Title
Distance-oriented Space-filling Design Constructions for Gaussian Process Modeling

Author
YIN, Yuhao

Publication Date
2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Distance-oriented Space-filling Design Constructions for Gaussian Process Modeling

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Yuhao Yin

2022

ABSTRACT OF THE DISSERTATION

Distance-oriented Space-filling Design Constructions
for Gaussian Process Modeling

by

Yuhao Yin
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2022
Professor Hongquan Xu, Chair

Computer experiments are increasingly being used to build high-quality surrogate models for complex emulation systems. Space-filling designs, frequently employed for planning computer experiments, help explore the design space uniformly and thus effectively. Among those, maximin distance designs are heavily investigated for its intuitive meaning and asymptotic D-optimality for the Gaussian process modeling when observations are nearly independent. In this thesis, we propose two deterministic approaches to constructing maximin distance designs with flexible sizes and extra favorable structures efficiently, including one-dimensional projection uniformity and mirror-symmetry among design rows. Besides, both classes of designs are nearly column-orthogonal, which guarantees low correlation between factors and improves the identification of linear trend of factorial effects. Meanwhile, we propose a new Bayesian-inspired space-filling criterion for the Gaussian Process modeling by meticulously planning the prior imposed on the correlation parameters, which ensures a better quantification of the significance of different design factors. A systematic procedure is introduced to rigorously select the hyperparameter within and two metaheuristic algorithms together with our novel implementations are presented to search for the corresponding

optimal design in a timely fashion. Furthermore, we illustrate the merits of this newly-introduced criterion in terms of its space-filling properties against other existing measures and Gaussian Process model-fitting performances against extensive simulation functions potentially with many inert factors.

The dissertation of Yuhao Yin is approved.

Arash A. Amini

Weng Kee Wong

Yingnian Wu

Hongquan Xu, Committee Chair

University of California, Los Angeles

2022

This work is completely dedicated to my beloved parents and grandparents.

For their profound and everlasting love

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor Hongquan Xu, for his relentless support and mentorship throughout the course of my Ph.D. study. His respect and tolerance have granted me the most precious opportunity for self-reflection at every stage of the research project, and his knowledge and enthusiasm have encouraged me in any challenging moment of the academic research. This dissertation would have never been possible without his great patience and expertise.

My appreciation also extends to the whole doctoral committee, including Professors Arash A. Amini, Weng Kee Wong and Yingnian Wu. Their graduate courses have considerably broadened my horizons and revealed the beauty of statistics from various perspectives. Moreover, they are truly inspiring in shaping my research methodology and proactive in critiquing my results.

Lastly, I would like to thank my girlfriend and families for their enduring understanding and encouragement along the journey. They make me restore faith in despair and are my eternal spiritual support.

VITA

2013–2017 B.S. in Mathematics, Nankai University, Tianjin, China.

2017–2018 Reader/Graduate Reader, Department of Statistics, UCLA.

2018–2022 Teaching Assistant/Associate/Fellow, Department of Statistics, UCLA.

2020.6–9 Data Scientist Intern, Blizzard Entertainment, Inc., Irvine, CA.

2021.6–9 Machine Learning Engineer Intern, Meta Platforms, Inc., Menlo Park, CA.

CHAPTER 1

Introduction

Computer experiments are increasingly being used to build high-quality surrogate models for intricate emulation systems. A general experimental design for planning computer experiments is the space-filling design, which evenly spreads design points over the design region and thus explores the region efficiently. Representative space-filling designs include the Latin hypercube designs (LHDs) (McKay et al., 1979), minimax and maximin distance designs (Johnson et al., 1990), and uniform designs (Fang et al., 2006; Fang et al., 2018). Orthogonality between factors is another important criterion when designing computer experiments, which guarantees low correlation between factors and improves the identification of linear trend of factorial effects (Ye, 1998). Meanwhile, mirror-symmetry of design points is also deemed as a good merit of a design because it guarantees that main and interaction effects are uncorrelated and can be accurately identified (Ye et al., 2000). Recently, hybrid designs, such as maximin LHDs (Wang, Xiao, et al., 2018; W. Zhou et al., 2020), orthogonal-maximin LHDs (Joseph & Hung, 2008), orthogonal symmetric LHDs (Wang, Sun, et al., 2018), and orthogonal uniform designs (X.-R. Zhang et al., 2020), are also being extensively investigated and constructed to integrate merits of different design criteria.

Santner et al. (2018) comprehensively surveyed many space-filling measures and found that the maximin distance criterion which maximizes the minimal distance among all pairs of points, is preferable to the other criteria. Its corresponding optimal design, maximin distance designs, is asymptotically D-optimal for Gaussian process modeling when observations are nearly independent (Johnson et al., 1990). Metaheuristic algorithms, such as simulated annealing (Joseph & Hung, 2008; Morris & Mitchell, 1995), particle swarm optimization (Chen et al., 2013; Moon et

al., 2011), and the threshold-accepting method (Xiao & Xu, 2018), have been adopted to search for good maximin LHDs. Ba et al. (2015) developed the `SLHD` R package to generate maximin distance sliced LHDs with great efficiency. Moreover, many other algorithms have been proposed to construct maximin LHDs, see Lin and Tang (2015) for a review. However, aforementioned stochastic algorithms are not competent for constructing large designs due to their computational complexity, yet large designs are commonly needed in computer experiments. Consequently, sophisticated systematic construction methods are highly valuable for being able to generate optimal maximin LHDs with large sizes efficiently. Y. Zhou and Xu (2015) proposed to construct maximin LHDs by linear level permutation based on good lattice point sets. Xiao and Xu (2017) proposed methods to construct LHDs with large L_1-distances via Costas' arrays. Wang, Xiao, et al. (2018) constructed a series of maximin LHDs via Williams transformations of good lattice point designs, some of the constructed designs are optimal under the maximin L_1-distance criterion and have small pairwise correlations between columns. He (2019) proposed a new method to construct maximin distance designs from interleaved lattices. W. Zhou et al. (2020) used the rotation method to construct maximin L_2-distance LHDs based on a 2^2 full factorial design and a series of saturated two-level regular designs. Li et al. (2021) proposed an easy-to-use method for constructing maximin distance designs based on some carefully selected small designs.

The maximin distance criterion, however, tends to place a large portion of points at the corners and on the boundaries of the domain, severely undermining its space-filling characteristic on low-dimensional projections and thus making it undesirable in the case when only a few design factors are active. To fill the gap, Joseph et al. (2015) proposed maximum projection designs, which claims to maximize space-filling properties on projections to all subsets of factors. By focusing on two-dimensional projection uniformity, Sun et al. (2019) proposed a new design criterion, called uniform projection criterion. They pointed out that uniform projection designs generated under this criterion scatter points uniformly in all dimensions and have good space-filling properties in terms of distance, uniformity and orthogonality. Moreover, they showed that maximin L_1-equidistant designs are uniform projection designs, and provided a method to construct uniform projection

designs based on good lattice point sets when the number of rows is an odd prime.

In Chapter 2, we will propose a general and easy-to-implement method for generating maximin L_1-distance LHDs. The resulting designs are also Latin squares, which are widely used in designs of experiments and other fields, see for example, Hedayat et al. (1999) and Keedwell and Dénes (2015). Theoretical results show that some of the constructed designs are both maximin L_1-distance and equidistant designs, which means that their pairwise L_1-distances are all equal, and they are also uniform projection designs; while others are asymptotically optimal under the maximin L_1-distance criterion.

In Chapter 3, we will propose a systematic approach to constructing mirror-symmetric maximin L_1-distance designs. The proposed method first constructs a class of maximin balanced designs via a piece-wise linear transformation, akin to the Williams transformation used in Wang, Xiao, et al. (2018) but modified for our purpose. We then rotate the generated balanced designs to obtain LHDs. It is well-known that rotation keeps orthogonality and therefore has been extensively used to construct orthogonal LHDs from regular designs (Pang et al., 2009; Steinberg & Lin, 2006; Wang, Sun, et al., 2018). We will show that rotation can also keep the maximin distance optimality of the generated balanced designs, providing a class of maximin LHDs. In addition, when the design size is relatively big (say, 100), we typically do not need as many levels as in an LHD to learn about the simulation system. In this case, the generated maximin balanced designs (without rotation) can be directly used for designing the experiment.

In Chapter 4, we will introduce a new optimal design criterion, so-called the Bayesian-inspired distance criterion, for the Gaussian Process modeling, following the footsteps of the maximum projection criterion. We meticulously impose a more reasonable (exponential) prior on the correlation parameters and discuss systematic approaches to optimally selecting the hyperparameter within. Intrinsic connections are revealed between this new Bayesian-inspired distance criterion and other existing space-filling ones. Moreover, we present two classes of metaheuristic algorithms, along with our novel implementations, to efficiently search for the corresponding optimal design once the hyperparameter is fully determined. Extensive numerical results are demonstrated for comparing

speed and quality of solutions of different optimization algorithms, as well as illustrating merits of our proposed criterion in a comprehensive manner.

Chapter 5 summarizes the whole dissertation, detailing the contribution of our work within each chapter respectively.

CHAPTER 2

Construction of Maximin L_1-Distance Latin Hypercube Designs

A maximin distance design maximizes the separation distance between design points and is asymptotically D-optimal for Gaussian process modeling when observations are nearly independent (Johnson et al., 1990). A Latin hypercube design (LHD) accommodates as many levels as the design size and therefore allows the study of complex systems. Maximin LHDs that integrate the merits of maximin distance designs and LHDs are commonly used for designing intricate computer experiments.

This chapter proposes a general and systematic method for generating maximin L_1-distance LHDs with large row and column sizes. The resulting designs are also Latin squares, which are widely used in designs of experiments and other fields, see Hedayat et al. (1999) and Keedwell and Dénes (2015) for examples of their applications. The generated Latin squares can be further expanded or pruned to cater to different experimentation demands. Theoretical results show that some of the constructed designs are both maximin L_1-distance and equidistant designs; while others are asymptotically optimal under the maximin L_1-distance criterion. Besides, we provide lower bounds of the L_1-distances of the constructed LHDs for more general cases through numerical computations.

2.1 Notation and Preliminaries

For a positive integer b, let $\mathbb{Z}_b$ denote the set $\{1, \ldots, b\}$. Given any two integers a and b, $\gcd(a, b)$ denotes the greatest common divisor of a and b. If $\gcd(a, b) = 1$, then a is coprime to b. For any

real number r, $\lfloor r \rfloor$ is the integer part of r.

A Latin square of order n is an $n \times n$ square matrix with n^2 entries of n different elements, none of them occurring twice within any row or column of the matrix. An isotopism of a Latin square L permutes the rows, the columns and the elements of L, resulting in another Latin square which is said to be isotopic to L; these two Latin squares belong to the same isotopy class (an isotopy class of Latin squares is an equivalence class for the isotopy relation). A Latin hypercube design, denoted by LHD(N, s), is an $N \times s$ matrix, in which each column is a uniform permutation on $\mathbb{Z}_N$ and all the columns are obtained independently. A Latin square of order n is a special LHD(n, n) when its n different elements are taken from $\mathbb{Z}_n$.

For an integer $q \geq 1$, define $d_q(\boldsymbol{x}, \boldsymbol{y}) = \Big(\sum_{i=1}^{s} |x_i - y_i|^q \Big)^{1/q}$ as the L_q-distance of any two row vectors $\boldsymbol{x} = (x_1, \ldots, x_s)$ and $\boldsymbol{y} = (y_1, \ldots, y_s)$. In this paper, we take $q = 1$. Define the L_1-distance of design D to be

$$d_1(D) = \min\{d_1(\boldsymbol{x}, \boldsymbol{y}) : \boldsymbol{x} \neq \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{y} \in D\}.$$

A maximin L_1-distance design D^* is defined to be the design that satisfies

$$d_1(D^*) = \max_D d_1(D)$$

among all possible candidate designs.

2.2 Construction Method

For a positive integer N, the number of positive integers that are less than and coprime to N is $\phi(N)$, where $\phi(\cdot)$ is the Euler function. It is easy to see that $\phi(N)$ is even for any integer $N > 2$. Define a generator vector $\boldsymbol{h}$ as

$$\boldsymbol{h} = (h_1, \ldots, h_n), \tag{2.1}$$

with $1 = h_1 < \cdots < h_n \leq \lfloor N/2 \rfloor$, and $\gcd(h_i, N) = 1$ for $i = 1, \ldots, n$, $n = \phi(N)/2$. The vector $\boldsymbol{h}$ consists of the first $\phi(N)/2$ elements of the generator vector for the $N \times \phi(N)$ good lattice point sets. Taking $\boldsymbol{h}$ given in (2.1) as the generator vector, we obtain an $n \times n$ square matrix $L = (r_{ij})$ with its (i, j)th element r_{ij} defined by

$$r_{ij} = \min\{h_i * h_j (\text{mod } N),\ N - h_i * h_j (\text{mod } N)\},\ i, j = 1, \ldots, n. \tag{2.2}$$

Lemma 2.1. *The $n \times n$ matrix L constructed in* (2.2) *is a Latin square of order n with n different elements $\{h_1, \ldots, h_n\}$.*

Table 2.1: Latin squares constructed by (2.2) for $N = 11$ and 22.

$N = 11$					$N = 22$				
1	2	3	4	5	1	3	5	7	9
2	4	5	3	1	3	9	7	1	5
3	5	2	1	4	5	7	3	9	1
4	3	1	5	2	7	1	9	5	3
5	1	4	2	3	9	5	1	3	7

Example 2.1. Let $N = 11$ and 22, then $n = \phi(N)/2 = 5$, $\boldsymbol{h} = (1, 2, 3, 4, 5)$ for $N = 11$ and $\boldsymbol{h} = (1, 3, 5, 7, 9)$ for $N = 22$. The Latin squares constructed by (2.2) are listed in Table 2.1.

For the Latin square L constructed by (2.2), replace each of its element h_i with i for $i = 1, \ldots, n$, and denote the obtained matrix as D, then D is both an LHD(n, n) and a Latin square of order n with n different elements in $\mathbb{Z}_n$. The following example shows that design D performs well under the maximin L_1-distance criterion.

Example 2.2. Take the Latin square for $N = 22$ in Table 2.1 as an example. Replace each of its

element h_i with i for $i = 1, \ldots, n$, i.e., $1 \to 1,\ 3 \to 2,\ 5 \to 3,\ 7 \to 4,\ 9 \to 5$, then we have

$$\begin{pmatrix} 1 & 3 & 5 & 7 & 9 \\ 3 & 9 & 7 & 1 & 5 \\ 5 & 7 & 3 & 9 & 1 \\ 7 & 1 & 9 & 5 & 3 \\ 9 & 5 & 1 & 3 & 7 \end{pmatrix} \longrightarrow \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 5 & 4 & 1 & 3 \\ 3 & 4 & 2 & 5 & 1 \\ 4 & 1 & 5 & 3 & 2 \\ 5 & 3 & 1 & 2 & 4 \end{pmatrix}.$$

It is easy to see that the generated matrix is both an LHD$(5, 5)$ and a Latin square of order 5. Furthermore, it can be calculated that the L_1-distances of these two LHD$(5, 5)$'s obtained when $N = 11$ and 22 are both equal to $10 = (5 + 1)5/3$.

For an LHD(N, s), its average pairwise L_1-distance is $(N + 1)s/3$ (Y. Zhou & Xu, 2015). While the minimum pairwise L_1-distance cannot exceed the integer part of the average, hence the upper bound of L_1-distance of any LHD(N, s) is $d_{upper} = \lfloor (N + 1)s/3 \rfloor$. It can be verified that the LHDs obtained in Example 2.2 are maximin L_1-distance designs. Inspired by this, we propose the following method for constructing maximin distance LHDs.

Algorithm 2.1 (Construction of maximin L_1-distance LHD(n, n))**.**

Step 1. For a given integer N, obtain the generator vector $\boldsymbol{h} = (h_1, \ldots, h_n)$ by (2.1), where $n = \phi(N)/2$.

Step 2. Generate the $n \times n$ Latin square L by (2.2), each of its rows and columns is a permutation on $\{h_1, \ldots, h_n\}$.

Step 3. Replace each element h_i in L with i for $i = 1, \ldots, n$, denote the obtained LHD(n, n) by D.

In fact, $N - \boldsymbol{h}$ can also be used as the generator vector in Algorithm 2.1. It can be verified that the obtained design are the same as the one constructed with generator vector $\boldsymbol{h}$.

For the LHD(n, n) D constructed by Algorithm 2.1, the following result shows that it has significantly restricted number of distinct pairwise L_1-distances.

Lemma 2.2. *The pairwise L_1-distances of the LHD(n, n) D generated by Algorithm 2.1 take at most $\lfloor n/2 \rfloor$ different values.*

Example 2.3. For $N = 21$, consider the LHD$(6, 6)$ D constructed by Algorithm 2.1, where

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

Let $\boldsymbol{l}_i$ be its ith row. It is easy to check that

$$\begin{aligned} &d_1(\boldsymbol{l}_1, \boldsymbol{l}_2) = d_1(\boldsymbol{l}_1, \boldsymbol{l}_6) = d_1(\boldsymbol{l}_2, \boldsymbol{l}_3) = d_1(\boldsymbol{l}_3, \boldsymbol{l}_5) = d_1(\boldsymbol{l}_4, \boldsymbol{l}_5) = d_1(\boldsymbol{l}_4, \boldsymbol{l}_6) = 12, \\ &d_1(\boldsymbol{l}_1, \boldsymbol{l}_3) = d_1(\boldsymbol{l}_1, \boldsymbol{l}_4) = d_1(\boldsymbol{l}_2, \boldsymbol{l}_5) = d_1(\boldsymbol{l}_2, \boldsymbol{l}_6) = d_1(\boldsymbol{l}_3, \boldsymbol{l}_4) = d_1(\boldsymbol{l}_5, \boldsymbol{l}_6) = 14, \\ &d_1(\boldsymbol{l}_1, \boldsymbol{l}_5) = d_1(\boldsymbol{l}_2, \boldsymbol{l}_4) = d_1(\boldsymbol{l}_3, \boldsymbol{l}_6) = 18. \end{aligned}$$

The pairwise L_1-distances between rows in D take 3 different values.

In fact, for the LHD(n, n) D constructed by Algorithm 2.1, the number of different values of the pairwise L_1-distances between its rows is far less than $\lfloor n/2 \rfloor$ in most cases. For a given positive integer n, as there may be more than one LHD(n, n) that can be constructed by Algorithm 2.1, Figure 2.1 plots the maximum number of distinct values of the pairwise L_1-distances between different rows among all possible designs for each n ($n \leq 800$). From Figure 2.1, it is easy to see that there are few designs with $\lfloor n/2 \rfloor$ different values of the pairwise L_1-distances; while in most cases, the number of distinct values of the pairwise L_1-distances is far less than $\lfloor n/2 \rfloor$.

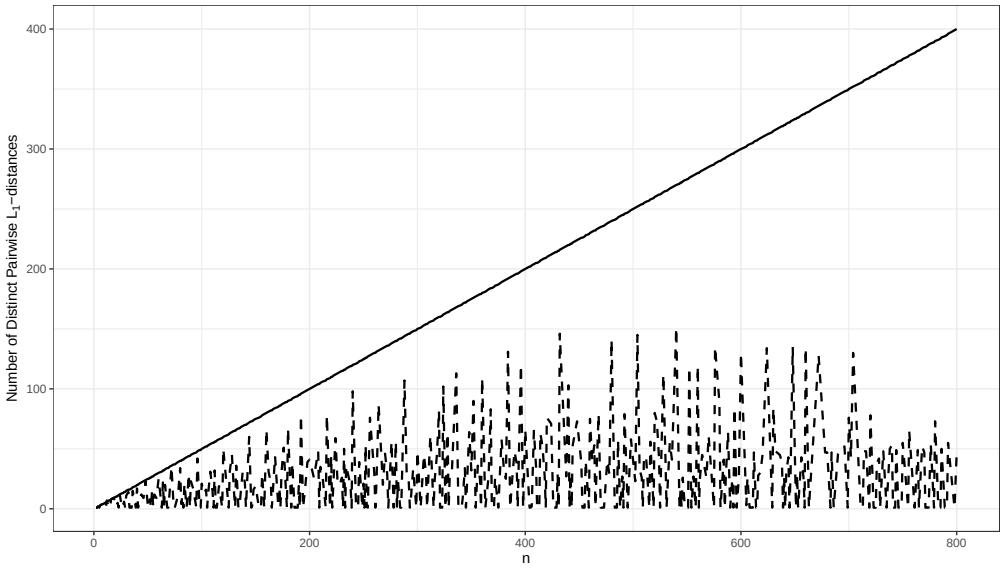

Figure 2.1: Maximum number of different values of the pairwise L_1-distances (dashed line) in the LHD(n, n)'s constructed by Algorithm 2.1, and the corresponding $\lfloor n/2 \rfloor$ (solid line).

For further clarification, consider $11 \leq N \leq 118$, Table 2.2 lists the possible LHD(n, n)'s generated by Algorithm 2.1 with different n values. Define the efficiency of an LHD(N, s) D under the maximin L_1-distance criterion as $d_1(D)/d_{upper}$ with $d_{upper} = \lfloor (N+1)s/3 \rfloor$ (Y. Zhou & Xu, 2015). It is obvious that $d_1(D)/d_{upper} \leq 1$, and a design with larger efficiency is preferable. For situations where $d_1(D)/d_{upper} < 1$, we select the largest $d_1(D)/d_{upper}$, and give the corresponding two smallest N's (if exist) with different $\#\{d_1(\boldsymbol{l}_i, \boldsymbol{l}_j)\}$(number of different pairwise L_1-distances for the same n). From Table 2.2, we can also see that the number of different values of the pairwise L_1-distances is far less than $\lfloor n/2 \rfloor$. And the LHD(n, n)'s constructed by the proposed method perform well under the maximin L_1-distance criterion.

Since there is more than one positive integer N that has the same value of the Euler function $\phi(\cdot)$, for a given positive integer n, there is more than one possible LHD(n, n) constructed by Algorithm 2.1. For further exploring the overall performance of the proposed method under the maximin L_1-distance criterion, Figure 2.2 plots the minimum efficiency for each n ($n \leq 800$). It is straightforward to see that the minimum efficiency of the constructed LHD(n, n) converges to 1, as its number of rows (and columns), i.e. n, becomes larger. That is, the proposed method can be

Table 2.2: Pairwise L_1-distances of the LHD(n, n)'s generated by Algorithm 2.1.

N	n	$\#\{d_1(\boldsymbol{l}_i, \boldsymbol{l}_j)\}$	$d_1(D)$	$d_1(D)/d_{upper}$
11, 22	5	1	10	1
13, 26	6	1	14	1
17, 34	8	1	24	1
19, 38	9	1	30	1
25, 33	10	2, 3	34	0.94
23, 46	11	1	44	1
39	12	4	48	0.92
29, 58	14	1	70	1
31, 62	15	1	80	1
51	16	4	86	0.96
37, 74	18	1	114	1
41, 82	20	1	140	1
43, 86	21	1	154	1
69	22	5	162	0.96
47, 94	23	1	184	1
65	24	8	186	0.93
53, 106	26	1	234	1
81	27	3	244	0.97
87, 116	28	5, 6	262	0.97
59, 118	29	1	290	1

used to generate LHDs with more flexible sizes that have large L_1-distances.

2.3 Theoretical Results and Comparisons

The proposed method generates optimal LHDs under the maximin L_1-distance criterion for different values of N. Next, we will further explore the properties of the LHDs constructed by Algorithm 2.1 in different cases. Throughout the paper, we assume that p is an odd prime.

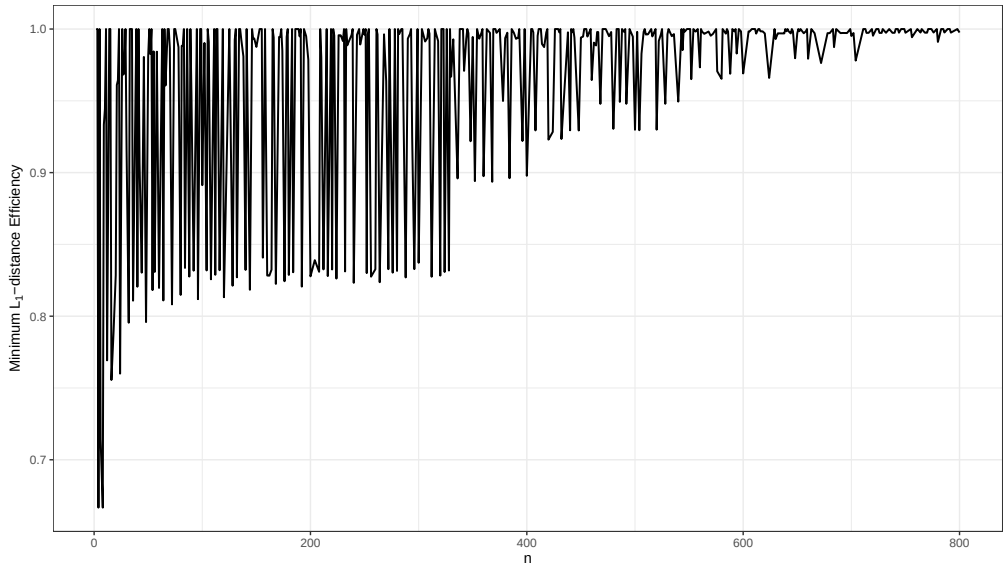

Figure 2.2: Minimum efficiencies of LHD(n, n)'s (solid line) generated by Algorithm 2.1 for general N's, $n = \phi(N)/2$.

2.3.1 $N = p$ and $2p$

When $N = p$ and $2p$, the generator vectors in (2.1) are $\boldsymbol{h} = (1, 2, \ldots, n)$ and $(1, 3, \ldots, 2n-1)$, respectively, where $n = \phi(N)/2 = (p-1)/2$; and it is easy to verify that integer 3 divides n or $n+1$ for $p \geq 5$. The following result holds for design D generated by Algorithm 2.1.

