

Systematic optimization methods for uniform designs

Liangwei Qi

Donghua University

Changxing Ma

University at Buffalo

Yongdao Zhou

ydzhou@nankai.edu.cn

NITFID, Nankai University

Research Article

Keywords: Good lattice point set, Optimization method, Quasi-Monte Carlo method, Wrap-around L2-discrepancy

Posted Date: September 19th, 2025

DOI: <https://doi.org/10.21203/rs.3.rs-7572223/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.
[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Statistics and Computing on April 29th, 2026. See the published version at <https://doi.org/10.1007/s11222-026-10891-z>.

Systematic optimization methods for uniform designs

Liangwei Qi^{1,2}, Changxing Ma³, Yongdao Zhou^{2*}

¹School of Mathematics and Statistics, Donghua University, Shanghai, 201620, China.

²NITFID, School of Statistics and Data science, Nankai University, Tianjin, 300071, China.

³Department of Biostatistics, University at Buffalo, Buffalo, 14214, NY, USA.

*Corresponding author(s). E-mail(s): ydzhou@nankai.edu.cn;
Contributing authors: lwqi@dhu.edu.cn; cxma@buffalo.edu;

Abstract

As a successful practice of the quasi-Monte Carlo method in computer experiments, uniform design aims to distribute points evenly on a restricted domain. Such a point set has low discrepancy and has enjoyed increasing popularity in applications. However, most designs obtained by the numerical optimization algorithms in literature are just nearly uniform, thus there is significant room for improvement. This paper reviews the existing work on uniform designs and then characterizes their structure. Deterministic construction methods for uniform designs with any number of levels are theoretically proposed, which break through a common limitation of setting the level number to be a prime or a prime power. Based on the above, we provide a systematic optimization algorithm to search for uniform designs under more general parameters for practical use. Numerical experiments show that the performance and runtime of the proposed algorithm are far superior to existing ones.

Keywords: Good lattice point set, Optimization method, Quasi-Monte Carlo method, Wrap-around L_2 -discrepancy

1 Introduction

Quasi-Monte Carlo (QMC) methods have been successfully used for high-dimensional integrals arising in many applications. Mathematicians have applied this method into approximate numerical calculations for multiple integrals (Harase 2015). The key to a QMC method is to find a set of evenly distributed points in the space. Uniform design, formally proposed by Fang (1980); Wang and Fang (1981), is an important type of QMC methods. It has been widely employed in numerous areas, such as the establishment of multiple interpolation formulas, the approximate solutions of systems of some integrals or differential equations, the global extremes of the functions, the approximate representation points for some multivariate distributions, and some problems for statistical inference, including multivariate normality test and the sphericity test (Fang et al. 2006; Winker and Fang 1997). Uniform designs also play a significant role in the field of the experimental design. Such designs have a number of desirable statistical properties, such as admissibility, minimaxity and robustness (Fang et al. 2006). In contrast to classical experimental designs such as orthogonal arrays, uniform designs are practically useful because they can investigate many high-level factors simultaneously using fairly economical experimental runs. Thousands of industrial experiments with unknown models have effectively verified the rationality and feasibility of uniform designs; see Fang et al. (2018) for details.

The wrap-around L_2 -discrepancy (WD), introduced by Hickernell (1998), is a popularly used uniformity measure and provides a productive way to construct uniform designs. A design with WD-value arriving at its minimum is called uniform under WD (Fang et al. 2005). It was shown that constructing uniform designs is an NP-hard problem (Zhou et al. 2012). The construction methods are able to be mainly divided into deterministic construction methods and numerical optimization algorithms. The former is dedicated to theoretically obtaining designs with WD-value arriving at or asymptotically arriving at the lower bound; while the latter relies on existing fast algorithms to find designs with relatively small discrepancy quickly, and the resulted ones are usually nearly uniform.

As a popular used deterministic construction method, the good lattice point (GLP) method, proposed by Korobov (1959) for numerical evaluation of multivariate integrals, is the first theoretical method used for constructions of point sets with low discrepancy. The designs generated by GLP method are called the GLP sets. It was further proved that GLP sets tend to have low discrepancy (Hua and Wang 1981). The promotions of GLP method are gradually emerging, such as the GLP method with a power generator (PGLP) (Korobov 1959) and the cutting method (Ma and Fang 2004). The level permutation was shown to be able to improve the space-filling property of regular designs (Tang et al. 2012; Zhou and Xu 2015). The GLP method was further extended by linear codes (Qi et al. 2023), based on which a deterministic construction for uniform designs was provided when the number of levels is prime. Nevertheless, the existing literature still lacks construction methods for the cases of the level number being a non-prime or a non-prime power.

Many numerical optimization methods, such as the simulated annealing algorithm (Joseph and Hung 2008; Morris and Mitchell 1995), the threshold-accepting heuristic method (Fang et al. 2018; Winker and Fang 1997) and the integer programming

method (Fang and Ma 2001; Zhou et al. 2013), have been used to search uniform designs. Among these, the cutting method is compatible with WD due to its wrap-around property (Ma and Fang 2004). It cuts out a Latin hypercube design (LHD) with lower WD-value from a larger existing uniform one. Besides, one of the most efficient procedures for searching uniform designs in reality is a package in the statistical software R, named “UniDOE” developed by Zhang and Li (2017), which is based on a stochastic and adaptive threshold-accepting algorithm with flexible initialization, adaptive threshold and stochastic evolution. It is able to find a nearly uniform design in a short period of time; while as the design size increases, the runtime increases exponentially with the searched WD-value far away from its lower bound.

In this paper, we reveal the stability of the structure of uniform designs, and provide several deterministic construction methods for uniform designs whose WD-values directly arrive at the lower bound, including cases in which the level number is not a prime or a prime power. Moreover, a fast calculation for WD-value is given. Based on above, we propose a systematic optimization method (SOM) for searching nearly uniform designs with WD-value close to the lower bound under general parameters. Such an optimization method is easy to implement, and obtains better uniformity than the existing algorithms. The experimental results show that SOM is much more effective and more time saving than the UniDOE package.

The organization of this paper is as detailed below. Section 2 presents the necessary notations and existing popular construction methods. Section 3 characterizes the structure of uniform designs, and Section 4 proposes deterministic constructions for them, even when the level number is not prime or prime power. In Section 5, we display the flow of SOM and describe its details. Section 6 compares the optimization effect and the runtime of the SOM and the UniDOE package under the same parameters. Conclusions and discussions are given in Section 7. For brevity, all proofs are included in Appendix.

2 Preliminaries

2.1 Notations

An *asymmetrical* (or *mixed-level*) design of N runs, n factors and with levels $s_1, \dots, s_n$, is denoted by an $(N, s_1 \cdots s_n)$ -design $D = (d_{kl})_{N \times n}$. It is a set of N row vectors, or an $N \times n$ matrix where each row corresponds to a run and each column to a factor. The l th column takes the values from $\mathbb{Z}_{s_l} = \{0, \dots, s_l - 1\}$ where $l = 1, \dots, n$. An $(N, s_1 \cdots s_n)$ -design is *balanced* (or *U-type*), denoted by $U(N; s_1 \cdots s_n)$, if each level appears equally often in every column, which requires N to be a multiple of $s_1, \dots, s_n$. Denote the set of all $U(N; s_1 \cdots s_n)$'s as $\mathcal{U}(N; s_1 \cdots s_n)$. If s_l is equal to each other, i.e. $s_1 = \dots = s_n = s$, an (N, s^n) -design is called as *symmetrical*. A *Latin hypercube design* (LHD) is a symmetrical balanced design with $N = s$. For an LHD with N runs, an s -level balanced array can be obtained by replacing $\{i(N/s), i(N/s) + 1, \dots, (i+1)(N/s) - 1\}$ by the integer i for $i \in \mathbb{Z}_s = \{0, \dots, s - 1\}$ where N is a multiple of s , and this is referred to as the *level-collapse* method. By mapping $i \rightarrow (2i + 1)/(2s_l)$ for $i \in \mathbb{Z}_{s_l}$, the rows of a $U(N; s_1 \cdots s_n)$ are transformed to N points in $C^n = [0, 1]^n$. Denote the transformed design as $\tilde{U}(N; s_1 \cdots s_n)$ and the corresponding set as $\mathcal{U}(N; s_1 \cdots s_n)$.

A design $D \in \mathcal{U}(N; s_1 \cdots s_n)$ is called a *uniform design* under the predetermined discrepancy measure, if it has the minimum discrepancy value over $\mathcal{U}(N; s_1 \cdots s_n)$, i.e. its discrepancy arrives at the lower bound. For the sake of distinction, we refer to a design with discrepancy value close to the lower bound as *nearly uniform*. In this paper, we mainly focus on the common-used *wrap-around L_2 -discrepancy* (WD).

Next we take an overview of the relevant previous work on WD. For a design $D = (d_{kl})_{N \times n} \in \mathcal{U}(N; s^n)$, its WD-value usually refers to that of the transformed form $\tilde{D} = (x_{kl})_{N \times n} \in \mathcal{U}(N; s^n)$. Its squared WD-value is detailed below,

$$\text{WD}^2(D) = -\left(\frac{4}{3}\right)^n + \frac{1}{N^2} \sum_{k=1}^N \sum_{j=1}^N \prod_{l=1}^n \left(\frac{3}{2} - |x_{kl} - x_{jl}| + |x_{kl} - x_{jl}|^2 \right). \quad (1)$$

Obviously, WD is a function of products of $3/2 - |x_{kl} - x_{jl}| (1 - |x_{kl} - x_{jl}|)$'s, where the values of $|x_{kl} - x_{jl}|$'s is limited to a specific set for a balanced design (Fang et al. 2005). Note that we have $|d_{kl} - d_{jl}| (s - |d_{kl} - d_{jl}|) = s^2 |x_{kl} - x_{jl}| (1 - |x_{kl} - x_{jl}|)$, which naturally induces the connection between WD and the *Lee distance*. For two elements $d_1, d_2 \in \mathbb{Z}_s$, their Lee distance is defined as $\text{Lee}(d_1, d_2) = \min(|d_1 - d_2|, s - |d_1 - d_2|)$; see Zhou et al. (2008) for details and its derived Lee discrepancy. More precisely, the distance $|d_1 - d_2|$ and $s - |d_1 - d_2|$ have the same meaning under WD. Taking $s = 5$ as an example, $|d_{kl} - d_{jl}| = 1$ or 4 results in the same value of $|x_{kl} - x_{jl}| (1 - |x_{kl} - x_{jl}|)$ as $1 \times (s - 1) = 4 \times (s - 4)$. Then WD can be studied more thoroughly and conveniently by the Lee distance, and the definition of *equivalence columns* is proposed (Qi et al. 2023).

