	7.2 Proof of Theorem 1	29
	7.3 Proof of Theorem 2	32
Re	eferences	35

# LIST OF FIGURES

1.1 Comparison of 9-run 2-factor LHDs	. 2
---------------------------------------	-----

# LIST OF TABLES

4.1	Comparisons of constructions of maximin LHDs	17
4.2	Comparison on the MmLE and LP method for constructing four-	
	level maximin designs	20
4.3	Comparison with UD-homepage designs	22
5.1	Comparison of one-phase and two-phase MmLE method	24

#### Acknowledgments

I wish to thank my committee members for their precious time. Special thanks to Professor Hongquan Xu, my instructor and committee chair for his time reviewing, encouraging, and patience for the whole process. Thanks to Professor Qing Zhou and Professor Jingyi Li for agreeing to serve on my committee. I would like to thank my department and graduate division for allowing me to conduct my research and providing any assistance requested. The research was supported in part by National Science Foundation DMS grant 1407560.

## CHAPTER 1

### Introduction

Computer experiments play a key role in modern era of product and technology developments. Since deterministic models are used in computer experiments, their outputs are subject to no randomness. Thus, replicate points under any design projection are undesirable. McKay, Beckman and Conover (1979)[MBC79] propose the Latin hypercube designs (LHDs) whose columns are some distinct permutations of number 1, 2, ..., n (*n* is the number of runs). LHDs are widely used in computer experiments due to their unique point projections on any one dimension. Space-filling designs is another important class of designs suitable for computer experiments. The space-filling property, which spreads the design points out across the entire experimental region, can bound the bias between the meta-models and the true models in computer experiments (Fang, Li, and Sudjianto (2005)[FLS05]). Thus, in practice space-filling Latin hypercube designs are usually considered as the most desirable for computer experiments.

Constructing space-filling LHDs, especially ones with large run and factor sizes, is challenging. Different methods and various optimality criteria have been proposed. Some researchers construct orthogonal LHDs, e.g., Bursztyn and Steinberg (2002) [BS02], Cioppa and Lucas (2007)[CL07], Sun et al. (2010) [SLL10] and Yang and Liu (2012)[YL12]. However, orthogonal LHDs are not necessarily space-filling, e.g. design (a) in Figure 1.1. Hickernell (1998) [Hic98] defines several discrepancy criteria via reproducing kernel Hilbert spaces and claims that designs with smaller discrepancy values are more space-filling. Among these discrepancy criteria, the following defined centered L2 discrepancy (CD) is the most widely accepted.

$$(CD(D_n))^2 = (\frac{13}{12})^k - \frac{2}{n} \sum_{i=1}^n \prod_{j=1}^k [1 + \frac{1}{2}|x_{i,j} - 0.5| - \frac{1}{2}|x_{i,j} - 0.5|^2] + \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \prod_{l=1}^k [1 + \frac{1}{2}|x_{i,l} - 0.5| + \frac{1}{2}|x_{j,l} - 0.5| - \frac{1}{2}|x_{i,l} - x_{j,l}|]$$

where  $x_{i,j}$  is the element of the  $i^{th}$  row and  $j^{th}$  column from the design matrix  $D_n$ . However, Zhou, Fang and Ning (2012) [ZFN13] find the curse of dimensionality that CD optimal designs concentrate more points in the center for high dimensional cases.



Figure 1.1: Comparison of 9-run 2-factor LHDs

Johnson et al. (1990) [JMY90] propose the maximin distance criterion which interpolates design points over the domain in a way that the minimum distance between pairs of points is maximized. They have also established an equivalence of the maximin distance criterion and an entropy criterion in a Bayesian setting. Maximin distance designs ensure that no design points are too close to each other. Due to the goodness and robustness of the maximin distance criterion in measuring designs' space-filling property, we aim at constructing maximin distance designs in this article. Morris and Mitchell (1995)[MM95] propose a scalar value:

$$\phi_p = \left(\sum_{i=2}^n \sum_{j=1}^{i-1} \frac{1}{d_{i,j}^p}\right)^{\frac{1}{p}} , \qquad (1.1)$$

where  $d_{i,j}$  is the distance between the  $i^{th}$  and  $j^{th}$  row of the design. When p becomes large enough,  $\phi_p$  is asymptotically identical to the maximin distance criterion. They also adopt a simulated annealing algorithm to search maximin distance LHDs. Joseph and Hung (2008) propose a multi-objective optimization approach to generate orthogonal maximin LHDs (OMLHDs). OMLHDs are searched towards a multi-objective criterion combining the  $\phi_p$  value defined in (1.1) and the average column-wise correlation of the design. Ba et al (2014) [BBM14] propose an efficient algorithm, available as the R package "SLHD", to generate maximin LHDs with flexible run and factor sizes. The algorithms by Joseph and Hung (2008) and Ba et al (2014) are currently most efficient ways in generating space-filling LHDs with flexible run and factor sizes.

To achieve stratification on multi-dimensional projections, Tang (1993) [Tan93] proposes the orthogonal array based LHDs (OALHDs). He first randomizes the rows, columns and symbols of a randomly chosen orthogonal array (OA) and then randomly expands its levels to generate a LHD. Yet, OALHDs can be bad towards space-filling property, for example, design (b) in Figure 1.1. Even the orthogonal OALHDs are still not space-filling, e.g., design (c) in Figure 1.1. As a following research, Leary, Bhaskar and Keane (2003) [LBK03] adopt a simulated annealing algorithm to search maximin OALHDs. Their results are still not satisfactory, since they fail to find out suitable initial OAs (refer to part 4.1).

Besides space-filling LHDs, space-filling fractional factorial designs are also of important use. In some combinatory drug experiments, researchers need to explore various models on the data due to the inherent complexity of the biological system. Under such situations, space-filling designs are desirable. In addition, since the number of levels for every drug should be relatively small in practice, spacefilling fractional factorial designs with relatively small levels should be used (Zhou and Xu, (2014)[ZX14]). To find maximin fractional factorial designs, Zhou and Xu (2014)[ZX14] propose a level permutation method. They first select existing orthogonal arrays with sequentially minimized generalized word length pattern. Then they permute their levels and select the maximin distance designs.

In this article, we propose a maximin level expansion (MmLE) method which can efficiently generate maximin distance LHDs and factorial designs with flexible run and factor sizes. It includes three steps: first finding generalized minimum aberration (GMA) designs (with sequentially minimized generalized word length patterns), then selecting maximin design by doing level permutations and finally performing level expansion (refer to part 3.1 for details). The first two steps, similar to Zhou and Xu (2014) [ZX14], give the best initial designs for level expansion and the third step includes Tang's level expansion idea. In part 4.2, we present examples to show that designs by our MmLE method have better maximin distance property than those by Zhou and Xu (2014)[ZX14]. Moreover, their permutation method requires suitable existing OAs which are hard to find in many cases whereas our MmLE method is more flexibile in designs' run and factor sizes. We also develop theories on the relationships of the space-filling property between the initial and generated designs. In addition, we prove the relationships between the initial designs' generalized word length patterns and the generated designs' average space-filling property. Compared with Tang (1993), this article generalizes Tang's level expansion idea, provides an efficient construction of maximin distance OALHDs, and fill in some theoretical blanks of Tang's work.

The article is organized as follows: in part 2, we develop theories to justify the method. In part 3, we introduce the 3-step procedure, the threshold accepting (TA) searching algorithm and the justifications for the MmLE method. In part 4, examples are given to show that our MmLE method outperforms the ordinary level expansion method by Leary, Bhaskar and Keane (2003) [LBK03], the OMLHD method by Joseph and Hung (2008), the R package "SLHD" by S Ba, the level permutation method by Zhou and Xu (2014)[ZX14], and the uniform designs by Kaitai Fang and his colleagues. In part 5, we introduce the multi-phase MmLE

method for constructing large maximin designs. Part 6 concludes and all proofs are given in the Appendix.

### CHAPTER 2

### Some Theoretical Results

Denote  $D(n, s^k)$  as an *n*-run, *k*-factor and *s*-level (labelled as 1, 2, ..., s) balanced design where each level appears n/s times in every column. From  $D(n, s^k)$  we can generate a set of designs  $D'(n, (ms)^k)$  with ms levels by the following level expansion procedure. For each column in design D, we replace the n/s positions of entry l (l = 1, 2, ..., s) by a random sequence of n/(ms) replicates of numbers: (l-1)m+1, (l-1)m+2, ..., (l-1)m+m. Here n, k, s, m are all positive integers which are larger than 1, and n is divisible by ms. When m = n/s, the generated designs D' are LHDs.