Theorem 2.1. *For $N = p$ or $2p$, $n = \phi(N)/2 = (p-1)/2$, the LHD(n, n) D generated by Algorithm 2.1 is a maximin L_1-distance* LHD*, with its pairwise L_1-distances between rows all equal to $n(n+1)/3$.*

Remark 1. (i) Theorem 2.1 suggests that when N is an odd prime or twice an odd prime, the pairwise L_1-distances of D are all equal to a constant. We name such a design as an *equidistant* LHD, which is obviously a maximin L_1-distance LHD. (ii) Hence, by Theorem 3 in Sun et al. (2019), the constructed designs when $N = p$ and $2p$ are also uniform projection designs, which have good space-filling properties not only in two, but also in all dimensions. (iii) When $N = p$, the LHD(n, n) D is the same as the design H constructed in Wang, Xiao, et al. (2018); then by Theorem 7 in Wang, Xiao, et al. (2018), we have that the average pairwise correlation between

columns of D, denoted by $\rho_{ave}(D)$, satisfies: $\rho_{ave}(D) < 2/(n-1)$.

Example 2.4. For both $N = 13$ and 26, $n = \phi(N)/2 = 6$, the generator vectors are $\boldsymbol{h} = (1, 2, 3, 4, 5, 6)$ and $\boldsymbol{h} = (1, 3, 5, 7, 9, 11)$, respectively. Table 2.3 lists the two LHD$(6, 6)$'s generated by Algorithm 2.1.

Table 2.3: Two LHD$(6, 6)$'s D_1 and D_2 generated by Algorithm 2.1 for $N = 13$ and 26.

D_1						D_2					
1	2	3	4	5	6	1	2	3	4	5	6
2	4	6	5	3	1	2	5	6	3	1	4
3	6	4	1	2	5	3	6	1	5	4	2
4	5	1	3	6	2	4	3	5	2	6	1
5	3	2	6	1	4	5	1	4	6	2	3
6	1	5	2	4	3	6	4	2	1	3	5

It can be calculated that the pairwise L_1-distances between rows of each design are all equal to 14, which implies that both D_1 and D_2 are equidistant and maximin L_1-distance LHDs. In addition, if we permute the rows, columns and elements of D_1 respectively according to the permutation given in (2.3):

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 4 & 5 & 2 & 3 & 6 \end{pmatrix}, \tag{2.3}$$

the obtained design is D_2, that is, D_1 and D_2 are equivalent (i.e., they belong to the same isotopy class). This may not be true in general; see Table 2.2.

Consider the two equidistant LHDs D_1 and D_2 for $N = p$ and $2p$ constructed by Algorithm 2.1, let

$$D^* = [D_1, D_2], \tag{2.4}$$

we have the following result.

Theorem 2.2. *The* LHD$(n, 2n)$ D^* *defined in* (2.4) *is also equidistant and thus a maximin* L_1*-distance LHD, with its pairwise* L_1*-distances between rows all equal to* $2n(n+1)/3$*, where* $n = (p-1)/2$.

Remark 2. The 1st and $(n+1)$th columns in D^* constructed by (2.4) are the same, denote the design obtained by deleting its $(n+1)$th column as D^*_{-1}. Because when we delete one column from an LHD with n rows, its L_1-distance will reduce at most by $n-1$, thus we have that the L_1-distance of the LHD$(n, 2n-1)$ D^*_{-1} satisfies $d_1(D^*_{-1}) \geq (2n^2 - n + 3)/3$. In addition, it is easy to obtain that $d_1(D^*_{-1})/d_{upper} > 1 - 1/(n+1)$, which means D^*_{-1} is an asymptotically optimal LHD, where $d_{upper} = \lfloor (n+1)(2n-1)/3 \rfloor$.

Theorem 2.2 is obvious from the equidistant property of the LHDs constructed by Algorithms 2.1 when $N = p$ and $2p$. Furthermore, if there are more than two equidistant LHDs with the same number of rows, larger maximin distance LHD can be generated by concatenating columns in a similar way, which is also equidistant.

Example 2.5 (Example 2.4 continued)**.** Consider $p = 13$, the two LHD$(6, 6)$'s D_1 and D_2 generated by Algorithm 2.1 for $N = p$ and $2p$ are listed in Table 2.3. From Theorem 2.1, it is easy to know that they are both equidistant LHDs with $d_1(D_1) = d_1(D_2) = 14$. The corresponding LHD$(6, 12)$ D^* constructed in (2.4) is also equidistant with $d_1(D^*) = 28$, which attains the upper bound of L_1-distance. As the first columns in each of the two designs listed in Table 2.3 are the same, we can obtain an LHD$(6, 11)$ D^*_{-1} by deleting one of the repeated columns, it can be calculated that $d_1(D^*_{-1}) = 23$, which is very close to the corresponding upper bound $d_{upper} = 25$.

For any LHD(n, n) constructed by Algorithm 2.1, by adding a row with n elements all being $n+1$, the obtained design has the same L_1-distance as the original LHD(n, n), and the following result holds.

Lemma 2.3. *Let D be an equidistant LHD(n, n) constructed by Algorithm 2.1 for $N = p$ or $2p$, and D' be the LHD$(n+1, n)$ obtained by adding a row of $(n+1)$'s to D. Then $d_1(D') = d_1(D) = (n+1)n/3$, and*

$$d_1(D')/d_{upper} \geq 1 - 1/(n+2) \to 1 \text{ as } n \to \infty,$$

where $d_{upper} = \lfloor (n+2)n/3 \rfloor$.

Lemma 2.3 is obvious and it shows that D' is an asymptotically optimal design under the maximin L_1-distance criterion. In addition, when we delete any column from an equidistant LHD(n, n) D, its L_1-distance will reduce at most by $n-1$. Repeat this deleting one column at a time procedure multiple times, we have the following result.

Lemma 2.4. *Let D be an equidistant LHD(n, n) constructed by Algorithm 2.1. Deleting its any k_c columns yields an LHD$(n, n - k_c)$, denoted by D'. Then*

$$d_1(D')/d_{upper} \geq 1 - 2k_c/(n - k_c).$$

If k_c is a fixed constant not increasing with n, then $d_1(D')/d_{upper} \to 1$ as $n \to \infty$; that is, designs obtained by deleting columns from an equidistant LHD are asymptotically optimal LHDs with different sizes under the maximin L_1-distance criterion. Similar results also hold for deleting columns from any (asymptotically) optimal design under the maximin L_1-distance criterion.

2.3.2 $N = 2^t$ and $2^t p$

When $N(\geq 16)$ is double even, i.e., $N/2$ is an even integer, according to Lemma 1 in Elsawah et al. (2021), we have $n = \phi(N)/2 = \phi(N/2)$ and n is even. For design D generated by Algorithm 2.1, denote D' as the submatrix of D that consists of its first $n/2$ columns, then the following result can be obtained from Theorem 5 in Elsawah et al., 2021. We omit the proof.

Theorem 2.3. *For any double even integer $N(\geq 16)$, let $D = (l_{ij})$ be the LHD(n, n) generated by Algorithm 2.1, $n = \phi(N)/2$. We have the following results:*

(i) the elements in D satisfy $l_{ij} + l_{i(n+1-j)} = n + 1$ and $l_{ij} + l_{(n+1-i)j} = n + 1$ for any $i, j = 1, \ldots, n$, which implies

$$D = \begin{pmatrix} A_1 & n + 1 - A_2 \\ n + 1 - A_3 & A_4 \end{pmatrix},$$

where A_1 is the $n/2 \times n/2$ leading principal submatrix of D, and A_2, A_3 and A_4 can be obtained from A_1 by reversing the orders of columns, rows and both, respectively;

(ii) denote D' as the $n \times n/2$ submatrix of D that consists of its first $n/2$ columns, i.e. $D' = \begin{pmatrix} A_1 \\ n+1-A_3 \end{pmatrix}$, then D' is an LHD$(n, n/2)$, and

$$d_1(D') = d_1(D)/2.$$

Theorem 2.3 (i) shows that, when $N(\geq 16)$ is double even, the corresponding LHD(n, n) generated by Algorithm 2.1 has a fold-over or mirror-symmetric structure with respect to both rows and columns.

Example 2.6. Consider double even integers $N = 28$ and 32, the corresponding LHD$(6, 6)$ and LHD$(8, 8)$ constructed by Algorithm 2.1 are listed in Table 2.4. Divide each of the two LHDs into four blocks as shown in Table 2.4, then it is easy to verify that property (i) in Theorem 2.3 holds.

Table 2.4: LHD(n, n)'s constructed by Algorithm 2.1 for double even integers $N = 28$ and 32.

D_1: LHD$(6, 6)$ for $N = 28$						D_2: LHD$(8, 8)$ for $N = 32$							
1	2	3	4	5	6	1	2	3	4	5	6	7	8
2	4	6	1	3	5	2	5	8	6	3	1	4	7
3	6	2	5	1	4	3	8	4	2	7	5	1	6
						4	6	2	8	1	7	3	5
4	1	5	2	6	3								
5	3	1	6	4	2	5	3	7	1	8	2	6	4
6	5	4	3	2	1	6	1	5	7	2	4	8	3
						7	4	1	3	6	8	5	2
						8	7	6	5	4	3	2	1

Let D_1' and D_2' be the 6×3 and 8×4 submatrices consisting of the first-half columns of D_1 and D_2 respectively, it can be calculated that $d_1(D_1') = d_1(D_1)/2 = 6,\ d_1(D_2') = d_1(D_2)/2 = 11$.

When $N = 4p$, $n = \phi(N)/2 = p - 1$, the corresponding generator vector $\boldsymbol{h}$ consists of $p - 1$

elements $\{2j-1, j=1,\ldots,p\}\backslash\{p\}$. When $N=2^t, n=\phi(N)/2=2^{t-2}$, the corresponding generator vector $\boldsymbol{h}=(1,3,\ldots,2n-1)$. We have the following results for $N=2^t$ and $4p$.

Theorem 2.4. *Let D be the LHD(n,n) generated by Algorithm 2.1, $n=\phi(N)/2$. The following results hold:*

(i) *if $N=4p, p\geq 5$, then $n=\phi(N)/2=p-1$, and*

$$d_1(D)=\begin{cases} n^2/3, & \text{if } p \pmod 3 = 1, \\ (n^2+2)/3, & \text{if } p \pmod 3 = 2; \end{cases}$$

(ii) *if $N=2^t, t\geq 3$, then $n=2^{t-2}$, and*

$$d_1(D)=(n^2+2)/3.$$

In addition, for both cases, we have $d_1(D)/d_{upper}\geq 1-1/(n+1)$, where $d_{upper}=\lfloor (n+1)n/3\rfloor$.

It is possible to establish similar theoretical results for the constructed LHD(n,n)'s when $N=2^tp\,(t>2)$ with more elaborate arguments, details are omitted here. Figure 2.3 plots the efficiencies of the LHD(n,n)'s generated by Algorithm 2.1 when $N=2^tp$ ($t=3,4$ and $16<p<200$). It is easy to see that the constructed designs perform well under the maximin L_1-distance criterion.

Corollary 2.1. *From Theorems 2.3 and 2.4, the following results hold for the LHD$(n,n/2)$ D':*

(i) *if $N=4p, p\geq 5$, then $n=\phi(N)/2=p-1$, and*

$$d_1(D')=\begin{cases} n^2/6, & \text{if } p \pmod 3 = 1, \\ (n^2+2)/6, & \text{if } p \pmod 3 = 2; \end{cases}$$

(ii) *if $N=2^t, t\geq 4$, then $n=2^{t-2}$, and*

$$d_1(D')=(n^2+2)/6.$$

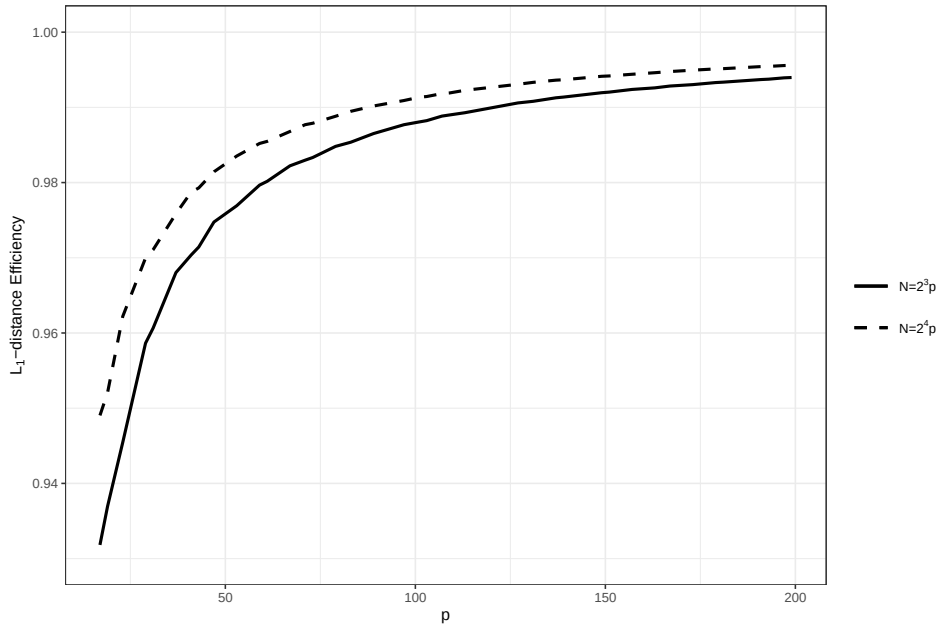

Figure 2.3: Efficiencies of LHD(n, n)'s generated by the proposed method for $N = 2^3p$ (solid line) and 2^4p (dashed line).

As the upper bound of $d_1(D')$ is $d_{upper} = \lfloor (n+1)n/6 \rfloor$, it is easy to verify that $d_1(D')/d_{upper} \to 1$ as $n \to \infty$ for each case listed in Corollary 2.1; that is, the LHD$(n, n/2)$ D' is an asymptotically optimal design under the maximin L_1-distance criterion. More generally, when N is double even, for each LHD(n, n) constructed by the proposed method, the corresponding submatrix that consists of its first $n/2$ columns is asymptotically optimal under the maximin L_1-distance criterion, as long as the LHD(n, n) itself is asymptotically optimal.

In Figure 2.4, we compare the efficiencies of LHD$(p-1, (p-1)/2)$'s generated by the linear permutation good lattice point sets ("LP-GLP", Y. Zhou & Xu, 2015), R package `SLHD` ("SLHD", Ba et al., 2015), and the proposed method ("Our Method") in Algorithm 2.1 for $5 \leq p < 200$. Since the last row of a $p \times (p-1)$ good lattice point set D is $(0, \ldots, 0)$, then the last row of the linear permutation good lattice point set D_b is $(b, \ldots, b)$ for $b = 0, 1, \ldots, p-1$. We utilize the leave-one-out method given in Wang, Xiao, et al. (2018) to generate an LHD$(p-1, p-1)$ based on each D_b, then p LHD$(p-1, (p-1)/2)$'s can be constructed by taking the first $(p-1)/2$ columns of each design. Among these p designs, we choose the one with the largest L_1-distance for comparison.

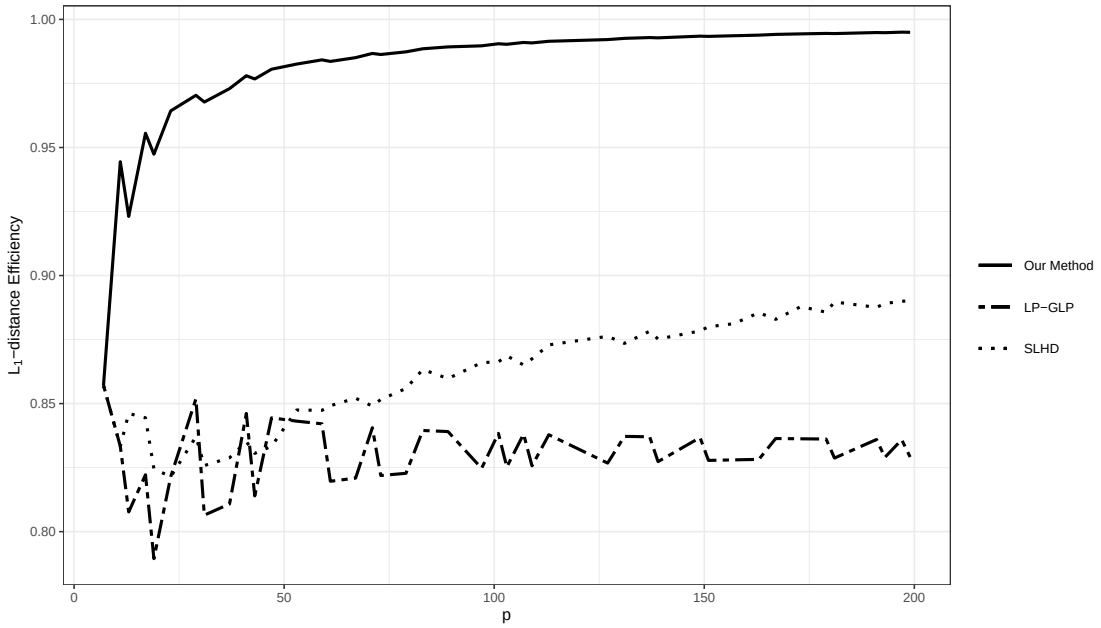

Figure 2.4: Efficiencies of LHD$(p-1,(p-1)/2)$'s generated by different methods: the new method (solid line), LP-GLP (twodash line) and SLHD (dotted line).

The `SLHD` package generates optimal designs under the maximin L_2-distance criterion, so we run the command `maximinSLHD()` with option $t = 1$ and default settings for 100 times, and choose the design with the largest L_1-distance. From the LHD$(p-1, p-1)$ generated by Algorithm 2.1 when $N = 4p$, we choose its first $(p-1)/2$ columns to obtain an LHD$(p-1,(p-1)/2)$, as stated in Theorem 2.3, for comparison. It can be seen from Figure 2.4 that the proposed method outperforms the other two methods as p becomes larger. Moreover, the proposed method generates LHD$(p-1,(p-1)/2)$'s without computer search for any given p.

2.3.3 Numerical studies

In this subsection, we further explore the properties of the LHD(n, n) D obtained from Algorithm 2.1 for more general N with $n = \phi(N)/2$ through simulations.

Figure 2.5 shows efficiencies of the LHD(n, n)'s generated by Algorithm 2.1 for $N = 5p, 7p, 11p$ and $13p$ $(13 < p < 200)$, with $n = 2(p-1), 3(p-1), 5(p-1)$ and $6(p-1)$, respectively.

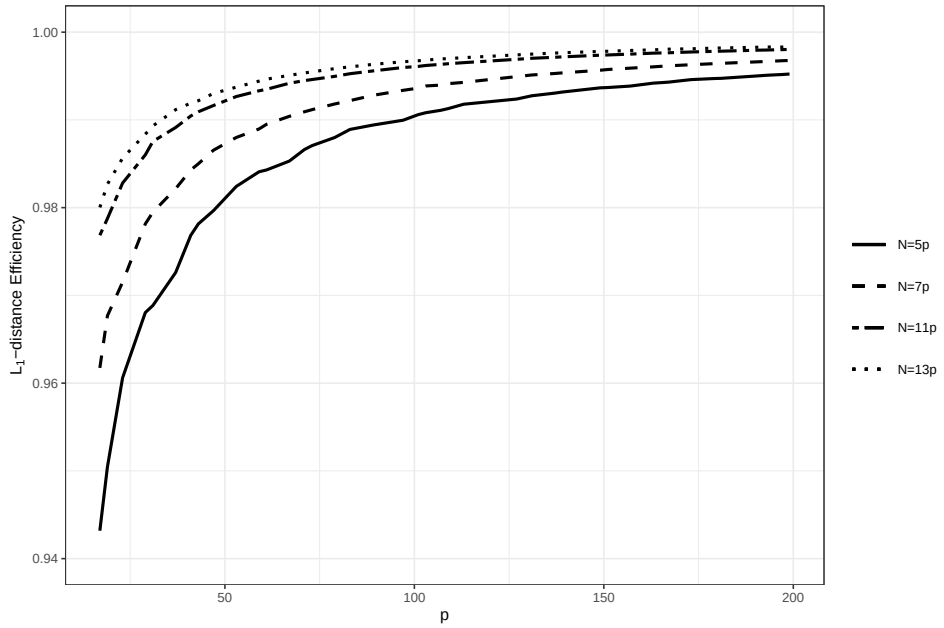

Figure 2.5: Efficiencies of LHD(n, n)'s generated by the proposed method for $N = 5p$ (solid line), $7p$ (dashed line), $11p$ (twodash line) and $13p$ (dotted line).

Figure 2.6 shows efficiencies of the LHD(n, n)'s generated by the proposed method for $N = p^2$ and p^3 ($5 \leq p < 50$), with $n = p(p-1)/2$ and $p^2(p-1)/2$, respectively. It is easy to see that the generated LHDs are all asymptotically maximin L_1-distance designs, and $d_1(D)$ approaches d_{upper} as p becomes larger. In general, more simulations can show that when $N = p_1 p_2$ or p_1^m (p_1, p_2 are odd primes, $m \geq 2$), the LHD(n, n)'s generated by Algorithm 2.1 are all asymptotically optimal designs under the maximin L_1-distance criterion. In addition, more asymptotically optimal LHDs with different sizes can be obtained by deleting columns (see Lemma 2.4) or deleting rows (see Theorem 9 in Wang, Xiao, et al. (2018)) from the constructed LHDs.

Furthermore, we give the following results on the L_1-distance of the constructed LHD(n, n) D for different N values, and we have verified the results up to $p = 1000$.

$$d_1(D) \geq \begin{cases} \lfloor (4p^2 - 10p)/3 \rfloor + 2, & \text{when } N = 5p, n = 2(p-1), \\ 3p^2 - 7p + 6, & \text{when } N = 7p, n = 3(p-1). \end{cases}$$

Through simulations we find that the lower bound is achieved by some N's for either of the two

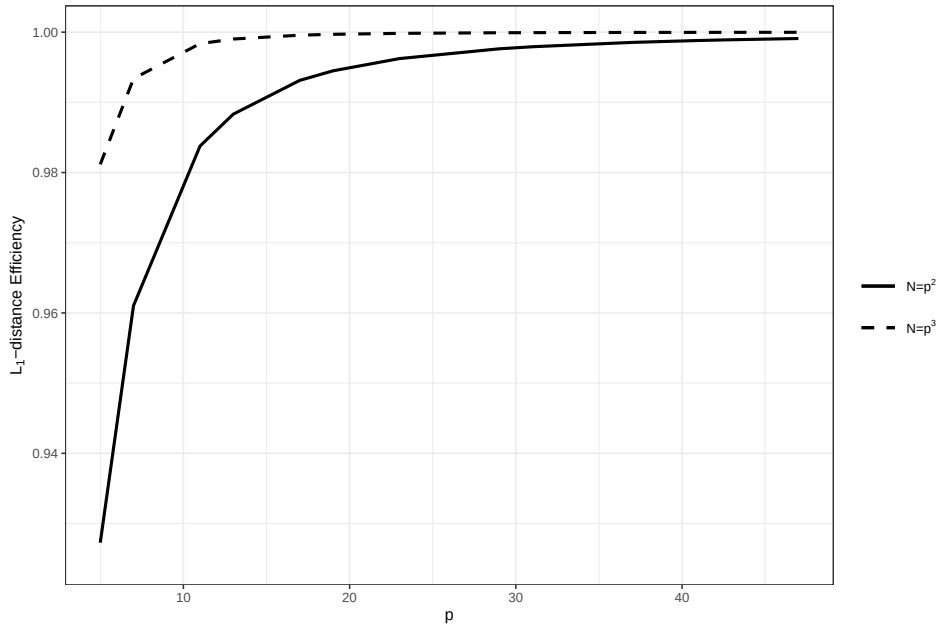

Figure 2.6: Efficiencies of LHD(n, n)'s generated by the proposed method for $N = p^2$ (solid line) and p^3 (dashed line).

cases. Moreover, it can be calculated that the corresponding upper bounds for $N = 5p$ and $7p$ are $d_{upper} = \lfloor (4p^2-6p+2)/3 \rfloor$ and $3p^2-5p+2$ respectively. Thus, the efficiencies of the LHD(n, n)'s generated by Algorithm 2.1 for $N = 5p$ and $7p$ satisfy

$$d_1(D)/d_{upper} > \begin{cases} 1 - 2p/(2p^2 - 3p + 1), & \text{when } N = 5p, n = 2(p-1), \\ 1 - 2p/(3p^2 - 5p + 2), & \text{when } N = 7p, n = 3(p-1), \end{cases}$$

which implies that $d_1(D)/d_{upper} \to 1$ as $n \to \infty$ for design D generated by Algorithm 2.1 when $N = 5p$ or $7p$.

2.4 Concluding Remarks

In this chapter, we propose a general and easy-to-implement method for constructing maximin L_1-distance LHDs. Theoretical results and numerical studies show that the proposed method can be used to generate (asymptotically) optimal LHDs that perform well under the maximin L_1-distance

criterion. When $N = p$ and $2p$, the constructed LHDs are all equidistant LHDs, thus maximin L_1-distance LHDs and uniform projection designs. Moreover, larger equidistant LHDs can be constructed by two or more equidistant LHDs with the same number of rows. Section 2.3.3 provides some lower bounds of the L_1-distances of the constructed LHDs for more general N's through numerical computations, further theoretical supports are possible with more elaborate arguments.

2.5 Appendix: Proofs

Proof of Lemma 2.1. Denote L as $L = (\boldsymbol{r}_1^T, \ldots, \boldsymbol{r}_n^T)^T$, where $\boldsymbol{r}_i$ is the ith row of L and ^{T} is the notation for transpose. It is obvious that $\boldsymbol{r}_1 = \boldsymbol{h}$ and $L^T = L$. Therefore, to prove L is a Latin square, it is sufficient to verify that each $\boldsymbol{r}_i$ $(i = 1, \ldots, n)$ is a permutation on the set $\{h_1, \ldots, h_n\}$.