Definition 1. For the column vectors $v \in \mathbb{Z}_s^{k \times 1}$ and $s_k = (s, \dots, s)^\top$ of length k , we call v and $s_k - v$ as a pair of equivalence columns, denoted as $E_v = \{v, s_k - v\}$.

For convenience, let $E_v = \{v\}$ if $s_k = 2v$ and $E_{0_k} = \{0_k\}$. An equivalence of this sort classifies and pairs the all possible column vectors from $\mathbb{Z}_s^{k \times 1}$. For instance, $(4, 3, 2, 1, 0)^\top$ and $(4, 5, 6, 7, 0)^\top$ in $\mathbb{Z}_8^{5 \times 1}$ form an equivalence column pair, and $(4, 4, 4, 4, 4)^\top$ is self-equivalent. It is evident that the equivalence column of any column vector always exists and is unique. Moreover, if two columns are equivalent, the Lee distances between any two corresponding rows are the same. For $(4, 3, 2, 1, 0)^\top$ and $(4, 5, 6, 7, 0)^\top$ when $s = 8$, $\text{Lee}(3, 1) = \text{Lee}(5, 7)$ and $\text{Lee}(2, 0) = \text{Lee}(6, 0)$. This causes that the equivalence columns contribute equal values in the calculation of WD. Next, the count of equivalent columns is established. For a vector $v \in \mathbb{Z}_s^{N \times 1}$, the occurrences of E_v in a design $D = (v_1, \dots, v_n) \in \mathbb{Z}_s^{N \times n}$ is recorded as

$$\sigma_{E_v}(D) = \sum_{l=1}^n \mathbb{I}(v_l \in E_v). \quad (2)$$

Let $\sigma_v(D) = \sigma_{s_k - v}(D) = \sigma_{E_v}(D)$ for simplicity. Notably if $k = 1$, a column vector simplifies to an element. Correspondingly, the levels i and $s - i$ in $\mathbb{Z}_s$ are termed a pair of equivalence elements, recorded as $e_i = \{i, s - i\}$. Naturally, $e_{s/2} = \{s/2\}$ when s is even and $e_0 = \{0\}$. For example when $s = 6$, there are four pairs of

equivalence relations in $\mathbb{Z}_6$, $e_0 = \{0\}$, $e_1 = \{1, 5\} = \{1, -1\}$, $e_2 = \{2, 4\} = \{2, -2\}$ and $e_3 = \{3\} = \{-3\}$. Here we still use the symbol σ to represent the occurrences of e_i in a row. Specifically, for the k th run $r_k = (d_{k1}, \dots, d_{kn})$ in an s -level design $D = (d_{kl})_{N \times n}$, let $\sigma_i(r_k) = \sigma_{s-i}(r_k) = \sigma_{e_i}(r_k) = \sum_{l=1}^n \mathbb{I}(d_{kl} \in e_i)$. For the run $r = (1, 3, 0, 2, 1, 0, 1)$ in the sense of $s = 4$, it is concluded that $\sigma_0(r) = 2$, $\sigma_1(r) = \sigma_3(r) = 4$ and $\sigma_2(r) = 1$. According to the previously mentioned symbols, a particular row difference between design points is in the position to be given.

Definition 2. For an (N, s^n) -design D , we say that D satisfies the row difference condition, if the difference between any two distinct rows r_k and r_j , $r_{kj} = r_k - r_j \in \mathbb{Z}_s^{1 \times n}$, satisfies

$$\begin{cases} \sigma_0(r_{kj}) = \Delta_1, \sigma_1(r_{kj}) = \dots = \sigma_{\frac{s-1}{2}}(r_{kj}) = 2\Delta_2, & \text{odd } s; \\ \sigma_0(r_{kj}) = \Delta_1, \sigma_1(r_{kj}) = \dots = \sigma_{\frac{s}{2}-1}(r_{kj}) = 2\Delta_2, \sigma_{\frac{s}{2}}(r_{kj}) = \Delta_2, & \text{even } s, \end{cases} \quad (3)$$

where $\Delta_1 = (N - s)n / [(N - 1)s]$ and $\Delta_2 = Nn / [(N - 1)s]$.

This row difference (3) pushes forward an immense influence on the later research. In addition, the squared WD-value of a $\tilde{U}(N; s^n)$ has a lower bound, provided by Fang et al. (2005),

$$\begin{cases} \text{LB(WD}^2) = -\left(\frac{4}{3}\right)^n + \frac{1}{N} \left(\frac{3}{2}\right)^n + \frac{N-1}{N} \left(\frac{3}{2}\right)^{\Delta_1} \\ \quad \times \left\{ \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right] \dots \left[\frac{3}{2} - \frac{(s-1)(s+1)}{4s^2} \right] \right\}^{2\Delta_2}, & \text{odd } s; \\ \text{LB(WD}^2) = -\left(\frac{4}{3}\right)^n + \frac{1}{N} \left(\frac{3}{2}\right)^n + \frac{N-1}{N} \left(\frac{3}{2}\right)^{\Delta_1} \left(\frac{5}{4}\right)^{\Delta_2} \\ \quad \times \left\{ \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right] \dots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2} \right] \right\}^{2\Delta_2}, & \text{even } s. \end{cases} \quad (4)$$

On this basis, a sufficient condition for WD-value arriving at the minimum can be succinctly reformed as below (Fang et al. 2005).

Lemma 1. For an (N, s^n) -balanced design D , $\text{WD}^2(D)$ arrives at the lower bound (4), i.e. D is uniform under WD, if D satisfies the row difference condition (3).

Notice that Lemma 1 reveals the sufficiency but not the necessity. If a design does not meet (3), this does not mean that its WD-value fails to arrive at the minimum, because the lower bound (4) is not tight under some parameters. That is, (4) is always less than or equal to the actual minimum; and when WD^2 arrives at (4), this lower bound represents the minimum. Through Lemma 1, the uniformity verification has shifted from the complicated calculation of WD to a simple counting of row differences, which facilitates the judgment of whether a design is uniform.

Additionally, for some hyperparametric experiments with particularly multiple factors but limited number of runs, we apply a currently popular design, *component orthogonal array* (COA), to arrange a uniform experimental plan. A COA, denoted by $\text{COA}(\lambda N(N-1), N)$, is a $[\lambda N(N-1)] \times N$ matrix, where each row is a permutation of $\mathbb{Z}_N$, and for any two columns, each of the level combinations (i_1, i_2) with $i_1 \neq i_2 \in \mathbb{Z}_N$ appears λ times as a row; see [Huang \(2021\)](#) for details. Because of the pairwise balance between any two columns in the orders, COAs have broad generality for the order-of-addition models.

2.2 Existing constructions for uniform designs

This subsection introduces some popular constructions for uniform designs, including deterministic construction methods and numerical optimization methods. Existing uniform design tables have been mainly collected on the Web site <http://web.stat.nankai.edu.cn/cms-ud/UD/UniformDesign.html>.

2.2.1 Deterministic construction methods

For the one-factor experiments within a continuous range of levels on $[0, 1]$, the set of shifted points with equal spacing in between them is recommended by [Fang and Ma \(2001\)](#). While for the cases of multiple factors, the canonical design space $C^n = [0, 1]^n$ is too complicated and the discrepancy is in general a multimodal function on this space, which makes traditional optimization methods infeasible. To overcome this difficulty, the considered candidate set of designs is often chosen to be a well-structured subset of all possible ones. Symmetrical balanced designs from $\mathcal{U}(N; s^n)$ are often employed as the candidate design space.

Here we briefly introduce the GLP method, the PGLP method and their promotion by [Qi et al. \(2023\)](#). The key idea of the GLP method is to use a generator vector $r = (d_1, \dots, d_n)$. For $k = 1, \dots, N$, the k th row of the design is obtained by $kr = (kd_1, \dots, kd_n) \bmod N$, where the operation modulo N causes elements to fall into $\{0, \dots, N-1\}$. When d_l and N are coprime, i.e. their greatest common divisor $\gcd(d_l, N)$ is 1, the l th column becomes a permutation of $\{0, \dots, N-1\}$. Note that there is always a row of a GLP set being $(0, \dots, 0)$. However, choosing the best generator vector in the GLP method is time-consuming even for moderate N . It is suggested using generator vectors of the form $r = (1, d, \dots, d^{n-1}) \bmod N$, where $\gcd(d^l, N) = 1$ for $l = 1, \dots, n-1$ and $d < N$ ([Korobov 1959](#)). Such a method is called PGLP method, and has been successfully applied by [Wang and Fang \(1981\)](#) in searching low discrepancy designs. It is suitable for finding some nearly uniform designs with many runs but few factors. Obviously the power generator vector $r = (1, d, \dots, d^{n-1})$ satisfying the above requirements is usually not unique. Here we denote the maximum length of all possible generator vectors in the PGLP method as $\gamma(N)$. Obviously, this method can only be implemented when the required column numbers does not exceed $\gamma(N)$. Nevertheless, neither the GLP method nor the PGLP method can guarantee that the resulted discrepancy is minimized. [Qi et al. \(2023\)](#) extended the generator vector to an m -dimensional generator matrix by linear codes. For an s^m full factorial

design X on $\mathbb{Z}_s$ with a prime s , a generator matrix G is formed by selecting multiple columns from $X^\top$ according to their equivalent column relationships, and then the product $D = XG$ is calculated on $\mathbb{Z}_s$. The resulted designs, as they proved, are uniform under WD. The example below illustrates their ideas.

Example 1. For the $(9, 3^2)$ -full factorial design X ,

$$X = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}^\top,$$

and two alternative generator matrices G and G_h ,

$$G = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \end{pmatrix}, \quad G_h = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix},$$

$D_a = XG$ and $D_b = XG_h$ as below are checked to satisfy the row difference condition (3) and are thus uniform under WD,

$$D_a = XG = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 2 & 0 & 1 & 2 \\ 2 & 1 & 0 & 2 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \\ 0 & 0 & 2 & 2 & 2 & 1 & 1 & 1 \\ 1 & 2 & 2 & 0 & 1 & 1 & 2 & 0 \\ 2 & 1 & 2 & 1 & 0 & 1 & 0 & 2 \end{pmatrix}, \quad D_b = XG_h = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 \\ 2 & 0 & 2 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 \\ 2 & 1 & 0 & 2 \\ 0 & 2 & 2 & 2 \\ 1 & 2 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix}.$$

As expected, if the generator matrix degenerates to a row vector, i.e. $m = 1$, the resulted design actually becomes a uniform GLP set.