**Example 1** As an illustration, here we perform the level expansion procedure to generate  $D'(8, 4^2)$  from  $D(8, 2^2)$ . For each column of design D, we first replace all 4 entries of 1 in D with a random permutation of numbers: 1, 1, 2, 2, and then replace all 4 entries of 2 with a random permutation of numbers: 3, 3, 4, 4, thus generating a 4-level design D'. In all we have 1296 possible D's.

$$D = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 1 & 2 \end{pmatrix}^{T} \Rightarrow D' = \begin{pmatrix} 1 & 2 & 1 & 2 & 3 & 4 & 4 & 3 \\ 1 & 3 & 2 & 4 & 1 & 4 & 2 & 3 \end{pmatrix}^{T} \text{ or } \dots$$

Denote  $x_{i,l}$  as the  $(i^{th}, l^{th})$  element and  $x_i$  as the  $i^{th}$  row of the design matrix D. After the level expansion process, they become  $x'_{i,l}$  and  $x'_i$  of the generated design D' respectively. Denote  $h_{i,j}$  as the Hamming distance (number of positions that the corresponding entries in the pair of rows are different) between rows  $x_i$  and  $x_j$ . Define  $d_{il,jl} = |x_{i,l} - x_{j,l}|$ . Denote the L-1 distance between two rows  $x_i$  and  $x_j$  as  $d_{i,j} = \sum_{l=1}^{k} d_{il,jl}$ . Denote  $d_{min}(D)$  as the minimum L-1 distance among all pairs of rows in design D. In the same way, we define  $h'_{i,j}$ ,  $d'_{il,jl}$ ,  $d'_{i,j}$  and  $d_{min}(D')$ for the derived design D' respectively. For any balanced design D, we define the L-1 distance distribution as

$$B_l(D) = n^{-1} \#\{(i,j) : d_{i,j} = l; x_i, x_j \in D, i, j = 1, 2, \dots, n\} .$$
(2.1)

It is easy to show that  $B_0(D) \ge 1$  and a design without repeated runs has  $B_0(D) =$ 1. The maximin distance design is defined as the one which sequentially minimizes the distance distribution  $B_0(D), B_1(D), B_2(D), B_3(D), \ldots$  Thus, designs with smaller  $\phi_p$  values defined in (1.1) have better distance distributions.

**Lemma 1** (a) The upper and lower bounds for the pairwise L-1 distances of the generated design D' are:  $(i, j = 1, 2, ..., n \text{ and } i \neq j)$ 

$$md_{i,j} - (m-1)h_{i,j} \leq d'_{i,j} \leq md_{i,j} + (m-1)k$$
.

(b) The upper and lower bounds for the minimum pairwise L-1 distance of the generated design D' are:

$$md_{min}(D) - (m-1)h_{max}(D) \leq d_{min}(D') \leq md_{min}(D) + (m-1)k$$
,

where  $h_{max}(D)$  is the largest pairwise Hamming distance in design D.

Given n, s and k, from different initial designs  $D(n, s^k)$ , by level expansion we can generate different sets of designs  $D'(n, (ms)^k)$ . By Lemma 1, the upper bound for  $d_{min}(D')$  is solely determined by  $d_{min}(D)$ . Thus, if we start from the maximin distance initial design D and can generate a derived design  $D'_{optimal}$  with  $d_{min}(D'_{optimal}) = md_{min}(D) + (m-1)k$ , then it's clear that  $D'_{optimal}$  has the largest minimum L-1 distance among all possible D's from all possible initial designs D. The lower bound of  $d_{min}(D')$  is also positively correlated with  $d_{min}(D)$ . Therefore, the maximin initial designs can improve the possible best and worst cases of the generated designs by level expansion. From a specific initial design D, by the level expansion process we have  $N_1 = \left(\frac{(n/s)!}{(\frac{n}{ms}!)^m}\right)^{sk}$  possible generated D'. In the following Theorem 1, the expectation and variance are given towards all these  $N_1$  possible D'.

**Theorem 1** For any different  $i^{th}$  and  $j^{th}$  row  $(i, j = 1, 2, ..., n \text{ and } i \neq j)$ , the expectation and variance of the pairwise L-1 distances of the generated designs D' via level expansion have the following relationship with the pairwise L-1 distance of the initial balanced design D.

$$E(d'_{i,j}) = md_{i,j} + (k - h_{i,j})\gamma \quad and \quad Var(d'_{i,j}) = C_{1,0} + C_{1,1}h_{i,j} ,$$
  
where  $\gamma = \frac{n(m^2 - 1)}{3m(n - s)}, \ C_{1,0} = \frac{n(m^2 - 1)(m^2n + 2n - 3m^2s)}{18m^2(n - s)^2}k \ and \ C_{1,1} = \frac{m^2 - 1}{18m^2(n - s)^2}(2m^2n^2 - 2n^2 - 3sm^2n + 3m^2s^2) > 0.$ 

Two-level initial designs  $D(n, 2^k)$  are useful to generate maximin designs with even number of runs via level expansion. Since the Hamming distances  $h_{i,j}$  are equal to the L-1 distances  $d_{i,j}$  for two-level designs, by Theorem 1 we have:  $E(d'_{i,j}) = \frac{2m^2(n-3)+n}{3m(n-2)}d_{i,j} + \frac{nk(m^2-1)}{3m(n-2)}$  and  $\operatorname{Var}(d'_{i,j}) = C_{1,0} + C_{1,1}d_{i,j}$ . Thus, for any pair of rows the expectation and variance of  $d'_{i,j}$  are solely determined by  $d_{i,j}$  from the initial design. Another useful class of initial designs is saturated orthogonal arrays, which provide the largest number of factors k ( $k = \frac{n-1}{s-1}$ ) given n while keeping the orthogonality. Mukerjee and Wu (1995)[MW95] show that the Hamming distances are constants n/s for all saturate OAs. By Theorem 1 we have:  $E(d'_{i,j}) = md_{i,j} + \frac{n-s}{s(s-1)}\gamma$  and  $\operatorname{Var}(d'_{i,j}) = C_{1,0} + C_{1,1}\frac{n}{s}$ . Thus, the average  $d'_{i,j}$  is also solely determined by  $d_{i,j}$  and the variance is constant. In the second step of MmLE method in part 3.1, we search best candidates for initial designs through level permutation, which don't change  $h_{i,j}$  for arbitrary i and j. Thus, in general a candidate initial deign D with better distance distribution will generate D's with better distance distribution on average. Xu and Wu (2001) [XW01] define the generalized word length pattern  $(A_1, A_2, \ldots, A_k)$  for fractional factorial designs on k factors. The values of  $A_j(D)$  represent the total aliasing between the general mean and all *j*-factor interactions in the full ANOVA model. The generalized minimum aberration (GMA) designs by Xu and Wu (2001) [XW01] sequentially minimize designs' generalized word length patterns. If we permute the levels of designs, their generalized word length patterns do not change, but their distance distributions can be improved. Our MmLE method combines the level permutation process and the level expansion process. The following two lemmas are from Xu (2003) and Zhou and Xu (2014) respectively.