Let $\boldsymbol{r}_i = (r_{i1}, \ldots, r_{in})$. For $k = 1, \ldots, n$, we have $r_{ik} = \min\{h_i * h_k (\text{mod } N),\ N - h_i * h_k (\text{mod } N)\}$. It is easy to check that $r_{ik} \leq \lfloor N/2 \rfloor$ and $\gcd(r_{ik}, N) = 1$, thus r_{ik} is an element of the set $\{h_1, \ldots, h_n\}$. As $\gcd(h_i, N) = 1$ $(1 \leq i \leq n)$, for any two entries r_{ik} and r_{iw} $(k \neq w)$, it is easy to obtain that $r_{ik} \neq r_{iw}$, otherwise, at least one of the following conditions holds: (1) N divides h_i, (2) N divides $h_k - h_w$, (3) N divides $h_k + h_w$, which leads to a contradiction. Consequently, each $\boldsymbol{r}_i$ is a permutation on the set $\{h_1, \ldots, h_n\}$, which completes the proof. □

To prove Lemma 2.2, we need the following results first. For the LHD(n, n) D constructed by Algorithm 2.1, let $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n$ be its 1st to nth rows, and α_i be the bijection from $\boldsymbol{l}_1 = (1, \ldots, n)$ to $\boldsymbol{l}_i = (l_{i1}, \ldots, l_{in})$ with $\alpha_i(k) = l_{ik}$ for $k = 1, \ldots, n$, $i = 1, \ldots, n$. α_1 is obviously an identity mapping. Further,

Lemma 2.5. *The transformation set $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ is a commutative group.*

Proof. Let $G = \{\alpha_1, \ldots, \alpha_n\}$, G is a commutative group if the following conditions hold:

(C1) if $\alpha, \beta \in G$, then $\alpha\beta \in G$;

(C2) the identity mapping is in G;

(C3) if $\alpha \in G$, then its inverse mapping α^{-1} is in G;

(C4) for any $\alpha, \beta \in G$, the equality $\alpha\beta = \beta\alpha$ holds.

Item (C2) holds obviously as $\alpha_1 (\in G)$ is an identity mapping, so only (C1), (C3) and (C4) need to be verified.

It is easy to see that the elements of Latin square L in (2.2) satisfy $r_{ik} = \min\{\pm h_i * h_k (\mathrm{mod}\ N)\}$. Suppose p is an odd prime, we can prove the lemma in two cases.

(i) When $N = p$ $(p \geq 5)$, $n = (p-1)/2$. The generator vector is $\boldsymbol{h} = (1, \ldots, n)$, thus the design $D = (l_{ij})_{n \times n}$ constructed by Algorithm 2.1 is the same as L. For $i = 1, \ldots, n$, we have

$$\alpha_i(k) = l_{ik} = r_{ik} = \min\{\pm i * k (\mathrm{mod}\ N)\},\ k = 1, \ldots, n.$$

Choose another transformation α_j $(j \neq i)$ from G, then $\alpha_j(k) = \min\{\pm j * k (\mathrm{mod}\ N)\}$ for $k = 1, \ldots, n$. The resultant of α_i and α_j can then be expressed as

$$\begin{aligned} \alpha_j\alpha_i(k) &= \alpha_j(\min\{\pm i * k (\mathrm{mod}\ N)\}) \\ &= \min\{\pm j * i * k (\mathrm{mod}\ N)\} \\ &= \alpha_i\alpha_j(k), \end{aligned}$$

where $k = 1, \ldots, n$, item (C4) holds. Since

$$\begin{aligned} \min\{\pm j * i * k (\mathrm{mod}\ N)\} &= \min\{\pm (j * i (\mathrm{mod}\ N)) * k (\mathrm{mod}\ N)\} \\ &= \min\{\pm \min\{\pm j * i (\mathrm{mod}\ N)\} * k (\mathrm{mod}\ N)\} \\ &= \min\{\pm w * k (\mathrm{mod}\ N)\}, \end{aligned}$$

where $w = \min\{\pm j * i\ (\mathrm{mod}\ N)\} \in Z_n$; it is easy to verify that $\alpha_j\alpha_i(k) = \alpha_i\alpha_j(k) = \alpha_w(k)$, i.e., $\alpha_j\alpha_i \in G$, item (C1) holds.

For each α_i, there exists a unique integer j_0 $(1 \leq j_0 \leq n)$ such that $\min\{\pm j_0 * i (\mathrm{mod}\ N)\} = 1$,

α_{j_0} and α_i satisfy the following equality:

$$\begin{aligned}\alpha_{j_0}\alpha_i(k) = \alpha_i\alpha_{j_0}(k) &= \min\{\pm j_0 * i * k(\text{mod } N)\} \\ &= \min\{\pm \min\{\pm j_0 * i(\text{mod } N)\} * k(\text{mod } N)\} \\ &= k,\end{aligned}$$

where $k = 1, \ldots, n$. Then α_{j_0} is the inverse mapping of α_i, and for each α_i in G, its inverse mapping is also in G, item (C3) holds.

(ii) When $N \neq p$, $n = \phi(N)/2$. From Lemma 2.1, for any two integers i and k $(1 \leq i, k \leq n)$, there exists a unique integer t $(1 \leq t \leq n)$ satisfying

$$h_t = \min\{\pm h_i * h_k(\text{mod } N)\},$$

which means $\alpha_i(k) = t$. In addition, for each h_i, there exists a unique integer i' $(1 \leq i' \leq n)$ such that

$$\min\{\pm h_i * h_{i'}(\text{mod } N)\} = h_1 = 1.$$

Then similar to the discussions in case (i), it easy to verify that items (C1), (C3) and (C4) hold.

In conclusion, G is a commutative group. This completes the proof. □

For any two distinct rows $\boldsymbol{l}_i$ and $\boldsymbol{l}_j (i < j)$ from D, reorder the elements of $\boldsymbol{l}_i$ such that its elements are in the increasing order, i.e., $\boldsymbol{l}_i$ is transformed to $\boldsymbol{l}_1$, correspondingly take the same permutation on the elements of row $\boldsymbol{l}_j$, denote the newly obtained row by $\boldsymbol{l}'_j$. From Lemma 2.5 and the definition of L_1-distance criterion, it is easy to verify that $\boldsymbol{l}'_j$ is still a row of D, and $d_1(\boldsymbol{l}_i, \boldsymbol{l}_j) = d_1(\boldsymbol{l}_1, \boldsymbol{l}'_j)$. Hence, the pairwise L_1-distances between rows in D take at most $n-1$ different values.

Lemma 2.5 also implies that each transformation in the set $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ has an inverse mapping. Suppose $\alpha_i = \alpha_j^{-1}$, then $d_1(\boldsymbol{l}_1, \boldsymbol{l}_i) = d_1(\boldsymbol{l}_1, \boldsymbol{l}_j)$. Therefore, the pairwise L_1-distances of the LHD(n, n) D generated by Algorithm 2.1 take at most $\lfloor n/2 \rfloor$ different values, which are

included in the set $\{d_1(\boldsymbol{l}_1, \boldsymbol{l}_i), 2 \leq i \leq n\}$. This concludes the proof of Lemma 2.2.

Proof of Theorem 2.1. For a given integer N, define $w(x)$ as the modified Williams' transformation in Wang, Xiao, et al. (2018), that is

$$w(x) = \begin{cases} 2x, & \text{if } x < N/2; \\ 2(N-x), & \text{if } x \geq N/2. \end{cases}$$

When $N = p$, the generator vector in (2.1) is $\boldsymbol{h} = (1, \ldots, n)$, where $n = \phi(N)/2 = (p-1)/2$. Hence the LHD(n, n) D generated by Algorithm 2.1 is the same as L in (2.2), and it can be verified that D is also the same as the design H constructed in Wang, Xiao, et al. (2018) by modified Williams' transformation. Then the result follows from Theorem 4 of Wang, Xiao, et al. (2018).

For $N = 2p,\ n = \phi(N)/2 = (p-1)/2$, let $U = (x_{ij})$ be the $N \times \phi(N)$ good lattice point design with generator vector $(1, 3, \ldots, p-2, p+2, \ldots, N-1)$. With proper row and column permutations, U is equivalent to

$$\begin{pmatrix} 2C + p \\ 2C \end{pmatrix} (\text{mod } N)$$

where C is the $p \times (p-1)$ good lattice point design.

Then $w(U)$ is equivalent to

$$\begin{pmatrix} w(2C \oplus p) \\ w(2C) \end{pmatrix},$$

where $2C \oplus p = 2C + p \ (\text{mod } N)$. According to Theorem 1 and proof of Theorem 8 in Wang, Xiao, et al. (2018), the following result holds for the ith and kth rows, denoted by $\boldsymbol{r}_i$ and $\boldsymbol{r}_k$, in $w(2C)$,

$$d_1(\boldsymbol{r}_i, \boldsymbol{r}_k) = \frac{2(p^2-1)}{3}, \text{ for } i \neq k,\ i \neq p, k \neq p, \text{and } i + k \neq p. \tag{2.5}$$

Moreover, it can be verified that (2.5) also holds for $w(2C \oplus p)$.

In addition, when $N = 2p$, it can be verified that for the $n \times n$ Latin square L generated in (2.2),

the following results hold: (i) its n elements are $\{1, 3, \ldots, p-2\}$; thus (ii) the L_1-distance of any two distinct rows in L is two times that of the corresponding rows in LHD(n, n) D constructed by Algorithm 2.1; (iii) under column permutations, L is equivalent to the submatrix of $w(2C \oplus p)/2$ that consists of its $((p+1)/2)$th to $(p-1)$th columns and 1st, 3rd, $\ldots$, $(p-2)$th rows. Hence, according to (2.5) and properties of good lattice point design U, for any two distinct rows in D, their L_1-distance equals

$$\frac{p^2-1}{12} = \frac{(n+1)n}{3},$$

which means that $d_1(D) = (n+1)n/3$, thus the theorem holds. □

Proof of Theorem 2.4. (i) For $N = 4p$, $n = \phi(N)/2 = p-1$, and the corresponding generator vector defined in (2.1) is $\boldsymbol{h} = (1, 3, \ldots, p-2, p+2, \ldots, 2p-1)$. Denote rows of the LHD$(n, n)$ D constructed by Algorithm 2.1 as $\boldsymbol{l}_1, \ldots, \boldsymbol{l}_n$. It is easy to see that the $p-1$ elements of $\boldsymbol{l}_1$ are $l_{1j} = j$ for $j = 1, \ldots, p-1$. For $\boldsymbol{l}_2$, its $p-1$ elements are

$$l_{2j} = \begin{cases} 2 + 3(j-1), & \text{for } j = 1, \ldots, (p-1)/6; \\ (p+1)/2 + 3[j - (p+5)/6], & \text{for } j = (p+5)/6, \ldots, (p-1)/3; \\ (p+5)/2 + 3[(p-1)/2 - j], & \text{for } j = (p+2)/3, \ldots, (p-1)/2; \\ p - 3[j - (p-1)/3], & \text{for } j = (p+1)/2, \ldots, 2(p-1)/3; \\ 3 + 3[j - (2p+1)/3], & \text{for } j = (2p+1)/3, \ldots, 5(p-1)/6; \\ 3[j - 2(p-1)/3] - 1, & \text{for } j = (5p+1)/6, \ldots, p-1, \end{cases}$$

when $p \pmod 3 = 1$, or

$$l_{2j} = \begin{cases} 2 + 3(j-1), & \text{for } j = 1, \ldots, (p+1)/6; \\ (p+3)/2 + 3[j - (p+7)/6], & \text{for } j = (p+7)/6, \ldots, (p+1)/3; \\ (p+5)/2 + 3[(p-1)/2 - j], & \text{for } j = (p+4)/3, \ldots, (p-1)/2; \\ 3 + 3[2(p-2)/3 - j], & \text{for } j = (p+1)/2, \ldots, 2(p-2)/3; \\ 1 + 3[j - (2p-1)/3], & \text{for } j = (2p-1)/3, \ldots, (5p-7)/6; \\ (p+1)/2 + 3[j - (5p-1)/6], & \text{for } j = (5p-1)/6, \ldots, p-1, \end{cases}$$

when $p \pmod 3 = 2$. It can be calculated that $d_1(\boldsymbol{l}_1, \boldsymbol{l}_2) = n^2/3$ (if $p \pmod 3 = 1$) or $d_1(\boldsymbol{l}_1, \boldsymbol{l}_2) = (n^2+2)/3$ (if $p \pmod 3 = 2$).

For $\boldsymbol{l}_3$, it can be verified that its $p-1$ elements are

$$l_{3j} = \begin{cases} 3 + 5(j-1), & \text{for } j = 1, \ldots, n/10; \\ 2 + 5(j-1), & \text{for } j = n/10+1, \ldots, n/5; \\ 4 + 5(2n/5 - j), & \text{for } j = n/5+1, \ldots, 3n/10; \\ 5 + 5(2n/5 - j), & \text{for } j = 3n/10+1, \ldots, 2n/5; \\ 1 + 5(j - 2n/5 - 1), & \text{for } j = 2n/5+1, \ldots, n/2; \\ n + 1 - l_{3(n+1-j)}, & \text{for } j = n/2+1, \ldots, n, \end{cases}$$

when $p \pmod 5 = 1$, and the corresponding L_1-distance $d_1(\boldsymbol{l}_1, \boldsymbol{l}_3) = n^2/3 + 4n/15 > d_1(\boldsymbol{l}_1, \boldsymbol{l}_2)$. Similarly, for p with other values or other rows in D, it can be verified that $d_1(\boldsymbol{l}_1, \boldsymbol{l}_i) \geq d_1(\boldsymbol{l}_1, \boldsymbol{l}_2)$ $(i = 3, \ldots, n)$ via some tedious calculations (details are omitted here). Therefore, the L_1-distance of design D equals the L_1-distance between its first two rows, i.e., $d_1(D) = d_1(\boldsymbol{l}_1, \boldsymbol{l}_2) = n^2/3$ (if $p \pmod 3 = 1$) or $d_1(D) = d_1(\boldsymbol{l}_1, \boldsymbol{l}_2) = (n^2+2)/3$ (if $p \pmod 3 = 2$).

(ii) For $N = 2^t, n = \phi(N)/2 = 2^{t-2}$, the corresponding generator vector is $\boldsymbol{h} = (1, 3, \ldots, 2n-1)$. Results on $d_1(D)$ can be proved similarly via some tedious calculations, so we omit the details.

In addition, for the constructed LHD(n, n) D in both cases, the upper bound of its L_1-distance

is $d_{upper} = \lfloor (n+1)n/3 \rfloor$, it is easy to verify that $d_1(D)/d_{upper} \geq 1 - 1/(n+1)$. This completes the proof. $\square$

CHAPTER 3

Construction of Mirror-Symmetric Maximin L_1-Distance Designs

Mirror-symmetry of design points is a good merit of a design because it guarantees that main and interaction effects are uncorrelated and can be accurately identified (Ye et al., 2000). This chapter proposes a systematic approach to constructing mirror-symmetric maximin L_1-distance LHDs. The proposed method first constructs a class of maximin balanced designs via a piece-wise linear transformation, akin to the Williams transformation used in Wang, Xiao, et al. (2018) but modified for our purpose. We then rotate the generated balanced designs to obtain LHDs. It is well-known that rotation keeps orthogonality and therefore has been extensively used to construct orthogonal LHDs from regular designs (Pang et al., 2009; Steinberg & Lin, 2006; Wang, Sun, et al., 2018). We will show that rotation can also retain the maximin distance optimality of the generated balanced designs, providing a class of maximin LHDs. In addition, when the design size is relatively big (say, 100), we typically do not need as many levels as in an LHD to learn about the simulation system. In this case, the generated maximin balanced designs (without rotation) can be directly used for designing the experiment.

3.1 Notation, Background and Preliminary Results

For an $N \times n$ design matrix $D = (x_{ij})_{1\le i\le N, 1\le j\le n}$, the L_1-distance between the ith and kth rows is defined as $d_{ik}(D) = \sum_{j=1}^{n} |x_{ij} - x_{kj}|$. The L_1-distance of design D, denoted by $d(D)$, is the minimum L_1-distance between any two distinct rows of D, that is, $d(D) = \min\{d_{ik}(D) : i \neq$

$k,\ i,k = 1, \cdots, N\}$. The maximin distance criterion (Johnson et al., 1990) is to maximize $d(D)$ among all possible designs.

An $N \times n$ design with s levels is balanced if all levels appear equally often, that is, each level appears exactly N/s times. In this paper, the levels of the generated designs are denoted by $-(s-1)/2, -(s-3)/2, \ldots, (s-3)/2, (s-1)/2$. An LHD is a special balanced design with $s = N$. A design D is mirror-symmetric if for any row x in D, its mirror-symmetric point, $-x$, is also a row of D.

To define the distance efficiency, we need an upper bound of L_1-distance. For an $N \times n$ balanced design, the average pairwise L_1-distance between rows is $N(s^2-1)n/[3s(N-1)]$ (Y. Zhou & Xu, 2015). Because the minimum pairwise L_1-distance cannot exceed the integer part of the average, we have the following lemma.

Lemma 3.1. *For any $N \times n$ balanced design D with s levels,*

$$d(D) \leq d_{\text{upper}} = \left\lfloor \frac{N(s^2-1)n}{3s(N-1)} \right\rfloor,$$

where $\lfloor x \rfloor$ is the integer part of x.

For an even N, the d_{upper} in Lemma 3.1 is achievable with a balanced mirror-symmetric design, yet for an odd N, this d_{upper} is not achievable. This is because when N is odd, a mirror-symmetric design must include the center $(0, \ldots, 0)$ so that its L_1-distance is restricted. We derive a tight upper bound when N is odd in the following.

Theorem 3.1. *For any $N \times n$ mirror-symmetric balanced design D with s levels, where N is an odd number,*

$$d(D) \leq d^*_{\text{upper}} = \left\lfloor \frac{N(s^2-1)n}{4s(N-1)} \right\rfloor.$$

Clearly, the d^*_{upper} in Theorem 3.1 is smaller than the d_{upper} in Lemma 3.1. Specifically, for an $N \times n$ LHD, $d_{\text{upper}} = \lfloor (N+1)n/3 \rfloor$ in Lemma 3.1, and $d^*_{\text{upper}} = \lfloor (N+1)n/4 \rfloor$ in Theorem 3.1. Theorem 3.1 implies that maximin distance designs cannot be mirror-symmetric when N is odd.

On the other hand, we will show that maximin distance designs can be constructed from mirror-symmetric designs when N is even.

For clarity, we define two types of distance efficiencies, based on Lemma 3.1 and Theorem 3.1, as follows:

$$d_{\text{eff}}(D) = d(D)/d_{\text{upper}} \text{ and } d^*_{\text{eff}}(D) = d(D)/d^*_{\text{upper}}.$$

When N is odd, $d_{\text{eff}}(D)$ is about $3/4$ or less for any mirror-symmetric design D, so we use $d^*_{\text{eff}}(D)$ to assess the distance efficiency among mirror-symmetric designs.

With these results, our goal is to construct balanced designs and LHDs with $d_{\text{eff}}(D)$ and/or $d^*_{\text{eff}}(D)$ close or equal to 1.

3.2 Main Construction Methods

We first introduce a transformation that will be used in the construction. Let p be an odd prime throughout the paper. For $x \in \{0, 1, \ldots, p-1\}$, define

$$\varphi(x) = \begin{cases} 2x, & \text{for } 0 \le x < p/4; \\ -2x + p, & \text{for } p/4 < x < 3p/4; \\ 2x - 2p, & \text{for } 3p/4 < x < p. \end{cases} \tag{3.1}$$

Then φ defines a one-to-one map from $x = 0, \ldots, p-1$ to $-(p-1)/2, \ldots, (p-1)/2$. For example, for $p = 3$, φ maps $(0, 1, 2)$ to $(0, 1, -1)$; for $p = 5$, φ permute the levels $(0, 1, 2, 3, 4)$ to $(0, 2, 1, -1, -2)$. Figure 3.1 shows the cases for $p = 7$ and 9. The permutation is well defined and for any pair of $x_1 \neq x_2$, $\varphi(x_1) \neq \varphi(x_2)$. In fact, for instance, if $0 \le x_1 < p/4$, $p/4 < x_2 < 3p/4$ and $\varphi(x_1) = \varphi(x_2)$, then $p = 2(x_1 + x_2)$, which contradicts with the fact that p is odd. It is easy to see that the permutation has the property

$$\varphi(p - x) = -\varphi(x). \tag{3.2}$$

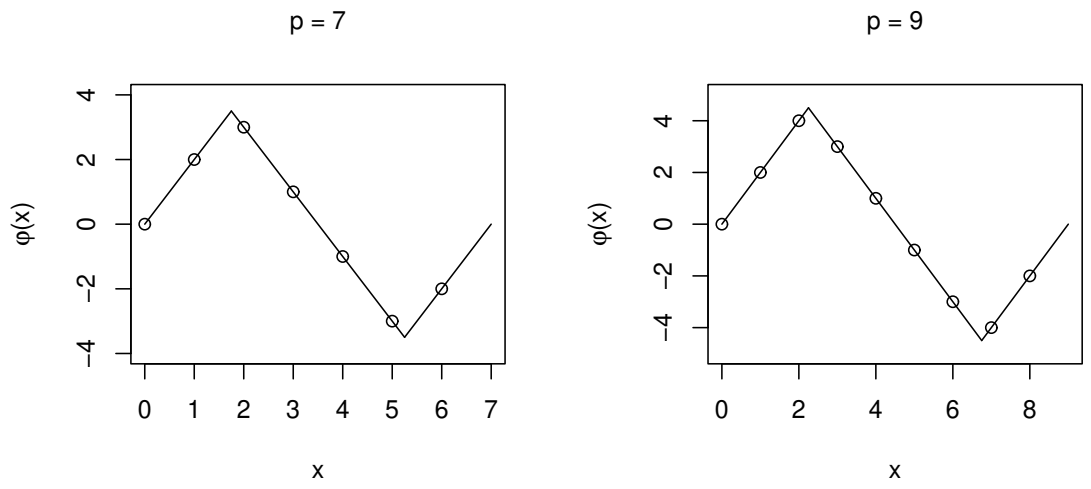

Figure 3.1: The permutation $\varphi(x)$ in (3.1) for $p = 7$ and 9.

It is worth noting that φ is, in fact, a piece-wise linear transformation, similar to the Williams transformation applied to the construction of LHDs in Butler (2001) and Wang, Xiao, et al. (2018). Aside from it, φ is ready to be deployed in our methods directly for constructing mirror-symmetric maximin distance designs. Now we present the main algorithm for constructing mirror-symmetric balanced designs and LHDs via the φ transformation.

Algorithm 3.1 (Construction of balanced designs and LHDs)**.**

Step 1. Let X be a p-level full factorial design with 2 columns:

$$X = \left[\begin{array}{cccc|cccc|c|cccc} 0 & 1 & \cdots & p-1 & 0 & 1 & \cdots & p-1 & \cdots\cdots & 0 & 1 & \cdots & p-1 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & \cdots\cdots & p-1 & p-1 & \cdots & p-1 \end{array}\right]^\top.$$

Step 2. Let $D = (x_{ij})$ be the design obtained by deleting the first column of $XX^\top \pmod p$ and

$$E = \varphi(D) = \varphi(x_{ij}). \tag{3.3}$$

Step 3. Define a rotation matrix

$$T_2 = \begin{bmatrix} p & -1 \\ 1 & p \end{bmatrix},$$

and let $T = \text{diag}\{T_2, \cdots, T_2\}$ *with* T_2 *repeating* $(p^2-1)/2$ *times. Let*

$$L = ET. \tag{3.4}$$

The design E and L generated in Steps 2 and 3 of Algorithm 3.1 both have $N = p^2$ rows and $n = p^2 - 1$ columns. We next show that E is a mirror-symmetric maximin balanced design and L is a mirror-symmetric maximin LHD.

3.2.1 Balanced designs

The design E in (3.3) is balanced and mirror-symmetric. First, E is balanced because D is balanced and φ is a one-to-one map. To see that E is mirror-symmetric, rearrange the order of rows in X as

$$X = [\mathbf{0}_2, G, p - G]^\top \pmod p,$$

where $G = [G_2, 2G_2, \cdots, ((p-1)/2)G_2]$ and

$$G_2 = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & 2 & \cdots & p-1 \end{bmatrix}.$$

Then

$$XX^\top = \begin{bmatrix} 0 & 0 & 0 \\ 0 & G^\top G & p - G^\top G \\ 0 & p - G^\top G & G^\top G \end{bmatrix} \pmod p,$$

and

$$E = \begin{bmatrix} \varphi(0) & \varphi(0) \\ \varphi(G^\top G) & \varphi(p - G^\top G) \\ \varphi(p - G^\top G) & \varphi(G^\top G) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ \varphi(G^\top G) & -\varphi(G^\top G) \\ -\varphi(G^\top G) & \varphi(G^\top G) \end{bmatrix} \tag{3.5}$$

where the second equation in (3.5) holds due to the property (3.2). It is clear that the ith and $(i+(N-1)/2)$th rows of E in (3.5) are mirror-symmetric, for $i = 2, \ldots, (N+1)/2$, and therefore E is mirror-symmetric. The following theorem summarizes these results and the property of E in terms of the L_1-distance and efficiency.

Theorem 3.2. *Let p be an odd prime. The design E constructed in* (3.3) *is a $p^2 \times (p^2-1)$ mirror-symmetric balanced design with p levels. For $i \neq k$,*

$$d_{ik}(E) = \begin{cases} (p-1)p(p+1)/4 & \textit{if } i = 1 \textit{ or } k = 1, \\ (p-1)p(p+1)/2 & \textit{if the } i\textit{th and } k\textit{th rows are mirror-symmetric}, \\ (p-1)p(p+1)/3 & otherwise. \end{cases}$$

Then $d(E) = (p-1)p(p+1)/4$, $d_{\text{eff}}(E) = 0.75$ and $d^*_{\text{eff}}(E) = 1$.