2.2.2 Numerical optimization methods

Up to present, most existing uniform designs with general parameters are attained through numerical optimization approaches. Despite the difficulty of searching the global optimal design with the minimum discrepancy in the discrete point set of R^{Nn} , plenty of scholars have made significant contributions to finding nearly uniform designs on the candidate set by heuristic global optimization algorithms, as introduced in Section 1. Such strategies sacrifice the minimum discrepancy to make the problem tractable. However, they are seldom efficient for large-scale designs due to the computational complexity, but large-scale designs are desired for computer experiments; for example, many simulation models involving hundreds of factors were employed by Morris (1991). Therefore, there is significant room for improvement in reducing the discrepancy and the runtime.

For designs with flexible parameters, especially for large-scale ones, we make use of the cutting method. For a large even moderate size of factors n , the GLP method

fails to generate a uniform design, as the maximum factor size has to be less than $\phi(N)/2 + 1$ where ϕ is the Euler function (Fang et al. 2018). While a nearly uniform LHD $U(N; N^n)$ with a non-prime N can be cut out from a larger one, as Ma and Fang (2004) suggested. Concretely, for an existing uniform LHD $D_0 \in \mathcal{U}(p; p^n)$, where $p \geq N$ and p or $p+1$ is a prime, let $\mathcal{P}$ be a proper range of $[0, 1]^n$ such that there are N points of D_0 fell on $\mathcal{P}$, and denote the set of these N points as D . From the theory of QMC methods, it is clear that D is uniformly scattered over $\mathcal{P}$. Therefore, the problem shifts to choosing a suitable rectangle $\mathcal{P}$ in $[0, 1]^n$, and after which the selected N points form a uniform or nearly uniform design D by linear transformations. It should be mentioned that the rectangle has wrap-around property; see Ma and Fang (2004) for details.

Many of the existing nearly uniform design tables are searched out by the UniDOE package in R, which is highly favored by engineers in practical projects. The input parameters include the run size N , the column size n , the number of levels s , the chosen discrepancy and the iterations, where N has to be a multiple of s . This package is able to search for a nearly uniform design relatively quickly. However, in practical applications, it has been found that this algorithm is extremely susceptible to the initial design and thus often falls into unfavorable local optima, whose WD-value is often far from its minimum.

3 Properties of uniform designs

This section focuses on uniform designs satisfying the row difference condition (3), proposes detailed descriptions of their structure and provides several construction methods, especially when factors are multitudinous. Firstly, the scale of a uniform design is characterized as below.

Corollary 1. *If an (N, s^n) -balanced design D satisfies the row difference condition (3), then*

$$\begin{cases} n = \lambda \left(\frac{N-1}{2} \right), & \text{if } N \text{ is odd,} \\ n = \lambda (N-1), & \text{if } N \text{ is even,} \end{cases} \quad (5)$$

where λ is a positive integer.

Corollary 1 is a generalization of Proposition 3 of Qi et al. (2023). This result establishes a keynote that, the allowable values for N and n are restricted when a uniform design satisfying (3) exists. For a given N , the minimum factor size for the experiment is recommended to be $(N-1)/2$ or $N-1$, depending on that N is odd or even. Furthermore, the same-scale uniform designs can be juxtaposed into one whose column number is increased several-fold. As an example, an (N, s^{2n}) -design (D, D) with doubled column size is uniform, based on the assumption that the (N, s^n) -design D satisfies (3). While it disregards the confounding among factors, so the juxtaposed designs usually require appropriate row permutations before column juxtaposition.

When n comes to be more flexible, the required design is able to be cut out from the one after some necessary random row permutations and column juxtapositions. For instance, if a $(9, 3^{11})$ -nearly uniform design is needed and here a $(9, 3^4)$ -uniform one XG_h has been obtained, as in Example 1, we firstly get a $(9, 3^{12})$ -uniform design $D_0 = (XG_h^{(1)}, XG_h^{(2)}, XG_h^{(3)})$ where $XG_h^{(i)}$ is a random row permutation design of XG_h , and select the optimal 11 columns from D_0 . Of course, setting $D_0 = (XG, XG_h^{(1)})$ is also advisable, in which XG is the $(9, 3^8)$ -design in Example 1. Here $\binom{12}{11}$ cutting frequencies may be acceptable, but the computations will become unbearable as the design size increases. The conclusion below provides a method to greatly reduce computational complexity of this problem.

Theorem 1. *If an (N, s^{n_1}) -design D_1 satisfies the row difference condition (3), arbitrarily juxtapose an (N, s^{n_2}) -design D_2 to D_1 , then for the resulted design (D_1, D_2) with $n = n_1 + n_2$ columns, there is a linear relationship between the squared WD-values of (D_1, D_2) and D_2 . Here $\text{WD}^2([D_1, D_2])$ is equal to*

$$-\left(\frac{4}{3}\right)^n + \frac{1}{N} \left(\frac{3}{2}\right)^n + K_s \left[\text{WD}^2(D_2) + \left(\frac{4}{3}\right)^{n_2} - \frac{1}{N} \left(\frac{3}{2}\right)^{n_2} \right], \quad (6)$$

where K_s is equal to

$$\begin{cases} \left(\frac{3}{2}\right)^{\Delta_1} \left\{ \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right] \cdots \left[\frac{3}{2} - \frac{(s-1)(s+1)}{4s^2} \right] \right\}^{2\Delta_2}, & \text{odd } s; \\ \left(\frac{3}{2}\right)^{\Delta_1} \left(\frac{5}{4}\right)^{\Delta_2} \left\{ \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right] \cdots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2} \right] \right\}^{2\Delta_2}, & \text{even } s, \end{cases} \quad (7)$$

and $\Delta_1 = (N-s)n_1/[(N-1)s]$ and $\Delta_2 = Nn_1/[(N-1)s]$.

Clearly, $\text{WD}^2([D_1, D_2])$ is proportional to $\text{WD}^2(D_2)$ as K_s is always positive. Then before column juxtapositions, we only need to focus on the uniformity of D_2 . Such as an example in which a $(9, 3^{4\lambda+3})$ -uniform design with a positive integer λ is required, we can firstly cut out an optimal $(9, 3^3)$ -sub-design from XG_h , and then juxtapose it and a $(9, 3^{4\lambda})$ -design $D_1 = (XG_h^{(1)}, \dots, XG_h^{(\lambda)})$ to form the final nearly uniform one. Compared with the previous example, this scheme reduces the calculation frequency from $\binom{12}{11}$ to $\binom{4}{3}$. In fact, the cutting frequency is irrelevant to the scale of D_1 , because the runtime is only spent cutting out D_2 from XG_h . Figure 3 in Section 6 shows this point. Additionally, the cutting frequency $\binom{4}{3}$ here can be reduced to once. That is, it is able to be directly constructed rather than selected out.

Corollary 2. *If an (N, s^n) -balanced design D_0 satisfies the row difference condition (3), deleting any column of D_0 results in the same WD-value. Denote the resulted*

(N, s^{n-1}) -balanced design as D , then if s is odd,

$$\begin{aligned} \text{WD}^2(D) = & -\left(\frac{4}{3}\right)^{n-1} + \frac{1}{N} \left(\frac{3}{2}\right)^{n-1} \\ & + \frac{N-s}{Ns} \left(\frac{3}{2}\right)^{\Delta_1-1} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2} \cdots \left[\frac{3}{2} - \frac{(s-1)(s+1)}{4s^2}\right]^{2\Delta_2} \\ & + \frac{2}{s} \left(\frac{3}{2}\right)^{\Delta_1} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2-1} \cdots \left[\frac{3}{2} - \frac{(s-1)(s+1)}{4s^2}\right]^{2\Delta_2} \\ & + \cdots \\ & + \frac{2}{s} \left(\frac{3}{2}\right)^{\Delta_1} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2} \cdots \left[\frac{3}{2} - \frac{(s-1)(s+1)}{4s^2}\right]^{2\Delta_2-1}; \end{aligned}$$

and if s is even,

$$\begin{aligned} \text{WD}^2(D) = & -\left(\frac{4}{3}\right)^{n-1} + \frac{1}{N} \left(\frac{3}{2}\right)^{n-1} \\ & + \frac{N-s}{Ns} \left(\frac{3}{2}\right)^{\Delta_1-1} \left(\frac{5}{4}\right)^{\Delta_2} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2} \cdots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2}\right]^{2\Delta_2} \\ & + \frac{1}{s} \left(\frac{3}{2}\right)^{\Delta_1} \left(\frac{5}{4}\right)^{\Delta_2-1} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2} \cdots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2}\right]^{2\Delta_2} \\ & + \frac{2}{s} \left(\frac{3}{2}\right)^{\Delta_1} \left(\frac{5}{4}\right)^{\Delta_2} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2-1} \cdots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2}\right]^{2\Delta_2} \\ & + \cdots \\ & + \frac{2}{s} \left(\frac{3}{2}\right)^{\Delta_1} \left(\frac{5}{4}\right)^{\Delta_2} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2}\right]^{2\Delta_2} \cdots \left[\frac{3}{2} - \frac{(s-2)(s+2)}{4s^2}\right]^{2\Delta_2-1}, \end{aligned}$$

where $\Delta_1 = (N-s)n/[(N-1)s]$ and $\Delta_2 = Nn/[(N-1)s]$. Moreover, $\text{WD}^2(D)$ converges to the lower bound (4) if $N \rightarrow \infty$, i.e. D is asymptotically uniform.

By this result, when an (N, s^{n-1}) -uniform design happens to be needed, any deletion of one column in D_0 is practicable. Continuing with the assumption that a $(9, 3^{11})$ -nearly uniform design is required, we gain a $(9, 3^{12})$ -uniform design $D_0 = (XG_h^{(1)}, XG_h^{(2)}, XG_h^{(3)})$ and delete any column directly. In this way, the squared WD-value is 6.46314. Under the same parameters, that obtained through UniDOE package with 2000 iterations is 6.52417, and the lower bound $\text{LB}(\text{WD}^2) = 6.41505$. Obviously, our result is closer to the minimum. Corollary 2 implies that, even if a minor modification is conducted on uniform designs, their structure ensures that the uniformity will not change much, notably when they are of a large size. However, it should be emphasized that if two or more columns are randomly deleted from D_0 , the resulted sub-designs usually have different WD-values, as the level combinations of the deleted columns may be diverse. Conversely, if we arbitrarily add a balanced column to a

uniform design, the WD-values of all possible resulted ones are the same and converge asymptotically to the minimum (4). The added column should avoid aliasing with effects valued in the original design. The end of this section reveals that, uniform designs with equivalent structures are able to be split into two uniform sub-ones.