**Lemma 2** The Hamming distances from design  $D(n, s^k)$  have the following relationships with the generalized word length pattern of design D

$$\sum_{1 \le i < j \le n} (k - h_{i,j}) = \frac{n^2 A_1(D) + kn^2 - ksn}{2s} ,$$
$$\sum_{1 \le i < j \le n} (k - h_{i,j})^2 = \frac{n^2 (2A_2(D) + (2k + s - 2)A_1(D) + k(k + s - 1)) - n(ks)^3}{2s^2}$$

**Lemma 3** Consider all possible level permutations of design  $D(n, s^k)$  and define  $\overline{B}_l(D)$  as the average of  $B_l(D)$  (defined in (2.1)). We have

$$\begin{split} \frac{1}{n} \sum_{l=1}^{(s-1)k} l \overline{B}_l(D) &= \frac{k(s^2-1)}{3s} - \frac{s+1}{3s} A_1(D) \ , \\ \frac{1}{n} \sum_{l=1}^{(s-1)k} l^2 \overline{B}_l(D) &= C_{0,0} + C_{0,1} A_1(D) + 2\left(\frac{s+1}{3s}\right)^2 A_2(D) \ , \\ where \ C_{0,0} &= k \frac{(s^2-1)(s^2+2ks^2-2k+2)}{18s^2} \ and \ C_{0,1} &= \frac{-(s+1)(3s^2+4(k-1)(s^2-1))}{18s^2}. \end{split}$$

Lemma 2 gives the relationship between design's Hamming distance and its generalized word length pattern. Lemma 3 shows the relationship between design's average distance distribution and its generalized word length pattern under all possible level permutations. By these two lemmas, we can prove the following Theorem 2. Given a design  $D(n, s^k)$ , by doing all possible level permutations we have a set of designs  $\{D_1, D_2, \ldots, D_{N_2}\}$  where  $N_2 = (s!)^k$ . From any  $D_p$  $(p = 1, 2, \ldots, N_2))$ , by level expansion we can generate a set of derived designs  $\{D'_{p,1}, D'_{p,2}, \ldots, D'_{p,N_1}\}$  where  $N_1 = \left(\frac{(n/s)!}{(\frac{n}{ms}!)^m}\right)^{sk}$ . Thus, when all possible level permutations towards the initial design  $D(n, s^k)$  are considered, by level expansion totally we have  $N_1 * N_2$  possible generated designs D' (denote this set as  $\Theta$ ). For a balanced design  $D(n, s^k)$ , it's easy to show that for all possible  $d'_{i,j}, \sum_{i\neq j=1}^n d'_{i,j} = \frac{kn^2(m^2s^2-1)}{3ms}$ . In the following Theorem 2, the expectations are taken over the set  $\Theta$ .

**Theorem 2** If all possible level permutations towards the initial design  $D(n, s^k)$ are considered, the relationships between the pairwise L-1 distances of the generated designs D' and the generalized word-length pattern of the initial design Dare

$$\sum_{i \neq j=1}^{n} E_{\Theta}((d'_{i,j})^2) = C_{2,0} + C_{2,1}A_2(D)$$

where  $C_{2,0}$  and  $C_{2,1} > 0$  are constants.

The quantity  $\sum_{i=1}^{n} \sum_{j=1}^{n} E_{\Theta}((d'_{i,j})^2)$  measures variance of the pairwise L-1 distances of D' in the set  $\Theta$ . Since for balanced designs the average distances are constants and  $C_{2,1} > 0$ , the variance is minimized when the initial designs are OAs  $(A_2 = 0)$ . Thus, when suitable OAs exist, we should choose OA initial designs for our MmLE algorithm. When suitable OAs don't exist, we should choose a balanced design with smallest  $A_2$  value. It's clear that this theorem gives the mean and variance of the pair-wise L-1 distances of the OALHDs by Tang (1993)[Tan93].

### CHAPTER 3

### The Maximin Level Expansion (MmLE) Method

#### 3.1 Procedures of MmLE

Benefiting from many existing orthogonal arrays, in this article we focus on MmLE method starting from low-level OAs to generate high-level maximin designs. We should also notice that MmLE method in fact can start from any designs, for example, supersaturated designs, nearly-OAs, randomly generated balanced designs or even unbalanced designs. Here we start from  $OA(n, s^{k_{max}})$  to generate  $D'(n, (ms)^k)$  where  $k \leq k_{max}$ . The MmLE method includes the following three steps.

- 1. Select the GMA k-column subset from  $OA(n, s^{k_{max}})$  and denote the generated design as  $D(n, s^k)$ .
- 2. If s > 2, select the maximin distance design from ones generated by doing all possible level permutations of D. Denote it as  $D_p(n, s^k)$ .
- For each column in D<sub>p</sub>, replace n/s positions of entry l (l = 1, 2, ..., s) by a random sequence of n/ms replicates of numbers: (l − 1)m + 1, (l − 1)m + 2..., (l − 1)m + m. Select the maximin one from all generated designs as D'(n, (ms)<sup>k</sup>).

In step 1, if the GMA design is not known from current literatures, we usually search it from saturated strength two  $OA(n, s^{k_{max}})$  where  $k_{max} = \frac{n-1}{s-1}$ . When  $\binom{k_{max}}{k}$  is small, we can enumerate and compare all subsets. When  $\binom{k_{max}}{k}$  is large,

we adopt a local searching method: randomly generate 5000 subset designs and select the GMA one. As a special case, for regular designs we choose minimum aberration (MA) designs in step 1. For step 2 and 3, we adopt the threshold accepting (TA) algorithm introduced in the following part 3.2 for the searching process.

#### 3.2 The Threshold Accepting Algorithm in MmLE

Here we adopt and modify the threshold accepting (TA) algorithm by Gilli et al (2006)[GKH06]. Compared with the simulated annealing algorithm by Kirkpatrick (1984)[Kir84], TA converges faster (Dueck and Scheuer (1990)[DS90]). Let  $\phi(D) = \phi_p(D)$  as the object function to be minimized in our TA algorithm. The steps for the TA algorithm are shown as follows:

- 1. Set up the predetermined value of  $n_{rounds}$ ,  $n_{steps}$  and  $n_{seq}$ .
- 2. Randomly choose the initial  $D_c$ .
- 3. For *i* from 1 to  $n_{seq}$ , randomly generate a new design  $D_n$  from the neighbours  $N(D_c)$ , denote the difference  $\delta_i = |\phi(D_c) \phi(D_n)|$ .
- 4. Compute the empirical distribution of  $\delta_i$ ,  $i = 1, 2, \ldots, n_{seq}$ , denote as F.
- 5. The threshold sequence  $\tau_r = F^{-1}\left(0.8 * \frac{n_{rounds} r}{n_{rounds}}\right), r = 1, 2, \dots, n_{rounds}$ .
- 6. for r from 1 to  $n_{rounds}$ , do
- 7. for i from 1 to  $n_{steps}$ , do
- 8. Randomly generate a new design  $D_n$  from the neighbours  $N(D_c)$  and compute difference  $\delta = \phi(D_n) - \phi(D_c)$
- 9. if  $\delta < \tau_r$  then let  $D_c = D_n$

- 10. end for
- 11. end for
- 12. Output solution  $D^{sol} = D_c$

Here  $n_{rounds}$  represents the number of thresholds,  $n_{steps}$  represents the number of steps per threshold and  $n_{seq}$  represents the number of random steps to compute the threshold sequence. In practice, based on the design size and time limits, normally we choose  $n_{rounds}$  from 25 to 75,  $n_{steps}$  from 2500 to 7500 and set  $n_{seq}$ equal to 2000. We apply this TA algorithm both in step 2 and 3 of our MmLE method. Denote the current design as  $D_c$ . In step 2,  $D_c$  is a random generated design by doing all possible level permutations of D from step 1. To generate neighbour designs  $D_n = N(D_c)$ , we randomly choose two distinct levels from a randomly chosen column of  $D_c$  and exchange all elements of these two levels. In step 3,  $D_c$  is a randomly generated design by doing level expansion of  $D_p$  from step 2. We define the neighbour design  $D_n = N(D_c)$  by exchanging the levels in two positions which have the same value in  $D_p$  but different values in  $D_c$ .

#### 3.3 Justifications for the procedures of MmLE

There are in all  $N_0 = \left(\frac{n!}{((n/(ms))!)^{ms}}\right)^k$  possible designs for  $D'(n, (ms)^k)$  (including isomorphic designs). By the level expansion procedure, we greatly reduce the number of possible designs from  $N_0$  to  $N_1 = \left(\frac{(n/s)!}{(\frac{n}{ms}!)^m}\right)^{sk}$ . However,  $N_1$  is still so large that the local searching method doesn't work well. In MmLE method, we adopt the TA algorithm in searching. If we can increase the average space-filling property of these  $N_1$  designs, we can improve the TA algorithm's efficiency and robustness. The purpose of step 1 and 2 is to find an optimal initial design which can improve the efficiency of the TA algorithm used in step 3.