Example 3.1. Let $p = 3$ and X be a 3-level full factorial:

$$X = [\mathbf{0}_2, G, 3 - G]^\top \pmod 3, \text{ where } G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}.$$

The design E generated by Algorithm 3.1 is given by

$$E = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & -1 & 0 & -1 & -1 \\ 0 & 1 & 1 & -1 & 0 & -1 & -1 & 1 \\ 1 & 1 & -1 & 0 & -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 & -1 & 1 & 0 & 1 \\ -1 & 0 & -1 & -1 & 1 & 0 & 1 & 1 \\ 0 & -1 & -1 & 1 & 0 & 1 & 1 & -1 \\ -1 & -1 & 1 & 0 & 1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 & 1 & -1 & 0 & -1 \end{bmatrix}.$$

It can be verified that E is balanced and mirror-symmetric. Furthermore, $d_{1k}(E) = 6$, $d_{ik}(E) = 12$ for $|i-k| = 4$ and $i, k > 1$, that is, for each mirror-symmetric pair of rows, and $d_{ik}(E) = 8$ for all other pairs of rows. Therefore, $d(E) = 6$. By Theorem 3.1, an upper bound of the L_1-distance for a 9×8 mirror-symmetric design with 3 levels is $9 \times (3^2 - 1) \times 8/[4 \times 3 \times (9-1)] = 6$. Hence, $d^*_{\text{eff}}(E) = 1$.

Example 3.1 is a toy example to illustrate the use of Algorithm 3.1. In real application, using a bigger prime p, we can obtain a larger design with many more levels. For example, with $p = 23$, we obtain a 529×528 balanced design with 23 levels each repeating 23 times. These many levels are typically adequate to study the complexity of perturbations within an intricate system.

We can further study the orthogonality of the constructed design E. Define the average pairwise correlation between columns by

$$\rho_{\text{ave}}(E) = \frac{\sum_{i \neq k} |\rho_{ik}(E)|}{n(n-1)}, \tag{3.6}$$

where $\rho_{ik}(E)$ is the correlation coefficient between the ith and kth columns of E. A low ρ_{ave} value indicates small correlation between design factors, and hence reduces the variance of coefficient

estimates for the linear trend of factorial effects.

Theorem 3.3. *For the $p^2 \times (p^2-1)$ design E constructed in Algorithm 3.1,*

$$\rho_{\text{ave}}(E) < \frac{2}{p^2-2} \to 0, \ \ as \ \ p \to \infty.$$

Although the average absolute column-wise correlation of design E converges to 0 as prime number p approaches infinity, there actually exists highly-correlated columns in it. Specifically, $\rho_{i(i+(N-1)/2)}(E) = -1$ for any $1 \le i < N/2$, where $N = p^2$.

A straightforward solution for this is to keep the first half columns of E, denoted as E_{half}, and randomly permute rows of the second half columns, $-E_{half}$. Furthermore, in order to retain the mirror-symmetry property, we propose first to shuffle rows 2 to $(N+1)/2$ and apply the same permutation to rows $(N+3)/2$ to N. This process will help reduce the maximum correlation of the twisted design without detriment to its distance. For instance, the design E in Example 3.1 has $d(E) = 6$, $\rho_{\text{ave}}(E) = 0.1429$, and $\max_{i \neq k} |\rho_{ik}(E)| = 1$. By permuting row indices of columns 5 to 8 from $(1,2,3,4,5,6,7,8,9)$ to $(1,3,5,4,2,7,9,8,6)$ and concatenating with columns 1 to 4, we get a new balanced mirror-symmetric design

$$E' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & -1 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 & -1 & 0 & -1 & -1 \\ 1 & 1 & -1 & 0 & -1 & -1 & 1 & 0 \\ 1 & -1 & 0 & -1 & 0 & -1 & -1 & 1 \\ -1 & 0 & -1 & -1 & 1 & -1 & 0 & -1 \\ 0 & -1 & -1 & 1 & 1 & 0 & 1 & 1 \\ -1 & -1 & 1 & 0 & -1 & 1 & -1 & 0 \\ -1 & 1 & 0 & 1 & 0 & 1 & 1 & -1 \end{bmatrix},$$

with $d(E') = 6$, $\rho_{\text{ave}}(E') = 0.2381$ and $\max_{i \neq k} |\rho_{ik}(E')| = 0.6667$. In general, the new balanced

design E' will have significantly smaller maximum absolute correlation, yet resembling average correlation compared to the original design E. This procedure can be analogously applied to mirror-symmetric LHDs generated in the following sections as well to further decorrelate design columns.

3.2.2 Mirror-symmetric LHDs

The following theorem summarizes the properties of L constructed in (3.4).

Theorem 3.4. *The design L constructed in* (3.4) *is a $p^2 \times (p^2-1)$ mirror-symmetric LHD with*

$$d^*_{\text{eff}}(L) = \frac{d(L)}{d^*_{\text{upper}}} \geq 1 - \frac{p+1}{p^2+1}, \ \ and \ \ \rho_{\text{ave}}(L) < \left(1 + \frac{2}{p}\right) \frac{2}{p^2-2}.$$

As $p \to \infty$, $d^*_{\text{eff}}(L) \to 1$ and $\rho_{\text{ave}}(L) \to 0$.

Example 3.2. Consider generating an LHD based on the design E constructed in Example 3.1. Let $T = \text{diag}\{T_2, T_2, T_2, T_2\}$ where

$$T_2 = \begin{bmatrix} 3 & -1 \\ 1 & 3 \end{bmatrix}.$$

Then

$$L = ET = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -3 & 4 & -2 & -1 & 3 & -4 & 2 \\ 3 & 1 & -2 & -4 & -3 & -1 & 2 & 4 \\ 4 & -2 & -1 & 3 & -4 & 2 & 1 & -3 \\ -2 & -4 & -3 & -1 & 2 & 4 & 3 & 1 \\ -1 & 3 & -4 & 2 & 1 & -3 & 4 & -2 \\ -3 & -1 & 2 & 4 & 3 & 1 & -2 & -4 \\ -4 & 2 & 1 & -3 & 4 & -2 & -1 & 3 \\ 2 & 4 & 3 & 1 & -2 & -4 & -3 & -1 \end{bmatrix}.$$

It can be verified that L is a mirror-symmetric LHD with $d(L) = 20$, $d^*_{\text{eff}}(L) = 1$, and $\rho_{\text{ave}}(L) = 0.1429$.

3.2.3 Leave-one-out extension

Considering that the first row of L in (3.4) is a zero vector, we can delete it and rearrange levels of L to obtain a new LHD. For example, deleting the first row of the constructed L in Example 3.2 and changing the levels $(-4, \ldots, -1, 1, \ldots, 4)$ to $(-3.5, \ldots, -0.5, 0.5, \ldots, 3.5)$, we obtain another LHD L_{-1} given by

$$L_{-1} = \begin{bmatrix} 0.5 & -2.5 & 3.5 & -1.5 & -0.5 & 2.5 & -3.5 & 1.5 \\ 2.5 & 0.5 & -1.5 & -3.5 & -2.5 & -0.5 & 1.5 & 3.5 \\ 3.5 & -1.5 & -0.5 & 2.5 & -3.5 & 1.5 & 0.5 & -2.5 \\ -1.5 & -3.5 & -2.5 & -0.5 & 1.5 & 3.5 & 2.5 & 0.5 \\ -0.5 & 2.5 & -3.5 & 1.5 & 0.5 & -2.5 & 3.5 & -1.5 \\ -2.5 & -0.5 & 1.5 & 3.5 & 2.5 & 0.5 & -1.5 & -3.5 \\ -3.5 & 1.5 & 0.5 & -2.5 & 3.5 & -1.5 & -0.5 & 2.5 \\ 1.5 & 3.5 & 2.5 & 0.5 & -1.5 & -3.5 & -2.5 & -0.5 \end{bmatrix}$$

with $d(L_{-1}) = 20$, which is close to the upper bound given in Lemma 3.1: $d_{\text{upper}} = \lfloor 3^2(3^2-1)/3 \rfloor = 24$. In general, deleting the first row of L in (3.4) and changing its levels x to $x - \text{sign}(x)/2$, we obtain a new $(p^2-1) \times (p^2-1)$ LHD, denoted by L_{-1}. The following result studies $d_{\text{eff}}(L_{-1})$ and $\rho_{\text{ave}}(L_{-1})$.

Theorem 3.5. *The design L_{-1} is a $(p^2-1) \times (p^2-1)$ mirror-symmetric LHD with*

$$d_{\text{eff}}(L_{-1}) = \frac{d(L_{-1})}{d_{\text{upper}}} \geq 1 - \frac{1}{p} - \frac{3}{p^2}, \ \ and \ \ \rho_{\text{ave}}(L_{-1}) < \left(10 + \frac{8}{p}\right) \frac{1}{p^2 - 2}.$$

As $p \to \infty$, $d_{\text{eff}}(L_{-1}) \to 1$ and $\rho_{\text{ave}}(L_{-1}) \to 0$.

3.2.4 Another class of mirror-symmetric LHDs

For any positive integer N, the number of positive integers that are less than and coprime to N is $\phi(N)$, where $\phi(\cdot)$ is the Euler function. Define a generate vector $\mathbf{h} = (h_1, h_2, \cdots, h_{\phi(N)})$ with $1 = h_1 < \cdots < h_{\phi(N)} = N - 1$ and $\gcd(h_i, N) = 1$ for $i = 1, \cdots, \phi(N)$. The $N \times \phi(N)$ good lattice point (GLP) sets D is given by,

$$D = M\mathbf{h} = (Mh_1, Mh_2, \cdots, Mh_{\phi(N)}) \pmod{N}, \tag{3.7}$$

where $M = (1, 2, \cdots, N)^\top$ is a column vector. Combining with the piece-wise linear transformation φ introduced before, we now present the construction method for another class of mirror-symmetric LHDs.

Algorithm 3.2 (Construction of mirror-symmetric LHDs based on GLP sets)**.**

Step 1. Let X be a $p \times (p-1)$ GLP design, where its kth row is given by,

$$k\mathbf{h} = (k, 2k, \cdots, (p-1)k) \pmod{p}.$$

Step 2. Let D be the design obtained by deleting the last constant row of X and

$$F = \varphi(D) - \textit{sign}(\varphi(D))/2. \tag{3.8}$$

The $(p-1) \times (p-1)$ design F detailed in (3.8) of Algorithm 3.2 is a mirror-symmetric LHD, according to property (3.2) of transformation φ. For example, when $p = 7$, the 6×6 mirror-

symmetric LHD constructed using Algorithm 3.2 is

$$F = \begin{bmatrix} 1.5 & 2.5 & 0.5 & -0.5 & -2.5 & -1.5 \\ 2.5 & -0.5 & -1.5 & 1.5 & 0.5 & -2.5 \\ 0.5 & -1.5 & 2.5 & -2.5 & 1.5 & -0.5 \\ -0.5 & 1.5 & -2.5 & 2.5 & -1.5 & 0.5 \\ -2.5 & 0.5 & 1.5 & -1.5 & -0.5 & 2.5 \\ -1.5 & -2.5 & -0.5 & 0.5 & 2.5 & 1.5 \end{bmatrix}$$

with $d(F) = 12$. We have the following theorem.

Theorem 3.6. *The design F is a $(p-1) \times (p-1)$ mirror-symmetric LHD with*

$$d(F) \geq \frac{p^2-1}{3} - (p-1) \;\; and \;\; d_{\text{eff}}(F) = \frac{d(F)}{d_{\text{upper}}} \geq 1 - \frac{2}{p}.$$

Theorem 3.6 can be proved using Theorem 1 of Wang, Xiao, et al. (2018) and the fact that the decrease of the L_1-distance between any two distinct rows cannot exceed $p-1$ after the level rearrangement.

Wang, Xiao, et al. (2018) provided a construction method for LHDs with $p-1$ runs and $p-1$ factors as well. A lower bound of L_1-distance efficiency of their design E_b^* is $1 - 2.43/p$ for primes $p \geq 7$, which is slightly worse than ours. For example, when $p = 89$, $d_{\text{eff}}(F) = 0.9890 > d_{\text{eff}}(E_b^*) = 0.9852$ and when $p = 103$, $d_{\text{eff}}(F) = 0.9903 > d(E_b^*) = 0.9886$, etc.

Notice that our design F is also mirror-symmetric among columns, i.e., $x_{\cdot j} = -x_{\cdot(p-j)}$ for any $1 \leq j \leq p-1$, where $x_{\cdot j}$ is the jth column of design F. Consequently, we can simply take the first half columns and the resulting $(p-1) \times (p-1)/2$ design F_{half} remains as a mirror-symmetric maximin L_1-distance LHD. However, for the $(p-1) \times (p-1)$ LHD E_b^* proposed by Wang, Xiao, et al. (2018), which subset of columns to select to maximize the resulting design distance is a non-trivial problem. To further illustrate the merits of F_{half}, we constructed a variety of LHDs with $p-1$ runs and $(p-1)/2$ factors using different algorithms, including the linearly permuted good

lattice point sets ("LP-GLP", Y. Zhou and Xu, 2015), Williams transformation of linearly permuted good lattice point sets ("WP-GLP", Wang, Xiao, et al., 2018), R package `SLHD` (Ba, 2015) and our proposed method. For the "LP-GLP" method, we apply the linear permutation $x \to x + (p-1)/2 \pmod{p}$ to the $p \times (p-1)$ good lattice point design following Tang and Xu (2014) and use the leave-one-out method to obtain a $(p-1) \times (p-1)$ LHD, which is mirror-symmetric among both rows and columns, and then we take the first half columns. The $(p-1) \times (p-1)$ LHD generated by the "WP-GLP" method is not mirror-symmetric among columns, and the design consisting of the first half columns has small distance. To make a meaningful comparison we randomly select half of its columns with 10,000 repetitions. The `SLHD` package targets at generating maximin L_2-distance designs, therefore we run the command `maximinSLHD()` with option $t = 1$ and other default settings 100 times for fair comparisons. Both median and maximum L_1-distances of LHDs constructed from these two methods are reported in Table 3.1.

Table 3.1 fully displays that our proposed method dominates all the other existing construction algorithms. For all primes $7 \leq p < 100$ investigated, our constructed design F_{half} consistently possesses the greatest L_1-distance, even compared to the maximum L_1-distances of "WP-GLP" and "SLHD" methods. Moreover, this result also holds for larger p, considering the asymptotic maximin L_1-distance optimality of F_{half}.

3.2.5 Further extension of the leave-one-out procedure

Mirror-symmetric LHDs constructed in previous sections have restricted structures, i.e., the number of design runs $N = p^2 - 1$ or $p-1$ and the number of design factors $n = N$ or $N/2$. Now we present a more flexible method to generate many asymptotic maximin L_1-distance mirror-symmetric LHDs with different N and n's.

Algorithm 3.3 (Construction of flexible mirror-symmetric LHDs)**.**

Step 1. Let D be an $N \times n$ mirror-symmetric LHD.

Step 2. Randomly delete one row x, as well as its mirror-symmetric row $-x$ from D. Rearrange

N	n	Our Method	LP-GLP	WP-GLP		SLHD	
				Median	Max	Median	Max
6	3	6	5	4	4	6	6
10	5	17	14	10	12	14	15
12	6	24	20	15	20	20	22
16	8	43	37	26	36	34.5	37
18	9	54	45	34	43	43	48
22	11	81	69	51	65	64	69
28	14	131	115	86	103	105	111
30	15	150	125	100	118	121	127
36	18	216	180	146	173	174	183
40	20	267	231	183	211	218	228
42	21	294	245	203	239	241	250
46	23	353	304	246	284	289	303
52	26	451	387	320	362	370.5	385
58	29	561	480	403	446	465	483
60	30	600	500	432	479	497	515
66	33	726	605	529	590	607.5	625
70	35	817	696	599	656	685	706
72	36	864	720	636	698	725	753
78	39	1014	845	752	824	857	877
82	41	1121	952	837	912	946	986
88	44	1291	1095	971	1047	1096.5	1127
96	48	1536	1280	1164	1255	1310	1352

Table 3.1: Comparison of the L_1-distances of $(p-1) \times (p-1)/2$ LHDs. *Our Method*, first half columns of F generated using Algorithm 3.2; *LP-GLP*, linearly permuted good lattice point set; *WP-GLP*, random half column sub-sampling from the Williams transformation of linearly permuted good lattice point set with 10,000 repetitions; *SLHD*, R function call `maximinSLHD(t=1)` with 100 repetitions.

levels of the remaining $N-2$ rows in the following way, such that the resulting $(N-2)\times n$ design D^* is still a mirror-symmetric LHD.

Denote $x=(x_1,x_2,\cdots,x_n)$. For any of the remaining $N-2$ rows $y=(y_1,y_2,\cdots,y_n)\notin\{x,-x\}$, set

$$y_i^*=\begin{cases} y_i-\textit{sign}(y_i) & \textit{if } |y_i|>|x_i|,\\ y_i & \textit{otherwise}. \end{cases} \tag{3.9}$$

where $y^*=(y_1^*,y_2^*,\cdots,y_n^*)$ is the corresponding row after level rearrangement in D^*.

Step 3. Repeat Step 2 k_r times, where k_r is a fixed constant, to generate a $(N-2k_r)\times n$ mirror-symmetric LHD.

Step 4. Randomly throw away k_c columns, where k_c is a fixed constant. The remaining $(N-2k_r)\times(n-k_c)$ design is still a mirror-symmetric LHD with high L_1-distance efficiency.

Step 2 of Algorithm 3.3 can be regarded as a generalization of the leave-one-out method introduced by Wang, Xiao, et al. (2018). The difference is that we leave out a pair of mirror-symmetric rows to keep the mirror-symmetric structure. Whenever two mirror-symmetric rows are deleted from the original $N\times n$ design, the distance will decrease at most by $2n$ after the level rearrangement detailed in (3.9), and whenever a column gets deleted, the L_1-distance of the remaining design will reduce at most by $N-1$. Hence, we have the following result.

Theorem 3.7. *Let D be an $N\times n$ mirror-symmetric LHD and D^* be the $(N-2k_r)\times(n-k_c)$ mirror-symmetric LHD generated using Algorithm 3.3. Then $d_{\text{eff}}(D^*)\geq d_{\text{eff}}(D)-6k_r/(N-2k_r)-3k_c/(n-k_c)$.*

Set the initial $N\times n$ mirror-symmetric LHD to be the first half columns of L_{-1}, denoted as $L_{-1,half}$, constructed in Section 3.2.3, or the first half columns of F, denoted as F_{half}, constructed in Section 3.2.4. According to the asymptotic maximin L_1-distance optimality of $L_{-1,half}$ and F_{half} and Theorem 3.7, for any constant $k_r\ll N$ and $k_c\ll n$, the resulting $(N-2k_r)\times(n-k_c)$ mirror-symmetric LHD generated by Algorithm 3.3 is still asymptotically optimal with respect

to the maximin L_1-distance criterion. Hence, Algorithm 3.3 is capable of constructing a series of asymptotically optimal mirror-symmetric LHDs with considerably more flexible design sizes. For example, when $p = 17,\ k_r = 5$ and $k_c = 4$, by deleting the first 5 rows, together with their corresponding mirror-symmetric rows and first 4 columns of $L_{-1,half}$, we obtain a 278×140 mirror-symmetric LHD with $d_{\text{eff}} = 0.9637$ after proper level rearrangement; when $p = 499,\ k_r = 8$ and $k_c = 9$, start from F_{half}, we obtain a 482×240 mirror-symmetric LHD with $d_{\text{eff}} = 0.9658$ by following the same procedure.

3.3 Numerical Comparisons

3.3.1 Distance efficiency and average pairwise correlations

We compare the distance efficiencies d_{eff} and average pairwise correlations ρ_{ave} of the LHDs constructed in Section 3.2 and the LHDs generated via an existing method, the R package `SLHD` provided by Ba et al. (2015). For fair comparisons, given each required size of design, we call the function `maximinSLHD()` 100 times with option $t = 1$ and default settings, and examine the boxplots of the values of d_{eff} and ρ_{ave} for the obtained LHDs.

For each odd prime p, Figure 3.2 compares the values of d_{eff} (left) and ρ_{ave} (right) of the $(p^2 - 1) \times (p^2 - 1)$ leave-one-out LHD constructed in Section 3.2.3 and the LHDs obtained via the `SLHD` package. All leave-one-out LHDs outperform the LHDs generated via the `SLHD` package in the sense of maximizing the d_{eff} and minimizing the ρ_{ave}. Specifically, the values of d_{eff} for all leave-one-out LHDs are above 95% and converges to 1 quickly as p increases, while values for the LHDs from the `SLHD` package are all below 95% and usually fluctuate around 90%. In addition, the values of ρ_{ave} for the leave-one-out LHDs are always smaller than those of the LHDs from the `SLHD` package and vanish quickly as p increases.

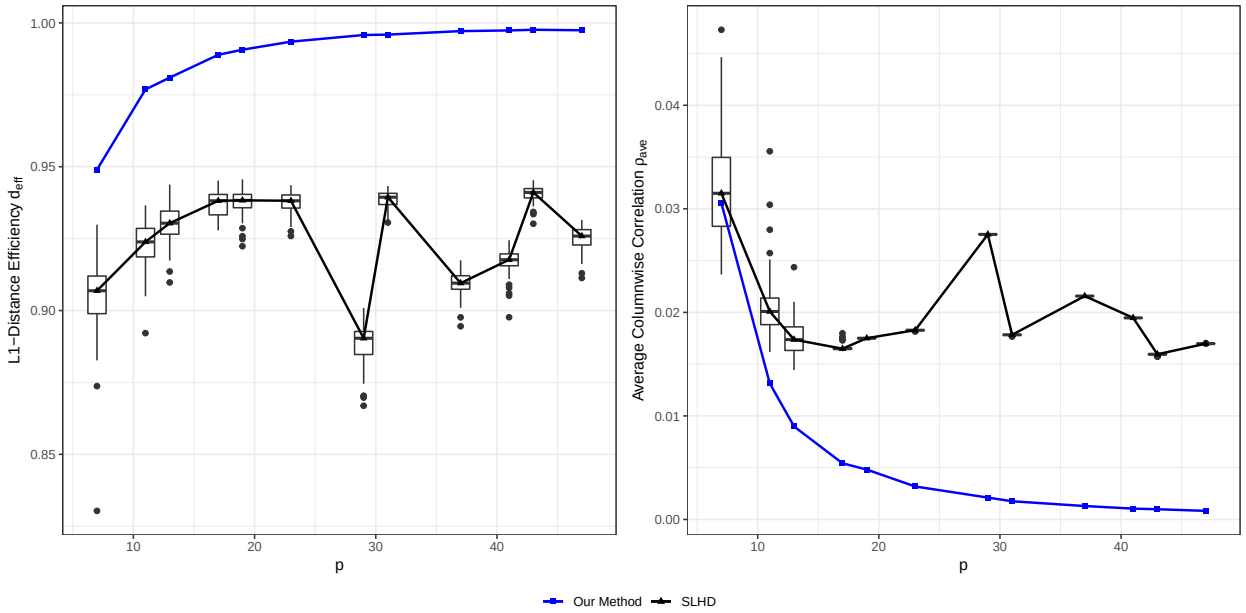

Figure 3.2: Comparison of the values of d_{eff} (left) and ρ_{ave} (right) of the leave-one-out LHDs constructed in Section 3.2.3 and the LHDs obtained via the `SLHD` package.

3.3.2 Space-filling properties under projections

Besides, we also compare the space-filling properties of the LHD L constructed in Section 3.2, under projection, with other designs constructed by existing methods. Note that mirror-symmetric columns in L are perfectly correlated, to further reduce the correlation, we take the first half columns of LHD L, and denote the resulting $p \times (p-1)/2$ LHD as L_{half}.

Some of the space-filling criteria considered here include Euclidean distance (the larger the better), maximum projection criterion $\psi(D)$ proposed by Joseph et al. (2015) (the smaller the better), relative centered L_2-discrepancy CD_2 (the smaller the better), and the average column-wise correlation ρ_{ave} defined in (3.6) (the smaller the better). The relative CD_2 is defined as the difference of CD_2 values between the design currently under investigation versus the baseline one. Here, we assign L_{half} as our baseline. For each projection dimension $k \leq n$, we evaluate all $\binom{n}{k}$ projected designs and determine the worst case scenarios for different space-filling criteria respectively.

Existing space-filling designs under comparison include uniform designs (Fang et al., 2006)

(ud), maximin distance designs (maximin), and maximum projection designs (maxpro). They are constructed using R packages `UniDOE` (A. Zhang et al., 2018) with discrepancy measure CD_2, `SLHD` (Ba, 2015) and `MaxPro` (Ba & Joseph, 2018), respectively.

Figure 3.3 and 3.4 present the final results for 25×12 ($p = 5$) and 49×24 ($p = 7$) space-filling designs under projection. When calculating the CD_2 value, we rescale the N levels $-(N-1)/2, \ldots, (N-1)/2$ to $[0, 1]$ using the transformation $x \to x/N + 0.5$, where $N = p^2$. Both figures exhibit the robustness of our proposed mirror-symmetric LHD against different space-filling criteria: first of all, L_{half} has the smallest average column-wise correlation among all optimal LHDs under all dimension projections, especially when $p = 5$, L_{half} is strictly orthogonal with $\rho_{\text{ave}}(L_{half}) = 0$; besides, for other criteria like L_2-distance, $\psi(D)$ or CD_2, L_{half} performs comparatively well and even outperforms the corresponding optimal LHDs (e.g., maximin distance designs for L_2-distance, maximum projection designs for $\phi(D)$, etc.) in low dimensions. This property can be super beneficial for conducting early-stage experiments with many inert factors.