Corollary 3. *For an (N, s^n) -design D with an even n and without repetitive columns, if D can be split into two $(N, s^{n/2})$ -designs, $D = (D_1, D_2)$, satisfying that for any column in D_1 , there is an equivalent column corresponding to it one by one in D_2 , then D satisfies the row difference condition (3) if and only if both sub-designs D_1 and D_2 satisfy (3).*

If the column size $n_0 \leq n/2$ is required in practice, then for an n -column design D with the above specific structure, the wanted (N, s^{n_0}) -nearly uniform one can be cut out from D_1 or D_2 , reducing the cutting frequency from $\binom{n}{n_0}$ to $\binom{n/2}{n_0}$.

4 Deterministic constructions for uniform designs with any level

For prime-level factorials, the GLP method was extended by linear codes (Qi et al. 2023) and plenty of uniform designs were constructed. Nonetheless, there are currently few deterministic constructions for non-prime-level cases. This section is committed to filling this gap. One prevailing way to arrange input settings of quantitative factors is to use LHDs. Here COA is regarded as a tool to construct a uniform LHD with any level.

Theorem 2. *For a positive integer λ , if an $(s, s^{\lambda s(s-1)})$ -LHD D is a transpose of a COA, i.e. $D^\top$ is a COA($\lambda s(s-1), s$), then $\text{WD}^2(D)$ arrives at its minimum (4).*

Example 2. *Consider a $(5, 5^{20})$ -LHD D below, obviously $D^\top$ is a COA(20, 5),*

$$D = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 & 1 & 4 & 0 & 1 & 2 & 3 & 1 & 2 & 3 & 4 & 0 & 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 & 3 & 4 & 0 & 1 & 2 & 2 & 3 & 4 & 0 & 1 & 1 & 2 & 3 & 4 & 0 \\ 3 & 4 & 0 & 1 & 2 & 1 & 2 & 3 & 4 & 0 & 4 & 0 & 1 & 2 & 3 & 2 & 3 & 4 & 0 & 1 \\ 1 & 2 & 3 & 4 & 0 & 2 & 3 & 4 & 0 & 1 & 3 & 4 & 0 & 1 & 2 & 4 & 0 & 1 & 2 & 3 \end{pmatrix}.$$

It can be checked that D satisfies (3), and thus it is uniform under WD.

The constructions of COAs with any number of components, breaking through the limitation that the component number was usually a prime or a prime power, has been proposed by Huang (2021). Therefore, uniform designs with any level number can be correspondingly constructed by Theorem 2. Despite this fact, it comes at the cost of an excessive number of factors, because a COA often introduces too many runs. For example, the minimum run size of a COA with 6 components is 60 (Yang et al. 2021), which means its transposition, a uniform LHD with 6 runs, includes at least 60

factors. It is generally less considered in real experiments. The following constructions contribute to the reduction of factor size.

Corollary 4. *Add a level to the end of each column of the transpose of a $\text{COA}(\lambda s(s-1), s)$, then the squared WD-value of the resulted $(s+1, (s+1)^{\lambda s(s-1)})$ -LHD arrives at its minimum (4).*

A $(6, 6^{20})$ -uniform LHD can be obtained by adding a level to the end of each column of the $(5, 5^{20})$ -LHD in Example 2. The column size is reduced to 20 from 60, whose effect is quite impressive. Next, we try to further reduce the scale $(6, 6^{20})$ to $(6, 6^{10})$. For an odd prime s , the following construction halves the factor size in Theorem 2 and Corollary 4.

Construction 1.

Step 1. *For an odd prime s , construct an $s \times [s(s-1)]$ array, namely D_0 , whose rows are labeled with all elements in $\mathbb{Z}_s$ and whose columns are labeled by the $s(s-1)$ polynomials over $\mathbb{Z}_s$ of degree one.*

Step 2. *Denote the polynomials (in the variable x , say) in Step 1 as $\phi_1, \dots, \phi_{s(s-1)}$. For $l = 1, \dots, s(s-1)$, ϕ_l can be expressed in the form $kx + a$ with $k = 1, \dots, s-1$ and $a = 0, \dots, s-1$. The element of D_0 in the row labeled $i \in \mathbb{Z}_s$ and the column labeled ϕ_l is defined to be $\phi_l(i)$, i.e. the value of the polynomial ϕ_l at i .*

Step 3. *Divide these $s(s-1)$ polynomials into two groups such that for any $k = 1, \dots, s-1$, (1) the polynomials $\{kx, kx+1, \dots, kx+(s-1)\}$ appear in the same group; (2) the polynomials $kx+a_1$ and $(s-k)x+a_2$ appear in different groups with $a_1, a_2 = 0, \dots, s-1$. Use either group to form an $s \times [s(s-1)/2]$ matrix, named D_1 .*

Step 4. *If needed, an additional level can be added at the end of each column in D_1 , that is, a run $(s, \dots, s)$ is added to D_1 . Denote the resulted $(s+1, (s+1)^{s(s-1)/2})$ -design as D_2 .*

Here D_0 is just the transpose of a $\text{COA}(s(s-1), s)$, which is the same as that created by Construction 1 of Huang (2021), so the first two steps propose a specific construction for an $(s, s^{s(s-1)})$ -uniform LHD, as an application of Theorem 2. Considering that s is an odd prime, for any polynomial ϕ_{l_1} in the set $\{\phi_1, \dots, \phi_{s(s-1)}\}$, there always exists a ϕ_{l_2} such that $\phi_{l_1} + \phi_{l_2} = 0$. As an example, for the polynomial $2x+1$ when $s=5$, $3x+4$ makes the sum of these two polynomials equal to 0, regardless of the value of x . Consequently, D_0 can always be split into two $(s, s^{s(s-1)/2})$ -sub-designs, ensuring the feasibility of Step 3.

Theorem 3. *For the $(s, s^{s(s-1)/2})$ - D_1 and $(s+1, (s+1)^{s(s-1)/2})$ - D_2 obtained by Construction 1, both $\text{WD}^2(D_1)$ and $\text{WD}^2(D_2)$ arrive at the minimum (4).*

Example 3. Let D_0 be the $(5, 5^{20})$ -LHD in Example 2. Then Construction 1 constructs a $(5, 5^{10})$ -LHD D_1 by Step 3, and a $(6, 6^{10})$ -LHD D_2 by Step 4,

$$D_1 = \begin{pmatrix} 0 & 2 & 4 & 3 & 1 \\ 1 & 3 & 0 & 4 & 2 \\ 2 & 4 & 1 & 0 & 3 \\ 3 & 0 & 2 & 1 & 4 \\ 4 & 1 & 3 & 2 & 0 \\ 0 & 4 & 3 & 1 & 2 \\ 1 & 0 & 4 & 2 & 3 \\ 2 & 1 & 0 & 3 & 4 \\ 3 & 2 & 1 & 4 & 0 \\ 4 & 3 & 2 & 0 & 1 \end{pmatrix}^T, \quad D_2 = \begin{pmatrix} 0 & 2 & 4 & 3 & 1 & 5 \\ 1 & 3 & 0 & 4 & 2 & 5 \\ 2 & 4 & 1 & 0 & 3 & 5 \\ 3 & 0 & 2 & 1 & 4 & 5 \\ 4 & 1 & 3 & 2 & 0 & 5 \\ 0 & 4 & 3 & 1 & 2 & 5 \\ 1 & 0 & 4 & 2 & 3 & 5 \\ 2 & 1 & 0 & 3 & 4 & 5 \\ 3 & 2 & 1 & 4 & 0 & 5 \\ 4 & 3 & 2 & 0 & 1 & 5 \end{pmatrix}^T.$$

Here D_1 and D_2 both satisfy (3) and are uniform under WD.

In line with Corollary 1, the minimum column size of a uniform LHD with 6 runs is 5 in theory. But due to the distinctiveness of non prime and non prime powers, the above $(6, 6^{10})$ -LHD is currently the best choice. Fortunately when $s = 4$, a $(4, 4^3)$ -uniform LHD obtained by Construction 1 minimizes the column size.

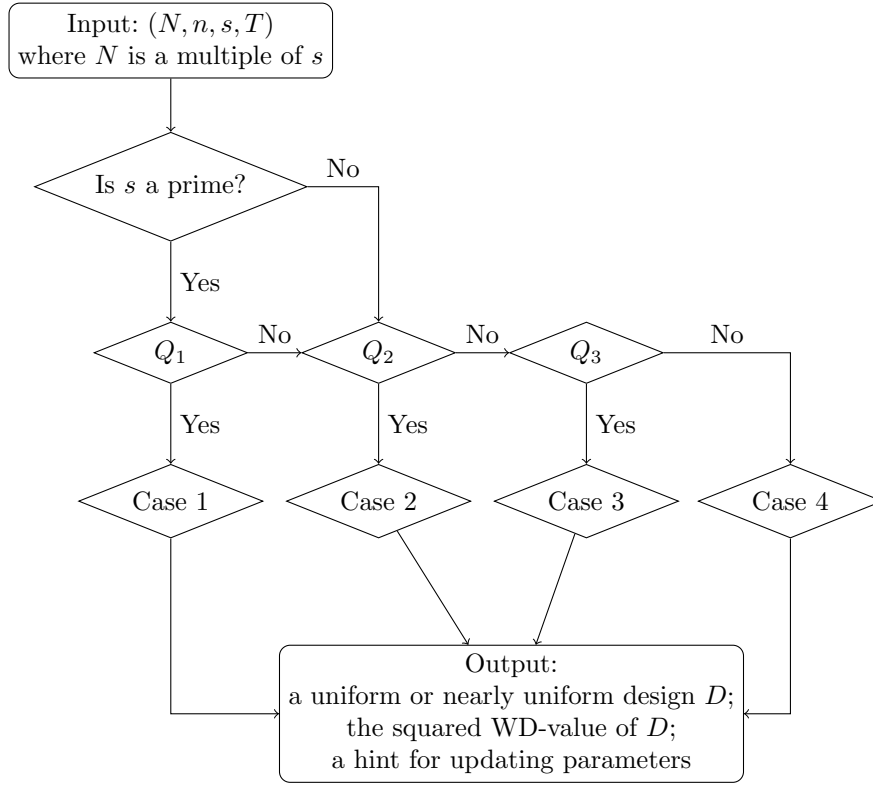
The above discuss the constructions of uniform LHDs, but more common situations with $N \neq s$ in practice are still an urgent need to be addressed. Next, we apply the level collapsion to the previously obtained designs for more general options of s .