Here we first show that generally speaking, the maximin design  ${\cal D}_p$  from step

2 will generate D's with best distance distribution on average in step 3 of our MmLE method. By Theorem 1, for any pair of rows i, j  $(i \neq j)$ , we have  $E(d'_{i,j}) = md_{i,j} - \gamma h_{i,j} + \gamma k$  where  $d_{i,j}$  and  $h_{i,j}$  are from  $D_p$  and  $d'_{i,j}$  is from D'. For any  $D_p$ where  $n \geq 2s$  and  $s \geq 2$ , it is easy to verify that  $m \geq 1.5\gamma$ . In fact, when k is relatively large,  $m \approx 3\gamma$ . In addition,  $d_{i,j}$  is always no less than  $h_{i,j}$ . Furthermore, in step 2 all  $D_p$  are generated from the same design D by level permutation which doesn't change the Hamming distance for any pair of rows. Thus,  $E(d'_{i,j})$ is dominated by  $d_{i,j}$ .

Zhou and Xu (2014)[ZX14] show that from GMA initial designs, we can get the best distance distribution designs on average by doing all possible level permutations. Thus choosing GMA design D in step 1 can benefit finding maximin design  $D_p$  in step 2. Further, the maximin initial design  $D_p$  can generate D's with best distance distribution on average in step 3. Moreover, by Theorem 2, when all-level permutations are considered (step 2), if the MmLE method starts from OA initial designs D in the first step, the variance of pairwise L-1 distances for all possible D's are minimized. We should also notice that the averages are always constants when balanced initial designs are used. In addition, if generated from OA initials, the derived designs D' can achieve some stratifications on twodimensional subspaces, since we can always collapse the levels in D' back to the initial OAs. For this reason, the average pairwise correlations of the generated designs are robust. Further, when suitable OAs do not exist, choosing GMA designs D which have the sequentially minimized  $A_1$  and  $A_2$  are clearly the best in step 1.

Next, we justify our method from a geometric point of view. To get a spacefilling n-run, k-factor design, a straightforward idea is that we divide the design space equally into n k-dimensional lattices, put one point into each lattice, and adjust every point's position within its lattice. This geometric structure of "one point within each lattice" can be achieved by performing level expansion to the full factorial initial designs. For example, see the LHD (b),(c) and (d) in Figure 1.1 generated by the level expansion process from full factorial design  $D(9, 3^2)$ . These three designs have only one point within each lattice formed by the solid lines, but their positions within the lattices are different. We should notice that by either level permutation or level expansion process, the "one point within each lattice" structure will not be changed, but the distance distribution of the design can be improved. Thus, by our MmLE method with full factorial initials, we can find the design with best distance distribution while keeping the "one point within each lattice" structure, e.g. the design (d) in Figure 1.1.

Furthermore, as a generalization, when  $n < s^k$ , an initial design with most sub-spaces that are full factorials is be ideal for our MmLE method, and GMA designs have such property in many cases. Box and Hunter (1961) [BH61] point out that any *p*-dimensional (p < r) projection of a 2-level regular design with resolution *r* is a full-factorial design. Additionally Chen (1998)[Che98] proves that for a two level regular design,  $\binom{n}{p} - \sum_{j=r}^{p} \binom{n-j}{p-j} A_j(D)$  *p*-dimensional projections  $(p = r, \ldots, \lfloor r + (r-1)/2) \rfloor$ ) are full-factorial designs and the above result can be generalized to fractional factorial designs with any levels of a prime power. Under these cases, since the GMA initials have largest resolutions and sequentially minimizes  $A_j(D)$   $(j = 1, 2, \ldots k)$ , they have the most parts that are full-factorials in *p*-dimensional projection spaces  $(p \leq \lfloor r + (r-1)/2) \rfloor$ ). We should notice that by all possible level permutations, designs' GMA properties keep unchanged while their distance distributions change.

### CHAPTER 4

### **Results and Comparison**

#### 4.1 Construction of Maximin LHDs

First, we compare our MmLE method with an ordinary level expansion (OdLE) method in generating maximin OALHDs. This OdLE method follows the logic of Leary, Bhaskar and Keane (2003)[LBK03] and Tang (1993)[Tan93]. It first randomly selects column-subset designs D from the corresponding saturated OAs, then performs level expansion towards D to generate LHDs, and finally searches the maximin one from all generated LHDs. In order to make a fair comparison, we replace the simulated annealing searching algorithm in Leary, Bhaskar and Keane (2003)[LBK03] with the more efficient TA algorithm used in this article. Table 4.1 lists some randomly chosen cases for comparison.

For all tables in this article, we use bold font to represent the better results. For "d(pairs)", d represents the smallest L-1 distance among all pairs of rows in the design matrix; "(pairs)" represents the number of pairs of which the distance is d. For the MmLE method here, in the 32, 48, 64 and 128 runs cases, we start from the respective 2-level minimum aberration initial designs by R package "FrF2"; in the 27, 54, 81 and 125 runs cases, we start from initial designs  $OA(27, 3^{13})$ ,  $OA(54, 3^{25})$ ,  $OA(81, 3^{40})$  and  $OA(125, 5^{31})$  respectively. The OdLE method here starts from the corresponding 2-level saturated OAs for the 32, 48, 64 and 128 runs cases, 3-level saturated OAs for the 27, 54 and 81 runs cases, and 5-level saturated OAs for the 125 runs cases. All OAs are from R-package "DoE.base".

Codes are run in R on a ThinkPad X1 laptop with a Intel 2.50GHz I7 CPU. Time used by MmLE method ranges from 5 minutes to half an hour for different cases here. For all cases, the OdLE method uses at least twice as much time as the MmLE method.

MmLE			OdLE	SLHD	OML	.HD	
n	k	d(pair)	$\psi_p$	d(pair)	d(pair)	d(pair)	$\psi_p$
27	9	$\boxed{72(2)}$	0.012	68(5)	55(1)	60(1)	0.025
32	20	205(1)	0.005	205(2)	184(1)	177(1)	0.012
48	31	476(4)	0.0026	474(3)	437(1)	409(1)	0.0067
54	20	329(1)	0.0034	317(1)	287(1)	279(1)	0.0083
54	25	425(3)	0.0022	399(1)	373(1)	360(1)	0.012
64	20	378(1)	0.0034	369(1)	332(1)	310(1)	0.0084
64	40	813(1)	0.0025	804(1)	771(1)	698(1)	0.0048
81	25	604(2)	0.0022	577(1)	516(1)	504(1)	0.0028
81	40	1016(1)	0.0016	962(1)	919(1)	899(1)	0.0016
125	23	797(1)	0.0021	640(1)	726(1)	668(1)	0.0021
125	31	1126(1)	0.0014	971(1)	1025(1)	955(1)	0.0076
128	49	1893(1)	0.0014	1873(1)	1801(1)	1643(1)	0.0057
128	64	2512(1)	0.0017	2479(1)	2497(1)	2239(1)	0.0061

Table 4.1: Comparisons of constructions of maximin LHDs

From Table 4.1, it's clear that the MmLE method generates better OALHDs than the OdLE method for all comparable cases. Compared with the MmLE method, OdLE method only includes the step 3, but does not have the step 1 and step 2 of the MmLE method. Thus, Table 4.1 also shows the usefulness of the first and second steps in the MmLE method which provide the optimal initial designs for the level expansion process in the step 3. We should notice that for 2-level initial designs, the step 2 can be skipped since level permutations do not change the design. Therefore, from the 32, 48, 64 and 128 run cases we can see the usefulness of step 1 alone. From the 54-run/25-factor, 81-run/40-factor and 125-run/31-factor cases, we can observe the usefulness of step 2 alone. From the rest cases, we can see the effects of the step 1 and 2 together.