3.3.3 Correlation among higher order terms

We now evaluate correlations among the higher order terms on LHD L and L_{-1} constructed in Section 3.2. Suppose design X has N runs and n factors. Following Georgiou (2009), we calculate the alias matrices for the first-order model associated with the two-factor interactions and pure quadratic terms, defined by

$$\begin{aligned} T &= (X_1^\top X_1)^{-1} X_1^\top X_{int}, \\ Q &= (X_1^\top X_1)^{-1} X_1^\top X_{quad}, \end{aligned} \tag{3.10}$$

where X_1 is the $N \times (n+1)$ regression matrix for the first-order model (including a column of ones for the intercept), X_{int} corresponds to the $N \times (n(n-1)/2)$ matrix containing all possible two-factor interactions and the $N \times n$ matrix X_{quad} includes all pure quadratic terms. Furthermore,

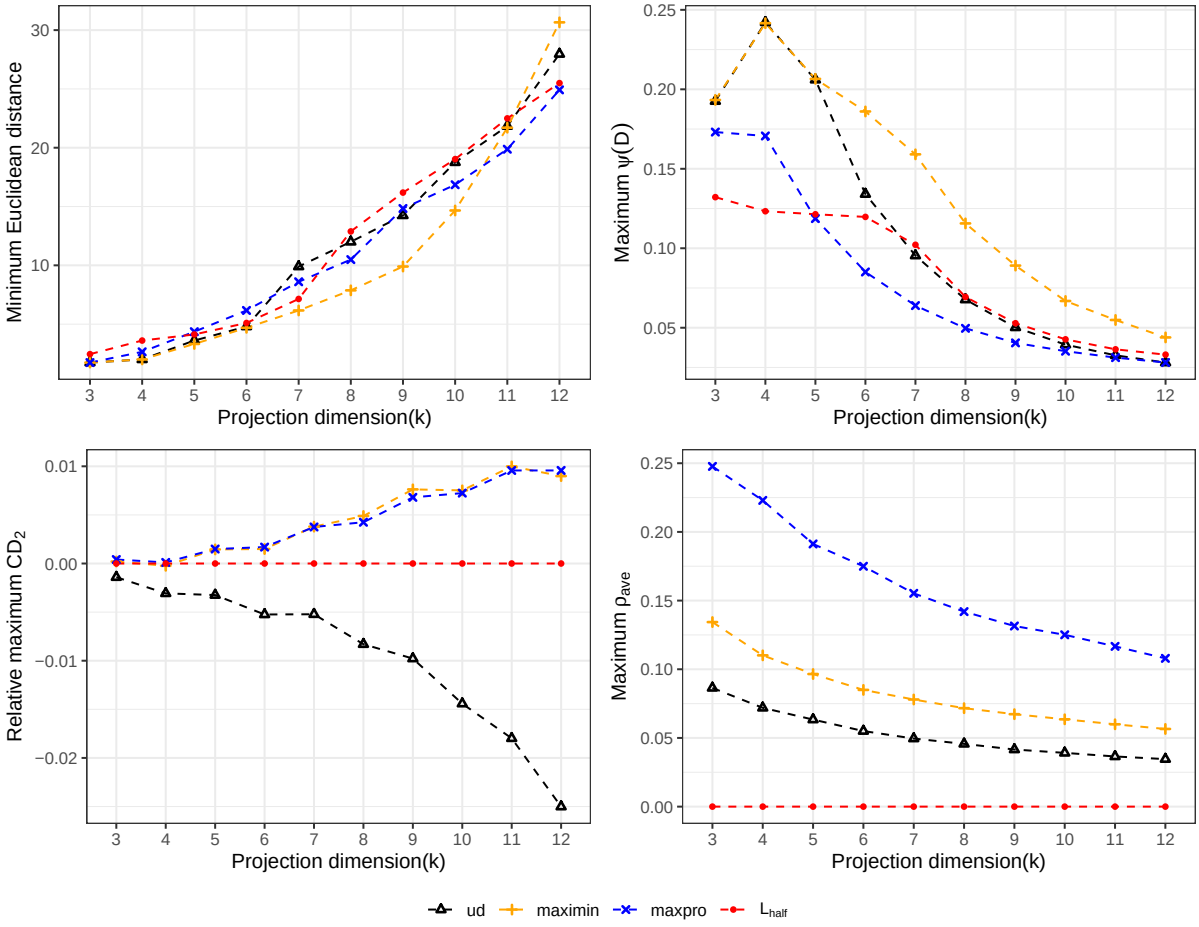

Figure 3.3: Comparisons of projection space-filling properties for 25×12 designs under minimum Euclidean distance (the larger the better), maximum $\psi(D)$ (the smaller the better), relative maximum CD_2 (the smaller the better), and maximum ρ_{ave} (the smaller the better) criteria.

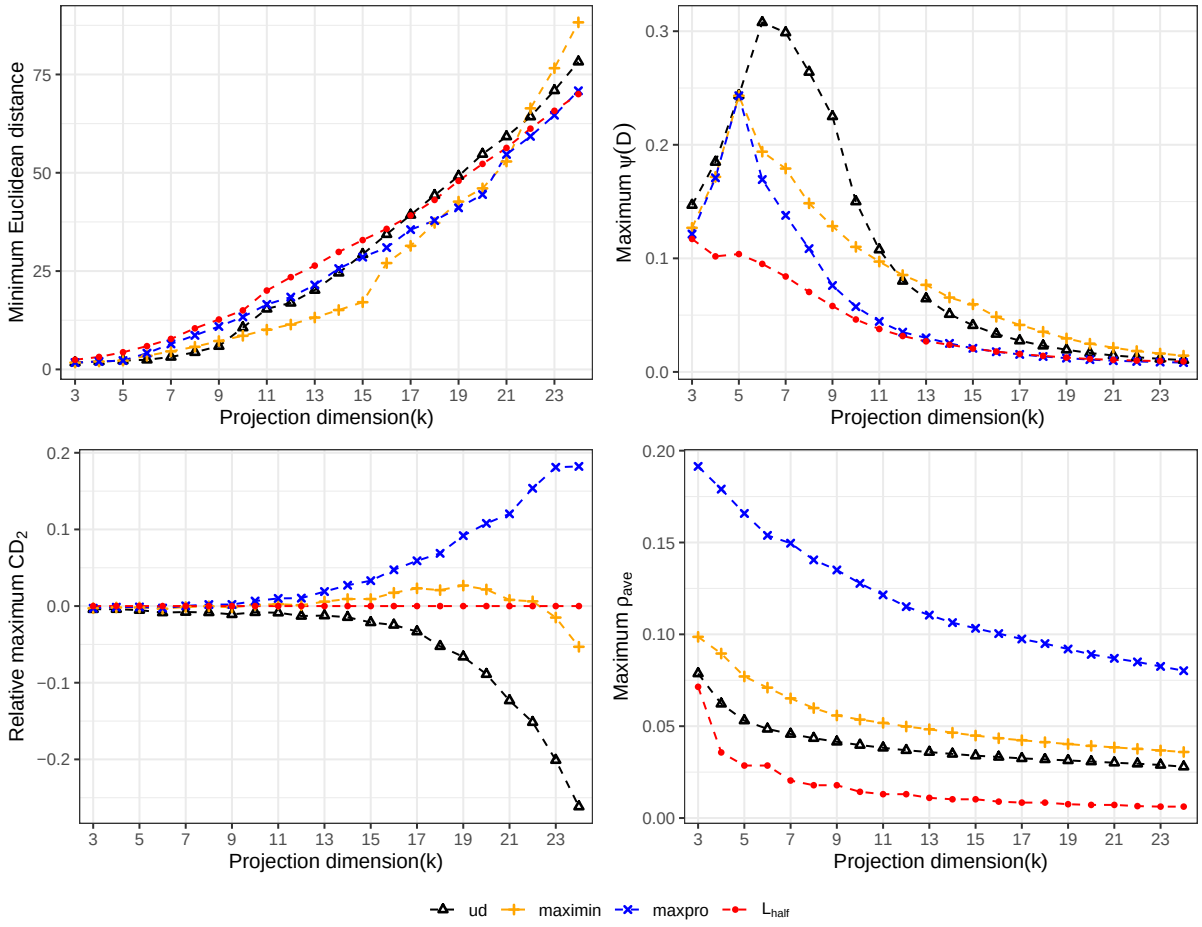

Figure 3.4: Comparisons of projection space-filling properties for 49×24 designs under minimum Euclidean distance (the larger the better), maximum $\psi(D)$ (the smaller the better), relative maximum CD_2 (the smaller the better), and maximum ρ_{ave} (the smaller the better) criteria.

define measures

$$
\begin{aligned}
E(|t|) &= \frac{\sum_{i=1}^{n+1}\sum_{j=1}^{n(n-1)/2}|t_{ij}|}{n(n^2-1)/2}, \quad \max|t| = \max_{i,j}|t_{ij}|; \\
E(|q|) &= \frac{\sum_{i=1}^{n+1}\sum_{j=1}^{n}|q_{ij}|}{n(n+1)}, \qquad \max|q| = \max_{i,j}|q_{ij}|.
\end{aligned}
\tag{3.11}
$$

Designs with relatively small absolute values in bias matrices T and Q are suitable for screening tasks. Consequently, designs with smaller $E(|t|), \max|t|, E(|q|)$ and $\max|q|$ values are deemed better in this regards.

For the LHD L constructed by (3.4) of Algorithm 3.1 and its leave-one-out extension L_{-1}, we delete the second half of the columns to further decrease the column-wise correlation and compute their bias on interactions and quadratic terms. Denote the resulting $p^2 \times (p^2-1)/2$ LHD as L_{half} and its leave-one-out version as $L_{-1,half}$. When $p = 5$ and 7, orthogonal LHDs with the same design size are constructed by Georgiou and Efthimiou (2014) using the Goethal-Seidel and Kharaghani arrays. A detailed comparison shall be made in terms of their design distance, as well as correlations among higher order terms. Notice that we rescale both classes of LHDs to range $[-1, 1]$ using the same transformation proposed by Georgiou and Efthimiou (2014) for convenience of measurement computations.

Table 3.2 summaries the numerical results. The LHDs from Georgiou and Efthimiou (2014) with different design sizes are given by Example 6, Example 1, Corollary 7 and Example 5 in their paper, respectively. It is worth noting that for all cases, our constructed LHDs L_{half} and $L_{-1,half}$ managed to achieve the lower bound of $E|q|$ and $\max|q|$, which are derived by Georgiou (2009) among orthgonal LHDs, yet LHDs L_{half} and $L_{-1,half}$ are not necessarily orthogonal in general. Furthermore, LHDs generated by our construction method tend to possess larger L_1-distances, as the increase of p.

N	n	LHD	$d(D)$	$E(\|t\|)$	max $\|t\|$	$E(\|q\|)$	max $\|q\|$
25	12	L_{half}	6.1667	0	0	0.0278	0.3611
		GE	6.5000	0	0	0.0278	0.3611
24	12	$L_{-1,half}$	7.8261	0	0	0.0279	0.3623
		GE	7.1304	0	0	0.0279	0.3623
49	24	L_{half}	12.3333	0	0	0.0139	0.3472
		GE	11.3333	0	0	0.0139	0.3472
48	24	$L_{-1,half}$	15.8298	0	0	0.0139	0.3475
		GE	11.2340	0	0	0.0139	0.3475

Table 3.2: Comparisons of L_1-distance ($d(D)$), bias on interactions ($E(|t|)$ and max $|t|$) and bias on quadratic terms ($E(|q|)$ and max $|q|$) between our constructed LHDs and those from Georgiou and Efthimiou (2014) (GE).

3.4 Concluding Remarks

In this chapter, we have proposed a series of deterministic construction methods for generating both mirror-symmetric maximin L_1-distance balanced designs and LHDs with a flexible number of runs and factors. The average correlation between columns of the constructed designs rapidly converge to zero with the increase of design sizes. Furthermore, first half columns of all constructed designs share the same distance efficiency as the full design, owing to the mirror-symmetry among design columns. The halved designs, coming with the extra mirror-symmetric property, have much greater L_1-distance efficiency than existing maximin designs found by state-of-art methods and algorithms. Furthermore, our constructed designs exhibit robust performance against many other space-filling criteria and under various dimension projections.

3.5 Appendix: Proofs

Proof of Theorem 3.1. Given the fact that the run size N is odd, the mirror-symmetric design $D = (x_{ij})_{1\le i\le N,\, 1\le j\le n}$ can always be represented in the form of $(0_n^T, X^T, -X^T)^T$, where 0_n is a row vector of n zeros, after proper row rearrangement.

As design D is balanced, all levels $\{-\frac{s-1}{2}, -\frac{s-3}{2}, \cdots, -1, 0, 1, \cdots, \frac{s-3}{2}, \frac{s-1}{2}\}$ appear exactly N/s times in each design column. The total L_1-distance between the first row (center point) and all other rows is

$$\begin{aligned}\sum_{i=2}^{N} d_{1i}(D) &= \sum_{i=2}^{N}\sum_{j=1}^{n} |x_{1j} - x_{ij}| = \sum_{j=1}^{n}\sum_{i=2}^{N} |x_{ij}| \\ &= n \cdot 2\left(|1| + \cdots + \left|\frac{s-3}{2}\right| + \left|\frac{s-1}{2}\right|\right) \cdot \frac{N}{s} = \frac{N(s^2-1)n}{4s}.\end{aligned}$$

It is obvious that $d(D)$ does not exceed the average distance between the center point (first row) and any other row, so

$$d(D) \le \min\{d_{1i}(D) : i = 2, \ldots, N\} \le \frac{N(s^2-1)n}{4s(N-1)}.$$

This completes the proof. □

Proof of Theorem 3.2. For the simplicity of description and derivation of theoretical properties of balanced design E in (3.3), we add the deleted constant zero column back to the design D in Step 2 of Algorithm 3.1, so that the resulting $E' = \varphi(XX^\top)$ becomes a $p^2 \times p^2$ square matrix with the same L_1-distance distribution as E in (3.3). We will use same notations D and E for designs retaining or deleting the first column in the following proofs, in order to simplify the notation slightly.

Denote $E = (\varphi(x_{ij}))_{1\le i\le N,\, 1\le j\le n}$, where $N = n = p^2$, and x_{ij} corresponds to the ith row and jth column element in design D of Algorithm 3.1.

- For the first case, note that $\varphi(x_i) = (\varphi(x_{i1}), \cdots, \varphi(x_{in}))$ for any $i \neq 1$ is a balanced row vector, with every level of $\{-(p-1)/2, \cdots, (p-1)/2\}$ appearing p times, and $\varphi(x_1) = (\varphi(x_{11}), \cdots, \varphi(x_{1n}))$ is a constant row vector with each component equal to $\varphi(0) = 0$. Therefore,

$$d_{ik}(E) = p\sum_{t=0}^{p-1} |t - (p-1)/2| = (p-1)p(p+1)/4, \quad when\ i = 1\ or\ k = 1;$$

- For the second case, when the ith and kth rows are mirror-symmetric, similarly according to the balance property of any design row $\varphi(x_i)$,

$$d_{ik}(E) = p\sum_{t=0}^{p-1}|t-(p-1-t)| = p\sum_{t=0}^{p-1}|2t-(p-1)| = (p-1)p(p+1)/2;$$

- For the last case, let's first discuss another property of the symmetric design D. Denote $X=(A,B)$, where A and B respectively are two design factors of the p-level 2-factor full factorial design X. Then all rows in D can be viewed as a linear space over finite field Z_p spanned by $A^\top$ and $B^\top$. The first row in D is the zero vector of this linear space, and any row in D can be represented as $c_1A^\top + c_2B^\top \pmod p$, where $c_1, c_2 \in Z_p = \{0,1,2,\ldots,p-1\}$. Consequently, we can divide all rows in D but the zero vector into $p+1$ groups, denoted as $\{R_1,\cdots,R_{p+1}\}$, such that row vectors within the same group are linearly dependent, and those between different groups are linearly independent. It is easy to show that the number of elements in every group R_t is $p-1$. We need to consider two cases.

  (i) For any two linearly independent row vectors, $x_i \in R_t$ and $x_k \in R_s$, $t \neq s$, $(x_i^\top, x_k^\top)$ form an orthogonal array of strength two and thus all level combinations appear exactly once. Furthermore, corresponding rows $\varphi(x_i)$ and $\varphi(x_k)$ in design E also share the same property, since the transformation (3.1) defines a one-to-one mapping over Z_p. Therefore,

$$\begin{aligned} d_{ik}(E) = \sum_{s=0}^{p-1}\sum_{t=0}^{p-1}|s-t| &= \sum_{s=0}^{p-1}\left(\sum_{t=0}^{s}(s-t)+\sum_{t=s+1}^{p-1}(t-s)\right) \\ &= \sum_{s=0}^{p-1}\left[\frac{(s+1)s}{2}+\frac{(p-s)(p-s-1)}{2}\right] \\ &= \sum_{s=0}^{p-1}s^2 + (1-p)\sum_{s=0}^{p-1}s + \frac{p^2(p-1)}{2} \\ &= (p-1)p(p+1)/3. \end{aligned}$$

  (ii) For L_1-distance between any two linearly dependent rows, note that every group R_t to-

gether with the zero vector is equivalent, up to column permutations, as p identical $p \times p$ good lattice point (GLP) designs concatenated by columns, where its generator vector $h = (0, 1, \cdots, p-1)$. On the other hand, the transformation (3.1) can be regarded as a special case of the linearly permuted Williams transformation, since $\varphi(x) = W(x \oplus (p-1)/4) - (p-1)/2$ if $p = 1 \pmod 4$ and $\varphi(x) = W(x \oplus (3p-1)/4) - (p-1)/2$ if $p = 3 \pmod 4$, where $x \oplus y = x + y \pmod p$. Then, according to Theorem 1 of Wang, Xiao, et al. (2018), for any two linear dependent row vector $x_i, x_k \in R_t$ that are not mirror-symmetric to each other, we have

$$d_{ik}(E) = (p^2 - 1)/3 \times p = (p-1)p(p+1)/3.$$

The proof is complete. □

Proof of Theorem 3.3. Note design $D = XX^\top$ is symmetric. Thus following the same logic as the proof of Theorem 3.2, we can similarly show that all columns of D also form a linear space over finite field Z_p spanned by A and B, where $X = (A, B)$. The first column in D is the zero vector of this linear space, and any column in D can be represented as $c_1 A + c_2 B \pmod p$, where $c_1, c_2 \in Z_p$.

Consequently, we can divide all columns, except for the first, in D into $p+1$ groups, denoted as $\{C_1, \cdots, C_{p+1}\}$, such that column vectors within the same group are linearly dependent, and those between different groups are linearly independent. It is easy to show that the number of elements in every group C_t is $p - 1$. Moreover, we perform the same grouping on corresponding columns of E and consider two cases.

(i) For any two linearly independent column vectors in design D, $x_i \in C_t$ and $x_k \in C_s$, where $t \neq s$, (x_i, x_k) form an orthogonal array of strength 2 and thus all level combinations appear exactly once. Furthermore, corresponding columns $\varphi(x_i)$ and $\varphi(x_k)$ in design E also share the same property, since the transformation (3.1) defines a one-to-one mapping over Z_p. Therefore, $\rho_{ik}(E) = 0$, given the fact that columns $\varphi(x_i)$ and $\varphi(x_k)$ are combinatorially orthogonal.

(ii) For correlation between two linearly dependent column vectors, note that every group C_t in

design D is equivalent, up to row permutations, to p identical $p \times (p-1)$ good lattice point (GLP) designs concatenated by rows, where its generator vector $h = (1, \cdots, p-1)$. Denote the GLP design as D_{glp}, and for any column index $i \in C_t$ of design D, denote its corresponding column index in D_{glp} as i'. We can easily proof that

$$\rho_{ik}(D) = \rho_{i'k'}(D_{glp}), \quad \text{where } i \neq k \in C_t.$$

Furthermore, as we have pointed out before, the transformation (3.1) is a special case of the linearly permuted Williams transformation. Then by applying Theorem 5 of Wang et al. (2018), we can derive, for any two non-identical column indices i, k within the same group C_t,

$$\sum_{i \neq k \in C_t} |\rho_{ik}(E)| = \sum_{i \neq k \in C_t} |\rho_{ik}(\varphi(D))| = \sum_{i' \neq k'} |\rho_{i'k'}(\varphi(D_{glp}))| < 2(p-1).$$

In summary,

$$\rho_{\text{ave}}(E) = \frac{\sum_{i \neq k} |\rho_{ik}(E)|}{(p^2-1)(p^2-2)} = \frac{\sum_{t=1}^{p+1} \sum_{i \neq k \in C_t} |\rho_{ik}(E)|}{(p^2-1)(p^2-2)} < \frac{2(p-1)(p+1)}{(p^2-1)(p^2-2)} = \frac{2}{p^2-2}.$$

□

Proof of Theorem 3.4. First of all, the design L in (3.4) is a Latin hypercube, since any two adjacent columns of design E in (3.3) forms an orthogonal array of strength two and T_2 is a rotation matrix. Apart from it, this LHD L is also mirror-symmetric, due to the fact that the balanced design E in (3.3) is mirror-symmetric and T is block-diagonal. As L is a $p^2 \times (p^2-1)$ mirror-symmetrical LHD, by Theorem 3.1,

$$d(L) \leq d^*_{\text{upper}} = (p^4-1)/4.$$

Denote $L = (l_{ij})$ and $E = (e_{ij})$. Since $L = ET$ and the rotation matrix T is block diagonal, for

any row index $i \neq k$, we have

$$
\begin{aligned}
d_{ik}(L) &= \sum_{j=1}^{p^2-1} |l_{ij} - l_{kj}| \\
&= \sum_{t=1}^{(p^2-1)/2} \left\{ |(pe_{i(2t-1)} + e_{i(2t)}) - (pe_{k(2t-1)} + e_{k(2t)})| + |(-e_{i(2t-1)} + pe_{i(2t)}) - (-e_{k(2t-1)} + pe_{k(2t)})| \right\} \\
&= \sum_{t=1}^{(p^2-1)/2} \left\{ |p(e_{i(2t-1)} - e_{k(2t-1)}) + (e_{i(2t)} - e_{k(2t)})| + |p(e_{i(2t)} - e_{k(2t)}) - (e_{i(2t-1)} - e_{k(2t-1)})| \right\} \\
&\geq \sum_{t=1}^{(p^2-1)/2} \left\{ p|e_{i(2t-1)} - e_{k(2t-1)}| - |e_{i(2t)} - e_{k(2t)}| \right\} + \left\{ p|e_{i(2t)} - e_{k(2t)}| - |e_{i(2t-1)} - e_{k(2t-1)}| \right\} \\
&= \sum_{j=1}^{p^2-1} \{ p|e_{ij} - e_{kj}| - |e_{ij} - e_{kj}| \} = (p-1) d_{ik}(E).
\end{aligned}
\tag{3.12}
$$

Furthermore, as a direct result from Theorem 3.2,

$$
d(L) = \min_{i \neq k} d_{ik}(L) \geq (p-1) d(E) = (p-1)^2 p(p-1)/4.
$$

Consequently,

$$
d^*_{\text{eff}}(L) = \frac{d(L)}{d^*_{\text{upper}}} \geq \frac{(p-1)^2 p(p-1)/4}{(p^4-1)/4} = 1 - \frac{p+1}{p^2+1} \to 1, \ \ as\ p \to \infty.
$$

In terms of the average pairwise correlation between columns of LHD L, we need to first prove the following result:

$$
\sum_{i \neq k} |\rho_{ik}(L)| \leq \frac{(p+1)^2}{p^2+1} \sum_{i \neq k} |\rho_{ik}(E)|. \tag{3.13}
$$

We regroup the summation of column-wise correlations on the left hand side of (3.13) as

$$\sum_{i\neq k}|\rho_{ik}(L)| = \sum_{t\neq s}\{|\rho_{(2t-1)(2s-1)}(L)| + |\rho_{(2t)(2s-1)}(L)| + |\rho_{(2t-1)(2s)}(L)| + |\rho_{(2t)(2s)}(L)|\} + 2\sum_{w=1}^{(p^2-1)/2}|\rho_{(2w-1)(2w)}(L)|,$$

where column indices $i \neq k \in \{1, 2, \cdots, p^2-1\}$, and $t \neq s \in \{1, 2, \cdots, (p^2-1)/2\}$. Since $E = (e_{ij})$ is a balanced design and $L = (l_{ij})$ is an LHD, $\sum_{j=1}^{p^2} e_{jk} = \sum_{j=1}^{p^2} l_{jk} = 0$, and $\sum_{j=1}^{p^2} e_{jk}^2 = c_0$ hold for every column k, where c_0 is a constant. For any t, columns $2t-1$ and $2t$ of design E are always combinatorially orthogonal; therefore, after rotation, they are still orthogonal and have correlation of 0. Besides,

$$\begin{aligned}
|\rho_{(2t-1)(2s-1)}(L)| &= \frac{\left|\sum_j (pe_{j(2t-1)} + e_{j(2t)})(pe_{j(2s-1)} + e_{j(2s)})\right|}{\sqrt{\sum_j (pe_{j(2t-1)} + e_{j(2t)})^2 \sum_j (pe_{j(2s-1)} + e_{j(2s)})^2}} \\
&= \frac{1}{(p^2+1)c_0}\left|\sum_j \{p^2 e_{j(2t-1)}e_{j(2s-1)} + pe_{j(2t-1)}e_{j(2s)} + pe_{j(2t)}e_{j(2s-1)} + e_{j(2t)}e_{j(2s)}\}\right| \\
&\leq \frac{1}{(p^2+1)c_0}\left\{p^2\left|\sum_j e_{j(2t-1)}e_{j(2s-1)}\right| + p\left|\sum_j e_{j(2t-1)}e_{j(2s)}\right|\right. \\
&\qquad \left. + p\left|\sum_j e_{j(2t)}e_{j(2s-1)}\right| + \left|\sum_j e_{j(2t)}e_{j(2s)}\right|\right\} \\
&= \frac{1}{p^2+1}\left\{p^2|\rho_{(2t-1)(2s-1)}(E)| + p|\rho_{(2t-1)(2s)}(E)| + p|\rho_{(2t)(2s-1)}(E)| + |\rho_{(2t)(2s)}(E)|\right\}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
|\rho_{(2t)(2s-1)}(L)| &\leq \frac{1}{p^2+1}\left\{p|\rho_{(2t-1)(2s-1)}(E)| + |\rho_{(2t-1)(2s)}(E)| + p^2|\rho_{(2t)(2s-1)}(E)| + p|\rho_{(2t)(2s)}(E)|\right\}, \\
|\rho_{(2t-1)(2s)}(L)| &\leq \frac{1}{p^2+1}\left\{p|\rho_{(2t-1)(2s-1)}(E)| + p^2|\rho_{(2t-1)(2s)}(E)| + |\rho_{(2t)(2s-1)}(E)| + p|\rho_{(2t)(2s)}(E)|\right\}, \\
|\rho_{(2t)(2s)}(L)| &\leq \frac{1}{p^2+1}\left\{|\rho_{(2t-1)(2s-1)}(E)| + p|\rho_{(2t-1)(2s)}(E)| + p|\rho_{(2t)(2s-1)}(E)| + p^2|\rho_{(2t)(2s)}(E)|\right\}.
\end{aligned}$$