Corollary 5. If an $(s, s^{\lambda s(s-1)})$ -LHD D_0 is the transpose of a COA, i.e. $D_0^T = \text{COA}(\lambda s(s-1), s)$ where $s = ks_0$ and $k \geq 2$, collapse the level s to s_0 , then the resulted $(s, s_0^{\lambda s(s-1)})$ -design D is balanced and $\text{WD}^2(D)$ arrives at its minimum (4).

Notice that s_0 is able to be any divisor of s , which results in a flexibility. In fact, the level collapsion also works for other uniform LHDs obtained previously.

Corollary 6. For the $(s+1, (s+1)^{\lambda s(s-1)})$ -LHD from Corollary 4 or the $(s+1, (s+1)^{s(s-1)/2})$ -LHD from Construction 1, if $s+1 = ks_0$ where $k \geq 2$, collapse the level $s+1$ to s_0 , then the resulted $(s+1, s_0^{\lambda s(s-1)})$ - or $(s+1, s_0^{s(s-1)/2})$ -design D is balanced and $\text{WD}^2(D)$ arrives at its minimum (4).

The proof of Corollary 6 is quite similar to that of Corollary 5, so it is omitted here. As Huang (2021) pointed out, a $\text{COA}(s(s-1), s)$ with $\lambda = 1$ always exists for an odd prime s . This implies that the column size of D in Corollary 6 can be reduced to $s(s-1)/2$, which is not unbearable. Given that odd prime numbers plus 1 are all composite, Corollary 6 thus has wide applicability.



Notes:

1. “ Q_1 ” means that “Is N a power of s ?”
2. “ Q_2 ” represents that “Is $n \leq \gamma(N)$ and $N/n > 10$?” where $\gamma(N)$ is the maximum length of all possible generator vectors in the PGLP method.
3. “ Q_3 ” means that “Is $N - 1$ a prime with $N > 3$ and n or $n - 1$ or $n + 1 \bmod (N - 1)(N - 2)/2 = 0$?”

Fig. 1 Flowchart of SOM

5 SOM for uniform designs

This section offers a *systematic optimization method* (SOM) to seek uniform or nearly uniform designs. Its flowchart is detailed in Figure 1, in which different processes determine the applications of different theories. It always assumes that N is a multiple of s in the input parameters (N, n, s, T) . Here T is the maximum of iterations, with the fact that the actual iterations in SOM is always less than or equal to T . In each case, SOM outputs a uniform or nearly uniform design, its squared WD-value, and a hint of updating parameters for better results.

For some parameters, as in Case 1, Case 2 and Case 3, SOM firstly constructs a uniform design with N runs, and then cuts out the required n columns. With respect to more general parameters in Case 4, SOM firstly constructs a uniform GLP set

with larger size, and then cuts both rows and columns to get the desired nearly uniform design. Logically, the larger the parameter, the better the effect, assured by the structure of uniform designs.

Case 1: Cutting columns from a GLP set or its promotion

When s is a prime and N is a power of s , a uniform design, with $N-1$ columns when $s = 2$ and with $(N-1)/2$ columns when s is odd, is able to be directly constructed, referring to Corollary 1 and Qi et al. (2023). In the following, the optimization method in which s is an odd prime will be thoroughly analyzed; and that where $s = 2$ is quite similar and is thus omitted. Denote the above $(N, s^{(N-1)/2})$ -uniform design as XG_h . Since the factor size n is arbitrary, it is divided into three situations below for analysis.

If $n \bmod (N-1)/2 = 0$, an effective idea is to obtain $(XG_h^{(1)}, \dots, XG_h^{(\lambda)})$ where $\lambda = 2n/(N-1)$ and $XG_h^{(i)}$ is a random row permutation design of XG_h . Its squared WD-value, equal to the minimum (4), is directly given. Moreover, a hint “This WD-value arrives at its minimum!” is provided. And if $n+1$ or $n-1 \bmod (N-1)/2 = 0$, by Corollary 2 and its generalization, we randomly remove or add a balanced column on the (N, s^{n+1}) - or (N, s^{n-1}) -uniform design $(XG_h^{(1)}, \dots, XG_h^{(\lambda)})$ to gain an (N, s^n) -nearly uniform one. The squared WD-value is also directly provided, with a hint “Setting the column size as $n+1$ (or $n-1$) will be a better choice!”. For example, if a $(9, 3^7)$ -nearly uniform design is required, remove one column from the $(9, 3^8)$ -uniform design $(D_b^{(1)}, D_b^{(2)})$ where $D_b = XG_h$ as in Example 1. The resulted squared WD-value is 0.96708, compared with 0.97684, that searched out by UniDOE package through 2000 iterations, and the lower bound 0.95172 under the same parameters.

In the two situations above, in which the required design is directly constructed, the search time is saved and the WD-value is equal to or extremely close to the lower bound. Note the output designs are not always the same, since that there is a row permutation step. But its impact is almost able to be ignored.

While if $1 < n \bmod (N-1)/2 < [(N-1)/2 - 1]$, it is necessary to juxtapose some XG_h ’s to cover (N, n) , from which the required design is cut out. Let $\lambda = \min\{i : i \times (N-1)/2 \geq n\}$, so that $n_0 = \lambda(N-1)/2$ covers n . Let $D = (D_1, D_2)$ where $D_1 = (XG_h^{(1)}, \dots, XG_h^{(\lambda-1)})$ and D_2 is a sub-design with $n' = n - (\lambda-1)(N-1)/2$ optimal columns cut out from $XG_h^{(\lambda)}$. Now by Theorem 1, the problem becomes choosing the optimal n' columns from $XG_h^{(\lambda)}$, and there are $\binom{(N-1)/2}{n'}$ selecting plans overall, leading to a long runtime. In reality, the selection range can be reasonably reduced. We refer to and improve the cutting method (Ma and Fang 2004). List all consecutive sorted n' numbers in $\{1, \dots, (N-1)/2\}$ as the required column labels, and pick these corresponding columns in $XG_h^{(\lambda)}$. This scheme reduces the computations from $\binom{(N-1)/2}{n'}$ to $(N-1)/2 - n' + 1 = (N-2n'+1)/2$. Note that the actual iteration in SOM is $\min\{T, (N-2n'+1)/2\}$, which means that if $T < (N-2n'+1)/2$, SOM only compares the first T candidate sub-designs. It ensures that the runtime of SOM will not be too long. Although the candidate sets are greatly reduced, the actual numerical simulations show that there is little difference in the uniformity of obtained designs. Finally, a hint “Setting the column size as $(\lambda-1)(N-1)/2$ or $\lambda(N-1)/2$ will be a better choice!” is given. For example, to get a $(9, 3^{10})$ -nearly uniform design, regard the $(9, 3^4)$ -design D_b in Example 1 as XG_h and gain a $(9, 3^8)$ -design D_1 . Next, we

traverse all sets of consecutive sorted column labels with a length of $n' = 10 - 8 = 2$ in the list $\{1, 2, 3, 4\}$ and obtain $\{1, 2\}$, $\{2, 3\}$ and $\{3, 4\}$, then choose the one with the minimum WD-value from these 3 sub-designs of D_b (if $T \geq 3$), recorded as D_2 . The squared WD-value of (D_1, D_2) is 4.09795, compared with 4.10989, that searched out by the UniDOE package through 2000 iterations, and the lower bound 4.04843 under the same parameters.

Case 2: Cutting columns from a PGLP set

Case 1 is extremely effective for input parameters of $N/n \leq 2$, but it may be powerless when N/n is large. Here we apply the PGLP method to make up for this shortcoming. Denote the set of all possible lengths of generator vectors as $\{L_1, L_2, \dots\}$. Then the PGLP method becomes available only when $n \leq \gamma(N) = \max\{L_1, L_2, \dots\}$. Under the assumptions of Case 2, an $(N, N^{\gamma(N)})$ -nearly uniform PGLP set can be constructed, denoted as D_0 . Then we select the optimal n columns from D_0 and collapse the level N to s , similar to the third situation in Case 1. Note that here the actual iteration in SOM is $\min\{T, \gamma(N) - n + 1\}$. Finally, we recommend increasing s if $N > s$; or setting N and s to a larger prime if $N = s$.

For example, to obtain an $(1002, 1002^{60})$ -nearly uniform design, we have $\{L_1, L_2, L_3\} = \{166, 83, 2\}$, so $60 = n \leq \gamma(1002) = 166$. Let D_0 be an $(1002, 1002^{166})$ -nearly uniform LHD by the PGLP method. The squared WD-value of the optimal 60 columns cut out from D_0 is 35737930 ($T = 200$), compared with 36023529, that searched out by the UniDOE package through 2000 iterations, and 36624577, that cut out from an $(1009, 1009^{504})$ -uniform LHD by the following Case 4, and the lower bound 33712325 under the same parameters. A hint “Setting N and s as 1009 will be a better choice!” is given.

Case 3: Cutting columns from a transposed COA

For multi-factor experiments, the column size n is often much larger than the row size N . Under these parameters, applying the constructions in Section 4 is ideal. Record the uniform LHD with N runs and $(N - 1)(N - 2)/2$ columns in Construction 1 as XG_h . Implement the necessary random row permutations, column juxtapositions and level collapses to gain a uniform design $D_0 = (XG_h^{(1)}, \dots, XG_h^{(\lambda)})$ with N runs and s levels. If $n \bmod (N - 1)(N - 2)/2 = 0$, D_0 is exactly the desired result. Its squared WD-value, equal to the minimum (4), is directly given, and we provide a hint “This WD-value arrives at its minimum!” While if $n + 1$ or $n - 1 \bmod (N - 1)(N - 2)/2 = 0$, we randomly remove or add a balanced column on D_0 with $n + 1$ or $n - 1$ columns. The squared WD-value is also directly given by Corollary 2, with a hint “Setting the column size as $n + 1$ (or $n - 1$) will be a better choice!”.