Next, we compare our MmLE method with the R package "SLHD" by S Ba and the OMLHD method by Joseph and Hung (2008)[JH08] in generating spacefilling LHDs. The R package "SLHD" is available at https://cran.r-project. org/web/packages/SLHD/index.html. It generates maximin LHDs when setting option t = 1. Joseph and Hung (2008)[JH08] propose a multi-objective criterion  $(\psi_p)$ 

$$\psi_p = \omega \rho^2 + (1 - \omega) \frac{\phi_p - \phi_{p,lb}}{\phi_{p,ub} - \phi_{p,lb}} , \qquad (4.1)$$

where  $\phi_p$  is defined in (1.1) with p = 15,  $\rho^2$  is the average column-wise correlation,  $\omega$  is the weight which is set to 0.5,  $\phi_{p,lb}$  and  $\phi_{p,ub}$  are the smallest and largest possible  $\phi_p$  values. Joseph and Hung (2008)[JH08] argue that designs with smaller  $\psi_p$  values are more space-filling. They also provide a modified simulated annealing algorithm to search the OMLHDs which have the smallest  $\psi_p$  values for given nand k. Table 4.1 lists some randomly chosen cases for comparison. We apply R package "SLHD" in software R and generate OMLHDs by applying the C++ code from Ying Hung's homepage. For all cases in Table 4.1, the time used by either the SLHD method or the OMLHD method is at least three times as much as the time used by our MmLE method.

From Table 4.1, it's clear that in all cases the MmLE method generates better LHDs than both the SLHD method and the OMLHD method towards the maximin distance criterion. Moreover, designs by our MmLE method also have better  $\psi_p$ values compared with those by the OMLHD method. We should notice that unlike the OMLHD method searching towards  $\psi_p$  directly, our MmLE method searches towards  $\phi_p$  criterion alone. In fact, designs by the MmLE method are born with small and robust pairwise correlations, since they are generated from OAs via level expansion and inherit orthogonality in big picture. In addition, it's clear that our MmLE method saves great computations compared with the OMLHD method, since the C++ language (used in OMLHD method) on Visual Studio runs much faster than R language (used in MmLE method) by nature. Above all, our MmLE method not only generates better LHDs, but also saves tremendous computations compared with the SLHD method and the OMLHD method.

#### 4.2 Construction of Maximin Factorial Designs

We first compare our MmLE method with the level permutation (LP) method by Zhou and Xu (2014)[ZX14] in generating maximin fractional factorial designs (FFDs). Zhou and Xu (2014)[ZX14] include a table of 10 maximin designs with  $n \leq 32$  that are comparable here. In Table 4.2(a) we list these small designs as the first ten cases. We also select another eight larger cases with  $n \geq 48$  to compare these two methods in Table 4.2 (b). All designs are 4-level FFDs. For the MmLE method, in the 16, 32, 64, and 128 run cases, 2-level MA initial designs are used; in the 48 and 80 run cases,  $OA(48, 2^{47})$  and  $OA(80, 2^{79})$  are used as the initials. For the LP method, in the 48, 64, 80 and 128-run cases, initial designs  $OA(48, 4^{13})$ ,  $OA(64, 4^{11})$ ,  $OA(80, 4^{11})$  and  $OA(128, 4^{40})$  are used respectively. Both methods' codes are run in R. For all cases, the LP method uses at least twice as much time as the MmLE method.

In Table 4.2 (a), for the first 9 cases both methods generate designs with the same smallest pairwise distances. For the last case in Table 4.2 (a) and all cases in Table 4.2 (b), the MmLE method generates better designs with larger smallest pairwise distances compared with the LP method. Furthermore, the LP method relies on existing OA initials that have the same number of runs, factors and levels as the generated designs. However, such OAs are often difficult to find or even do not exist. For example, there is no  $OA(24, 6^8)$  can be used to generate maximin

		(a)		_				
		MmLE	LP	-			(b)	
n	k	d(paris)	d(paris)	-			MmLE	LP
16	3	2(12)	2(12)		n	k	d(paris)	d(paris)
16	4	4(60)	4(56)		48	10	9(6)	8(3)
16	5	4(1)	4(4)		48	13	13(15)	12(10)
32	3	2(156)	2(156)		64	9	8(395)	6(38)
32	4	2(8)	2(8)		64	11	10(77)	9(12)
32	5	4(100)	4(106)		80	7	5(177)	4(48)
32	6	5(48)	5(58)		80	11	9(1)	8(29)
32	7	6(24)	6(28)		128	29	30(42)	29(79)
32	8	8(132)	8(128)		128	40	43(1)	40(2)
32	9	9(62)	8(6)					

Table 4.2: Comparison on the MmLE and LP method for constructing four-levelmaximin designs

 $D'(24, 6^8)$ . Compared with the LP method, our MmLE method has much more flexibility in design size, since we can start from low-level designs to generate highlevel designs. For example, we can start from a 2-level  $OA(24, 2^{23})$  to generate a 24-run and 6-level design with up to 23 factors. Above all, compared with the LP method, our MmLE method can generate better maximin FFDs with more flexibility in design size and within shorter computing time.

Next, we compare the designs from our MmLE method with some existing uniform designs listed on the Uniform Design Homepage (http://uic.edu.hk/ isci/). These uniform designs (UD-page designs) are searched by Kaitai Fang and his collaborators towards CD criterion. In order to make a fair comparison, here we also include another MmLE-CD method. The only difference between the original MmLE and the modified MmLE-CD methods is that in the step 3 of the latter we search designs toward CD criterion. In Table 4.3, we select some 4-level and 6-level designs for comparison. For both methods we start from the initial designs  $OA(32, 2^{31})$ ,  $OA(40, 2^{39})$  and  $OA(48, 2^{47})$  to generate the 4-level designs,  $OA(36, 3^{13})$ ,  $OA(48, 2^{47})$ ,  $OA(54, 3^{18})$  and  $OA(60, 2^{30})$  to generate the 6-level designs for the corresponding cases.

From Table 4.3, it's clear that designs by the MmLE method are always better towards maximin distance criterion compared with the UD-page designs. Further, for all cases here MmLE-CD can generate better CD designs than the respective UD-page designs. Though the Maximin distance and CD criteria are different in expression, both measure some space-filling geometric properties. It's clear that the 3-step procedures of the MmLE method are efficient and generate better space-filling designs compared with the UD-page designs.

			Mm	nLE	MmLE-CD		UD-page designs	
n	k	$\mathbf{S}$	CD	d(pair)	CD	d(pair)	CD	d(pair)
32	7	4	0.074	6(18)	0.070	5(6)	0.071	4(1)
32	13	4	0.0343	13(5)	0.0343	13(5)	0.0344	12(2)
40	13	4	0.3186	13(4)	0.3067	12(5)	0.3068	11(1)
40	15	4	0.5080	16(56)	0.4969	13(1)	0.4987	14(1)
48	11	4	0.1841	10(11)	0.1758	8(1)	0.1767	7(1)
48	15	4	0.461	15(13)	0.447	13(1)	0.449	12(1)
36	12	6	0.1744	20(15)	0.1673	19(14)	0.1691	17(1)
48	12	6	0.1416	19(21)	0.1362	16(2)	0.1374	16(1)
54	9	6	0.0601	12(13)	0.0564	8(1)	0.0568	10(3)
54	12	6	0.1362	17(14)	0.1268	16(1)	0.1299	16(2)
60	9	6	0.0576	12(11)	0.0544	8(1)	0.0546	9(2)

Table 4.3: Comparison with UD-homepage designs

### CHAPTER 5

### Multi-phase Maximin Level Expansion Method

In this part, we briefly introduce the multi-phase MmLE method suitable to generate very large maximin designs. In part 3.1, we introduce the 3-step procedures for the MmLE method which only includes one phase. The multi-phase MmLE method shares the same step 1 and step 2 as the above one-phase MmLE. The difference lies in the step 3: instead of directly generating  $D'(n, (ms)^k)$  from  $D_p(n, s^k)$ , we gradually expand the levels from s to ms by t ( $t \ge 2$ ) phases. For example, in a two-phase (t = 2) MmLE method where  $m_1, m_2$  are integers and  $m_1 * m_2 = m$ , in step 3 we first generate the maximin design  $D_1(n, (m_1s)^k)$  via level expansion from  $D_p$  by step 2; then from  $D_1$  we generate the maximin design  $D_2(n, (m_2m_1s)^k)$  (which is the  $D'(n, (ms)^k)$ ) via level expansion again. By Theorem 1, from maximin initial designs we can generate designs with best distance distributions on average via level expansion. Thus, it's efficient to select the maximin design  $D_1$  in the first phase to be the initial design to generate  $D_2$  in the second phase. It's straight forward to generalize and justify this process for more phases.