Hence,

$$\begin{aligned}\sum_{i\neq k}|\rho_{ik}(L)| &= \sum_{t\neq s}\{|\rho_{(2t-1)(2s-1)}(L)| + |\rho_{(2t)(2s-1)}(L)| + |\rho_{(2t-1)(2s)}(L)| + |\rho_{(2t)(2s)}(L)|\} \\ &\leq \frac{(p+1)^2}{p^2+1}\sum_{t\neq s}\{|\rho_{(2t-1)(2s-1)}(E)| + |\rho_{(2t)(2s-1)}(E)| + |\rho_{(2t-1)(2s)}(E)| + |\rho_{(2t)(2s)}(E)|\} \\ &= \frac{(p+1)^2}{p^2+1}\sum_{i\neq k}|\rho_{ik}(E)|,\end{aligned}$$

which completes the proof of (3.13). Then, together with the result from Theorem 3.3, we have

$$\rho_{\text{ave}}(L) \leq \frac{(p+1)^2}{p^2+1}\rho_{\text{ave}}(E) \leq \frac{(p+1)^2}{p^2+1}\frac{2}{p^2-2} < \left(1+\frac{2}{p}\right)\frac{2}{p^2-2}.$$

□

Proof of Theorem 3.5. First, we can establish the following relationship of pairwise L_1-distances between balanced design $E = (e_{ij})$ and LHD $L = (l_{ij})$:

$$d_{ik}(L_{-1}) + (p^2-1) \geq d_{(i+1)(k+1)}(L) \geq (p-1)d_{(i+1)(k+1)}(E), \ \ \forall\, i \neq k.$$

The intermediate inequality $d_{(i+1)(k+1)}(L) \geq (p-1)d_{(i+1)(k+1)}(E)$ holds according to (3.12) in the proof of Theorem 3.4. Assuming the minimal distance of L_{-1} is achieved between the i_0 and k_0 rows,

$$d(L_{-1}) = d_{i_0k_0}(L_{-1}) \geq (p-1)d_{(i_0+1)(k_0+1)}(E) - (p^2-1) \geq \frac{(p-1)^2p(p+1)}{3} - (p^2-1).$$

By applying Lemma 3.1, we get $d_{\text{upper}}(L_{-1}) = \lfloor p^2(p^2-1)/3 \rfloor$. Therefore,

$$\begin{aligned}d_{\text{eff}}(L_{-1}) = \frac{d(L_{-1})}{d_{\text{upper}}(L_{-1})} \geq \frac{d(L_{-1})}{p^2(p^2-1)/3} &\geq \frac{p(p-1)^2(p+1)/3 - (p^2-1)}{p^2(p^2-1)/3} \\ &= 1 - \frac{1}{p} - \frac{3}{p^2} \to 1, \ as\, p \to \infty.\end{aligned}$$

Besides, denote the leave-one-out LHD $L_{-1} = (l'_{ij})$, where l'_{ij} corresponds to the element at the ith row and jth column of design L_{-1}. Then $\sum_{j=1}^{p^2-1} l'_{jk} = \sum_{j=1}^{p^2} l_{jk} = 0$ hold for any column k. Moreover,

$$\sum_{j=1}^{p^2-1}(l'_{jk})^2 = 2\left[\left(\frac{1}{2}\right)^2 + \left(\frac{3}{2}\right)^2 + \cdots + \left(\frac{p^2-2}{2}\right)^2\right] = \frac{p^2(p^2-1)(p^2-2)}{12},$$
$$\sum_{j=1}^{p^2}(l_{jk})^2 = 2\left[1^2 + 2^2 + \cdots + \left(\frac{p^2-1}{2}\right)^2\right] = \frac{p^2(p^2-1)(p^2+1)}{12},$$

also hold for any column k. Then for the absolute correlation between any two columns i and k of design L_{-1},

$$|\rho_{ik}(L_{-1})| = \frac{|\sum_j l'_{ji}l'_{jk}|}{\sqrt{\sum_j (l'_{ji})^2 \sum_j (l'_{jk})^2}} = \left|\sum_j l'_{ji}l'_{jk}\right| \Big/ \left(\frac{p^2(p^2-1)(p^2-2)}{12}\right).$$

The numerator

$$\begin{aligned}
|\sum_j l'_{ji}l'_{jk}| &= \Bigg| \sum_{j:\, l_{ji}>0,\, l_{jk}>0} \left(l_{ji}-\frac{1}{2}\right)\left(l_{jk}-\frac{1}{2}\right) + \sum_{j:\, l_{ji}>0,\, l_{jk}<0} \left(l_{ji}-\frac{1}{2}\right)\left(l_{jk}+\frac{1}{2}\right) \\
&\qquad + \sum_{j:\, l_{ji}<0,\, l_{jk}>0} \left(l_{ji}+\frac{1}{2}\right)\left(l_{jk}-\frac{1}{2}\right) + \sum_{j:\, l_{ji}<0,\, l_{jk}<0} \left(l_{ji}+\frac{1}{2}\right)\left(l_{jk}+\frac{1}{2}\right) \Bigg| \\
&= \Bigg| \sum_j l_{ji}l_{jk} - \frac{1}{2}\sum_{j:\, l_{ji}>0,\, l_{jk}>0}(l_{ji}+l_{jk}) + \frac{1}{2}\sum_{j:\, l_{ji}<0,\, l_{jk}<0}(l_{ji}+l_{jk}) \\
&\qquad + \frac{1}{2}\sum_{j:\, l_{ji}>0,\, l_{jk}<0}(l_{ji}-l_{jk}) - \frac{1}{2}\sum_{j:\, l_{ji}<0,\, l_{jk}>0}(l_{ji}-l_{jk}) + \sum_{j:\, l_{ji}l_{jk}>0}\frac{1}{4} - \sum_{j:\, l_{ji}l_{jk}<0}\frac{1}{4} \Bigg| \\
&\le |\sum_j l_{ji}l_{jk}| + \frac{1}{2}\sum_j \{|l_{ji}| + |l_{jk}|\} + \sum_j \frac{1}{4} \\
&= |\sum_j l_{ji}l_{jk}| + \frac{(p^2-1)(p^2+1)}{4} + \frac{p^2-1}{4}.
\end{aligned}$$

Therefore,

$$|\rho_{ik}(L_{-1})| = \frac{\left|\sum_j l'_{ji}l'_{jk}\right|}{\sqrt{\sum_j (l'_{ji})^2 \sum_j (l'_{jk})^2}} \leq \frac{p^2+1}{p^2-2}|\rho_{ik}(L)| + \frac{3(p^2+2)}{p^2(p^2-2)} < 2|\rho_{ik}(L)| + \frac{6}{p^2-2},$$

and

$$\rho_{\text{ave}}(L_{-1}) = \frac{\sum_{i\neq k}|\rho_{ik}(L_{-1})|}{(p^2-1)(p^2-2)} < \frac{\sum_{i\neq k}\left\{2|\rho_{ik}(L)| + \frac{6}{p^2-2}\right\}}{(p^2-1)(p^2-2)} = 2\rho_{\text{ave}}(L) + \frac{6}{p^2-2}.$$

By applying Theorem 3.4, we get

$$\rho_{\text{ave}}(L_{-1}) < 2\rho_{\text{ave}}(L) + \frac{6}{p^2-2} < 2\left(1+\frac{2}{p}\right)\frac{2}{p^2-2} + \frac{6}{p^2-2} = \left(10+\frac{8}{p}\right)\frac{1}{p^2-2}.$$

□

CHAPTER 4

Bayesian-Inspired Distance Designs

One major drawback of the maximin distance criterion is that it tends to place a large portion of points at the corners and on the boundaries of the domain, severely undermining its space-filling characteristic on low-dimensional projections and thus making it undesirable in the case when only a few design factors are active.

To fill the gap, Joseph et al. (2015) proposed maximum projection designs, which claims to maximize space-filling properties on projections to all subsets of factors. Denote any design with n runs and m factors as $D = (x_{ik})_{1\leq i\leq n, 1\leq k\leq m}$, and any row or design point within as $\mathbf{x}_i$, for $1 \leq i \leq n$. A maximum projection design aims to minimize the following criterion,

$$\psi(D) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{k=1}^{m} (x_{ik} - x_{jk})^2} \right\}^{1/m}. \tag{4.1}$$

Note that the denominator of Eq. (4.1) consists of products of Euclidean distances among all factors. Consequently, any two points, even if projected into low-dimensional subspaces, of a maximum projection design are guaranteed to be well-separated from each other.

Apart from it, the validity of maximum projection designs is further justified from the Gaussian process, or kriging modeling perspective, which is frequently used as a surrogate model for approximating sophisticated computer programs or physical processes. Specifically, we assume that the response surface can be fully depicted by a shifted stationary Gaussian process, i.e., $Y(x) = \mu + Z(x)$,

where $Z(x) \sim \mathcal{N}(0,\ \sigma^2 R(\alpha))$, with Gaussian correlation function,

$$R(\mathbf{x}_i, \mathbf{x}_j; \alpha) = \exp\left\{-\sum_{k=1}^{m} \alpha_k (x_{ik} - x_{jk})^2\right\},\ \ \alpha_k \in (0, \infty). \tag{4.2}$$

A popular optimal design criterion is the maximum entropy measure (Shewry & Wynn, 1987), which aims at finding a design that maximizes the determinant of the correlation matrix, $|R(\alpha)|$. Joseph et al. (2015) showed that with a noninformative prior, $p(\alpha_k) \propto 1$, imposed on every weight parameter α_k, the maximum projection design tends to minimize the expected sum of off-diagonal elements of the correlation matrix, $\sum_{i \neq j} R(\mathbf{x}_i, \mathbf{x}_j; \alpha)$. By applying Hadamard's inequality and Gershgorin's theorem, we get the following lower and upper bounds on the determinant of $R(\alpha)$,

$$\prod_{s=1}^{n} \left\{1 - \sum_{j \neq i_s} R(\mathbf{x}_{i_s}, \mathbf{x}_j; \alpha)\right\}_+ \leq |R(\alpha)| \leq 1, \tag{4.3}$$

where $i_1, \cdots, i_n \in \{1, \cdots, n\}$ are not necessarily distinctive and $\{x\}_+ = \max\{x, 0\}$. Thus, a maximum projection design will be guaranteed with decently large $|R(\alpha)|$ and perform well under the maximum entropy criterion as well. Besides, the maximum projection criterion $\psi(D)$ overcomes a major drawback of the maximum entropy criterion, i.e., being model-dependent on the unknown correlation parameter α, by adopting a Bayesian approach, and thus can be efficiently computed given any design alone.

4.1 Bayesian-Inspired Distance Criterion

The choice of noninformative prior, $p(\alpha_k) \propto 1$, is seemingly right, especially when there is no pre-existing domain knowledge to determine the importance of design factors. However, on second thought, assigning equal prior probability to α_k for being close to infinity versus moderately small appears a bit questionable. Indeed, this becomes more of an issue with the observation that each α_k quantifies the significance of factor, or dimension k, and thus cannot be extremely large. Numerous

Gaussian process model fitting results have delivered support for this argument, where most of maximum likelihood estimators of correlation parameters are small, with only a few exceptions.

In consequence, we deem it more appropriate to replace the noninformative prior with certain distribution that tends to allocate higher probability towards small values. The exponential prior, denoted as $\text{Exp}(\lambda_k)$, is one of those qualified candidates, with the rate parameter λ_k determining the curvature of its density,

$$p(\alpha_k) \propto \lambda_k e^{-\lambda_k \alpha_k}, \;\; \lambda_k > 0. \tag{4.4}$$

Thenceforth, the expected sum of off-diagonal elements of the correlation matrix now becomes,

$$\begin{aligned} \mathbb{E}_\alpha \left[\sum_{i=1}^{n} \sum_{j \neq i} R_{ij}(\alpha) \right] &= \sum_{i=1}^{n} \sum_{j \neq i} \mathbb{E}_\alpha \left\{ \prod_{k=1}^{m} \exp\left[-\alpha_k (x_{ik} - x_{jk})^2 \right] \right\} \\ &= \sum_{i=1}^{n} \sum_{j \neq i} \left\{ \prod_{k=1}^{m} \int \exp\left[-\alpha_k (x_{ik} - x_{jk})^2 \right] \cdot \lambda_k e^{-\lambda_k \alpha_k} \, d\alpha_k \right\} \\ &= \sum_{i=1}^{n} \sum_{j \neq i} \prod_{k=1}^{m} \frac{\lambda_k}{(x_{ik} - x_{jk})^2 + \lambda_k}, \end{aligned} \tag{4.5}$$

where $R_{ij}(\alpha)$ stands for the correlation between ith and jth design point, $R(\mathbf{x}_i, \mathbf{x}_j; \alpha)$. Likewise, our objective here is to minimize the expected summation (4.5), so as to approximately increase the lower bound on $|R(\alpha)|$.

When domain knowledge is absent or could be potentially misleading before any actual experimentation, all α_k's share an identical prior distribution, $\text{Exp}(\lambda)$, in which situation, minimizing Eq. (4.5) is equivalent to minimizing

$$\psi_\lambda(D) = \left\{ \frac{1}{\binom{n}{2}} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{1}{\prod_{k=1}^{m} [\lambda + (x_{ik} - x_{jk})^2]} \right\}^{1/m}, \tag{4.6}$$

for any fixed value of λ. This newly proposed design criterion $\psi_\lambda(D)$ can be viewed as a general version of the maximum projection criterion, with $\lambda = 0$ deteriorating into the original maximum projection criterion $\psi(D)$. Motivated by a Bayesian framework, we name $\psi_\lambda(D)$ detailed in (4.6)

as the Bayesian-inspired distance criterion, and the corresponding optimal design constructed by minimizing $\psi_\lambda(D)$ as the Bayesian-inspired distance design.

One advantage of our proposed criterion is being capable of constructing multi-level optimal designs, which is deemed cost-effective for conducting experiments with large run sizes. In contrast, the original maximum projection criterion defined in (4.1) fails to do so due to the intrinsic flaw of its objective definition.

4.2 Connection to Optimal Moments Criterion

On the other hand, another Gaussian process model optimal design criterion, known as the optimal moments criterion, also strives for the minimization of the sum of off-diagonal elements of a correlation matrix. It stems from the assumption that correlation between any pair of points can be written as a function of distance between the two. Mathematically,

$$R_{ij}(\alpha) := R(\mathbf{x}_i, \mathbf{x}_j; \alpha) = \exp\left(-\alpha \cdot d(\mathbf{x}_i, \mathbf{x}_j)\right). \tag{4.7}$$

When $d(\mathbf{x}_i, \mathbf{x}_j)$ is the squared Euclidean distance, $R_{ij}(\alpha)$ will correspond to the Gaussian correlation function illustrated in Eq. (4.2) with one sole parameter α. Meanwhile, $d(\mathbf{x}_i, \mathbf{x}_j)$ being Manhattan, or L_1-distance renders an exponential correlation. It is worth noting that all design factors are presumed to be equally significant in the optimal moments criterion derivation. In contrast, for the Bayesian-inspired distance criterion, although all α_k's share an identical prior, they can still vary, indicating different importance measures of design factors.

Starting from (4.7), we can rewrite $R_{ij}(\alpha)$ using its Taylor series expanded at point $\alpha = 0$,

$$R_{ij}(\alpha) = \exp(-d(\mathbf{x}_i, \mathbf{x}_j) \cdot \alpha) = \sum_{t=0}^{\infty} \frac{(-d(\mathbf{x}_i, \mathbf{x}_j) \cdot \alpha)^t}{t!}. \tag{4.8}$$

For any positive integer t, denote $M_t = \sum_{i=1}^{n} \sum_{j \neq i} d(\mathbf{x}_i, \mathbf{x}_j)^t$. The sum of off-diagonal elements

of the correlation matrix, for any given design $D = (x_{ij})$, now becomes

$$\sum_{i=1}^{n}\sum_{j\neq i} R_{ij}(\alpha) = \sum_{i=1}^{n}\sum_{j\neq i}\sum_{t=0}^{\infty}\frac{(-\alpha\cdot d(\mathbf{x}_i,\mathbf{x}_j))^t}{t!} = \sum_{t=0}^{\infty}\frac{(-\alpha)^t M_t}{t!}. \tag{4.9}$$

When $\alpha \to 0$, i.e., all design points are nearly dependent, minimization of (4.9) can be achieved by sequentially minimizing,

$$-M_1, M_2, -M_3, \cdots$$

Since for any integer $t > 0$, M_t corresponds to the t-th moment of vector $d(\mathbf{x}_i, \mathbf{x}_j)$, this optimization objective is named as the optimal moments criterion.

Meanwhile, for the Bayesian-inspired distance criterion derived from (4.5), when all α_k's share the same exponential prior with rate parameter λ, it is equivalently minimizing

$$\sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\frac{\lambda}{\lambda+(x_{ik}-x_{jk})^2} = \sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\frac{1}{1+(x_{ik}-x_{jk})^2/\lambda}. \tag{4.10}$$

Denote $y = (x_{ik}-x_{jk})^2/\lambda$. According to the Taylor series of $1/(1+y)$ at $y=0$, objective (4.10) can be further expanded as,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\frac{\lambda}{\lambda+(x_{ik}-x_{jk})^2} &= \sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\sum_{s=0}^{\infty}\left[-\frac{(x_{ik}-x_{jk})^2}{\lambda}\right]^s \\ = \sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\sum_{s=0}^{\infty}&(-1)^s(x_{ik}-x_{jk})^{2s}\lambda^{-s}. \end{aligned}\tag{4.11}$$

Notice that for any finite positive integer S, $\prod_{k=1}^{m}\sum_{s=0}^{S}(-1)^s(x_{ik}-x_{jk})^{2s}\lambda^{-s}$ is a product of m polynomials. In consequence, by expanding the product and collecting terms with the same polynomial degree, Equation (4.11) is simplified to,

$$\sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\sum_{s=0}^{\infty}(-1)^s(x_{ik}-x_{jk})^{2s}\lambda^{-s} = \sum_{i=1}^{n}\sum_{j\neq i}\sum_{t=0}^{\infty}\left(\sum_{(s_1,\cdots,s_m)\in S_t}\prod_{k=1}^{m}(-1)^{s_k}(x_{ik}-x_{jk})^{2s_k}\lambda^{-s_k}\right),$$

where $S_t = \{(s_1, \cdots, s_m) : s_1 + \cdots + s_m = t,\ 0 \leq s_1, \cdots, s_m \leq t\}$. Thus,

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\frac{\lambda}{\lambda+(x_{ik}-x_{jk})^2} &= \sum_{i=1}^{n}\sum_{j\neq i}\sum_{t=0}^{\infty}\left(\sum_{(s_1,\cdots,s_m)\in S_t}\prod_{k=1}^{m}(-1)^{s_k}(x_{ik}-x_{jk})^{2s_k}\lambda^{-s_k}\right) \\
&= \sum_{i=1}^{n}\sum_{j\neq i}\sum_{t=0}^{\infty}(-1)^t\lambda^{-t}\left(\sum_{(s_1,\cdots,s_m)\in S_t}\prod_{k=1}^{m}(x_{ik}-x_{jk})^{2s_k}\right) \\
&= \sum_{t=0}^{\infty}(-1)^t\lambda^{-t}\sum_{i=1}^{n}\sum_{j\neq i}\left(\sum_{(s_1,\cdots,s_m)\in S_t}\prod_{k=1}^{m}(x_{ik}-x_{jk})^{2s_k}\right).
\end{aligned}$$

Let

$$M_t = \sum_{i=1}^{n}\sum_{j\neq i} d(\mathbf{x}_i, \mathbf{x}_j)^t \text{ and } N_t = \sum_{i=1}^{n}\sum_{j\neq i}\sum_{k=1}^{m}(x_{ik}-x_{jk})^{2t},$$

where $d(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{m}(x_{ik}-x_{jk})^2$ is the squared Euclidean distance. Clearly $M_1 = N_1$. For level-balanced designs, N_t are constant for $t = 1, 2, \ldots$. Let

$$\Delta_t = \sum_{i=1}^{n}\sum_{j\neq i}\sum_{(s_1,\cdots,s_m)\in S_t}\prod_{k=1}^{m}(x_{ik}-x_{jk})^{2s_k}.$$

It is clear that $\Delta_0 = n(n-1)$ and $\Delta_1 = M_1 = N_1$. Next, we can express Δ_2 as follows:

$$\Delta_2 = \sum_{i=1}^{n}\sum_{j\neq i}\left[\frac{1}{2}\left(\sum_{k=1}^{m}(x_{ik}-x_{jk})^2\right)^2 + \frac{1}{2}\sum_{k=1}^{m}(x_{ik}-x_{jk})^4\right] = \frac{1}{2}M_2 + \frac{1}{2}N_2.$$

Combining these results, we have

$$\sum_{i=1}^{n}\sum_{j\neq i}\prod_{k=1}^{m}\frac{\lambda}{\lambda+(x_{ik}-x_{jk})^2} = n(n-1) - \lambda^{-1}M_1 + \lambda^{-2}(M_2+N_2)/2 + o(\lambda^{-3}).$$

When λ is large, minimizing the Bayesian-inspired distance criterion is the equivalent of minimizing M_2 among level-balanced designs. In other words, the optimal moments criterion can be viewed as another asymptotic case of our proposed Bayesian-inspired distance criterion when the

rate parameter λ is substantially large.

4.3 Influence of Different λ's on Design Properties

The Bayesian-inspired distance criterion, $\psi_\lambda(D)$, introduces a new unknown hyperparameter λ, which characterizes the shape of our exponential prior and needs to be determined ahead of constructing the optimal design. As the same predicament encountered in directly estimating the correlation parameter α_k, measurement of λ remains infeasible until the implementation of the actual experiment. Hence, this section is meant to study the varying Bayesian-inspired distance (BID) design properties corresponding to different λ's.

To start, we investigate the space-filling property of Latin hypercube BID designs with two factors. Each 80-run optimal design is constructed using optimization algorithms detailed in the next section. According to Figure 4.1, when two factors of interest exist, a Bayesian-inspired distance design with $\lambda = 0.01$ turns out to be the most space-filling. The shape of all 80 design points then gradually evolves into concentric circles and finally a perfect diamond as λ increases. Similar patterns are also observed in multi-level BID designs, according to Figure 4.2, under which circumstances the original maximum projection criterion fails to distinguish any design. Notice that when $\lambda = 0.05$, duplicated points start to appear in the optimal design. This coincides with the fact that a two-level design is optimal for continuous regions. For both cases, the M_1 values of constructed designs stay constant, while the M_2 values consistently decline alongside the rising of λ.

However, direct visualization of design points becomes almost infeasible for designs with more than two factors. Instead, we propose to select the best λ based on the robustness of Bayesian-inspired distance designs against misspecifications of the correlation function. For simplicity, we study the cases where misspecification only occurs within the Gaussian correlation structure defined by Equation (4.2), but the methodology can be easily generalized to arbitrary correlation functions.

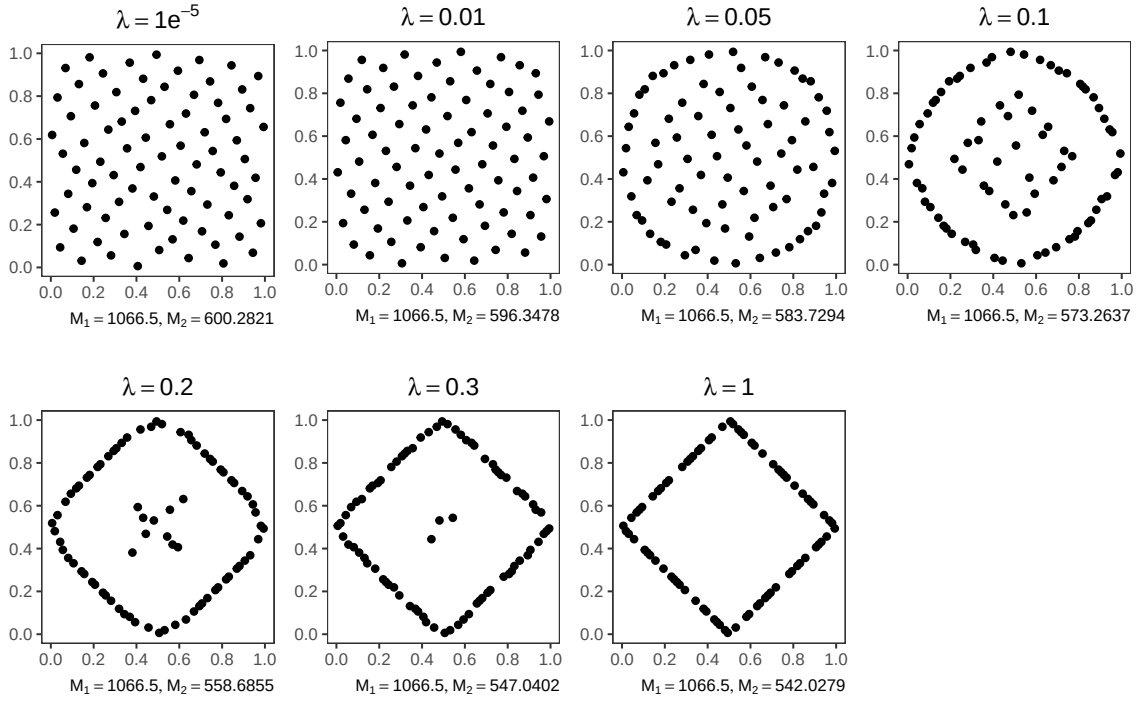

Figure 4.1: Visualization of two-dimensional Latin hypercube Bayesian-inspired distance designs with 80 runs against different λ's.