Case 3 is similar to Case 1 and requires no runtime for searching. For instance, if a $(6, 6^9)$ -uniform design is needed, it can be gained by deleting one column of the $(6, 6^{10})$ -uniform one from Theorem 3. Its squared WD-value is 2.22554; while under the same parameters, the searched design by the UniDOE package through 2000 iterations has a squared WD-value 2.25353, and the lower bound is 2.21982. Notice that the strategy in Case 3 is also effective when $N = s$ is a prime. But in this case, the transposed COA is a column juxtaposition of some uniform GLP sets, as a special circumstance of Case 1. Additionally, the condition “ n or $n - 1$ or $n + 1 \bmod (N - 1)(N - 2)/2 = 0$ ”

ensures that the structure of the resulted design will not differ significantly from D_0 . If this condition is not met, Case 3 is often less effective than following Case 4.

Case 4: Cutting both rows and columns from a larger-size GLP set

For more general parameters (N, n, s) , we firstly cut out a nearly uniform LHD from a larger existing uniform one, and then collapse the level N to s to gain the needed design. This cutting method is always able to be implemented, no matter s is prime or not. The primary task is to find a suitable uniform LHD to cover (N, n) . Let $p = \min\{i : N < i \text{ and } i \text{ is a prime}\}$ and p is clearly odd. Then a (p, p^{p-1}) - and a $(p, p^{(p-1)/2})$ -uniform LHD, still recorded as XG and XG_h respectively, can be constructed by the GLP method (Qi et al. 2023). Unlike Case 1 which only utilizes XG_h , the discussions below consider both XG and XG_h .

If $0 < n \bmod (p-1) \leq (p-1)/2$, firstly juxtapose as many XG 's as possible after random row permutations and then juxtapose a random row permutation design of XG_h to form a preliminary design D_0 . Specifically, let $\lambda = \max\{i : i \times (p-1) < n\}$ and obtain $D_0 = (XG^{(1)}, \dots, XG^{(\lambda)}, XG_h^{(1)})$ with $n_0 = (\lambda + 1/2)(p-1)$ columns. While for other situations, i.e. if $n \bmod (p-1) = 0$ or $> (p-1)/2$, just juxtapose as many XG 's as possible to form D_0 . Denote $\lambda = \min\{i : i \times (p-1) \geq n\}$ and get $D_0 = (XG^{(1)}, \dots, XG^{(\lambda)})$ with $n_0 = \lambda(p-1)$ columns. After acquiring a (p, p^{n_0}) -design D_0 , list all consecutive sorted N numbers in $\{1, \dots, p\}$ as the required row labels and such n numbers in $\{1, \dots, n_0\}$ as the required column labels; then pick the corresponding rows and columns in D_0 to form sub-designs. The levels of such sub-designs at present are bound to be in a mess, so we have to reorder them in each column and collapse N levels to s . Next, choose the optimal one with the minimum WD-value from $\min\{T, (p-N+1)(n_0-n+1)\}$ alternative sub-designs. The limited iteration prevents the runtime of SOM from being too long. Here we cut both rows and columns simultaneously, which is essentially consistent with the idea of Ma and Fang (2004). Finally, if $N = s$, a hint "Setting N and s as p will be a better choice!" is given; while if $N \neq s$, we recommend increasing the level s .

6 Numerical examples

Numerical experiments not only compare the squared WD-values obtained by SOM, the UniDOE package in R and the lower bound (4) under the same parameters, recorded as WD_{SOM} , WD_{UniDOE} and LB respectively; but also compare their runtime with the same iterations. Notice that LB is set to be the maximum of 0 and (4), as LB is always non negative. In order to display the *optimization effect* (OE) more intuitively, a reasonable criterion

$$OE = \begin{cases} 0, & \text{if } WD_{UniDOE} = LB; \\ \frac{WD_{UniDOE} - WD_{SOM}}{WD_{UniDOE} - LB} \times 100, & \text{if } WD_{UniDOE} > LB, \end{cases} \quad (8)$$

is proposed. Similar criterion can be found in Joseph and Hung (2008). As expected, $0 \leq OE \leq 100$. The larger OE is, the closer WD_{SOM} is to LB compared to WD_{UniDOE} . Notice that when $WD_{UniDOE} = WD_{SOM} > LB$, the designs found by two methods

share the same uniformity, but it does not imply that they are not optimal although $OE = 0$. For instance, WD_{SOM} and WD_{UniDOE} with 2000 iterations under the input parameter $(N, n, s) = (9, 2, 3)$ are both 0.04973, which is greater than $LB = 0.04467$, and the gained designs are both $(9, 3^2)$ -full factorial ones. However, as [Zhou et al. \(2012\)](#) pointed out, a full factorial design has the minimum WD-value over $\mathcal{U}(N; s^n)$. Such a phenomenon is because that LB is usually not tight enough.

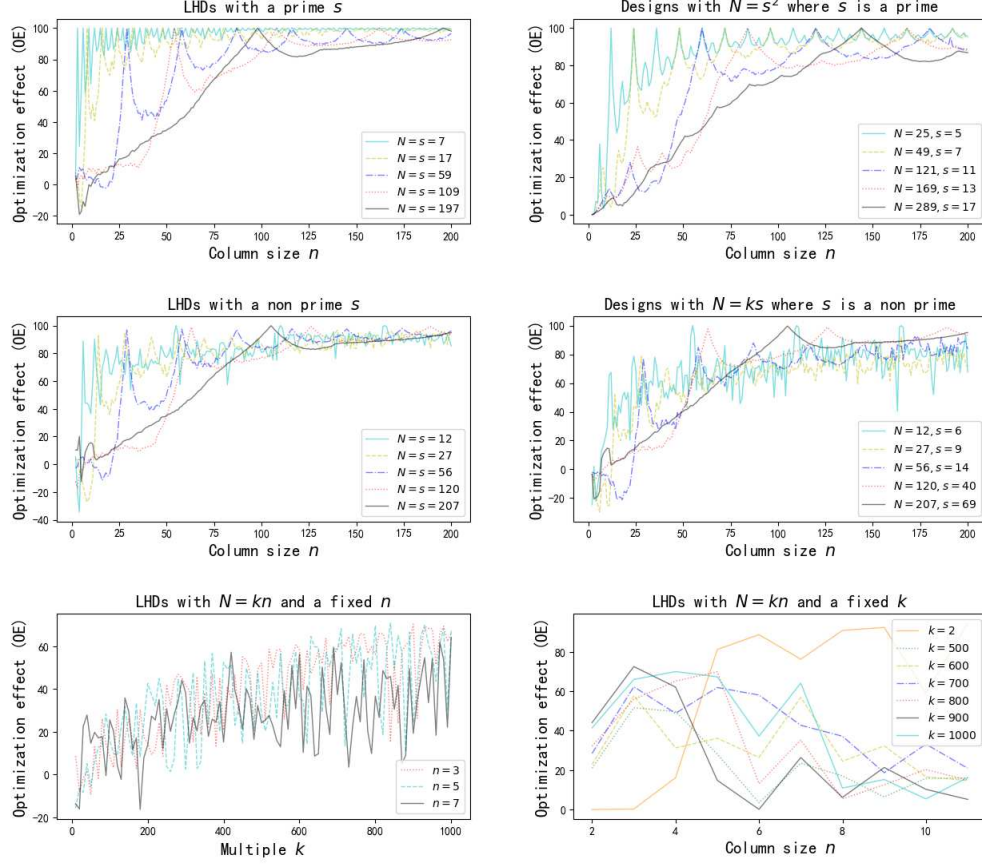


Fig. 2 Optimization effect

Numerous numerical examples have been conducted, as depicted in Figure 2. The iteration parameter T is 2000 for both WD_{SOM} and WD_{UniDOE} . The first two sub-graphs mainly show OEs of Case 1. As n increases, OE intermittently equals 100 like a harmonic wave, and the amplitude gradually decreases, leading to a steady trend towards 100. When $N = s = 59$ for example, a uniform design with $n = 29, 58, 87, \dots$, whose squared WD-value equals to LB , is directly constructed, so its OE is 100. When n approaches these values, the obtained designs still retain the uniform structure well,

so their OEs are close to 100. The middle two subgraphs mainly correspond to Case 4, which are similar to the first two ones. But the amplitude is not particularly significant due to the row cutting in Case 4. Additionally, OEs of Case 2 and Case 3 have actually been included in the middle subgraphs. For instance, when $n \leq 13$ and $N = s = 207$, there is a range of OE significantly greater than 0. This is because the designs cut out by the PGLP method perform well under WD when the row-column ratio is very large, as pointed out by Case 2. And when $N = s = 12$ and $n = 54, 55, 56, 109, 110, 111, \dots$, the design is directly constructed by Case 3, with its OE close to 100. The last two subgraphs respectively portray the situations where N increases with n unchanged and N and n increase proportionally. When N increases with n unchanged, few columns are cut out from a larger PGLP set, which leads to an unstable structure. As N increases, OE is generally greater than 0 and fluctuates upward. While in the last subgraph, overall SOM is better than UniDOE package. Especially when $N = 2n$, this ratio is extremely close to $n = (N - 1)/2$ in Corollary 1, and thus its OE gradually approaches 100.

At last, we compare the runtime of our SOM and the UniDOE package under the same iterations, outlined in Figure 3. It is widely recognized that the time complexity of calculating WD-value by (1) is $\mathcal{O}(N^2n)$, which is time-consuming. SOM converts its calculation from a simple loop to a matrix operation to shorten runtime. The specific algorithm based on R is given as Algorithm 1.

Considering that the maximum iterations T in the input parameters of SOM only limits the upper bound of iterations, rather than the actual iterations, the iterations of UniDOE package here is set to be the same as the actual iterations of SOM, in order to compare the runtime of these two more fairly and reasonably. Particularly, when SOM directly constructs the design, the corresponding iteration of UniDOE package is set to be 1. In Figure 3, it is apparent that SOM takes less time than UniDOE package. The former exhibits the staged changes as n increases, because the direct construction under some parameters takes no runtime. For example in the first sub-figure, when $N = s = 197$, the runtime tends to 0 if $n = 97, 98, 99, 195, 196, 197$. When n changes from 99 to 100, SOM immediately changes from direct construction to cutting out the optimal 100 columns from an $(197, 197^{196})$ -uniform LHD with cutting frequency of $196 - 100 + 1 = 97$ times. When n gradually increases from 100 to 194, the cutting frequency drops from 97 to 3. Therefore, its runtime always increases rapidly and then slowly decreases to 0 in each stage. The runtime of SOM is always bound by a trend as n changes. This ensures that the runtime will not explode even if abundant factors are required, guaranteed by Theorem 1.