The more phases (larger t) we use, the more restrictions are put on the searching space. In practice, choosing the number of phases t is a trade-off process. On one hand, the number of designs needed to be compared decreases dramatically for every one more phase. For example, to generate  $D'(16, 8^2)$  from  $D(16, 2^2)$ , for the one-phase MmLE method, we totally have  $4 * 10^{13}$  possible D's to be compared; while, for the two-phase MmLE method, we only need to compare  $1.7*10^6$  designs.

On the other hand, more restrictions on the searching space also means that we are more likely to miss good results. In Table 5.1, we compare the one-phase and two-phase MmLE method in generating maximin LHDs. For both methods, we start from the respective full factorial designs for the first 5 cases, and from  $OA(27, 3^{13})$ ,  $OA(32, 2^{31})$ ,  $OA(54, 3^{24})$ ,  $OA(64, 2^{63})$ ,  $OA(75, 5^9)$ ,  $OA(81, 3^{40})$  and  $OA(125, 5^{31})$  for the rest cases respectively. The time is recorded in seconds and "sequence" in Table 5.1 represents the level expansion path from s to n for the two-phase MmLE method.

		One-pl	nase	Two-phase		
n	k	d(pair)	time	d(pair)	time	sequence
32	5	37(1)	103	37(3)	101	$2 \rightarrow 8 \rightarrow 32$
64	6	83(1)	301	81(1)	306	$2 \rightarrow 8 \rightarrow 64$
27	3	14(4)	67	14(5)	107	$3 \rightarrow 9 \rightarrow 27$
81	4	50(1)	478	50(3)	490	$3 \rightarrow 9 \rightarrow 81$
125	3	38(5)	603	37(3)	950	$5 \rightarrow 25 \rightarrow 125$
27	10	79(1)	150	79(4)	179	$3 \rightarrow 9 \rightarrow 27$
32	15	151(1)	211	150(2)	218	$2 \rightarrow 8 \rightarrow 32$
54	12	173(1)	886	178(2)	806	$3 \rightarrow 9 \rightarrow 54$
54	20	309(1)	1346	322(2)	1275	$3 \rightarrow 9 \rightarrow 54$
64	40	805(1)	1062	813(1)	1026	$2 \rightarrow 8 \rightarrow 64$
75	8	133(1)	889	134(3)	810	$5 \rightarrow 25 \rightarrow 75$
81	40	1005(1)	1479	1014(1)	936	$3 \rightarrow 9 \rightarrow 54$
125	31	1111(1)	2085	1116(1)	1548	$5 \rightarrow 25 \rightarrow 125$

Table 5.1: Comparison of one-phase and two-phase MmLE method

From Table 5.1, we can observe that if run and factor sizes are relatively large, two-phase method is better, e.g. the last 6 cases. Otherwise, one-phase method works slightly better, e.g., the  $2^{nd}$ ,  $5^{th}$  and  $7^{th}$  cases. Generally speaking, when the available computation is limited compared to the design size, it's better to use more phases. In such case, we depend more on our theoretical results and geometrical concern to find the best design. Otherwise, choosing less phases is better.

### CHAPTER 6

### Conclusions

In this article, we propose the maximin level expansion (MmLE) method which can efficiently generate maximin distance LHDs and maximin distance multi-level factorial designs. This method is a combination and improvement of the level permutation method by Zhou and Xu (2014)[ZX14] and the level expansion method by Tang (1993) [Tan93]. The first two steps in our method is to find optimal initial designs for the one or multi-phase level expansion process in step 3. To justify our method, we establish a relationship of the L-1 distance distributions between the initial and generated designs via level expansion. When all possible level permutations are considered, we also give the mean and variance of the pairwise L-1 distances of the generated design in our method. Besides the mathematical justifications, we also show the geometrical interpretations of our method. Since from full factorial initials the expanded designs have the "one point within each lattice" geometric property, when  $n < s^k$  we want to select an initial design which has the most sub-sets that are full factorial. In this article, we focus on generating  $D'(n, (ms)^k)$  from  $OA(n, s_{max}^k)$  with our MmLE method. This procedure is very easy to be generalized for mixed level designs. Starting from a mixed-level initial deign, we can individually set the level expansion path for each factor. Thus in this way, we can generate mixed-level factorial designs. Moreover, when suitable OA initials are not found, the MmLE method works well for many kinds of initial designs, e.g. supersaturated designs, nearly OAs and et cetera. The run, factor and level sizes for designs by the MmLE method are rather flexible. We uses examples to show that our MmLE method outperforms the ordinary level expansion process by Leary, Bhaskar and Keane (2003)[LBK03], the SLHD package by S Ba, the OMLHD method by Joseph and Hung (2008), the level permutation method by Zhou and Xu (2014)[ZX14]. We also find many more space-filling designs compared with the existing uniform designs.

## CHAPTER 7

### Appendix

#### 7.1 Proof of Lemma 1

(a) For  $i \neq j$ , we have:

- 1. when  $x_{i,l} = x_{j,l}$ , min  $d'_{il,jl} = 0$  and max  $d'_{il,jl} = m 1$ .
- 2. when  $x_{i,l} \neq x_{j,l}$ ,  $\min d'_{il,jl} = m(d_{il,jl} 1) + 1$  and  $\max d'_{il,jl} = m(d_{il,jl} 1) + 2m 1$ .

Therefore, the largest possible  $d'_{i,j}$  is

$$max \ d'_{i,j} = (m-1)(k-h_{i,j}) + \sum_{l=1}^{h_{i,j}} (m(d_{il,jl}-1) + 2m-1) = md_{i,j} + k(m-1) \ .$$

The smallest possible  $d'_{i,j}$  is

min 
$$d'_{i,j} = 0 * (k - h_{i,j}) + \sum_{l=1}^{h_{i,j}} (m(d_{il,jl} - 1) + 1) = md_{i,j} - (m - 1)h_{i,j}$$
.

Thus, we have

$$md_{i,j} - (m-1)h_{i,j} \leq d'_{i,j} \leq md_{i,j} + k(m-1)$$

(b) Let  $x_a$  and  $x_b$  be the pair of rows in design D that forms the smallest L-1 pairwise distance  $d_{min}(D)$  (there could be many such pairs). Let  $x'_c$  and  $x'_d$  be the pair of rows in design D' that forms the smallest L-1 pairwise distance  $d_{min}(D')$  (there could be many such pairs).

$$d_{min}(D') = d'_{c,d} \leqslant d'_{a,b} \leqslant md_{a,b} + k(m-1) = md_{min}(D) + (m-1)k .$$

$$d_{min}(D') = d'_{c,d} \ge md_{c,d} - (m-1)h_{c,d} \ge md_{c,d} - (m-1)h_{max}(D)$$
$$\ge md_{a,b} - (m-1)h_{max}(D) = md_{min}(D) - (m-1)h_{max}(D) .$$

Thus, we have:

•

$$md_{min}(D) - (m-1)h_{max}(D) \leq d_{min}(D') \leq md_{min}(D) + (m-1)k$$
.

### 7.2 Proof of Theorem 1

We calculate the probability distribution for  $d_{il,jl}$  with its range from Lemma 1. For  $i \neq j$ ,

1. when  $x_{i,l} = x_{j,l}$ , the probability distribution is:

$$P(d'_{il,jl} = 0) = \frac{m\binom{n/(ms)}{2}}{m(m-1)(n/(ms))^2 + m\binom{n/(ms)}{2}} = \frac{n-ms}{m(n-s)},$$
$$P(d'_{il,jl} = t) = \frac{2(m-t)(n/(ms))^2}{m(m-1)(n/(ms))^2 + m\binom{n/(ms)}{2}} = \frac{2n(m-t)}{m^2(n-s)}$$

for t = 1, 2, ..., m - 1. Thus:

$$E(d'_{il,jl}) = \sum_{t=1}^{m-1} tP(d'_{il,jl} = t) = \frac{n(m^2 - 1)}{3m(n-s)} = \gamma , \qquad (7.1)$$

$$E((d'_{il,jl})^2) = \sum_{t=1}^{m-1} t^2 P(d'_{il,jl} = t) = \frac{n(m^2 - 1)}{6(n-s)} .$$
(7.2)

2. When  $x_{i,l} \neq x_{j,l}$ , define  $d_0 = m(d_{il,jl} - 1) + 1$ . The probability distribution is

$$P(d'_{il,jl} = d_0 + t) = \frac{t+1}{m^2}, \text{ for } t = 0, 1, \dots m-1$$

$$P(d'_{il,jl} = d_0 + t) = \frac{2m - t - 1}{m^2}$$
, for  $t = m - 1, m, \dots 2m - 2$ .