More precisely, with the same exponential prior assumption on Gaussian correlation parameter α_k's as in Section 4.1, by varying their universal rate hyperparameter λ, we compute determinants of the correlation matrix for any given design, given observations of α_k's sampled from Exp(λ) defined in Equation (4.4) respectively, and study its robustness against correlation misspecifications.

Figures 4.3 and 4.4 summarize the results for designs with two and eight factors, respectively. Given each rate hyperparameter λ, we sample 100 sets of α_k's, in order to account for the variability and measure the robustness of different designs. The y-axis in each figure represents the relative log-determinant, which is defined as $\log(|R(\alpha)|)/\sum_D |\log(|R_D(\alpha)|)|$, where $R_D(\alpha)$ is the correlation matrix for design D, and the summation in the denominator is taken over all designs under comparisons. We generate other types of space-filling designs, including uniform (ud), maximin distance (maximin) and maximum projection (maxpro) designs via existing R packages; see Section 4.5.2 for details.

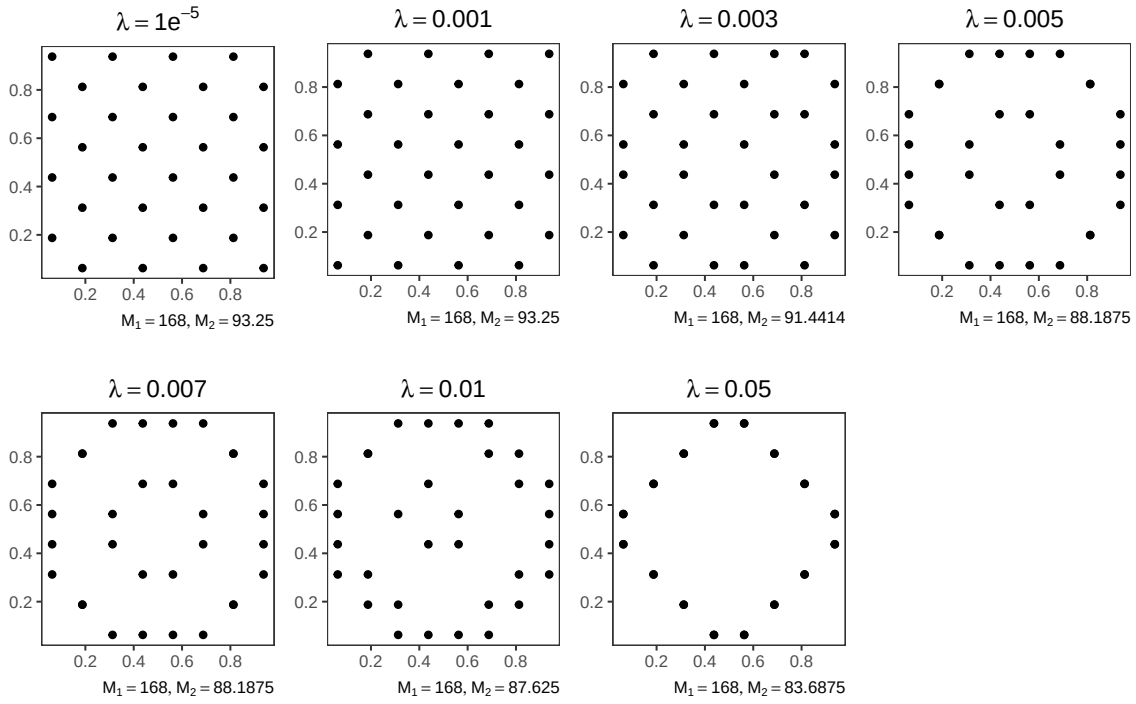

Figure 4.2: Visualization of two-dimensional multi-level balanced Bayesian-inspired distance designs with 32 runs and 8 levels against varying λ's.

Figure 4.3 illustrates that the BID design with $\lambda = 0.01$ achieves the largest determinant of the correlation matrix when the underlying true rate parameter is 0.01 or 0.1, which coincides with the conclusion drawn based on the space-filling visualization in Figure 4.1. Besides, both BID designs consistently outperform the maximin distance and maximum projection designs. For designs with eight factors, the proposed BID designs continue to dominate the other existing space-filling designs, with maximum projection designs being the worst in this case. Particularly, the Bayesian-inspired distance design with $\lambda = 0.1$ appears to be the most promising with respect to the maximum entropy criterion according to Figure 4.4.

In general, given the numbers of design runs and factors needed for the experiment, we can determine the optimal λ of the Bayesian-inspired distance criterion by following the aforementioned procedure.

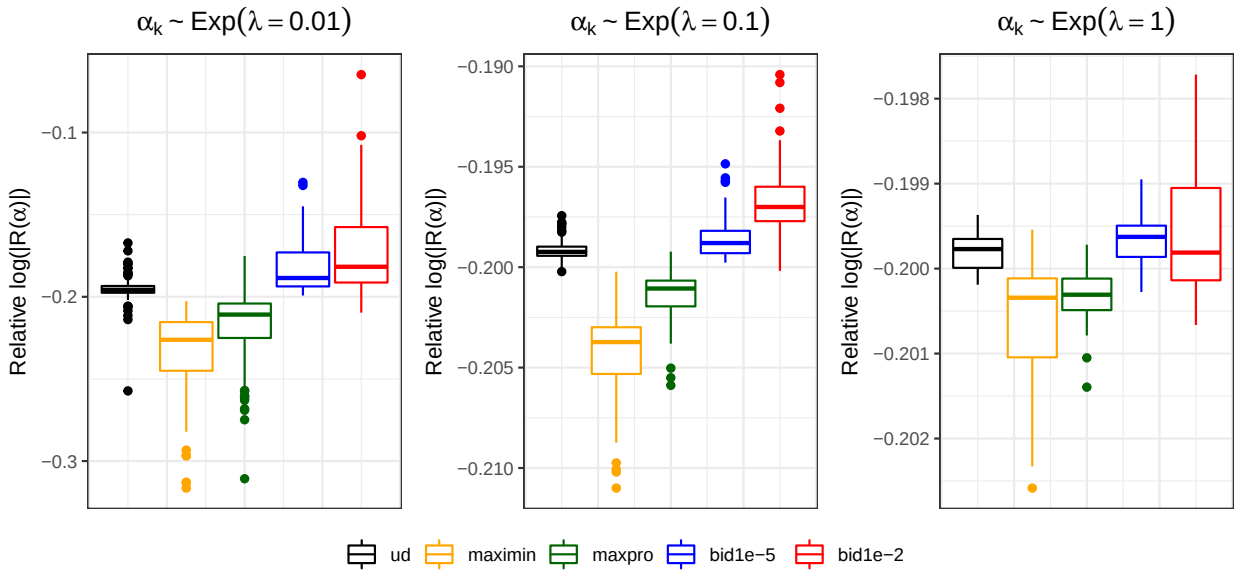

Figure 4.3: Comparisons of relative log-determinant of the correlation matrix among BID designs and other space-filling designs with two factors. `bid1e-5` and `bid1e-2` represent the BID designs with $\lambda = 1e^{-5}$ and 0.01 respectively. Each subplot considers a different true Gaussian correlation function, where the correlation parameter $\alpha_k \sim \text{Exp}(\lambda)$ with $\lambda = 0.01,\ 0.1$ and 1.

4.4 Optimization Algorithms for Design Construction

Bayesian-inspired distance designs are constructed by means of minimizing $\psi_\lambda(D)$ in (4.6) among all feasible designs. This, at first glance, can be easily solved using any continuous optimization algorithm. However, a direct optimization of $\psi_\lambda(D)$ is quite challenging, due to its enormous number of optimization variables, nm, and countless local minima. The latter dilemma is identified by the fact that $\psi_\lambda(D)$ will remain unchanged under arbitrary column or row permutations.

Alternatively, we shall narrow down the search space, by imposing certain restrictions onto the design structure. Among those, Latin hypercubes is a class of extensively investigated candidate designs, where each factor is composed of n equally spaced levels and thus ensures one-dimensional projection uniformity. Moreover, constructing an optimal Latin hypercube design (LHD) that minimizes $\psi_\lambda(D)$ turns into a combinatorial optimization problem, since the total number of LHDs with n runs and m factors is finite.

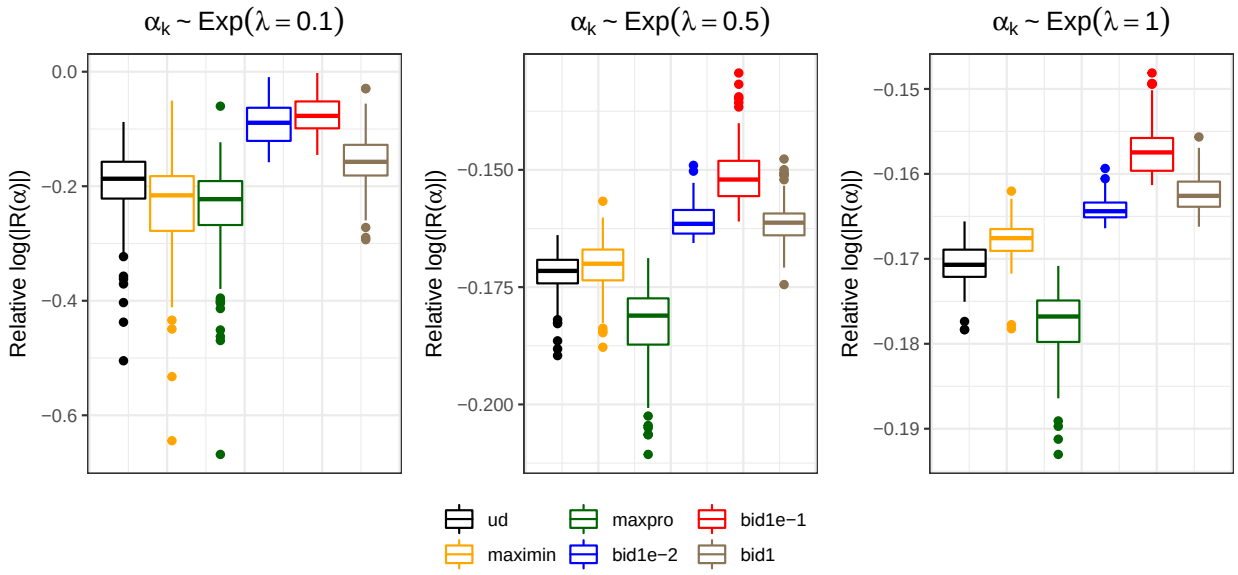

Figure 4.4: Comparisons of relative log-determinant of the correlation matrix among BID designs and other space-filling designs with eight factors. `bid1e-2`, `bid1e-1` and `bid1` correspond to the BID designs with $\lambda = 0.01, 0.1$ and 1 respectively. Each subplot considers a different true Gaussian correlation function, where the correlation parameter $\alpha_k \sim \text{Exp}(\lambda)$ with $\lambda = 0.1,\ 0.5$ and 1.

In the succeeding section, we will delve into two metaheuristic algorithms - simulated annealing and genetic algorithms, discuss our proposed implementations and compare performances to existing ones. It is worth noting that both algorithms can be easily generalized to construct multi-level balanced optimal designs (see Figure 4.2 for an illustrative example), which fills the gap of the `MaxPro` package (Ba & Joseph, 2018).

4.4.1 Simulated Annealing Algorithm

Simulated annealing (SA) is a class of stochastic metaheuristic algorithms, which progressively approximates the global optimum of an objective function. Its name comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to alter its physical properties. This optimization algorithm is sophisticatedly designed in a way to bypass the local optimum, by retaining positive probability of accepting worse solutions, so as to ensure a more extensive exploration of the entire solution space.

In the design literature, Morris and Mitchell (1995) successfully applied SA to construct the maximin Latin hypercube designs. In their implementation, a new candidate LHD was generated as a perturbation of the preceding one, by randomly swapping two distinct rows within a randomly selected column. As a consequence, the one-dimensional uniformity structure of a Latin hypercube design is preserved, since every column still remains as a permutation of n equally spaced levels.

This so-called exchange algorithm, when applied in our problem, manages to significantly reduce the computational complexity of the difference in $\psi_\lambda(D)$, from $O(n^2m)$ to $O(nm)$, and thus serves as a principal building block in our modified version of SA as well. The novelty of our implementation is that after a column of the current LHD, D_{curr}, is randomly selected, we opt to evaluate M different perturbations, and propose the next candidate design, say D_{next}, that minimizes $\psi_\lambda(D_{next}) - \psi_\lambda(D_{curr})$. This greedy implementation, although might seem more prone to local optima, is capable of finding good Bayesian-inspired distance designs at an accelerated rate. See Algorithm 1 for more detailed explanations.

Algorithm 1 Simulated annealing for Bayesian-inspired distance design constructions

procedure SA($n, m, \lambda, N_{iters}, M, T_0, c, T_{min}$)
  Initialization: $T \leftarrow T_0$, $D_{curr} \leftarrow$ A random LHD(n, m);
  for $t = 1 : N_{iters}$ **do**
    Randomly select a column index k_0;
    $D_{next} \leftarrow D_{curr}$, $\psi_\lambda\_min\_diff \leftarrow \infty$;
    for $s = 1 : M$ **do** ▷ pick the most promising among M perturbations
      Randomly sample two distinct row indices i_0, j_0;
      Generate D_{prop}, by swapping $(i_0, k_0)th$ with $(j_0, k_0)th$ element of D_{curr};
      if $\psi_\lambda(D_{prop}) - \psi_\lambda(D_{curr}) < \psi_\lambda\_min\_diff$ **then**
        $D_{next} \leftarrow D_{prop}$;
        $\psi_\lambda\_min\_diff \leftarrow \psi_\lambda(D_{prop}) - \psi_\lambda(D_{curr})$;
      end if
    end for
    if $runif(1) \leq \exp(-(\psi_\lambda(D_{next}) - \psi_\lambda(D_{curr}))/T)$ **then**
      $D_{curr} \leftarrow D_{next}$; ▷ proposal D_{next} got accepted
      $T \leftarrow \max\{c \cdot T, T_{min}\}$;
    end if
  end for
  Return D_{curr} as the constructed design.
end procedure

Algorithm 1 has several tuning parameters. N_{iters} is the total number of optimization iterations (default to 10,000), M is the number of perturbations considered for each randomly selected design column (default to 100), c is the temperature diminishing factor (default to 0.99), T_0 is the initial temperature (default to 100), and T_{min} corresponds to the minimal temperature allowed in the system (default to $1e^{-15}$). For all numeric results shown in Section 4.5, our modified SA algorithm is executed with default settings, without further adaptive parameter tuning.

4.4.2 Genetic Algorithm

Another metaheuristic, genetic algorithm (GA), is inspired by Charles Darwin's theory of natural evolution. As a metaphor of natural selection, it is commonly applied to generate high-quality solutions to complicated search and optimization problems, with the assistance of biologically-inspired operators, including mutation, crossover, etc. GAs are deemed as intelligent exploitation of random search, i.e., adaptively directing the search into better performance regions provided with historical data from preceding generation. Therefore, it is rarely snared in the local optimum and deemed appropriate for our optimal design construction purpose.

In terms of its actual implementation, each generation of GA consists of a set of $popSize$ candidate Latin hypercube designs. Each LHD, known as an individual, can be fully characterized by the vector of design factors, as an analogy to chromosome. After objective function evaluation, the $popSize * pElit$ most promising individuals are consistently retained, and the $popSize * pCross$ most fittest ones are used for reproduction, with $100 * pMut\%$ probability of mutation happening. Particularly, we manage to properly overwrite both mutation and crossover operators, to cater to the intrinsic Latin hypercube structure of each individual.

Mutation: The mutation of an individual can be achieved by randomly permuting some design column. This mechanism is adopted to ensure diversity in offspring population and better exploration of the solution space.

Crossover: When two individuals, or LHDs, are selected to produce offspring in the next

generation, columns of child LHD share equal probability of inheriting from either parent. Defining crossover in this way retains the one-dimensional projection uniformity structure in descendants naturally.

More implementation details shall be found in Algorithm 2. Likewise, we adopt default parameter values within, where $N_{iters} = 10,000$, $popSize = 100$, $pElit = 0.1$, $pCross = 0.2$ and $pMut = 0.1$, for both time saving and fair comparisons.

Carnell (2022) utilized the genetic algorithm to draw Latin hypercube samples in the `lhs` R package. They implemented the crossover operator by swapping a single column between the current fittest individual and another random one. Besides, their mutation is accomplished by switching merely two elements of a column. We claim that our novel implementation is more universal, where offspring are generated by mixing approximately half chromosomes from many promising ancestors, not necessarily the fittest. The mutation can potentially permute orders of all elements within the selected column instead of just two. We will present more elaborate numerical comparisons in Section 4.5.1.

4.5 Numerical Results

This section investigates the discrepancy between two optimization algorithms mentioned above in generating high-quality Bayesian-inspired distance (BID) designs and the disparity of the BID criterion compared to other space-filling measures.

4.5.1 Optimization Algorithms

Comparing performances among different stochastic optimization algorithms can sometimes be subtle. Some algorithms are more computationally efficient in finding a good but not necessarily the optimal solution, while others, although potentially slower, emphasize finding the real global optimum. For the optimal design construction purpose, we cherish both speeds and quality. In the

Algorithm 2 Genetic algorithm for Bayesian-inspired distance design constructions

procedure GA($n, m, \lambda, N_{iters}, popSize, pElit, pCross, pMut$)
    Initialize the current generation G_{curr} as a vector of $popSize$ random LHD(n, m)s;
    for $t = 1 : N_{iters}$ **do**
        $sort(G_{curr})$; ▷ sort the current generation based on $\psi_\lambda(D)$ criterion
        $G_{next} \leftarrow vector()$;
        for $i = 1 : popSize * pElit$ **do**
            ▷ retain $100 * pElit$% fittest individuals and pass them into the next generation
            $G_{next} \leftarrow c(G_{next}, G_{curr}[i])$;
        end for
        for $i = 1 : popSize * (1 - pElit)$ **do** ▷ rest are descendants of two promising parents
            $idx1, idx2 \leftarrow sample(popSize * pCross, 2, replace = F)$;
            $par1, par2 \leftarrow G_{curr}[idx1], G_{curr}[idx2]$;
            $child \leftarrow$ an $n \times m$ of 0s;
            for $k = 1 : m$ **do**
                Generate a random number u between 0 and 1;
                if $u < (1 - pMut)/2$ **then** ▷ inherit from parent 1
                    $child[, k] \leftarrow par1[, k]$;
                else if $u < 1 - pMut$ **then** ▷ inherit from parent 2
                    $child[, k] \leftarrow par2[, k]$;
                else ▷ mutate with probability $pMut$
                    $child[, k] \leftarrow sample(n, n, replace = F)$; ▷ a permutation of $1 : n$
                end if
            end for
            $G_{next} \leftarrow c(G_{next}, child)$;
        end for
        $G_{curr} \leftarrow G_{next}$;
    end for
    Return G_{curr} as the constructed design.
end procedure

sequential experimentation setup, optimal designs are adaptively proposed to cater to the varying number of design runs and factors in a timely fashion. While on the other hand, given design specifications as an initial design, we can afford more time and resources searching for that particular optimal design.

Algorithms under examination here include the simulated annealing algorithm proposed by Morris and Mitchell (1995) for maximin Latin hypercube design constructions (SA_{base}), its modified greedy version detailed in Algorithm 1 (SA), the baseline genetic algorithm with crossover and mutation operators defined by Carnell (2022) in `geneticLHS` for drawing Latin hypercube samples (GA_{base}), together with our proposed implementation of genetic algorithm in Algorithm 2 (GA). Hyperparameters within each algorithm are predetermined and deemed robust for diverse design specifications.

We fix $\lambda = 1$ in the BID criterion as a trade-off between the maximum projection and optimal moments criteria and evaluate different algorithm performances with respect to minimizing $\psi_1(D)$. Extensive design specifications are investigated, with different number of design factors $m = 4, 6, 8, 10$ and runs $n = 2m, 5m, 10m$. For each n and m, pairs of the aforementioned stochastic searching processes under comparison are executed for an equal period, with ten repetitions each for better variability measurement. It is also worth mentioning that we deliberately diminish the searching time when comparing SA_{base} and SA for a more significant difference. Conclusively, we visualize and report algorithm comparisons in the form of side-by-side boxplots in Figures 4.5 to 4.7.

Figures 4.5 and 4.6 deliver the message that our proposed implementations of SA and GA manage to achieve better $\psi_1(D)$ than their baseline version, respectively. Among all algorithms studied, GA_{base} leads to the worst or most considerable $\psi_1(D)$ as well as the highest variation. Besides, our proposed simulated annealing algorithm, SA, generally outperforms the genetic algorithm, GA, although these two exhibit comparable results when the design size is moderately small.

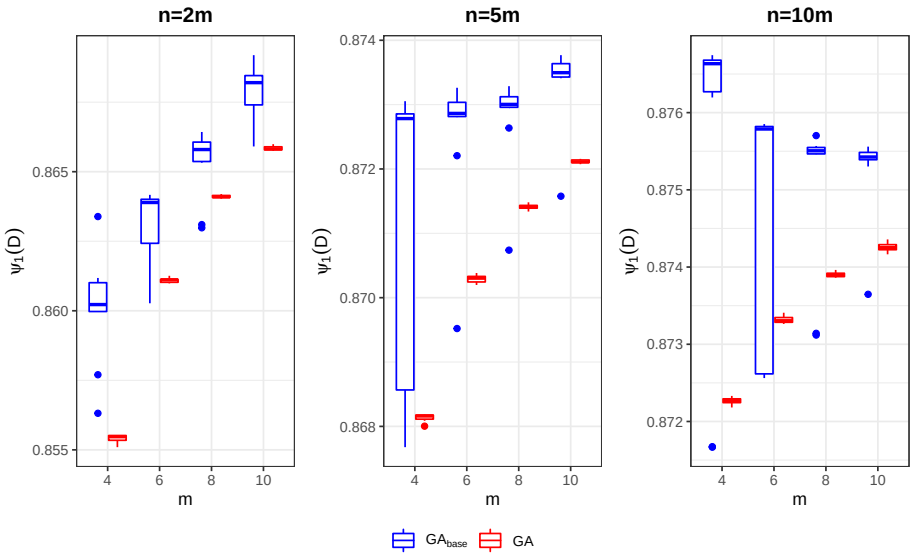

Figure 4.5: Comparisons of metaheuristic genetic algorithms in constructing Bayesian-inspired distance Latin hypercube designs ($\lambda = 1$) with disparate numbers of runs and factors.

4.5.2 Optimal Design Criteria

Furthermore, to justify the validity of the BID criterion, we consider comparisons among different optimal designs regarding their both space-filling properties and model-fitting performances. Existing space-filling designs considered include uniform designs (Fang et al., 2006) (ud), maximin distance designs (maximin), and maximum projection designs (maxpro). Additionally, random designs (rand) are adopted as a baseline for more pronounced comparisons.

Given a desired number of design runs n and factors m, the uniform Latin hypercube design is constructed using the R package `UniDOE` (A. Zhang et al., 2018) with centered L_2-discrepancy measure, the maximin Latin hypercube design is constructed via the `SLHD` package (Ba, 2015), and the maximum projection Latin hypercube design is generated with the help of the `MaxPro` package (Ba & Joseph, 2018). Last but not least, we run both Algorithm 1 and 2 in parallel and choose the Bayesian-inspired distance (bid) design with smaller $\psi_\lambda(D)$. Here, without loss of generality, we fix $\lambda = 1$ as well.

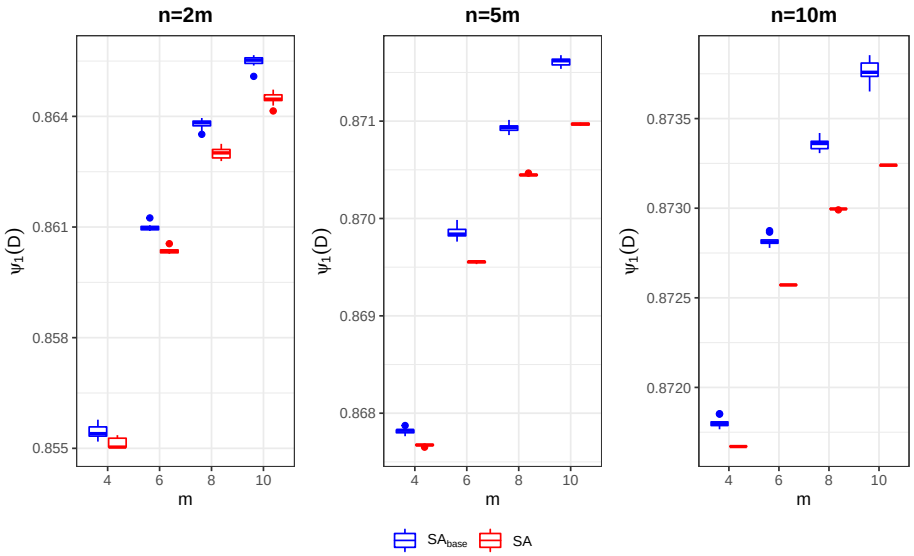

Figure 4.6: Comparisons of simulated annealing algorithms in constructing Bayesian-inspired distance Latin hypercube designs ($\lambda = 1$) with disparate numbers of runs and factors.

4.5.2.1 Space-filling property

An early-stage computer or physical experiment often involves many inert or less active factors (Woods & Lewis, 2017). Consequently, good space-filling properties in projection designs are desirable, especially for factor screening proposes.