When N increases with n unchanged and N and n increase proportionally, the maximum iterations T of SOM and that of UniDOE package in the last two subgraphs are both set to be 100. It is easy to see that, the runtime of UniDOE package shows an obvious linear growth trend with the increase of design scale; while the average runtime of SOM is much less than that of UniDOE package. Additionally, it is worth mentioning that in the actual use of UniDOE packages, the iterations typically take several thousand or even more, rather than dozens or hundreds here. When the iteration is set to be less than 2000, WD_{UniDOE} is generally worse than

Algorithm 1 A fast calculation for WD

Require: A balanced design D_0 **Ensure:** The squared WD-value of D_0

```
1: function WD2( $D_0$ )
2:    $N \leftarrow \text{nrow}(D_0)$ 
3:    $n \leftarrow \text{ncol}(D_0)$ 
4:   if  $n == 0$  then
5:      $result \leftarrow 0$ 
6:   else
7:      $s \leftarrow \text{apply}(D_0, 2, \text{max})$ 
8:     if  $\text{max}(s) \geq 1$  then
9:       if  $\text{min}(\text{apply}(D_0, 2, \text{min})) == 0$  then
10:         $s \leftarrow s + 1$ 
11:         $t \leftarrow \text{matrix}(\text{rep}(s, \text{each} = N), \text{nrow} = N, \text{byrow} = \text{TRUE})$ 
12:         $D \leftarrow (D_0 + 0.5)/t$ 
13:      else
14:         $t \leftarrow \text{matrix}(\text{rep}(s, \text{each} = N), \text{nrow} = N, \text{byrow} = \text{TRUE})$ 
15:         $D \leftarrow (D_0 - 0.5)/t$ 
16:      end if
17:    else
18:       $t \leftarrow \text{matrix}(\text{rep}(s, \text{each} = N), \text{nrow} = N, \text{byrow} = \text{TRUE})$ 
19:       $D \leftarrow D_0$ 
20:    end if
21:     $L \leftarrow \text{vector}("list", N - 1)$ 
22:    for  $i$  in  $1 : (N - 1)$  do
23:       $X_1 \leftarrow \text{matrix}(\text{rep}(D[i, ], N - i), \text{nrow} = N - i, \text{byrow} = \text{TRUE})$ 
24:       $A \leftarrow X_1 - D[(i + 1) : N, ]$ 
25:       $L[[i]] \leftarrow \text{sum}(\text{apply}(3/2 - \text{abs}(A) + A^2, 1, \text{prod}))$ 
26:    end for
27:     $S \leftarrow \text{sum}(\text{unlist}(L))$ 
28:     $result \leftarrow -(4/3)^n + (3/2)^n/N + 2 * S/(N^2)$ 
29:  end if
30:  return  $result$ 
31: end function
```

that in Figure 2. So in summary, SOM runs faster than UniDOE package under the same iteration, and the resulted designs of SOM are far more uniform.

7 Conclusions and discussions

This paper firstly characterizes uniform designs with the minimum WD-values and proposes deterministic constructions for uniform designs with any level. On the basis of these, we provide SOM for searching nearly uniform designs with WD-values closer to the minimum. Numerical experiments show that SOM is far superior to UniDOE package and approaches the lower bound more quickly.

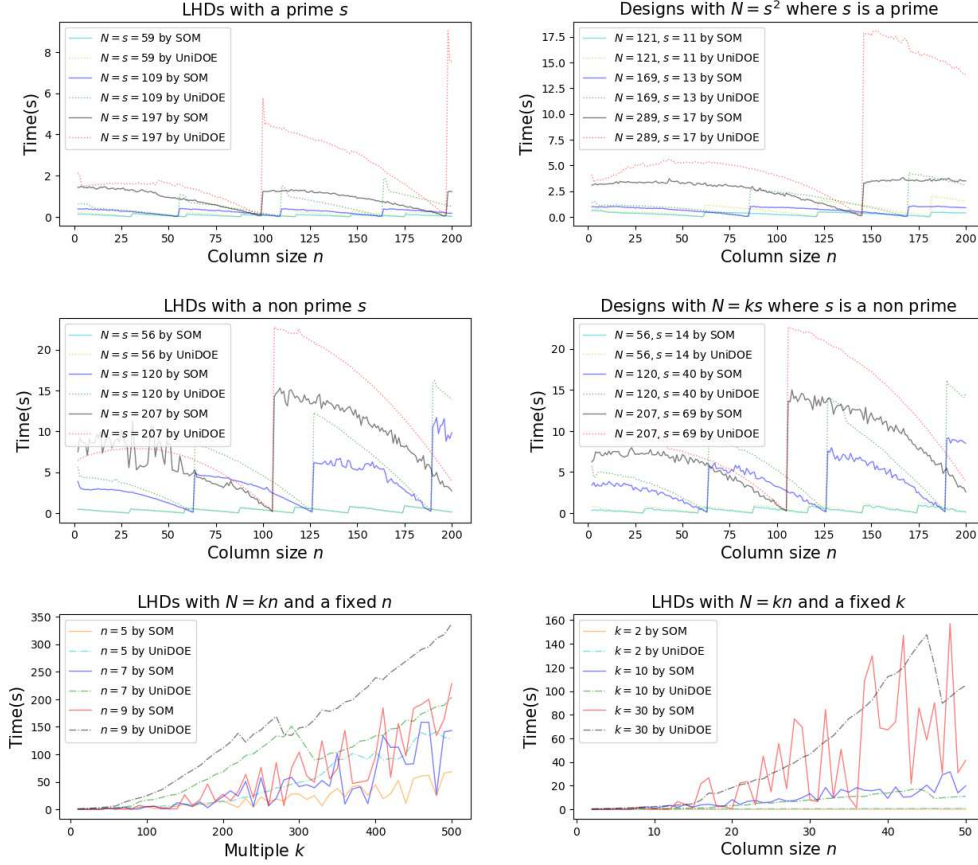


Fig. 3 Runtime(s)

For parameter inputs with large row-column ratio N/n , there may has a lot of room for improvement, which is worth for future study.

Statements and Declarations

- Competing interests

The authors declare no competing interests.

- Funding

This work was supported by the National Natural Science Foundation of China (12131001), the Fundamental Research Funds for the Central Universities, LPMC, and KLMDASR.

- Data availability

Data will be made available on request.

- Author contribution

First author: Writing-original draft, review & editing, investigation, visualization, software. Second author: Review, methodology, validation, funding acquisition. Third author: Writing-review & editing, methodology, funding acquisition.

Appendix A Proofs

Proof of Corollary 1: Let $N = ks$. If s is even, then $\Delta_1 = (k-1)n/(N-1)$ and $\Delta_2 = kn/(N-1)$ should be both integers by (3). So $n = kn - (k-1)n$ is a multiple of $N-1$. Similarly when s is odd, Δ_1 and $2\Delta_2$ are integers, resulting in $2n = 2kn - 2(k-1)n$ being a multiple of $N-1$. Therefore, if s and k are both odd, then $N-1$ is even and $n = \lambda(N-1)/2$; while if s is odd but k is even, $N-1$ is odd and $n = \lambda(N-1)$. $\square$

Proof of Theorem 1: For the design $[D_1, D_2] = (x_{kl})_{N \times n}$, we have that $\text{WD}^2([D_1, D_2]) = -(4/3)^n + \sum_{k,j}^N \prod_{l=1}^n (3/2 - |x_{kl} - x_{jl}| + |x_{kl} - x_{jl}|^2)/N^2 = -(4/3)^n + (3/2)^n/N + \sum_{k \neq j}^N \prod_{l_1=1}^{n_1} (3/2 - |x_{kl_1} - x_{jl_1}| + |x_{kl_1} - x_{jl_1}|^2) \prod_{l_2=n_1+1}^n (3/2 - |x_{kl_2} - x_{jl_2}| + |x_{kl_2} - x_{jl_2}|^2)/N^2$. Given that any two runs in the (N, s^{n_1}) -design D_1 satisfy (3), it is obviously that $\prod_{l_1=1}^{n_1} (3/2 - |x_{kl_1} - x_{jl_1}| + |x_{kl_1} - x_{jl_1}|^2)$ is a constant for any k and j , recorded as K_s . And combined with the the expression of $\text{WD}^2(D_2)$ by (1), the result is natural. $\square$

Proof of Corollary 2: Given that the distance distribution between any two rows in a balanced design satisfying (3) is fixed, the WD-values of the resulting sub-designs are clearly the same after removing any column. For a balanced s -level column, each level is repeated N/s times. So when s is an odd, among the total $\binom{N}{2}$ element pairs, there are $\binom{N/s}{2} \times s = N(N-s)/(2s)$ pairs with Lee distance equal to 0, and N^2/s pairs with Lee distance equal to 1, 2, ..., $(s-1)/2$ respectively. Based on (4), it is easy to obtain the expression of $\text{WD}^2(D)$. And when s is even, the idear is similar. Next, we prove that D is asymptotically uniform. When s is odd, the lower bound of an (N, s^{n-1}) -design, $\text{LB}(\text{WD}^2)$ in (4), can be obtained. Considering that $(N-1)/N = (N-s)/(Ns) + (2/s) \times [(s-1)/2]$, $\text{LB}(\text{WD}^2)$ can be expressed as

$$\begin{aligned} \text{LB}(\text{WD}^2) = & -\left(\frac{4}{3}\right)^{n-1} + \frac{1}{N} \left(\frac{3}{2}\right)^{n-1} + \frac{N-s}{Ns} \left(\frac{3}{2}\right)^{\frac{(N-s)(n-1)}{(N-1)s}} P^{\frac{2N(n-1)}{(N-1)s}} \\ & + \frac{2}{s} \left(\frac{3}{2}\right)^{\frac{(N-s)(n-1)}{(N-1)s}} P^{\frac{2N(n-1)}{(N-1)s}} + \dots + \frac{2}{s} \left(\frac{3}{2}\right)^{\frac{(N-s)(n-1)}{(N-1)s}} P^{\frac{2N(n-1)}{(N-1)s}}, \end{aligned}$$

where $P = [3/2 - 2(2s-2)/(4s^2)] \times \dots \times [3/2 - (s-1)(s+1)/(4s^2)]$. Note that the total $(s-1)/2$ terms from the fourth term to the last term are completely identical. Then we compare $\text{WD}^2(D)$ and $\text{LB}(\text{WD}^2)$ term by term. Specifically, the comparison between the third terms of two is scaled as follows,

$$\frac{(3/2)^{\frac{(N-s)n}{(N-1)s}-1} P^{\frac{2Nn}{(N-1)s}}}{(3/2)^{\frac{(N-s)(n-1)}{(N-1)s}} P^{\frac{2N(n-1)}{(N-1)s}}} = \left(\frac{3}{2}\right)^{-\frac{N(s-1)}{(N-1)s}} P^{\frac{2N}{(N-1)s}} \leq \frac{\left[(3/2)^{\frac{s-1}{2}}\right]^{\frac{2N}{(N-1)s}}}{(3/2)^{\frac{N(s-1)}{(N-1)s}}} = 1.$$