It's straight forward to verify that

$$\sum_{t=0}^{2m-2} tP(d'_{il,jl} = d_0 + t) = \sum_{x=1}^{m-1} (m-x) \left( \frac{m+x-1}{m^2} + \frac{m-x+1}{m^2} \right) = m-1$$

$$\sum_{t=0}^{2m-2} t^2 P(d'_{il,jl} = d_0 + t) = \sum_{x=1}^{m-1} (m-x) \left( \frac{(m+x-1)^2}{m^2} + \frac{(m-x)(m-x+1)}{m^2} \right)$$

$$= \frac{(m-1)(7m-5)}{6}.$$

Then we have the following for the level expanded designs:

$$E(d'_{il,jl}) = \sum_{t=0}^{2m-2} (d_0 + t) P(d'_{il,jl} = d_0 + t) = d_0 + \sum_{t=0}^{2m-2} t P(d'_{il,jl} = d_0 + t) = m d_{il,jl} ,$$
(7.3)

$$E((d'_{il,jl})^2) = \sum_{t=0}^{2m-2} (d_0 + t)^2 P(d'_{il,jl} = d_0 + t) = d_0^2 + 2d_0 \sum_{t=0}^{2m-2} t P(d'_{il,jl} = d_0 + t) + \sum_{t=0}^{2m-2} t^2 P(d'_{il,jl} = d_0 + t) = m^2 d_{il,jl}^2 + \frac{m^2 - 1}{6} .$$
(7.4)

Note that  $d_{il,jl} = 0$  when  $x_{i,l} = x_{j,l}$ . Combining (7.1) and (7.3), we have:

$$E(d'_{i,j}) = \sum_{l=1}^{k} E(d'_{il,jl}) = \sum_{l=1}^{k} md_{il,jl} + (k - h_{i,j})\gamma$$

Thus, we have

$$E(d'_{i,j}) = md_{i,j} + (k - h_{i,j})\gamma .$$
(7.5)

Next, combining (7.2) and (7.4), we have:

$$E\left(\sum_{l=1}^{k} (d'_{il,jl})^2\right) = \sum_{l=1}^{k} E((d'_{il,jl})^2) = m^2 \sum_{l=1}^{k} d^2_{il,jl} + \frac{m^2 - 1}{6} h_{i,j} + (k - h_{i,j}) \frac{n(m^2 - 1)}{6(n - s)}$$
(7.6)

Further, we have

$$E((d'_{i,j})^2) = E\left(\left(\sum_{l=1}^k d'_{il,jl}\right)^2\right) = E\left(\sum_{l=1}^k (d'_{il,jl})^2\right) + E\left(\sum_{p\neq q=1}^k d'_{ip,jp}d'_{iq,jq}\right).$$
(7.7)

For the latter part, note that  $d'_{ip,jp}$ ,  $d'_{iq,jq}$   $(p \neq q)$  are independently determined by the p, q columns in the original design D. Note that  $d_{il,jl} = 0$  when  $x_{i,l} = x_{j,l}$ . Combining (7.1) and (7.3), we have:

$$E\left(\sum_{p\neq q=1}^{k} d'_{ip,jp} d'_{iq,jq}\right) = \sum_{p\neq q=1}^{k} E(d'_{ip,jp}) E(d'_{iq,jq}) = m^{2} \sum_{p\neq q=1}^{k} d_{ip,jp} d_{iq,jq} + 2(k-h_{i,j})\gamma \sum_{l=1}^{k} m d_{il,jl} + (k-h_{i,j})(k-h_{i,j}-1)\gamma^{2}.$$
(7.8)

Combining (7.6), (7.7), (7.8), we have:

$$E((d'_{i,j})^2) = m^2 d^2_{i,j} + (k - h_{i,j})^2 \gamma^2 + 2m\gamma d_{i,j}(k - h_{i,j}) - \gamma^2 (k - h_{i,j}) - \frac{(m^2 - 1)}{6(n - s)} (sh_{i,j} - nk) ,$$

Then, we have:

$$Var(d'_{i,j}) = E\left((d'_{i,j})^2\right) - \left(E(d'_{i,j})\right)^2$$
$$= -\gamma^2(k - h_{i,j}) - \frac{s(m^2 - 1)}{6(n - s)}h_{i,j} + \frac{n(m^2 - 1)}{6(n - s)}k$$
$$= C_{1,0} + C_{1,1}h_{i,j}$$

where  $C_{1,0} = \frac{n(m^2-1)(m^2n+2n-3m^2s)}{18m^2(n-s)^2}k$  and  $C_{1,1} = \frac{m^2-1}{18m^2(n-s)^2}(2m^2n^2-2n^2-3sm^2n+3m^2s^2).$ 

Note that  $n \ge ms \ge 2s$  and  $m \ge 2$ . When  $n \ge 3s$ ,  $2m^2n^2 - 2n^2 - 3sm^2n + 3m^2s^2 = (m^2 - 2)n^2 + m^2n(n - 3s) + 3m^2s^2 > 0$ ; when n = 2s with m = 2,  $2m^2n^2 - 2n^2 - 3sm^2n + 3m^2s^2 = 12s^2 > 0$ . Thus  $C_{1,1} > 0$  in both cases.

#### 7.3 Proof of Theorem 2

As stated in part 2, from an initial design D, by all possible level permutations we can generate  $N_2$  possible designs (denote as  $\sigma(D)$  here). From each  $\sigma(D)$ , by level expansion we can generate  $N_1$  possible designs (denote as  $\pi(\sigma(D))$  here). Denote  $E_{\sigma}$  as the expectation towards designs generated by all possible level permutations. Denote  $E_{\pi}$  as the expectation towards designs generated by level expansion. Denote  $E_{\Theta}(d'_{i,j})$  as the expectation over all  $N_1 * N_2$  possible designs in set  $\Theta$  as defined in part 2. To simplify notations, we use  $d'_{i,j}$  to represent the L-1 distance in designs  $\pi(\sigma(D))$ ; use  $d^{\sigma}_{i,j}$  and  $h^{\sigma}_{i,j}$  to represent the L-1 and Hamming distances in designs  $\sigma(D)$ . Design D and  $\sigma(D)$  have the same Hamming distance distribution  $(h^{\sigma}_{i,j} = h_{i,j})$  and generalized word-length pattern.

For all balanced designs  $\pi(\sigma(D))$  with *n*-run, *k*-factor and *ms*-level, the average of all pair-wise L-1 distances are constants.

$$\sum_{i \neq j=1}^{n} E_{\Theta}(d'_{i,j}) = 2k \frac{n}{ms} \sum_{i=1}^{ms-1} i(ms-i) = \frac{kn^2(m^2s^2-1)}{3ms} .$$
 (7.9)

Thus we have:

$$\sum_{\substack{i\neq j=1}}^{n} E_{\Theta}((d'_{i,j})^2) = \sum_{\substack{i\neq j=1}}^{n} E_{\pi} E_{\sigma}((d'_{i,j})^2) = \sum_{\substack{i\neq j=1}}^{n} E_{\sigma} E_{\pi}((d'_{i,j})^2)$$
$$= m^2 \sum_{\substack{i\neq j=1}}^{n} E_{\sigma}(d^{\sigma}_{i,j})^2 + \sum_{\substack{i\neq j=1}}^{n} (k-h_{i,j})^2 \gamma^2 + 2mk\gamma \sum_{\substack{i\neq j=1}}^{n} E_{\sigma}(d^{\sigma}_{i,j}) - 2m\gamma \sum_{\substack{i\neq j=1}}^{n} E_{\sigma}(d^{\sigma}_{i,j})h_{i,j}$$
$$- \gamma^2 \sum_{\substack{i\neq j=1}}^{n} (k-h_{i,j}) - \frac{s(m^2-1)}{6(n-s)} \sum_{\substack{i\neq j=1}}^{n} h_{i,j} + \frac{n^2(n-1)(m^2-1)}{6(n-s)}k . \quad (7.10)$$