In this section, we compare projection properties of different space-filling designs under five criteria: Euclidean distance, maximum projection criterion $\psi(D)$ defined in (4.1), BID criterion $\psi_1(D)$ defined in (4.6), relative centered L_2-discrepancy CD_2 and averaged column-wise correlation ρ_{ave}. The relative CD_2 value is computed as the difference of the CD_2 values between the corresponding design and the BID design. For each projection dimension $k \leq m$, we evaluate all $\binom{m}{k}$ projected designs and determine the worst case scenario respectively.

For illustrating purposes, results for 25×12 space-filling Latin hypercube designs are reported in Figure 4.8. Similar patterns persist in designs with varying sizes as well. First of all, not very surprisingly, random designs perform worst in all space-filling measures under every single projection dimension. Aside from that, every optimal design comparatively outperforms the others under

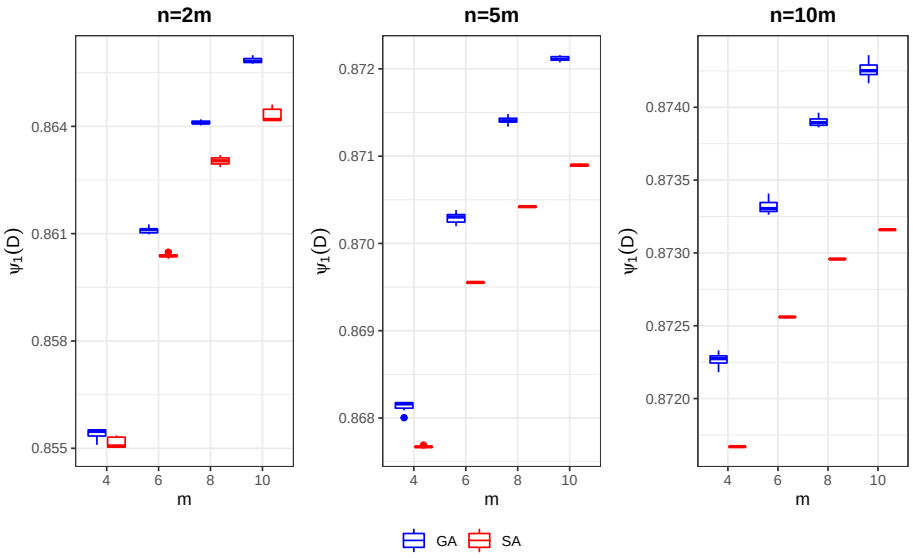

Figure 4.7: Comparisons of our proposed optimization algorithms in constructing Bayesian-inspired distance Latin hypercube designs ($\lambda = 1$) with disparate numbers of runs and factors.

their own space-filling measure both in the full and projected spaces, e.g., the uniform design, even under projection, achieves the smallest relative maximum CD_2 values. Furthermore, our proposed BID design demonstrates an appealing and robust pattern against different criteria and dimension sparsity, especially in terms of the averaged correlation between factors. Therefore, it is deemed robust and a promising initial design for various experimentation purposes.

4.5.2.2 Model-fitting performance

Apart from good space-filling properties in low dimensions, we are equally, if not more, interested in Gaussian Process (GP) model-fitting performances upon various optimal designs. More specifically, for every row, $\mathbf{x}_i$, in an optimal design, we conduct either physical or computer experiments and observe its corresponding response, y_i. Aggregating all $\{(\mathbf{x}_i, y_i) : i = 1, \cdots, n\}$ as the training dataset, we then use them to fit a GP model. The evaluation of the model-fitting performance is based on,

$$\text{SRMSE}_{test} = \frac{\sqrt{(1/N)\sum_{i=1}^{N}(\hat{y}(\mathbf{x}_{test}^{(i)}) - y(\mathbf{x}_{test}^{(i)}))^2}}{\sqrt{(1/N)\sum_{i=1}^{N}(\bar{y} - y(\mathbf{x}_{test}^{(i)}))^2}}, \tag{4.12}$$

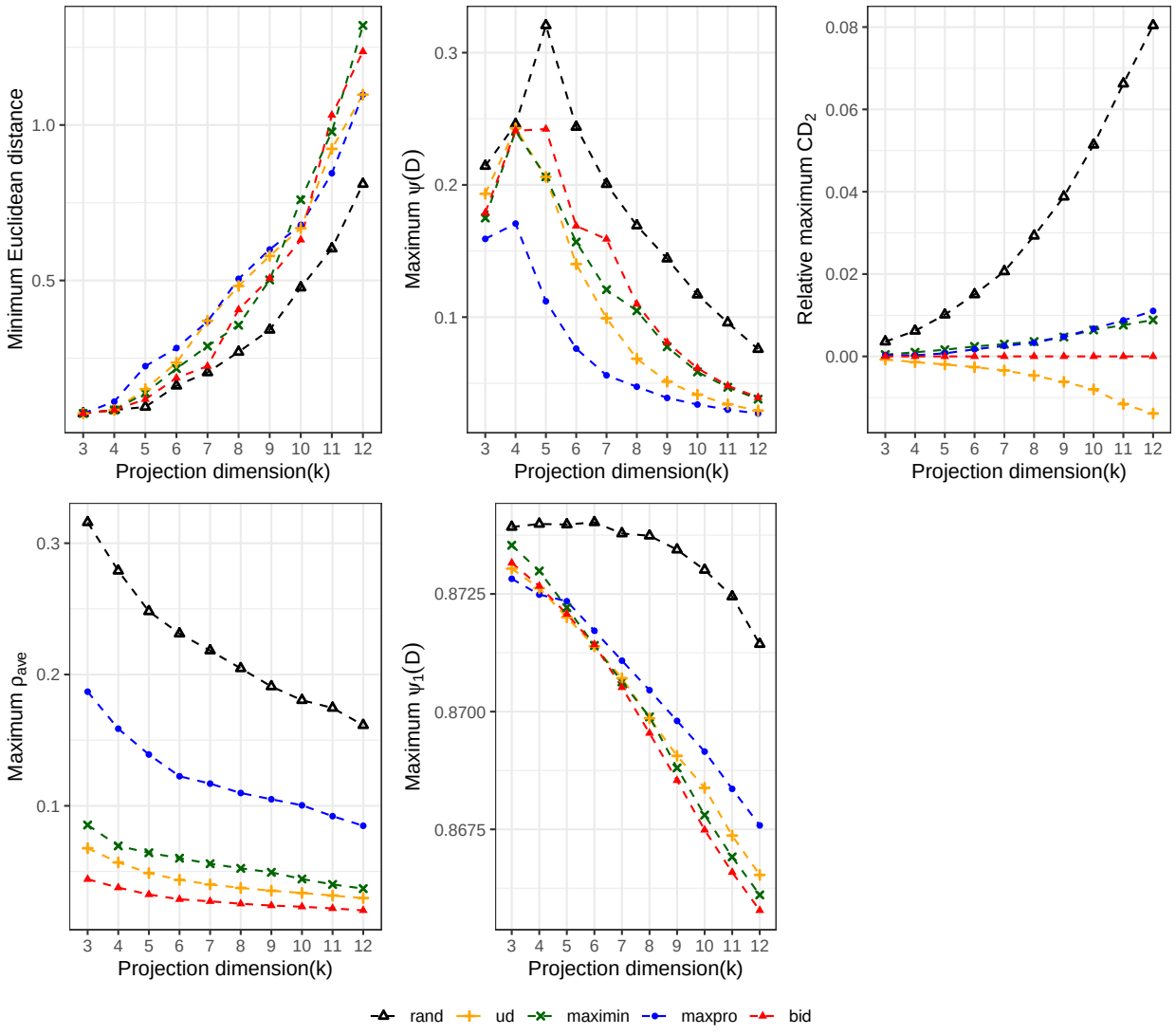

Figure 4.8: Comparisons of projection design properties under minimum Euclidean distance (the larger the better), maximum $\psi(D)$ (the smaller the better), relative maximum CD_2 (the smaller the better), maximum ρ_{ave} (the smaller the better), and maximum $\psi_1(D)$ (the smaller the better) criteria.

where $\mathbf{x}_{test}^{(i)}$ is the ith randomly selected test data point, $y(\mathbf{x}_{test}^{(i)}))$ represents its true response, and $\hat{y}(\mathbf{x}_{test}^{(i)})$ corresponds to its predicted response using our fitted GP model. Last but not least, $\bar{y}$ is the average of responses in the training dataset, $(1/n)\sum_{i=1}^{n} y_i$, which can be regarded as the prediction from an intercept only model. Thus, Equation (4.12) is a standardized version of the testing root mean squared error, which puts SRMSE_{test} roughly on $[0, 1]$ whatever the scale of y, with 1 indicating no better performance than the trivial predictor $\bar{y}$, and smaller values of SRMSE_{test} are desired.

As for the GP model itself, we impose a Gaussian covariance structure, and estimate correlation parameters by maximum likelihood via `DiceKriging` R package (Roustant et al., 2012). No nugget effect is added, since experiments considered are deterministic, with no observational noise. Finally, prediction made at any new test site is the universal kriging mean.

We consider a suite of simulation experiments instead of actual physical or computer ones and apply the GP model as a function approximation, in analogy to the experiment emulation. Simulation functions under investigation here include,

(a) Borehole (Worley, 1987): $f(\mathbf{x}) = \frac{2\pi T_u(H_u - H_l)}{\ln(r/r_w)\left[1 + \frac{2LT_u}{\ln(r/r_w)r_w^2 K_w} + \frac{T_u}{T_l}\right]}$, with

- $r_w = (0.15 - 0.05) * x_1 + 0.05$, $r = (50000 - 100) * x_2 + 100$,
- $T_u = (115600 - 63070) * x_3 + 63070$, $H_u = (1110 - 990) * x_4 + 990$,
- $T_l = (116 - 63.1) * x_5 + 63.1$, $H_l = (820 - 700) * x_6 + 700$,
- $L = (1680 - 1120) * x_7 + 1120$, $K_w = (12045 - 9855) * x_8 + 9855$;

(b) Linear with decreasing coefficients (Linkletter et al., 2006): $f(\mathbf{x}) = \sum_{k=1}^{8} 0.2\, x_k / 2^{k-1}$;

(c) Quadratic with two-way interactions: $f(\mathbf{x}) = \sum_{k=1}^{8} \left(\sum_{l \le k} x_l\right) x_k$;

(d) Trigonometric (Oakley & O'Hagan, 2004): $f(\mathbf{x}) = \mathbf{a_1}^\top \mathbf{x} + \mathbf{a_2}^\top \sin(\mathbf{x}) + \mathbf{a_3}^\top \cos(\mathbf{x}) + \mathbf{x}^\top \mathbf{M} \mathbf{x}$ with $\mathbf{a_1}, \mathbf{a_2}, \mathbf{a_3}, \mathbf{M}$ predetermined as constant vectors and matrix respectively;

where $\mathbf{x} = (x_1, x_2, \cdots, x_8)$ with $x_i \in [0, 1]$ for all $i = 1, \cdots, 8$.

For each simulation function, we consider scenarios when it possesses $k \leq m$ active variables (factors) by fixing the rest at their medium level (i.e., $x_i = 0.5$) as constants and calculate SRMSE_{test} based on $N = 1000$ randomly selected test data points respectively for different optimal designs.

Specifically, we compare GP model-fitting performances among five classes of 70×8 space-filling designs as in the preceding section, including random, maximin distance, maximum projection, uniform, and BID designs. Notice that all optimal design criteria remain unchanged after arbitrary column rearrangement. However, assigning a different design column to one particular factor will lead to slightly varying model-fitting and prediction results. Consequently, we opt to randomly assign columns of design matrices to input variables, with 100 repetitions to account for the variation.

Figure 4.9 summarizes our simulation study, highlighting the merits of our proposed BID designs concerning the GP model prediction. First of all, among all functions investigated, the BID design renders a decently small, if not the smallest, average testing SRMSE. Moreover, when there only exists exceptionally few active factors in the system, fitting a GP model with the BID design tends to be more stable than using other space-filling designs, i.e., assigning different portions of design columns to those unknown active factors will not cause dramatically diverging prediction accuracy.

Consequently, our proposed Bayesian-inspired distance designs are deemed as a robust initial design for the GP modeling, especially when inert factors exist in the system and cannot be identified before conducting the experiment.

4.6 Concluding Remarks

This chapter introduces a new optimal design criterion, the Bayesian-inspired distance criterion, for the Gaussian Process modeling by meticulously planning a more reasonable (exponential) prior

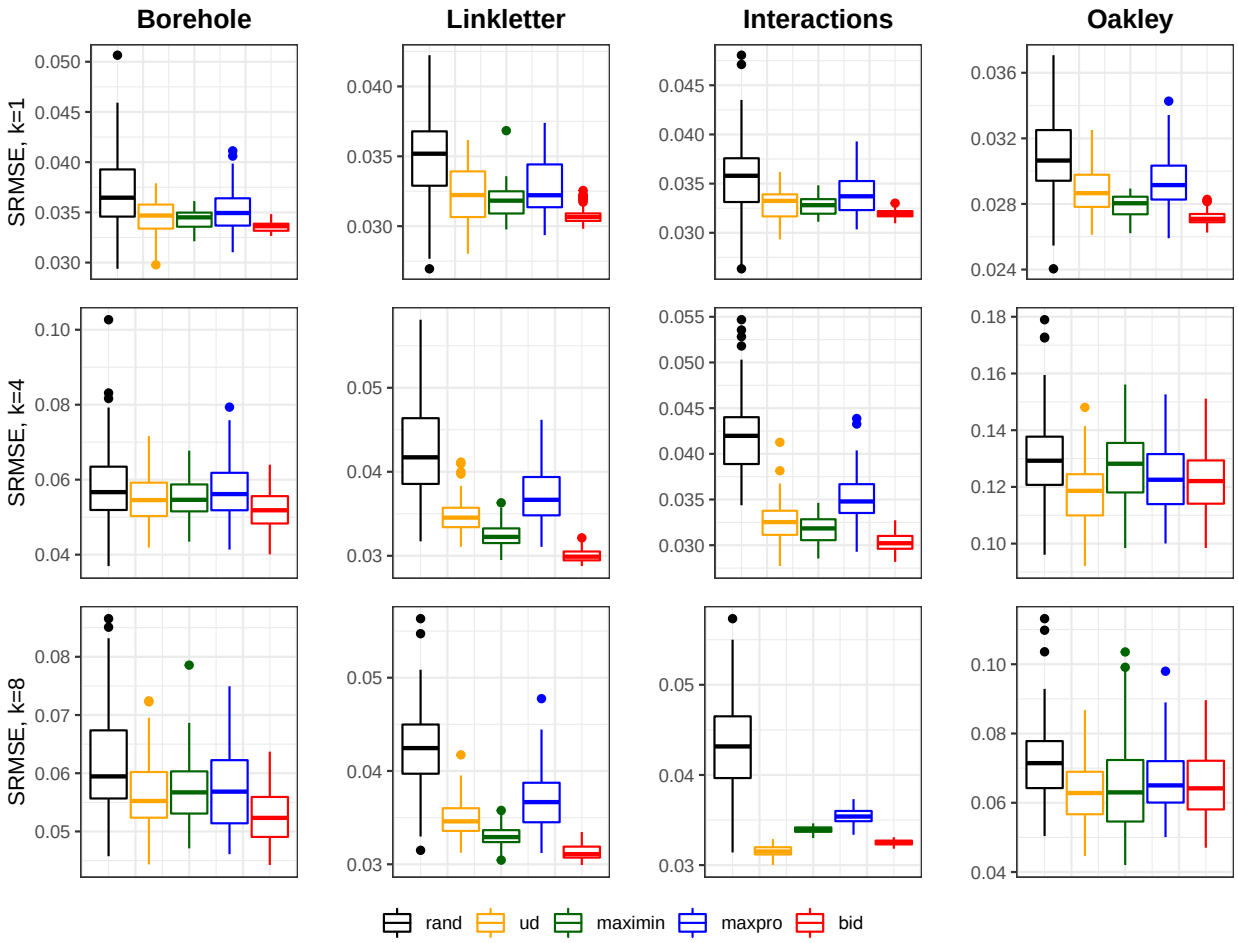

Figure 4.9: Extensive comparisons of GP model-fitting performances on various physical and simulation functions. The Bayesian-inspired distance design is constructed with rate parameter $\lambda = 0.1$, following the footsteps of tuning procedures proposed in Section 4.3.

imposed on the correlation parameters within. We revealed the intrinsic connections between this new criterion ψ_λ to other existing ones, including maximum projection and optimal moments criteria. Systematic approaches are introduced to selecting the best rate parameter λ of the exponential prior, regardless of whether direct visualization of the resulting optimal design is viable or not, given the design size. Moreover, we present two classes of metaheuristic algorithms, along with our novel implementations, to efficiently search for the corresponding optimal design once the rate parameter λ is fully determined. Extensive numerical results are demonstrated for comparing speed and quality of solution of different optimization algorithms, as well as illustrating merits of the Bayesian-inspired distance designs in a comprehensive manner.

CHAPTER 5

Conclusion

Gaussian Process models are increasingly being employed as surrogate models for approximating complex emulation systems due to their flexibility and capability of fitting non-smooth black-box functions. Maximin distance designs, well-known as a class of space-filling designs, propose to maximize the separation distance between any pair of design points. It is asymptotically D-optimal for the Gaussian Process modeling when certain conditions among observations are met. Existing algorithmic search for maximin distance designs has severely reduced performance when the design size grows moderately large. Consequently, we concentrate on systematic and easy-to-implement construction methods that generate maximin distance designs in a deterministic fashion.

The first class of maximin distance designs constructed in Chapter 2 are Latin squares, which is a special case of Latin hypercube designs (LHDs). Theoretical results show that some of the constructed designs are both (asymptotically) maximin L_1-distance and equidistant designs, which means that their pairwise L_1-distances are all equal and, therefore, they are also uniform projection designs. Aside from it, we illustrate procedures on generating more flexible LHDs when the number of design runs are substantially different from that of design factors, while retaining the maximin distance property. Chapter 3 introduces another class of maximin distance designs with the extra mirror-symmetry property among design points, which guarantees uncorrelated estimates of main and interaction effects. The proposed method first constructs a class of maximin balanced designs via a piece-wise linear transformation, and further rotates it to generate LHDs. We show via mathematical proofs that the rotation step keeps the maximin distance optimality of the generated balanced designs, providing a class of maximin LHDs. Both classes of maximin distance designs

are also (nearly) column-orthogonal, and we further highlight their robustness against different space-filling criteria through numerical comparisons.

Chapter 4 points out one major drawback of the maximin distance criterion, which is its poor space-filling performance on low-dimensional projections. This makes it undesirable in the case when many inert or less active factors exist in an early-stage experiment. To tackle this problem, we introduce a new optimal design criterion, so-called the Bayesian-inspired distance criterion, following the footsteps of the maximum projection criterion. The main contribution and novelty of our proposed criterion reflect in a more reasonable (exponential) prior imposed on the correlation parameters. We provide justifications of it over the original noninformative prior and discuss systematic approaches to optimally selecting the hyperparameter within. Moreover, we present novel implementations of two classes of metaheuristic algorithms to efficiently search for the corresponding optimal design once the hyperparameter is fully determined. Extensive numerical results are demonstrated for comparing speed and quality of solutions from different searching algorithms. The Bayesian-inspired distance criterion and its optimal designs look promising under many space-filling criteria and low-dimensional projections. Moreover, comprehensive simulation functions are investigated for verification of its accurate and consistent Gaussian Process model fitting and predictions.

BIBLIOGRAPHY

Ba, S. (2015). *Slhd: Maximin-distance (sliced) latin hypercube designs* [R package version 2.1-1]. https://CRAN.R-project.org/package=SLHD

Ba, S., & Joseph, V. R. (2018). *Maxpro: Maximum projection designs* [R package version 4.1-2]. https://CRAN.R-project.org/package=MaxPro

Ba, S., Myers, W. R., & Brenneman, W. A. (2015). Optimal sliced Latin hypercube designs. *Technometrics*, *57*(4), 479–487.

Butler, N. A. (2001). Optimal and orthogonal Latin hypercube designs for computer experiments. *Biometrika*, *88*(3), 847–857.

Carnell, R. (2022). *Lhs: Latin hypercube samples* [R package version 1.1.5]. https://github.com/bertcarnell/lhs

Chen, R.-B., Hsieh, D.-N., Hung, Y., & Wang, W. (2013). Optimizing Latin hypercube designs by particle swarm. *Statistics and computing*, *23*(5), 663–676.

Elsawah, A. M., Fang, K.-T., & Deng, Y. H. (2021). Some interesting behaviors of good lattice point sets. *Communications in Statistics - Simulation and Computation*, *50*(11), 3650–3668.

Fang, K.-T., Li, R., & Sudjianto, A. (2006). *Design and modeling for computer experiments*. Chapman; Hall/CRC.

Fang, K.-T., Liu, M.-Q., Qin, H., & Zhou, Y.-D. (2018). *Theory and application of uniform experimental designs*. Singapore; Beijing: Springer; Science Press.

Georgiou, S. D. (2009). Orthogonal Latin hypercube designs from generalized orthogonal designs. *Journal of Statistical Planning and Inference*, *139*(4), 1530–1540.

Georgiou, S. D., & Efthimiou, I. (2014). Some classes of orthogonal Latin hypercube designs. *Statistica Sinica*, *24*(1), 101–120.

He, X. (2019). Interleaved lattice-based maximin distance designs. *Biometrika*, *106*(2), 453–464.

Hedayat, A., Sloane, N., & Stufken, J. (1999). *Orthogonal arrays: Theory and applications*. Springer, New York.

Johnson, M. E., Moore, L. M., & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, *26*(2), 131–148.

Joseph, V. R., Gul, E., & Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, *102*(2), 371–380.

Joseph, V. R., & Hung, Y. (2008). Orthogonal-maximin Latin hypercube designs. *Statistica Sinica*, 171–186.

Keedwell, A. D., & Dénes, J. (2015). *Latin squares and their applications (2nd edition)*. New York: Academic Press.

Li, W., Liu, M.-Q., & Tang, B. (2021). A method of constructing maximin distance designs. *Biometrika*, *108*(4), 845–855.

Lin, C. D., & Tang, B. (2015). Latin hypercubes and space-filling designs. In A. Dean, M. Morris, J. Stufken, & D. Bingham (Eds.), *Handbook of design and analysis of experiments* (pp. 593–625). Chapman; Hall/CRC.

Linkletter, C., Bingham, D., Hengartner, N., Higdon, D., & Ye, K. Q. (2006). Variable selection for gaussian process models in computer experiments. *Technometrics*, *48*(4), 478–490.

McKay, M., Beckman, R., & Conover, W. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*, 239–245.

Moon, H., Dean, A., & Santner, T. (2011). Algorithms for generating maximin Latin hypercube and orthogonal designs. *Journal of Statistical Theory and Practice*, *5*(1), 81–98.

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, *43*(3), 381–402.

Oakley, J. E., & O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: A bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *66*(3), 751–769.

Pang, F., Liu, M.-Q., & Lin, D. K. J. (2009). A construction method for orthogonal Latin hypercube designs with prime power levels. *Statistica Sinica*, *19*(4), 1721–1728.

Roustant, O., Ginsbourger, D., & Deville, Y. (2012). DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, *51*(1), 1–55. https://www.jstatsoft.org/v51/i01/

Santner, T. J., Williams, B. J., & Notz, W. I. (2018). *The design and analysis of computer experiments (2nd edition)*. New York: Springer.

Shewry, M. C., & Wynn, H. P. (1987). Maximum entropy sampling. *Journal of Applied Statistics*, *14*(2), 165–170.

Steinberg, D. M., & Lin, D. K. J. (2006). A construction method for orthogonal Latin hypercube designs. *Biometrika*, *93*(2), 279–288.

Sun, F., Wang, Y., & Xu, H. (2019). Uniform projection designs. *The Annals of Statistics*, *47*(1), 641–661.

Tang, Y., & Xu, H. (2014). Permuting regular fractional factorial designs for screening quantitative factors. *Biometrika*, *101*(2), 333–350.

Wang, L., Sun, F., Lin, D. K. J., & Liu, M.-Q. (2018). Construction of orthogonal symmetric Latin hypercube designs. *Statistica Sinica*, *28*, 1503–1520.

Wang, L., Xiao, Q., & Xu, H. (2018). Optimal maximin L_1-distance Latin hypercube designs based on good lattice point designs. *The Annals of Statistics*, *46*(6B), 3741–3766.

Woods, D., & Lewis, S. (2017). Design of experiments for screening. https://doi.org/10.1007/978-3-319-12385-1_33

Worley, B. A. (1987). Deterministic uncertainty analysis. https://doi.org/10.2172/5534706

Xiao, Q., & Xu, H. (2017). Construction of maximin distance Latin squares and related Latin hypercube designs. *Biometrika*, *104*(2), 455–464.

Xiao, Q., & Xu, H. (2018). Construction of maximin distance designs via level permutation and expansion. *Statist. Sinica*, *28*, 1395–1414.

Ye, K. Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. *Journal of the American Statistical Association*, *93*(444), 1430–1439.

Ye, K. Q., Li, W., & Sudjianto, A. (2000). Algorithmic construction of optimal symmetric latin hypercube designs. *Journal of Statistical Planning and Inference*, *90*, 145–159.

Zhang, A., Li, H., Quan, S., & Yang, Z. (2018). *Unidoe: Uniform design of experiments* [R package version 1.0.2]. https://CRAN.R-project.org/package=UniDOE

Zhang, X.-R., Liu, M.-Q., & Zhou, Y.-D. (2020). Orthogonal uniform composite designs. *Journal of Statistical Planning and Inference*, *206*, 100–110.

Zhou, W., Yang, J.-F., & Liu, M.-Q. (2020). Optimal maximin l2-distance latin hypercube designs. *Journal of Statistical Planning and Inference*, *207*, 113–122.

Zhou, Y., & Xu, H. (2015). Space-filling properties of good lattice point sets. *Biometrika*, *102*(4), 959–966.