And the ratio between the fourth terms of two is as follows,

$$\frac{(3/2)^{\frac{(N-s)n}{(N-1)s}} \left[3/2 - 2(2s-2)/(4s^2)\right]^{\frac{2Nn}{(N-1)s}-1} Q^{\frac{2Nn}{(N-1)s}}}{(3/2)^{\frac{(N-s)(n-1)}{(N-1)s}} P^{\frac{2N(n-1)}{(N-1)s}}}$$

$$\begin{aligned}
&= \left(\frac{3}{2}\right)^{\frac{N-s}{(N-1)s}} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right]^{\frac{-(s-2)N+s}{(N-1)s}} Q^{\frac{2N}{(N-1)s}} \\
&\leq \left(\frac{3}{2}\right)^{\frac{N-s}{(N-1)s}} \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right]^{\frac{-(s-2)N+s}{(N-1)s}} \left\{ \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right]^{\frac{s-3}{2}} \right\}^{\frac{2N}{(N-1)s}} \\
&= \left(\frac{3}{2}\right)^{\frac{N-s}{(N-1)s}} / \left[\frac{3}{2} - \frac{2(2s-2)}{4s^2} \right]^{\frac{N-s}{(N-1)s}} \leq \left[\frac{3}{2} / \left(\frac{3}{2} - \frac{1}{s} \right) \right]^{\frac{N-s}{(N-1)s}},
\end{aligned}$$

where $Q = [3/2 - 4(2s-4)/(4s^2)] \times \cdots \times [3/2 - (s-1)(s+1)/(4s^2)]$. Let $N = ks$. Then $N \rightarrow \infty$ naturally leads to $s \rightarrow \infty$, so we have $(N-s)/[(N-1)s] = (k-1)/(N-1) \rightarrow 0$ and $(3/2)/(3/2 - 1/s) \rightarrow 1$. The comparisons of the other items are similar. Therefore, $\text{WD}^2(D)/\text{LB}(\text{WD}^2) \leq 1$ is concluded. Obviously $\text{WD}^2(D) \geq \text{LB}(\text{WD}^2)$, resulting in $\text{WD}^2(D) \rightarrow \text{LB}(\text{WD}^2)$. While s is even, the proof is similar. $\square$

Proof of Corollary 3: The necessity is guaranteed by the property of column juxtaposition. As for the sufficiency, the distribution of the row difference in D_1 is the same as that of D_2 according to the condition. So if D satisfies (3) with the parameters $\Delta_1 = (N-s)n/[(N-1)s]$ and $\Delta_2 = Nn/[(N-1)s]$, then the difference of any two rows in D_1 or D_2 satisfies (3) with $\Delta_1 = (N-s)n/[2(N-1)s]$ and $\Delta_2 = Nn/[2(N-1)s]$. $\square$

Proof of Theorem 2: Considering the combinatorial properties of a COA, for any two rows of D , we always have $\sigma_0 = 0$, $\sigma_1 = \cdots = \sigma_{(s-1)/2} = 2\lambda s$ if s is odd, and $\sigma_1 = \cdots = \sigma_{(s-2)/2} = 2\lambda s$ and $\sigma_{s/2} = \lambda s$ if s is even. It can be easily verified to satisfies (3). $\square$

Proof of Corollary 4: The transpose of a $\text{COA}(\lambda s(s-1), s)$, denoted as D_0 , is a sub-design of the resulting $(s+1, (s+1)^{\lambda s(s-1)})$ -LHD, denoted as D . For any two rows in D_0 with s levels, $\sigma_0 = 0$ and $\sigma_1 = 2\lambda(s-1)$ in the sense of $s+1$ levels. The occurrences of other equivalence elements can be similarly obtained, which satisfies (3) in the sense of $s+1$ levels. On the other hand, for any row in D_0 and the added row $(s, s, \dots, s)$ in D , the conclusion also holds, guaranteed by the balance of COA. $\square$

Proof of Theorem 3: D_0 in Construction 1 is always able to be split into two $(s, s^{s(s-1)/2})$ -sub-designs, one of whom is D_1 . Then by Corollary 3, D_1 satisfies (3) and is thus uniform. As for D_2 , Step 3 in Construction 1 ensures that each row in D_1 is balanced, then the discussion for D_2 is similar to the proof of Corollary 4. $\square$

Proof of Corollary 5: The balance of D is guaranteed by that of D_0 . Assuming that $\lambda = 1$, any two runs of D_0 can be expressed as a juxtaposition of the following,

$$\begin{pmatrix} d & d & \cdots & d & d & \cdots & d \\ 0 & 1 & \cdots & d-1 & d+1 & \cdots & s-1 \end{pmatrix},$$

where $d = 0, 1, \dots, s-1$. After the level collapse from s to s_0 , any two runs of D can be expressed as k repeated juxtapositions of the following,

$$\begin{pmatrix} d_0 & \cdots & d_0 & \cdots & d_0 & \cdots & d_0 & \cdots & d_0 & \cdots & d_0 \\ 0 & \cdots & 0 & \cdots & d_0 & \cdots & d_0 & \cdots & s_0-1 & \cdots & s_0-1 \end{pmatrix},$$

where $0, \dots, d_0-1, d_0+1, \dots, s_0-1$ are repeated k times respectively and d_0 is repeated $k-1$ times. Then for any two runs of the $(s, s_0^{s(s-1)})$ -design D , $\sigma_0 = (k-1)ks_0 = (k-1)s$,

$\sigma_1 = \cdots = \sigma_{(s_0-1)/2} = (2k)ks_0 = 2ks$ when s is odd, and $\sigma_1 = \cdots = \sigma_{s_0/2-1} = 2ks$ and $\sigma_{s_0/2} = ks$ when s is even. The discussion is similar when $\lambda \geq 2$. Note that in sense of new parameters, $\Delta_1 = (N - s_0)n/[(N - 1)s_0] = (s - s_0)[\lambda s(s - 1)]/[(s - 1)s_0] = \lambda(k - 1)s$ and $\Delta_2 = Nn/[(N - 1)s_0] = s[\lambda s(s - 1)]/[(s - 1)s_0] = \lambda ks$ and so on. It can be checked that D satisfies (3) as above. $\square$

References

- Fang, K.: The uniform design: application of number-theoretic methods in experimental design. *Acta Mathematicae Applicatae Sinica* **3**(4), 363–372 (1980)
- Fang, K., Liu, M., Qin, H., Zhou, Y.: *Theory and Application of Uniform Experimental Designs*. Science Press and Springer, Beijing (2018)
- Fang, K., Li, R., Sudjianto, A.: *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC, London (2006)
- Fang, K., Ma, C.: Wrap-around l_2 -discrepancy of random sampling, latin hypercube and uniform designs. *Journal of Complexity* **17**(4), 608–624 (2001)
- Fang, K., Tang, Y., Yin, J.: Lower bounds for wrap-around L_2 -discrepancy and constructions of symmetrical uniform designs. *Journal of Complexity* **21**(5), 757–771 (2005)
- Harase, S.: Quasi-monte carlo point sets with small t -values and wafom. *Applied Mathematics and Computation* **254**, 318–326 (2015)
- Hickernell, F.J.: Lattice rules: How well do they measure up? In: Hellekalek, P., Larcher, G. (eds.) *Random and Quasi-Random Point Sets*. Lecture Notes in Statistics, vol. 138, pp. 109–166. Springer, New York (1998)
- Huang, H.: Construction of component orthogonal arrays with any number of components. *Journal of Statistical Planning and Inference* **213**, 72–79 (2021)
- Hua, L., Wang, Y.: *Applications of Number Theory to Numerical Analysis*. Springer, Berlin (1981)
- Joseph, V.R., Hung, Y.: Orthogonal maximin latin hypercube designs. *Statistica Sinica* **18**, 171–186 (2008)
- Korobov, N.M.: The approximate computation of multiple integrals. *Doklady Akademii nauk SSSR* **124**, 1207–1210 (1959)
- Ma, C., Fang, K.: A new approach to construction of nearly uniform designs. *International Journal of Materials and Product Technology* **20**, 115–126 (2004)
- Morris, M.D., Mitchell, T.J.: Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference* **43**, 381–402 (1995)

- Morris, M.D.: Factorial plans for preliminary computational experiments. *Technometrics* **33**, 161–174 (1991)
- Qi, L., Liu, Z., Zhou, Y.: Deterministic construction methods for uniform designs. *Journal of Statistical Planning and Inference* **226**, 30–38 (2023)
- Tang, Y., Xu, H., Liu, D.K.J.: Uniform fractional factorial designs. *The Annals of Statistics* **40**(2), 891–907 (2012)
- Wang, Y., Fang, K.: A note on uniform distribution and experimental design. *Chinese Science Bulletin* **26**(6), 485–489 (1981)
- Winker, P., Fang, K.: Application of threshold-accepting to the evaluation of the discrepancy of a set of points. *SIAM Journal on Numerical Analysis* **34**, 2028–2042 (1997)
- Yang, J., Sun, F., Xu, H.: A component-position model, analysis and design for order-of-addition experiments. *Technometrics* **63**(2), 212–224 (2021)
- Zhou, Y., Fang, K., Ning, J.: Constructing uniform designs: a heuristic integer programming method. *Journal of Complexity* **28**(2), 224–237 (2012)
- Zhou, Y., Fang, K., Ning, J.: Mixture discrepancy for quasi-random point sets. *Journal of Complexity* **29**, 283–301 (2013)
- Zhang, A., Li, H.: Unidoe: An r package for constructing uniform design of experiments via stochastic and adaptive threshold accepting algorithm. Technical Report (2017)
- Zhou, Y., Ning, J., Song, X.: Lee discrepancy and its applications in experimental designs. *Statistics and Probability Letters* **78**(13), 1933–1942 (2008)
- Zhou, Y., Xu, H.: Space-filling properties of good lattice point sets. *Biometrika* **102**(4), 959–966 (2015)