By Lemma 2, we can easily get the following equations for the balanced design D where  $A_1 = 0$ :

$$\sum_{i \neq j=1}^{n} h_{i,j} = \frac{kn^2(s-1)}{s} , \qquad (7.11)$$

$$\sum_{i \neq j=1}^{n} (k - h_{i,j}) = \frac{kn(n-s)}{s} , \qquad (7.12)$$

$$\sum_{i \neq j=1}^{n} h_{i,j}^2 = \frac{n^2}{s^2} \{ 2A_2(D) + (s-1)k[1 + (s-1)k] \} .$$
 (7.13)

$$\sum_{i \neq j=1}^{n} (k - h_{i,j})^2 = \frac{n^2}{s^2} (2A_2(D) + k(k+s-1)) - nk^2 .$$
 (7.14)

By Lemma 3, for a balanced design we have:

$$\sum_{i \neq j=1}^{n} E_{\sigma}(d_{i,j}^{\sigma}) = n \sum_{l=1}^{(s-1)k} l \overline{B_l(D)} = \frac{kn^2(s^2 - 1)}{3s} , \qquad (7.15)$$

$$\sum_{i \neq j=1}^{n} E_{\sigma}(d_{i,j}^{\sigma})^2 = n \sum_{l=1}^{(s-1)k} l^2 \overline{B_l(D)} = n^2 (C_{0,0} + 2\left(\frac{s+1}{3s}\right)^2 A_2(D)) .$$
(7.16)

By all possible level permutations of D, if the two positions  $x_{i,l}$  and  $x_{j,l}$  have the same level, they are always the same  $(d_{il,jl} = 0)$ . If they are different, the average of all cases is:

$$E_{\sigma}(d_{il,jl}) = \frac{2 * (1 * (s-1) + 2 * (s-2) + \ldots + (s-1) * 1)}{s(s-1)} = \frac{s+1}{3}.$$

Since all possible level permutations do not change the Hamming distance  $h_{i,j}$  between the  $i^{th}$  and  $j^{th}$  row, given  $h_{i,j}$ ,

$$E_{\sigma}(d_{i,j}^{\sigma}) = \sum_{l=1}^{k} E_{\sigma}(d_{il,jl}) = 0 * (k - h_{i,j}) + \frac{s+1}{3}h_{i,j} = \frac{s+1}{3}h_{i,j}.$$

Thus, together with (7.13) we have:

$$\sum_{i \neq j=1}^{n} E(d_{i,j}^{\sigma})h_{i,j} = \frac{s+1}{3} \sum_{i \neq j=1}^{n} h_{i,j}^{2}$$
$$= \frac{(s+1)n^{2}}{3s^{2}} \{2A_{2}(D) + (s-1)k[1+(s-1)k]\}.$$

Combining (7.10), (7.11), (7.12), (7.14), (7.15) and (7.16), we have:

$$\sum_{i \neq j=1}^{n} E_{\Theta}((d'_{i,j})^2) = C_{2,0} + C_{2,1}A_2(D),$$

where

$$\begin{split} \frac{C_{2,0}}{n^2} &= m^2 C_{0,0} + \frac{k\gamma^2}{ns^2} (nk + ns - n - ks^2) + \frac{2mk^2\gamma(s^2 - 1)}{3s} \\ &- \frac{2mk\gamma(s^2 - 1)}{3s^2} (1 + sk - k) - \frac{k(n - s)\gamma^2}{ns} + \frac{k(m^2 - 1)}{6(n - s)} (n - s) \;, \end{split}$$

and

$$\frac{C_{2,1}}{2n^2} = m^2 \left(\frac{s+1}{3s}\right)^2 + \frac{n^2(m^2-1)^2}{9m^2s^2(n-s)^2} - \frac{2n(m^2-1)(s+1)}{9s^2(n-s)}$$
$$= \frac{s+1}{9s^2(n-s)} \{m^2[(s-1)n-s(s+1)]+2n\} + \frac{n^2(m^2-1)^2}{9m^2s^2(n-s)^2} .$$

Since  $n \ge ms \ge 2s$ , we have  $(s-1)n - s(s+1) \ge 2s(s-1) - s(s+1) = s(s-3)$ . Thus, when  $s \ge 3$ , we have  $C_{2,1} > 0$ . When s = 2, it's easy to calculate that  $C_{2,1} > 0$  for all  $n \ge ms$ .

#### References

- [BBM14] Shan Ba, William A Brenneman, and William R Myers. "Optimal Sliced Latin Hypercube Designs." *Technometrics*, 2014.
- [BH61] George EP Box and J Stuart Hunter. "The  $2^{k-p}$  Fractional Factorial Designs." *Technometrics*, **3**(3):311–351, 1961.
- [BS02] Dizza Bursztyn and David M Steinberg. "Rotation designs: orthogonal first-order designs with higher order projectivity." Applied Stochastic Models in Business and Industry, 18(3):197–206, 2002.
- [Che98] Hegang Chen. "Some projective properties of fractional factorial designs." *Statistics & Probability Letters*, **40**(2):185–188, 1998.
- [CL07] Thomas M Cioppa and Thomas W Lucas. "Efficient nearly orthogonal and space-filling Latin hypercubes." *Technometrics*, **49**(1), 2007.
- [DS90] Gunter Dueck and Tobias Scheuer. "Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing." Journal of Computational Physics, **90**(1):161–175, 1990.
- [FLS05] Kai-Tai Fang, Runze Li, and Agus Sudjianto. Design and Modeling for Computer Experiments. CRC Press, 2005.
- [GKH06] Manfred Gilli, Evis Këllezi, and Hilda Hysi. "A data-driven optimization heuristic for downside risk minimization." Swiss Finance Institute Research Paper, (06-2), 2006.
- [Hic98] Fred Hickernell. "A generalized discrepancy and quadrature error bound." Mathematics of Computation of the American Mathematical Society, 67(221):299–322, 1998.
- [JH08] V Roshan Joseph and Ying Hung. "Orthogonal-maximin Latin hypercube designs." *Statistica Sinica*, **18**(1):171, 2008.
- [JMY90] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. "Minimax and maximin distance designs." Journal of Statistical Planning and Inference, 26(2):131–148, 1990.
- [Kir84] Scott Kirkpatrick. "Optimization by simulated annealing: Quantitative studies." Journal of statistical physics, **34**(5-6):975–986, 1984.
- [LBK03] Stephen Leary, Atul Bhaskar, and Andy Keane. "Optimal Orthogonalarray-based Latin hypercubes." Journal of Applied Statistics, 30(5):585–598, 2003.

- [MBC79] Michael D McKay, Richard J Beckman, and William J Conover. "Comparison of three methods for selecting values of input variables in the analysis of output from a computer code." *Technometrics*, 21(2):239– 245, 1979.
- [MM95] Max D Morris and Toby J Mitchell. "Exploratory designs for computational experiments." Journal of Statistical Planning and Inference, 43(3):381–402, 1995.
- [MW95] Rahul Mukerjee and CF Jeff Wu. "On the existence of saturated and nearly saturated asymmetrical orthogonal arrays." *The Annals of Statistics*, **23**(6):2102–2115, 1995.
- [SLL10] Fasheng Sun, Min-Qian Liu, and Dennis KJ Lin. "Construction of orthogonal Latin hypercube designs with flexible run sizes." *Journal of Statistical Planning and Inference*, 140(11):3236–3242, 2010.
- [Tan93] Boxin Tang. "Orthogonal array-based Latin hypercubes." Journal of the American Statistical Association, 88(424):1392–1397, 1993.
- [XW01] Hongquan Xu and CFJ Wu. "Generalized minimum aberration for asymmetrical fractional factorial designs." Annals of Statistics, pp. 549–560, 2001.
- [YL12] JY Yang and Min-Qian Liu. "Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs." Statist. Sinica, 22:433–442, 2012.
- [ZFN13] Yong-Dao Zhou, Kai-Tai Fang, and Jian-Hui Ning. "Mixture discrepancy for quasi-random point sets." Journal of Complexity, 29(3):283– 301, 2013.
- [ZX14] Yong-Dao Zhou and Hongquan Xu. "Space-filling fractional factorial designs." Journal of the American Statistical Association, 109(507):1134–1144, 2014.